From Unstructured Data to In-Context Learning

A Comparative Study of Transformers and Word2Vec

Yixin Wang

Joint work with Kevin Christian Wibisono (UMich)

Language models excel at in-context learning

You

FJD -> Fiji

CAD -> Canada

JPY -> Japan

KRW -> ?

ChatGPT

KRW -> South Korea

You

How do you know South Korea is the answer? Explain your thought process.

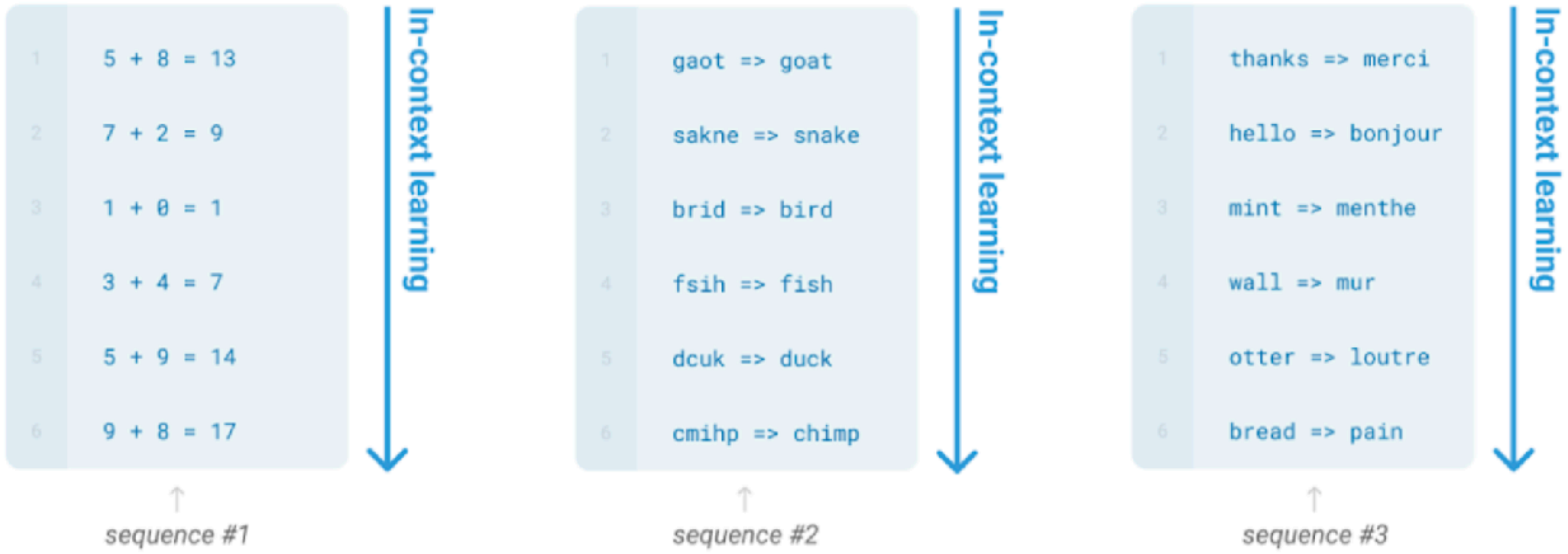
ChatGPT

The three-letter currency codes you provided (FJD, CAD, JPY) are ISO 4217 codes, which are international standard codes for currencies. In this case:

- FJD stands for Fijian Dollar (currency of Fiji),
- CAD stands for Canadian Dollar (currency of Canada),
- JPY stands for Japanese Yen (currency of Japan).

Following this pattern, KRW is the ISO 4217 code for the South Korean Won, the currency of South Korea. Therefore, KRW corresponds to South Korea based on the standard currency code system.

Language models excel at in-context learning



In-context learning: Produce correct answer for new tasks
without any parameter updates

From **next token prediction** to **in-context learning**

- **Training:**
  - next token prediction
  - **web-crawled text**
    - e.g. NY Times article
- **Test:**
  - Next token prediction
  - $x_1, f(x_1), x_2, f(x_2), x_3, f(x_3), x_4, ?$
  - Expected output: $f(x_4)$

Training Examples:

THE NEW YORK TIMES - June 15, 2023
"Man and Woman: Gender Terms in Modern Society"
By Sarah Chen
Understanding the relationship between man and woman in linguistics...

NEW YORKER - July 22, 2023
"From King to Queen: Royal Titles Explained"
By James Wilson
The distinction between king and queen in monarchical systems...

The Economist - August 5, 2023
"Size Matters: The Spectrum from Big to Small"
By Michael Roberts

Complete the Pattern:

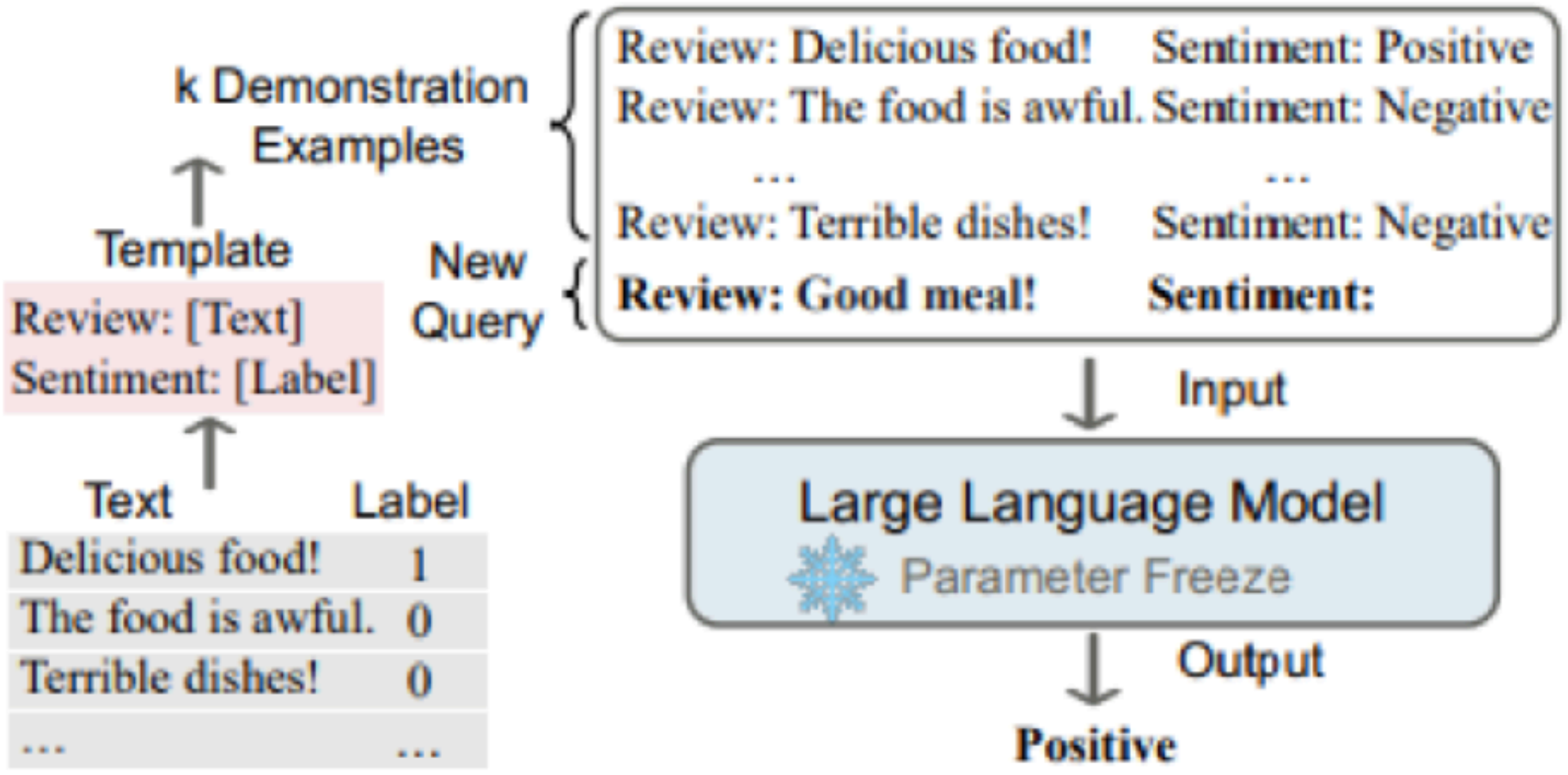
```
man, woman; king, queen; big, small; hot,cold
```

Pattern: Each pair shows opposing/complementary relationships
Prediction: Based on pattern of opposites across different domains

In-context Classification

Frame classification tasks as next token prediction in a sequence

- Train on web-crawled data
- Test on classification data



In-context Regression

Frame regression tasks as next token prediction in a sequence

- Train on sequences
  - $x_{11}^{\text{tr}}, y_{11}^{\text{tr}}, x_{12}^{\text{tr}}, y_{12}^{\text{tr}}, \ldots$
    - where $y_{1j}^{\text{tr}} = w_1^{\top} x_{1j} + \epsilon_{1j}$, $\epsilon_{1j} \sim N(0, \sigma^2)$
  - $x_{21}^{\text{tr}}, y_{21}^{\text{tr}}, x_{22}^{\text{tr}}, y_{22}^{\text{tr}}, \ldots$
    - where $y_{2j}^{\text{tr}} = w_2^{\top} x_{2j} + \epsilon_{2j}$, $\epsilon_{2j} \sim N(0, \sigma^2)$
  - ….
- Test $x_1^{\text{te}}, y_1^{\text{te}}, x_2^{\text{te}}, y_2^{\text{te}}, x_3^{\text{te}}, y_3^{\text{te}}, x_4^{\text{te}}, ?$
  - $y_i^{\text{te}} = w_{\text{new}}^{\top} x_i^{\text{te}} + \epsilon_i, \ \epsilon_i \sim N(0, \sigma^2)$
- Expected output: $\hat{y}_4^{\text{te}}$

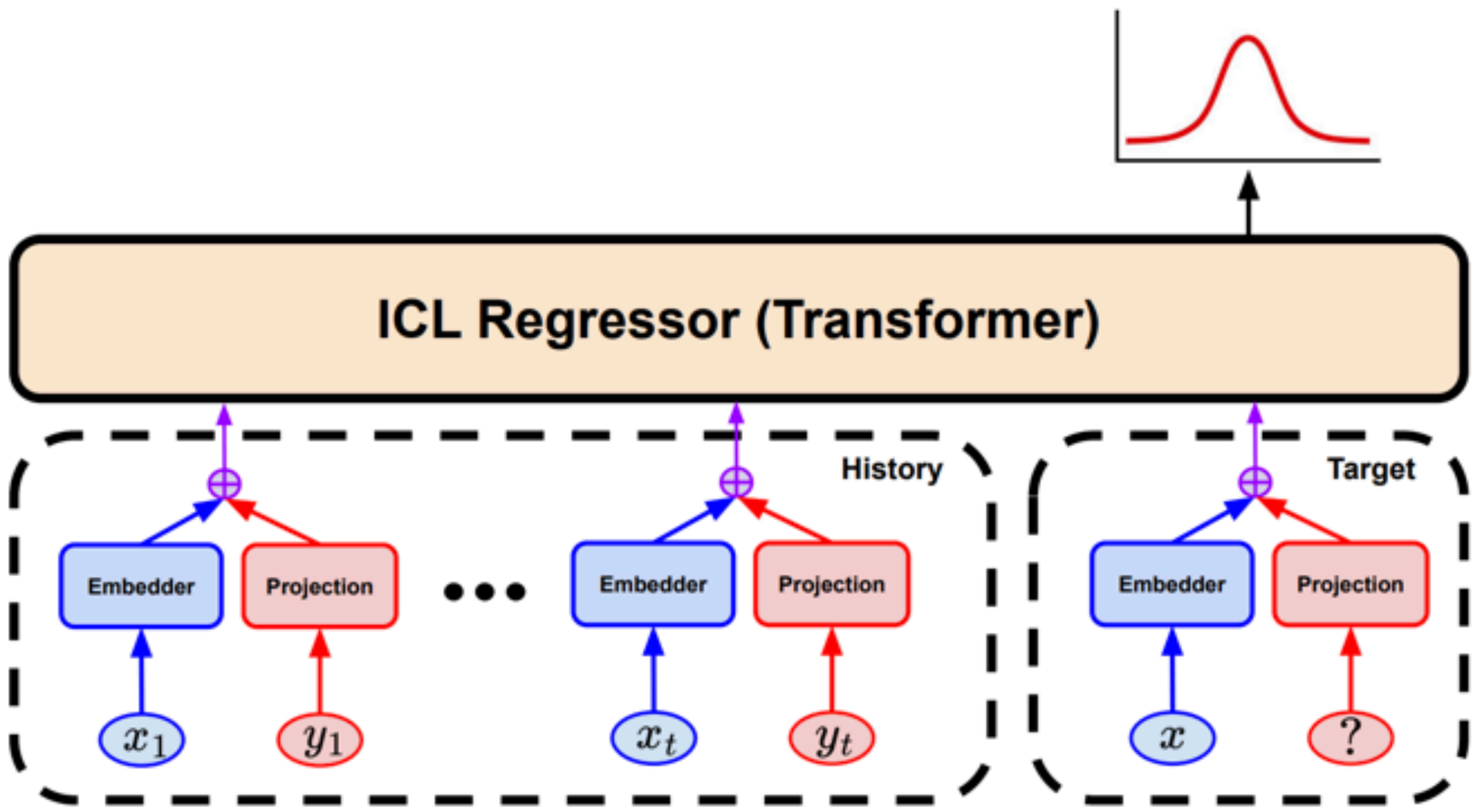
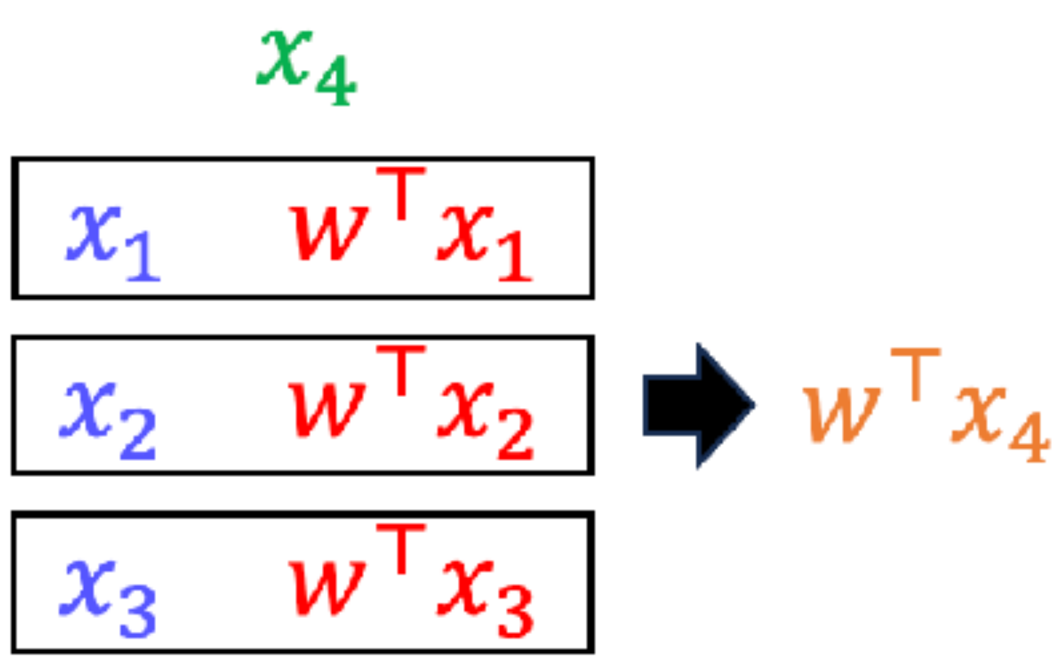


Figure 2 | Overview of our model. Most notably, candidates x are converted to language model embeddings to be ultimately used as fixed dimensional features.



From next token prediction to in-context learning

Why does in-context learning work?

- Training data typically consist of "unstructured" sentences
  - Not analogy, regression/classification, logical reasoning sequences

- Transformers-based LMs are trained to predict the next token

Training Examples:

THE NEW YORK TIMES - June 15, 2023

"Man and Woman: Gender Terms in Modern Society"

By Sarah Chen

Understanding the relationship between man and woman in linguistics...

NEW YORKER - July 22, 2023

"From King to Queen: Royal Titles Explained"

By James Wilson

The distinction between king and queen in monarchical systems...

The Economist - August 5, 2023

"Size Matters: The Spectrum from Big to Small"

By Michael Roberts

↓

Complete the Pattern:

`man, woman; king, queen; big, small; hot,`**`cold`**

Pattern: Each pair shows opposing/complementary relationships

Prediction: Based on pattern of opposites across different domains

Today

From next token prediction to in-context learning

- Is it real? Or did language model just memorize?
- What enables it?
- When does it fail?
- How does it help statistics?

Training Examples:

THE NEW YORK TIMES - June 15, 2023
"Man and Woman: Gender Terms in Modern Society"
By Sarah Chen
Understanding the relationship between man and woman in linguistics...

NEW YORKER - July 22, 2023
"From King to Queen: Royal Titles Explained"
By James Wilson
The distinction between king and queen in monarchical systems...

The Economist - August 5, 2023
"Size Matters: The Spectrum from Big to Small"
By Michael Roberts

Complete the Pattern:

`man, woman; king, queen; big, small; hot,`**`cold`**

Pattern: Each pair shows opposing/complementary relationships
Prediction: Based on pattern of opposites across different domains

Spoiler Alert

- It is real.
- In-context learning is not specific to transformer.
- Can replicate it with more classical language models (word2vec)
  - continuous-bag-of-words (CBOW) in word2vec
  - latent-factor-model-like
- Transformer = CBOW + (stacked) mixture-of-experts
- Extend to non-natural-language sequence data and tabular data

Is in-context learning from unstructured data real?

In-context Logical Reasoning

A curious experiment

- [40, 1, 4, 10, 3, 6, 30] -> ?
- [12, 5, 1, 10, 40, 3, 4] -> ?

Wibisono & W., NeurIPS M3L 2023]

In-context Logical Reasoning

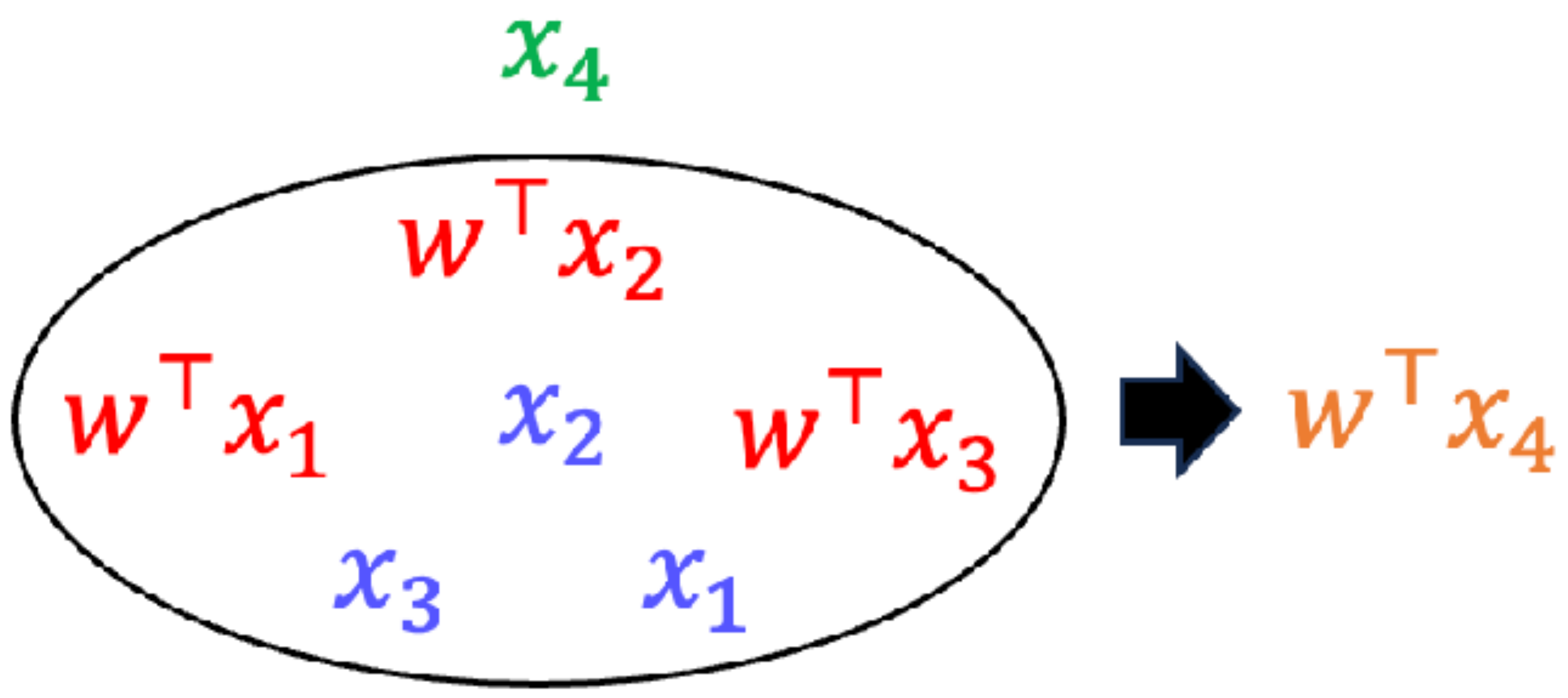
A curious experiment

- [40, 1, 4, 10, 3, 6, 30] -> ?
- [12, 5, 1, 10, 40, 3, 4] -> ?

Wibisono & W., NeurIPS M3L 2023]

In-context Logical Reasoning

A curious experiment



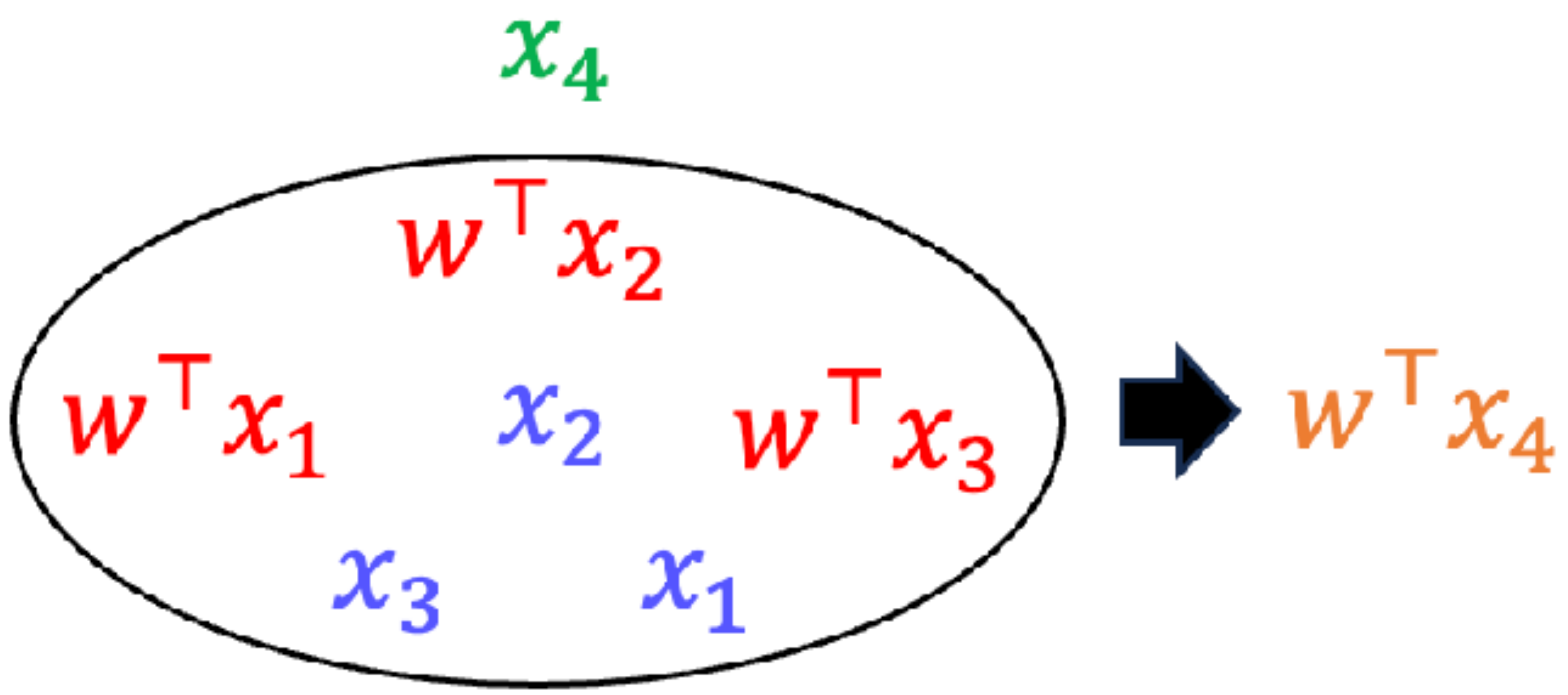
- [40, 1, 4, 10, 3, 6, 30] -> 60
- [12, 5, 1, 10, 40, 3, 4] -> 20

- **(1) Find x-y pairing**
- **(2) identify "odd-one-out"** x_4^{te}
- **(3) predict** y_4^{te}

Wibisono & W., NeurIPS M3L 2023]

In-context Logical Reasoning

A curious experiment



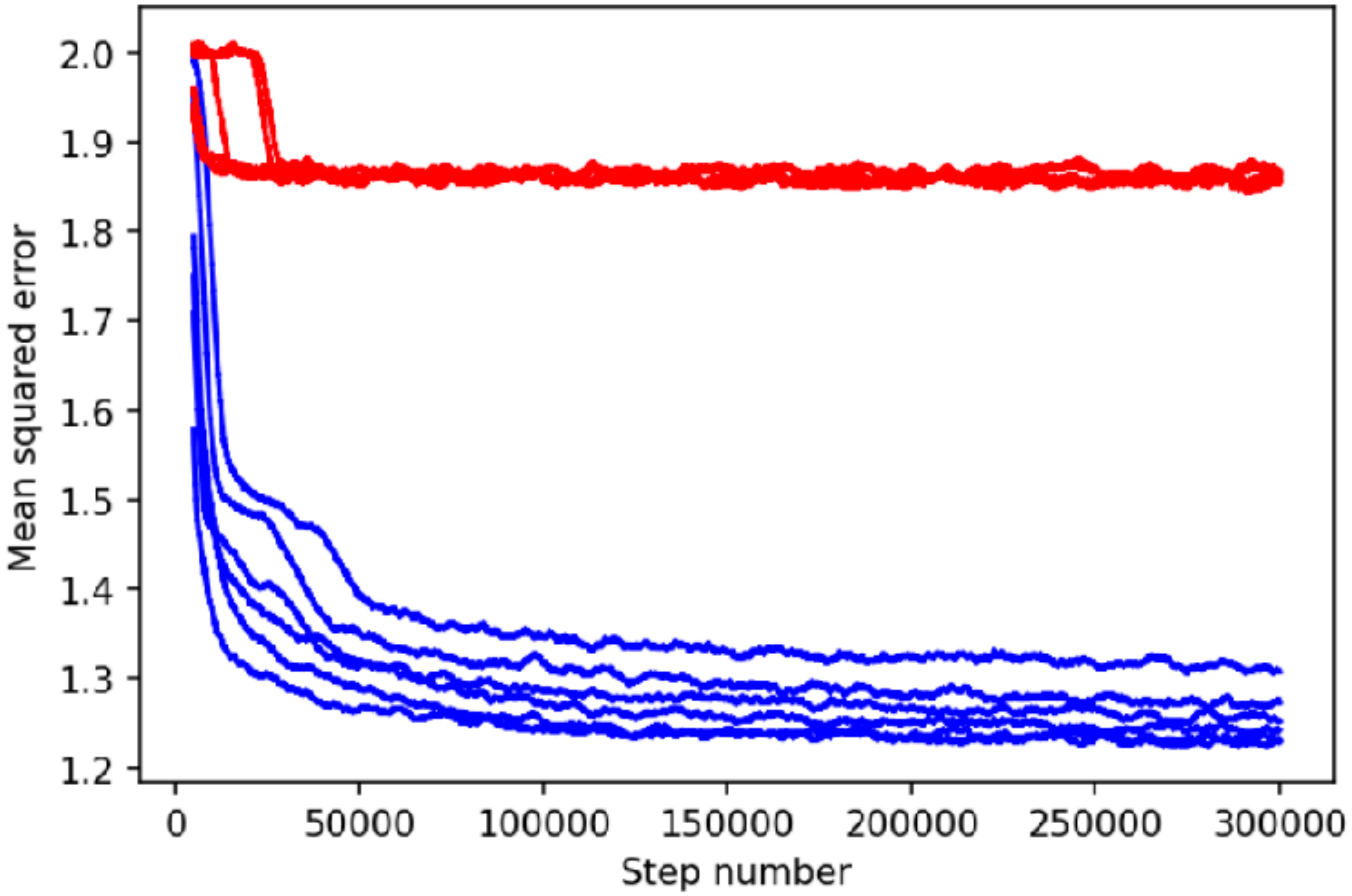
- Train on sequences (scrambled x-y order)
  - $x_{11}^{\text{tr}}, x_{14}^{\text{tr}}, y_{14}^{\text{tr}}, x_{12}^{\text{tr}}, x_{13}^{\text{tr}}, y_{11}^{\text{tr}}, x_{13}^{\text{tr}}, x_{12}^{\text{tr}}$
  - $x_{24}^{\text{tr}}, x_{22}^{\text{tr}}, y_{24}^{\text{tr}}, x_{23}^{\text{tr}}, y_{22}^{\text{tr}}, x_{21}^{\text{tr}}, x_{23}^{\text{tr}}, x_{21}^{\text{tr}}$
    - where $y_{ij}^{\text{tr}} = w_i^\top x_{ij} + \epsilon_{ij}$
- Test $x_4^{\text{te}}, x_3^{\text{te}}, y_2^{\text{te}}, y_1^{\text{te}}, x_2^{\text{te}}, y_3^{\text{te}}, x_1^{\text{te}}, ?$
  - $y_i^{\text{te}} = w_{\text{new}}^\top x_i^{\text{te}} + \epsilon_i,\ \epsilon_i \sim N(0, \sigma^2)$
- $w \sim N(0, I), X \sim N(0, \Lambda), \sigma^2 = 1$

- [40, 1, 4, 10, 3, 6, 30] -> 60
- [12, 5, 1, 10, 40, 3, 4] -> 20
- **(1) Find x-y pairing**
- **(2) identify "odd-one-out"** x_4^{te}
- **(3) predict** y_4^{te}

Wibisono & W., NeurIPS M3L 2023]

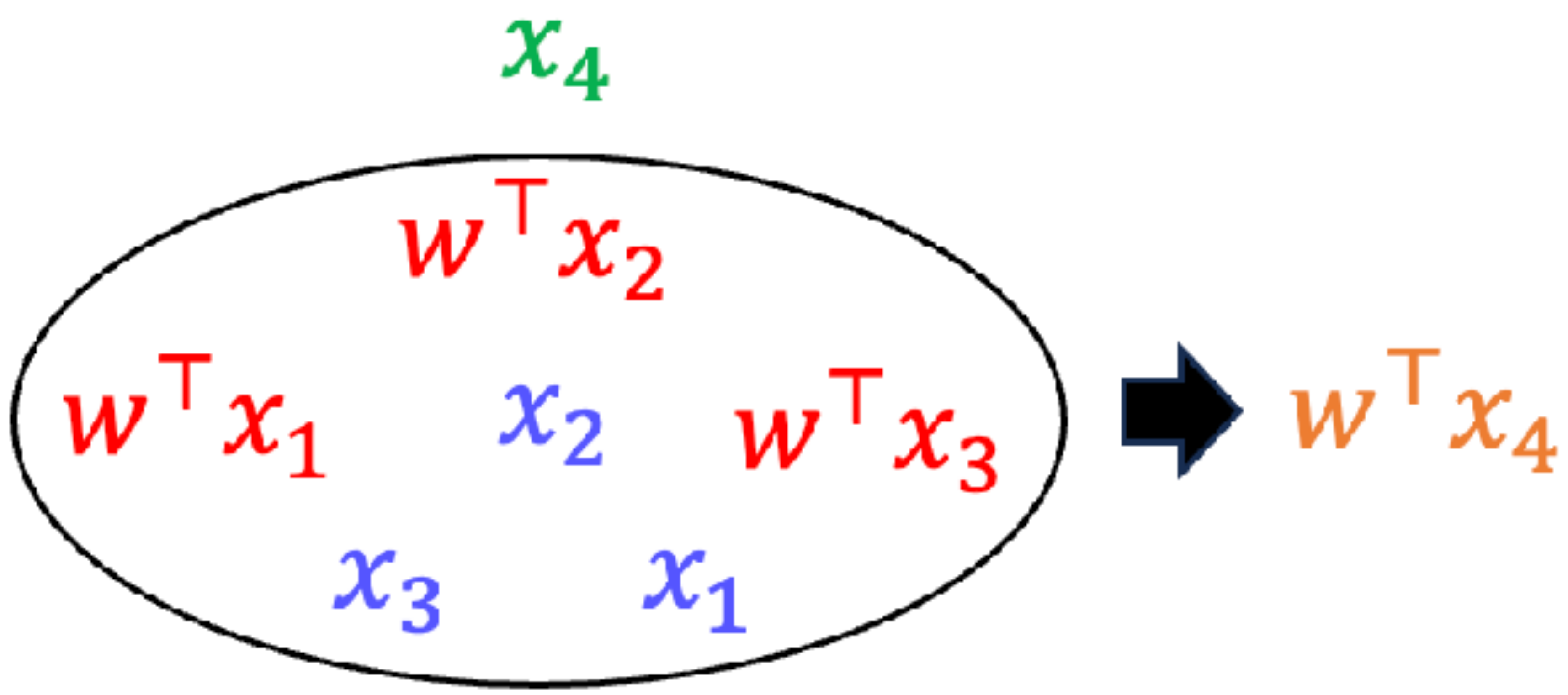
In-context Logical Reasoning

A curious experiment



$$y_i^{te} = w_{new}^\top x_i^{te} + \epsilon_i, \epsilon_i \sim N(0,1)$$

Transformer with Softmax/Linear attention



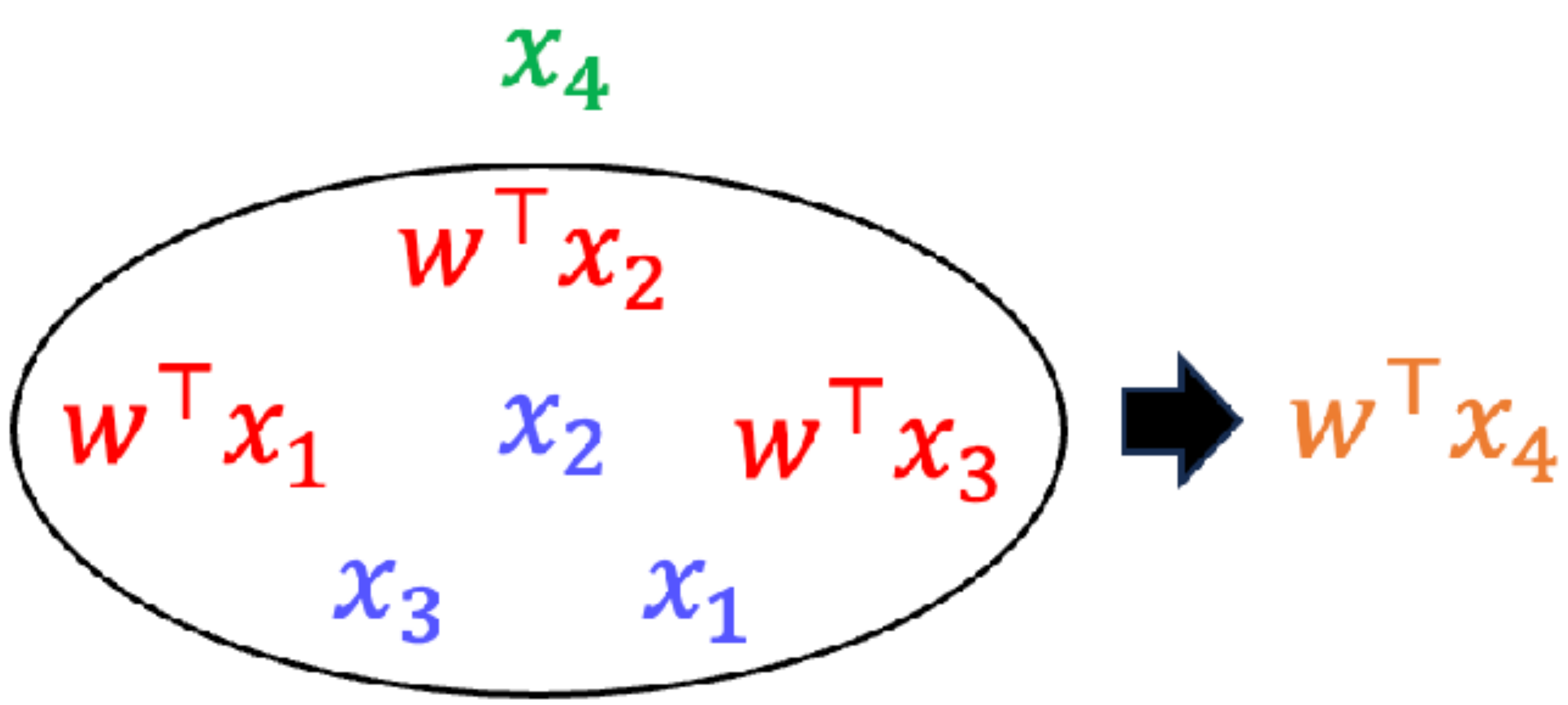
- **(1) Find x-y pairing**
- **(2) identify "odd-one-out"** x_4^{te}
- **(3) predict** y_4^{te}

Wibisono & W., NeurIPS M3L 2023]

In-context Logical Reasoning

Transformer learns x-y pairings

- Can transformer pair up x and y?
  - [40, 1, 4, 10, 3, 6, 30] -> 60
  - [12, 5, 1, 10, 40, 3, 4] -> 20



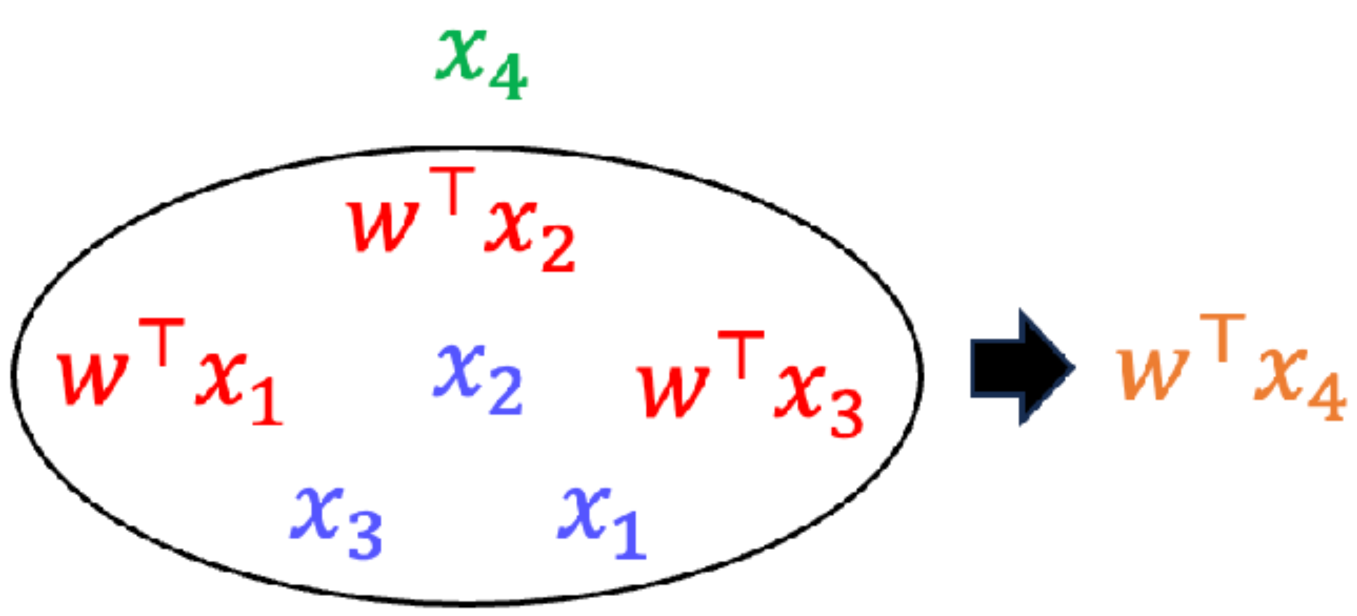
- Probing analysis with "attention"
  - 1 should pay more attention to 10 than 4 in the first
  - The opposite should happen in the second

- **(1) Find x-y pairing**
- **(2) identify "odd-one-out" x_4^{te}**
- **(3) predict y_4^{te}**

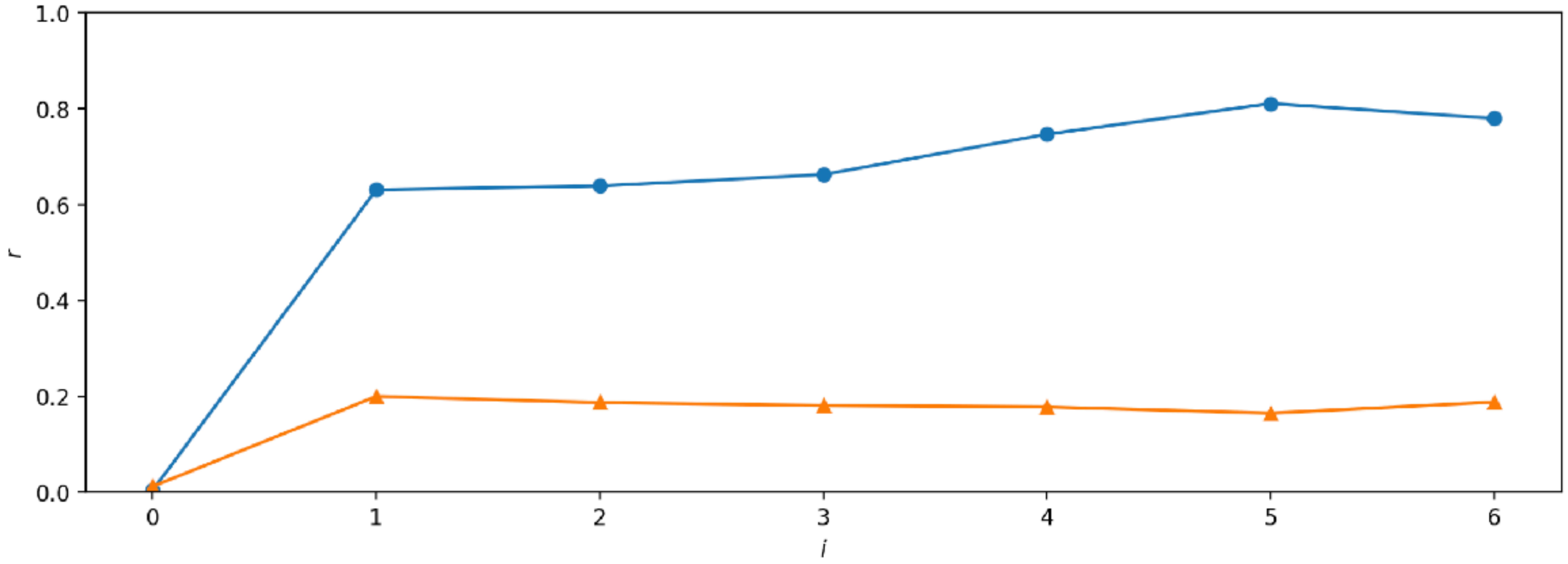
Wibisono & W., NeurIPS M3L 2023]

In-context Logical Reasoning

Transformer learns x-y pairings



- Can transformer pair up x and y?
  - [40, 1, 4, 10, 3, 6, 30] -> 60 , [12, 5, 1, 10, 40, 3, 4] -> 20
- **Yes, as early as in the first layer!**



Blue line: $\mathrm{corr}\left(w^\top x_1, \tilde{f}_i\left(r_i(x_1)\right)\right)$, correct pair

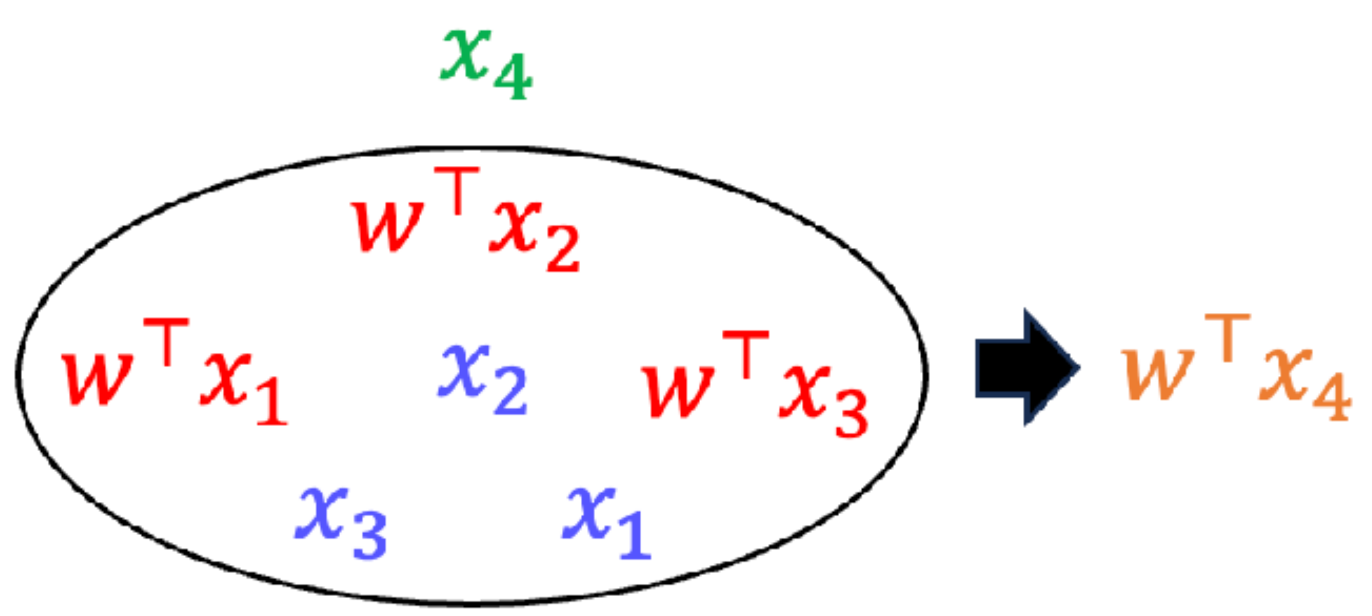
Orange line: $\mathrm{corr}\left(w^\top x_2, \tilde{g}_i\left(r_i(x_1)\right)\right)$, incorrect

i: Layer number

Wibisono & W., NeurIPS M3L 2023]

In-context Logical Reasoning

Transformer learns x-y pairings



- [40, 1, 4, 10, 3, 6, 30] -> 60
- [12, 5, 1, 10, 40, 3, 4] -> 20

- It seems real.
- Transformers really seems to be able to find repeated, heterogeneous, but related patterns in unstructured data.

- What enables it?
- Is it specific to transformers?

Wibisono & W., NeurIPS M3L 2023]

What enables in-context learning?

Result 1: **Co-occurrence modeling**

ICL for **word analogy completion** can arise by *only* modeling **co-occurrence information, without transformers**

In-context Word Analogy

Did it learn? Or did these words just co-occur often?

Training data:

In Canada, octane-95 gasoline costs CAD 7.00 (USD 5.20) per gallon, while the same quantity is priced at FJD 10.86 (USD 4.72) in Fiji..

K **You**

FJD -> Fiji

CAD -> Canada

JPY -> Japan

KRW -> ?

ChatGPT

KRW -> South Korea

[Wibisono & W., NeurIPS 2024]

In-context Learning on (country, capital city) pairs

- **Two types** of often co-occurring words: (**country, largest city), (country, capital city)**
  - 160 countries with population > 1M in 2022
    - 31 have largest city != capital city *(type A)*, e.g., Türkiye, Australia, USA.
    - 129 have largest city = capital city *(type B), e.g.,* Malaysia, Mexico, UK.

- Consider ICL task of the following form: c_1 d_1 ... c_6 d_6 c_7 *?*
  - c_i denotes a country, and d_i its capital city.
  - The desired task/relationship is clearly **country – capital city.**

[Wibisono & W., NeurIPS 2024]

In-context Learning on (country, capital city) pairs

- **Two types** of often co-occurring words: (**country, largest city), (country, capital city)**
  - 160 countries with population > 1M in 2022
    - 31 have largest city != capital city *(type A)*, e.g., Türkiye, Australia, USA.
    - 129 have largest city = capital city *(type B), e.g.,* Malaysia, Mexico, UK.

- **If transformer learns, then it should do equally well on both.**

- **Significant difference** in ICL accuracies for both country types.
  - Accuracy is 58% when c_7 is of type A; and 96% when c_7 is of type B.

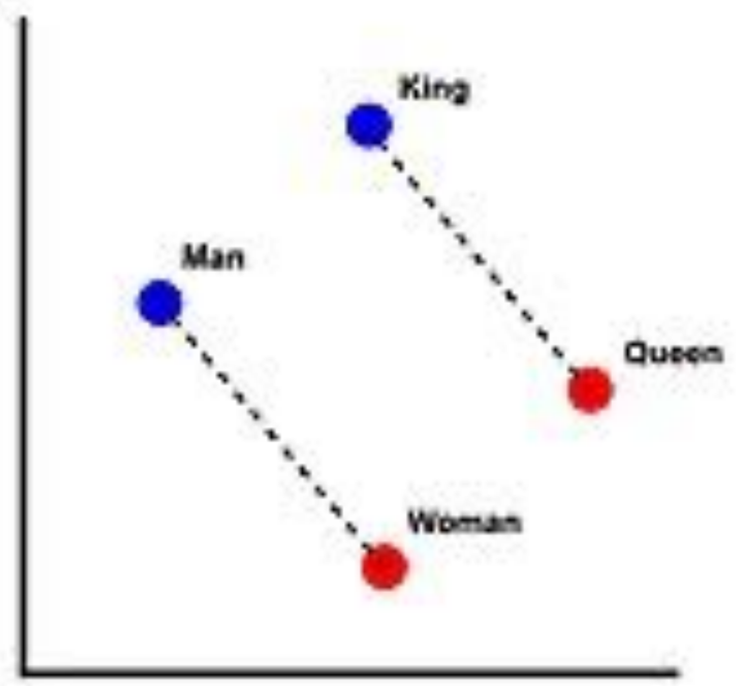
In-context learning with transformers output words that co-occurs more often.

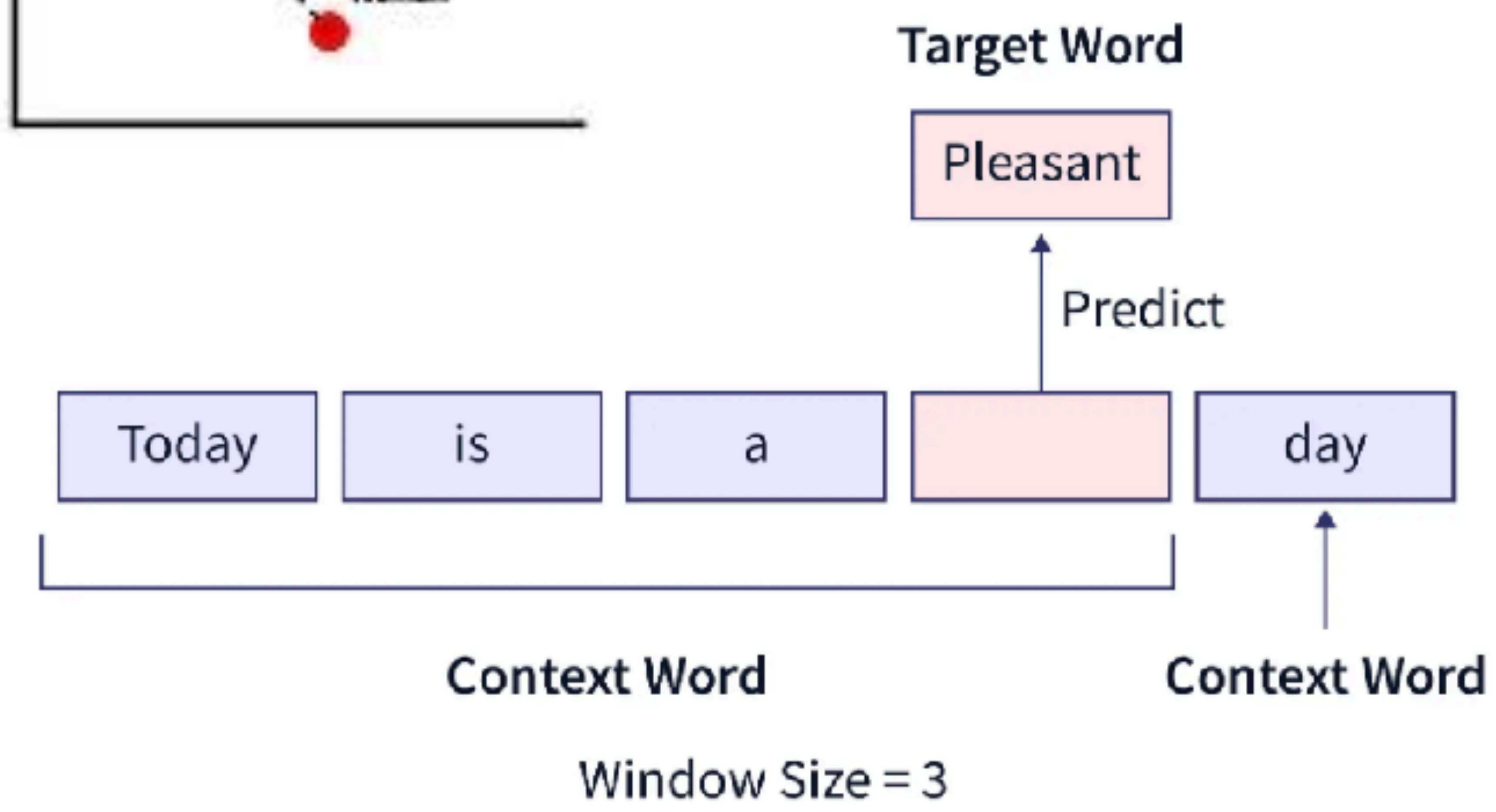
[Wibisono & W., NeurIPS 2024]

Many models can model co-occurrence

They can do in-context word analogy too

- Continuous Bag-of-words (CBOW) in word2vec
- Close to a latent factor model
- Given a sequence $x_1, x_2, \ldots, x_I$,
  - $p(x_i = k \mid \{x_j\}_{j \neq i}) \propto \exp(u_k^\top \sum_{j \neq i} v_{x_j})$
  - u_w, v_w are center and context embedding of word w





[Wibisono & W., NeurIPS 2024]

Transformer is not necessary for in-context word analogy

(p_0, p_1, p_2)	Clean $d_E = 10$	Clean $d_E = 100$
$(1/2, 1/2, 0)$	1	0.99
$(1/2, 0, 1/2)$	1	1
$(0, 1/2, 1/2)$	1	1
$(1/3, 1/3, 1/3)$	1	1

Training sentences:
The capital city of Indonesia is Jakarta.
Beijing functions as the capital of China.
Gas prices in India are around 100 Rupees per liter.
Cedi, the currency of Ghana, is facing depreciation.

ICL prompts:
Ghana Accra Nigeria Lagos China Beijing

India Rupee China Yuan Japan Yen

p_k denotes the probability of having exactly k pairs of (c_i, d_i) in the training corpus

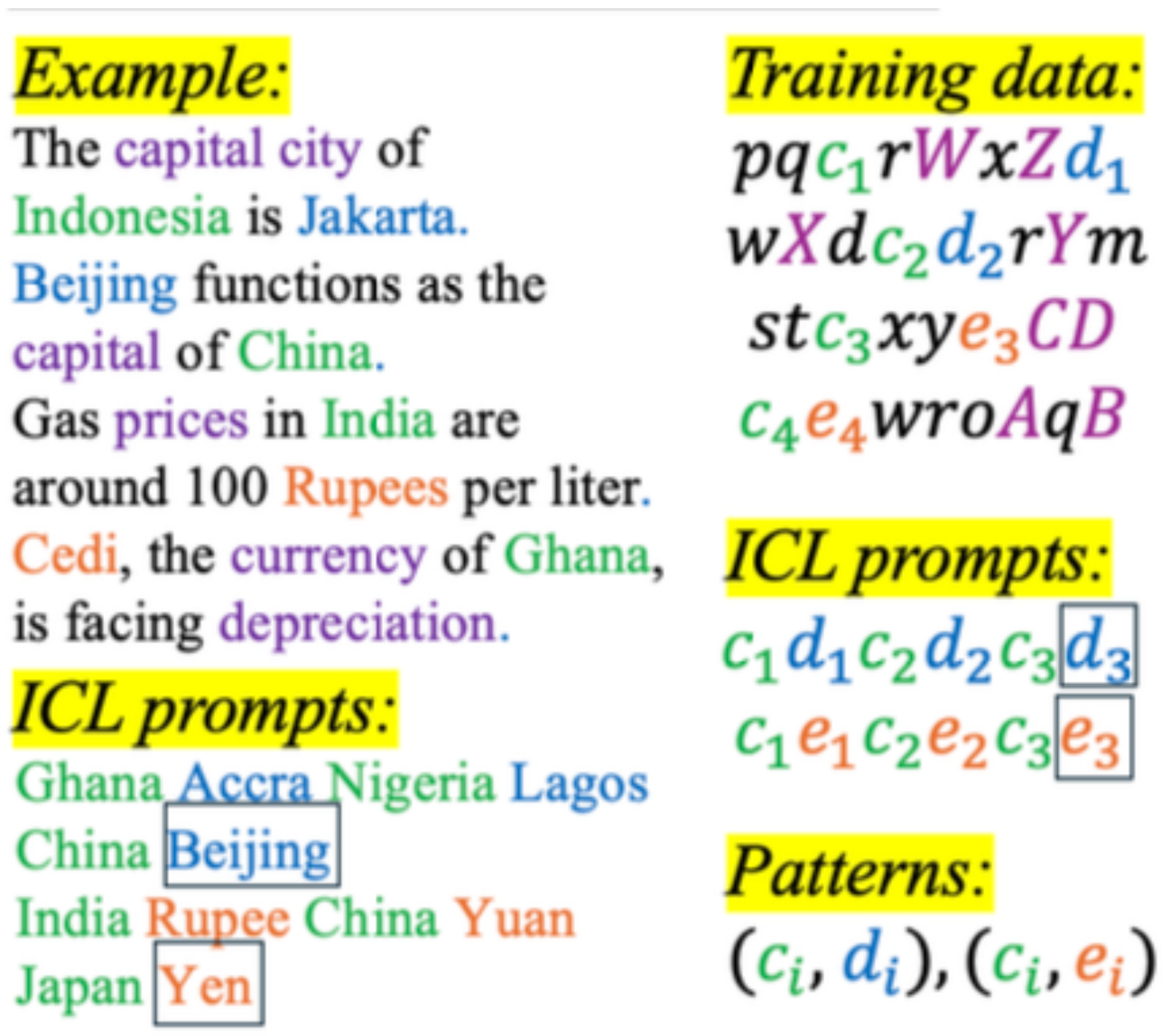
CBOW in Word2Vec can do in-context word analogy too

[Wibisono & W., NeurIPS 2024]

In-context word analogy

Co-occurrence modeling can induce in-context learning

- **Theorem (Informal)**
  - **Single relationship of interest** (e.g. country – capital city)
  - Vocabulary set: K countries and capital cities, L other words.
  - Each sentence (length S) contains one (country, capital city) pair.
  - Use continuous bag of words (CBOW) to model the sentences.
- **Then ICL succeeds as long as the number of examples is not greater than a bound that depends on K, L and S.**



[Wibisono & W., NeurIPS 2024]

In-context word analogy

Co-occurrence modeling can induce in-context learning

Theorem 1 (ICL on single-relationship word analogy tasks). *Let $K, L \geq S \geq 3$. Suppose each training sentence of length S is generated by selecting one (c_i, d_i) pair and $S-2$ distinct r_i's uniformly at random. We train a CBOW model with the squared loss and a sufficiently large embedding dimension on these sentences. Given a prompt $c_{i_1} d_{i_1} \cdots c_{i_\ell} d_{i_\ell} c_{i_{\ell+1}}$ with distinct i_k's, the model correctly predicts $d_{i_{\ell+1}}$ if and only if $2\ell + 1 < \frac{KL(S-1)^3}{(K+L)(S-2)^2(S-1)+K(S-2)(S-1)^2-2(S-2)^4}$.*

[Wibisono & W., NeurIPS 2024]

In-context word analogy

Co-occurrence modeling can induce in-context learning

- **Theorem (Informal)**
  - **Dual connected relationship** (e.g. country – capital city and currency)
    - Results also apply when both relationships are disjoint.
  - Vocabulary set: K countries, capital cities and currencies, L other words.
  - Two (country, capital city) or two (country, currency) pairs per sentence.

- **Then the model can identify each task from the in-context examples and ICL succeeds**

[Wibisono & W., NeurIPS 2024]

So…is co-occurrence modeling all of in-context learning?

Result 2: **Position information and nuisance structure matters**

ICL for logic reasoning tasks requires **proper modeling of positional information** and **blocked nuisance structure**

Positional information matter

Scenario 1 (clean):

- Training data: *aba cdc*
- ICL prompt: *aba cd?* (target = *c*)

ICL accuracy:

- No positional embeddings = 0%
- Learned positional embeddings = 100%
- Sinusoidal positional embeddings = 96%
- Rotary positional embeddings (RoPE) = 48%

[Wibisono & W., NeurIPS 2024]

Positional information matter

- **Theorem (Informal)**
  - Training sequence: $x_{i1}x_{i2}x_{i3}x_{i1}$
  - $x_{i1} \neq x_{i2} \neq x_{i3}$ randomly drawn from vocabulary
  - one-layer transformer model that, given $x_{i1}x_{i2}x_{i3}$, predicts the last x_{i1}
- **Impossible** to achieve zero loss if next token prediction does **not use positional information** $f(\{x_{i1}, x_{i2}\}, x_{i3})$
- **Possible** to achieve zero loss if next token prediction **uses** positional information $f(\{(x_{i1},1), (x_{i2},2)\}, (x_{i3},3))$

Result not specific to transformers!

Nuisance structures matter

Scenario 2 (one-noisy):

- Training data: *aba cdc*, with a nuisance token *n* inserted in between
  - anbancndc
  - abnacnndcn
- ICL prompt: *aba cd?* (target = *c*)

ICL accuracy:

- No positional embeddings = 0%
- Learned positional embeddings = 0%
- Sinusoidal positional embeddings = 10%
- Rotary positional embeddings (RoPE) = 0%

Example:

The capital city of Indonesia is Jakarta. Beijing functions as the capital of China. Gas prices in India are around 100 Rupees per liter. Cedi, the currency of Ghana, is facing depreciation.

[Wibisono & W., NeurIPS 2024]

Nuisance structures matter

Scenario 3 (block-noisy):

- Training data: *aba cdc*, with nuisance blocks $n_1n_2n_3$ inserted in between while preserving the *aba* and *cdc* blocks
  - aba$n_1n_2n_3$cdc
  - $n_4n_5n_6$abacdc
  - abacdc$n_7n_8n_9$
- ICL prompt: <u>*aba*</u> <u>*cdc*</u> <u>*ef?*</u> (target = <u>*e*</u>)

ICL accuracy:

- No positional embeddings = 0%
- **Learned positional embeddings = 100%**
- Sinusoidal positional embeddings = 55%
- Rotary positional embeddings (RoPE) = 3%

Example:

5,2,3,5. Predict the next number: 8,4,6,8. What number comes next? 2,1,3,2, 5,7,6,5. Guess the next number 1,2,3,2, 4,6,7,6. 2,9,7,9, 1,3,6,3. Extend the pattern please.

ICL prompts:

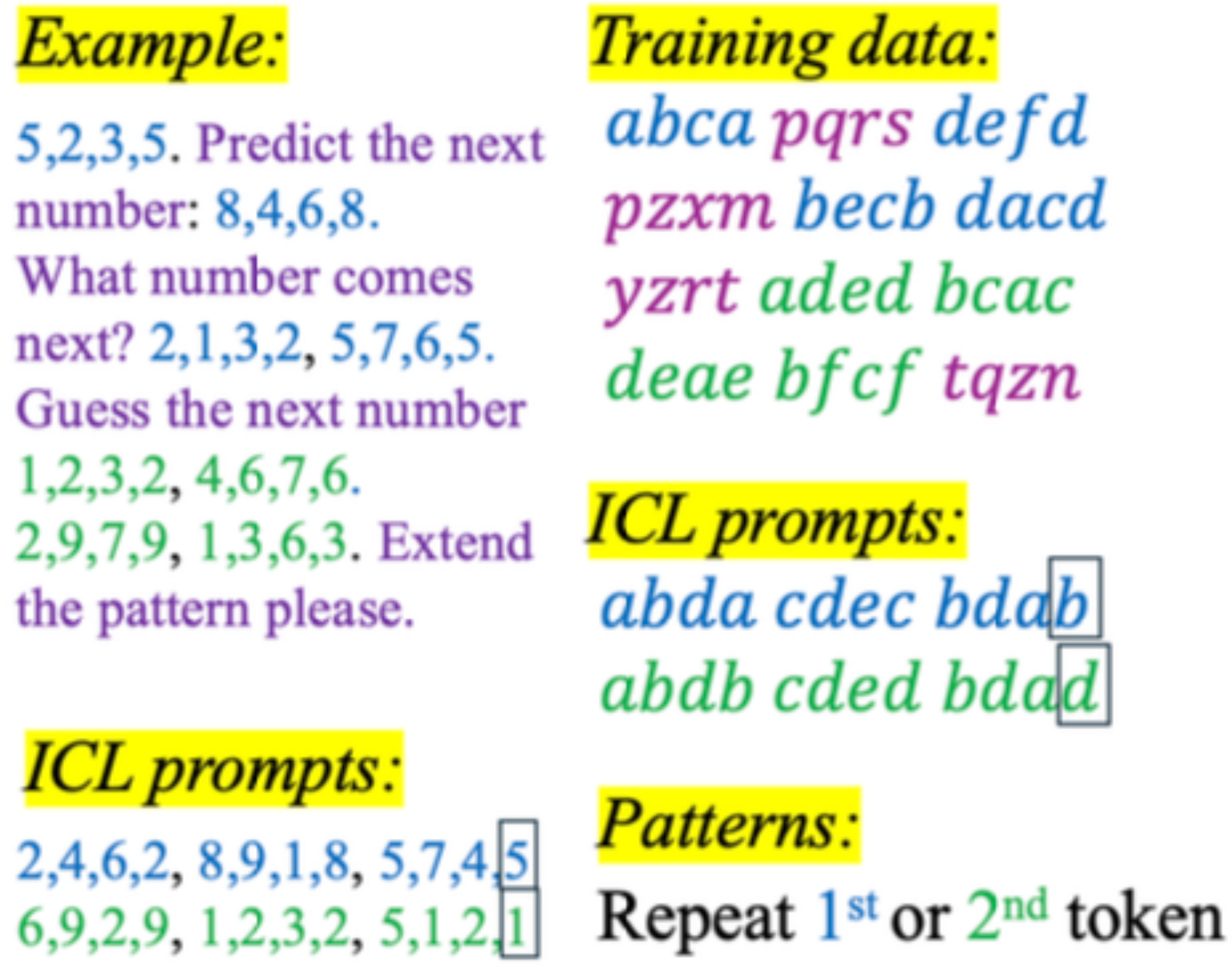
2,4,6,2, 8,9,1,8, 5,7,4,$\boxed{5}$
6,9,2,9, 1,2,3,2, 5,1,2,$\boxed{1}$

[Wibisono & W., NeurIPS 2024]

Nuisance structures matter

Blocked nuisance token structure facilitates ICL

- **Theorem (Informal)**
  - Sufficiently large autoregressive position-aware model that can achieve the minimum possible theoretical loss.
  - Training this model in the **one-noisy scenario** results in **zero** ICL accuracy.
  - Training this model in the **block-noisy scenario** results in **perfect** ICL accuracy.



[Wibisono & W., NeurIPS 2024]

Does in-context learning always work?

Result 3: LLMs require specific structures in the pre-training data to exhibit ICL ability

Failed scenario 1: Different repeating patterns

Training sentence: *aba cdc efe*

ICL prompts: *abb* *cdd* *ef?* (target = *f*)

Theorem (Informal):

ICL **fails** to output *f* irrespective of architectures.

This is related to the importance of **data diversity** for ICL

- See e.g., Raventos et al. (2023) and Yadlowsky et al. (2023).

In-context learning cannot reliably generalize to novel patterns.

[Wibisono & W., NeurIPS 2024]

Failed scenario 2: Pairs in fixed locations

Training sentences:

PEK is the airport code for Beijing.
ORD uniquely identifies the airport serving Chicago.
SIN denotes the primary airport for Singapore.
DEN is an airport located in Denver.

ICL prompts:

PEK Beijing ORD Chicago SIN $\boxed{\text{Singapore}}$

Theorem (Informal):
ICL **fails** irrespective of architectures.

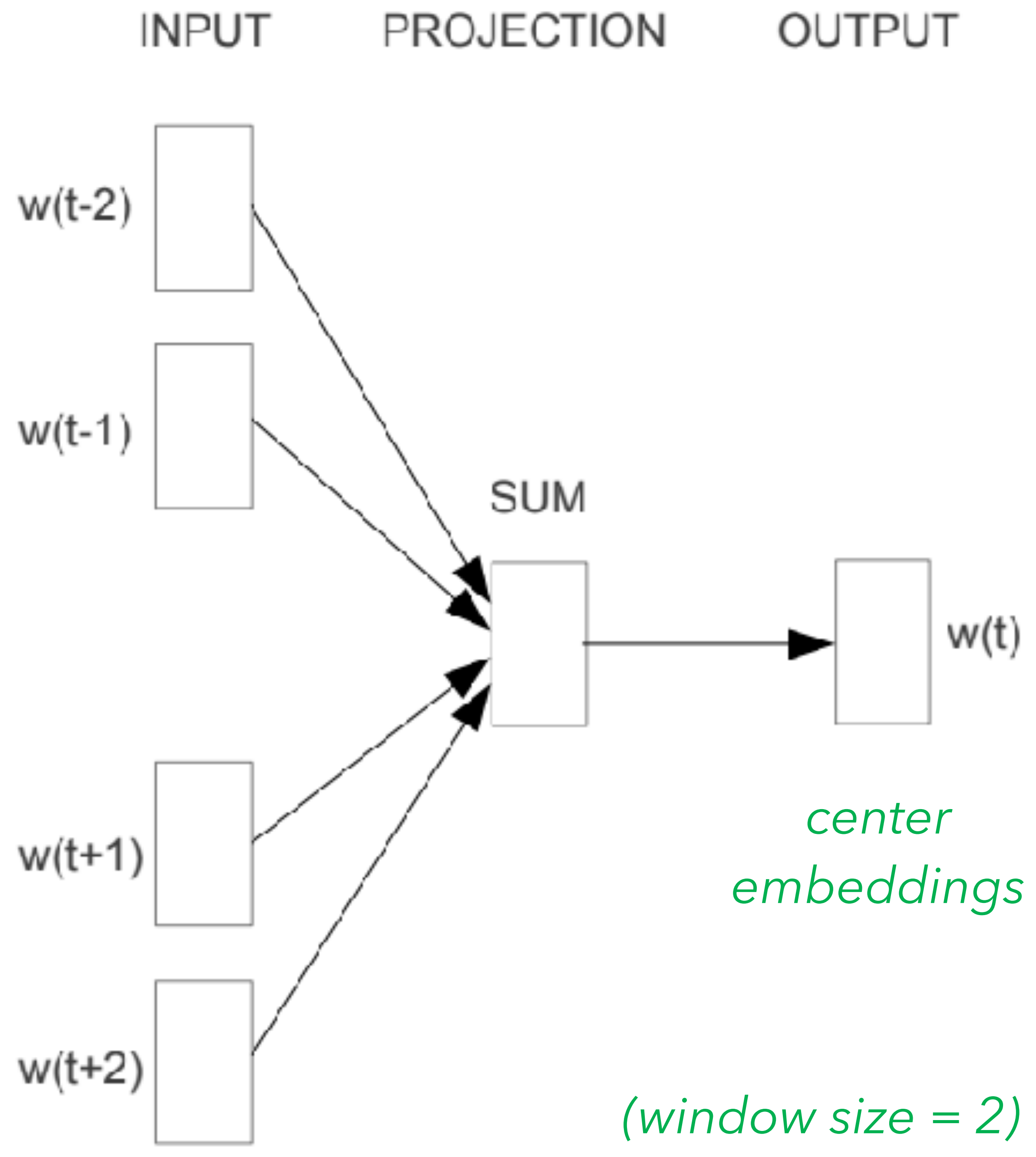
[Wibisono & W., NeurIPS 2024]

In-context learning is not specific to transformers.

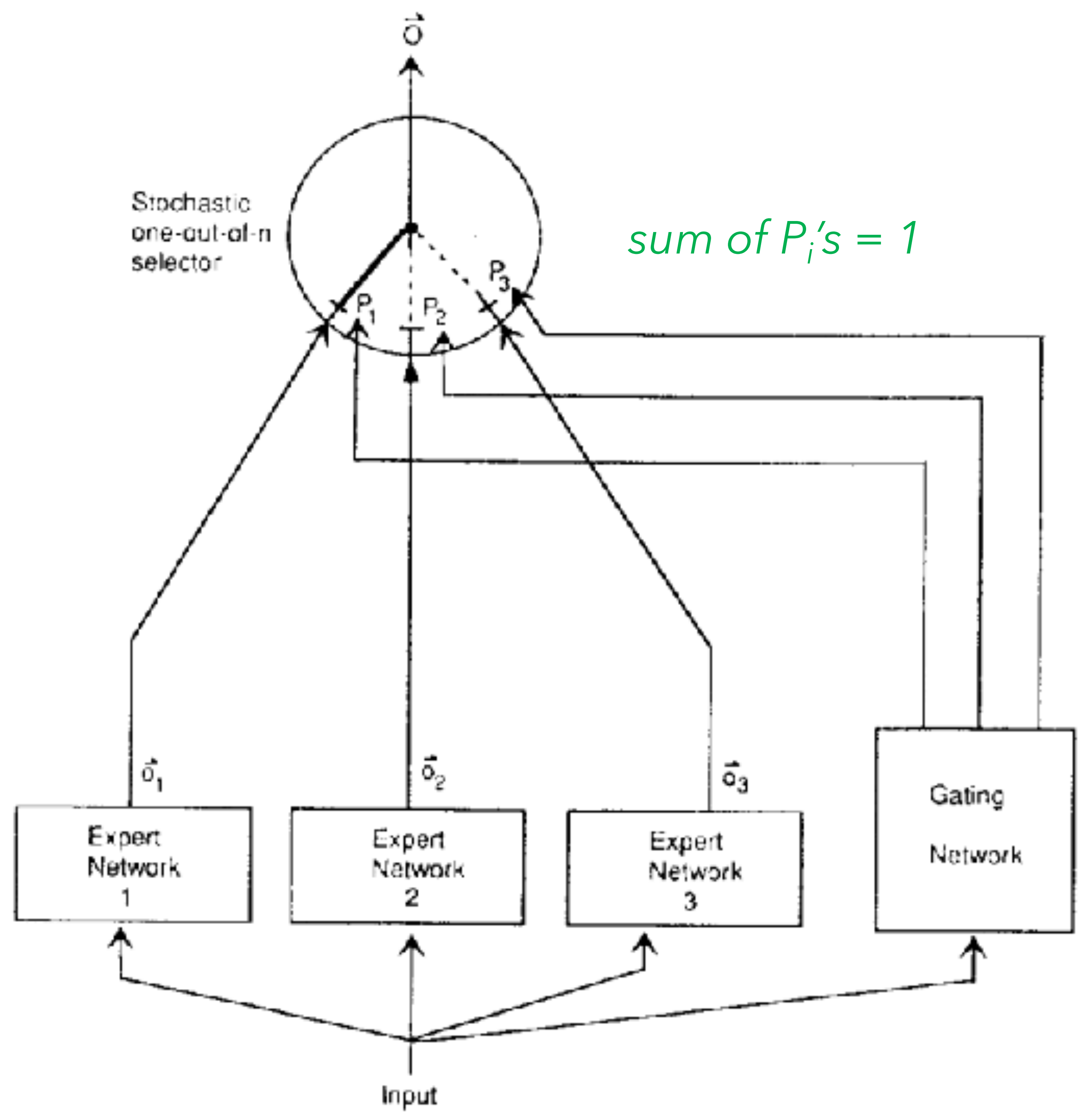
So, what is special about transformers?

Result: Transformer = CBOW + Mixture-of-experts

Bidirectional attention = CBOW + MoE weights



[Mikolov et al., 2013]



[Jacobs et al., 1991]

$$p(x_i = k \mid \{x_j\}_{j \neq i}) \propto \exp(u_k^\top \sum_{j \neq i} v_{x_j})$$

$$p(x_i = k \mid \{x_j\}_{j \neq i}) \propto \exp(\sum_k w_k f_j(\{x_j\}_{j \neq i}))$$

CBOW = Continuous bag of words

MoE = Mixture-of-experts

[Wibisono & W., UAI 2023]

Focus: Bidirectional attention

An encoder-only Transformer with three main ingredients:

- The self-attention mechanism
- Positional encodings
- The masked language model (MLM) objective

Example: Bidirectional Encoder Representations from Transformers (BERT)
[Devlin et al., 2019]

[Wibisono & W., UAI 2023]

First, what does *bidirectional* mean?

Task: Guess the *missing word*

The capital of North ? is Raleigh

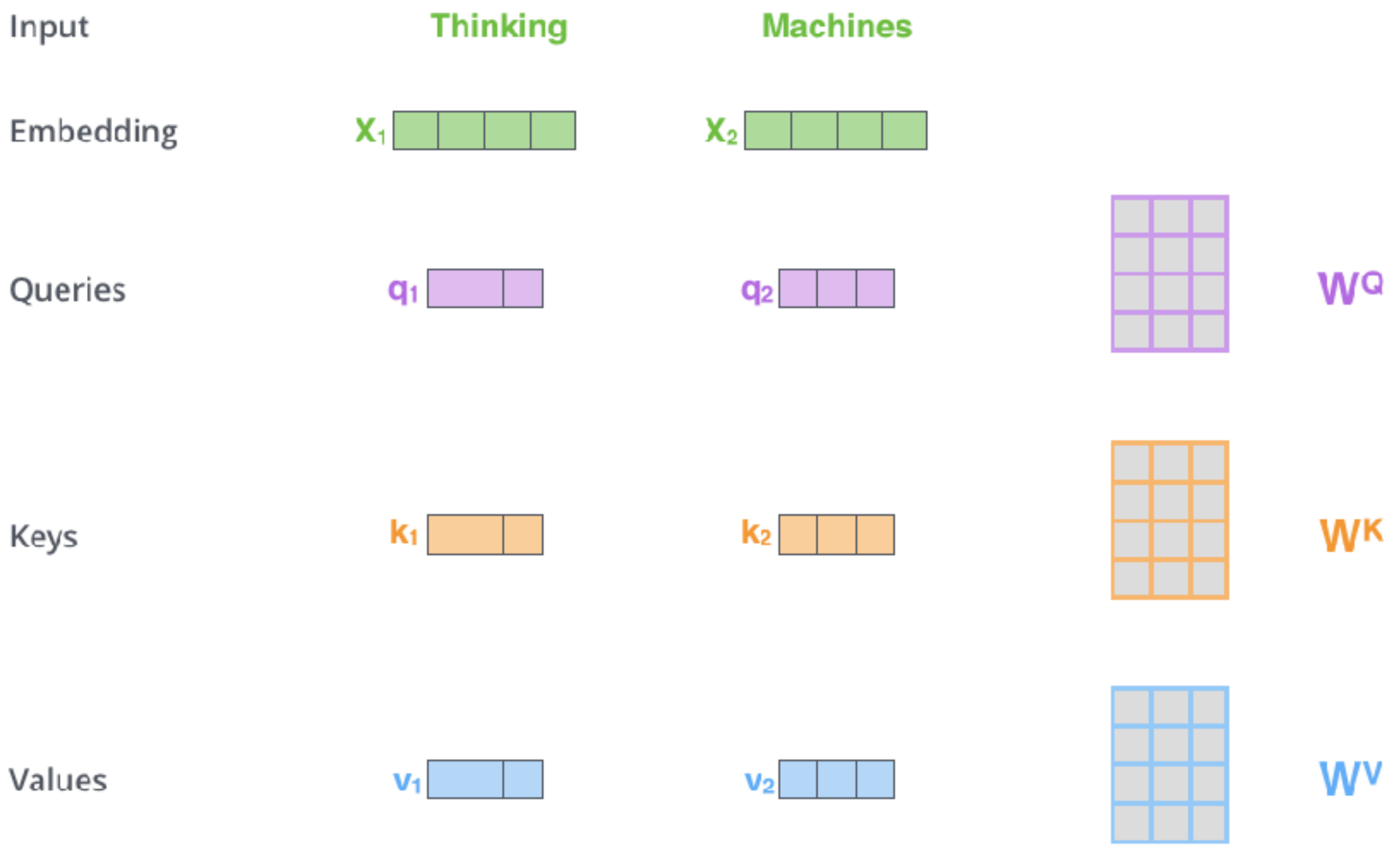
Bidirectional

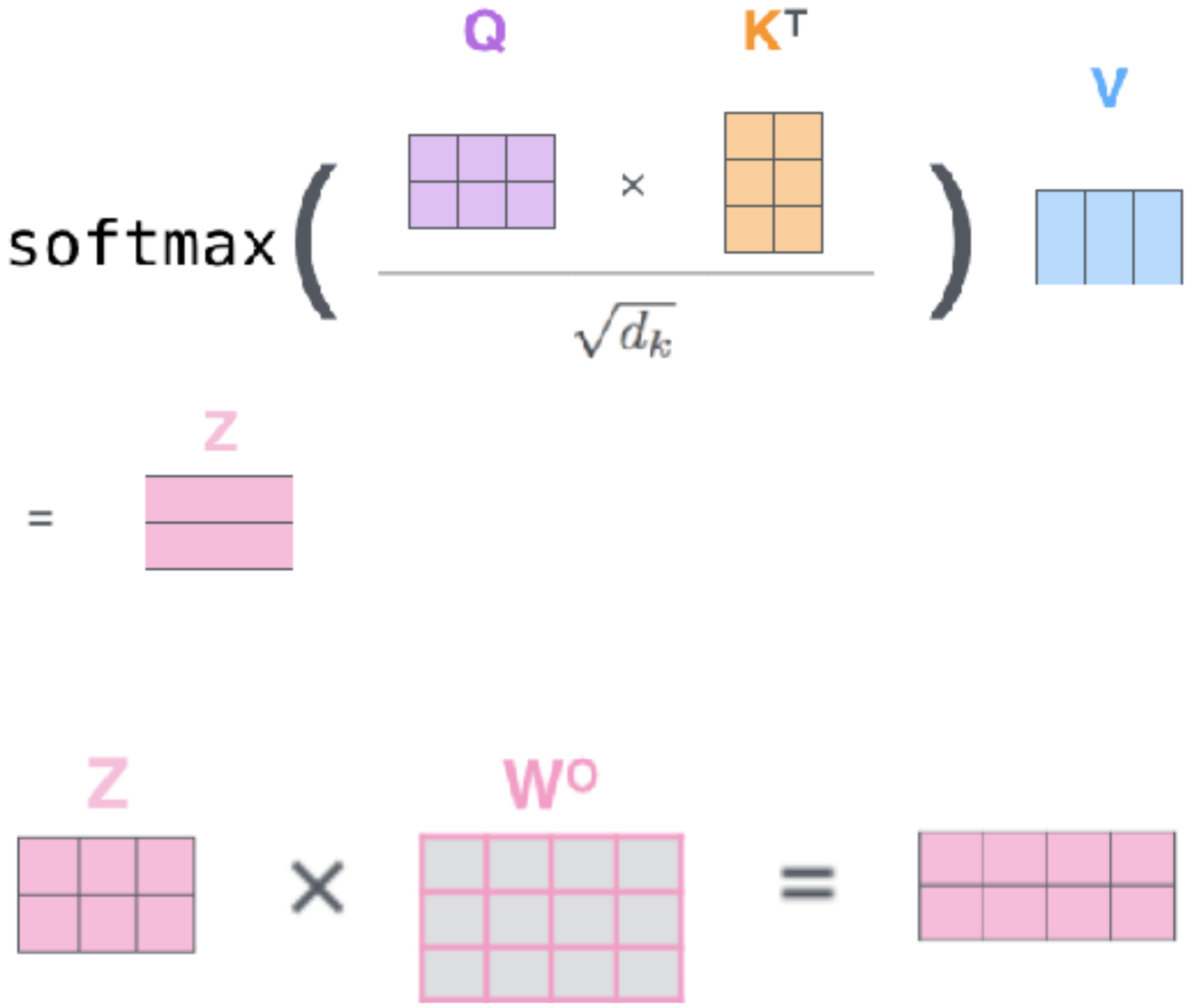
? = Carolina

CBOW is also bidirectional as it uses left and right contexts to predict the center word

Ingredient 1: Self-attention mechanism

Goal: To allow each word to "look at" other words in different positions





[Alammar, 2018]

Ingredient 2: Positional encodings

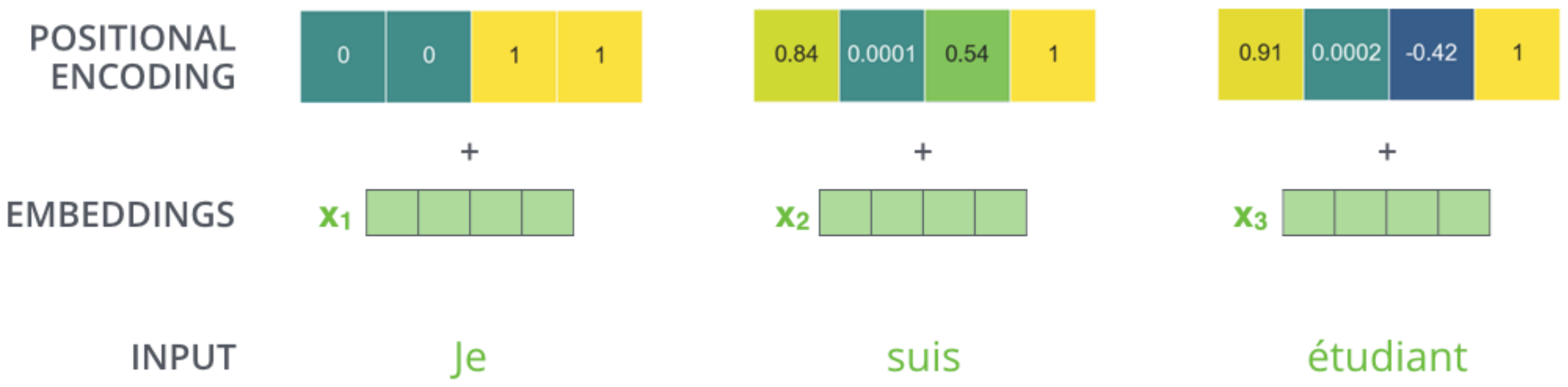
The self-attention mechanism is *order-invariant*

Input embedding: $[x_1, x_2, x_3]$

After self-attention: $[x_1', x_2', x_3']$

Input embedding: $[x_2, x_3, x_1]$

After self-attention: $[x_2', x_3', x_1']$



[Alammar, 2018]

Ingredient 3: The MLM objective

- A self-supervised learning objective

- *Goal:* Predict the randomly masked words via the cross-entropy loss

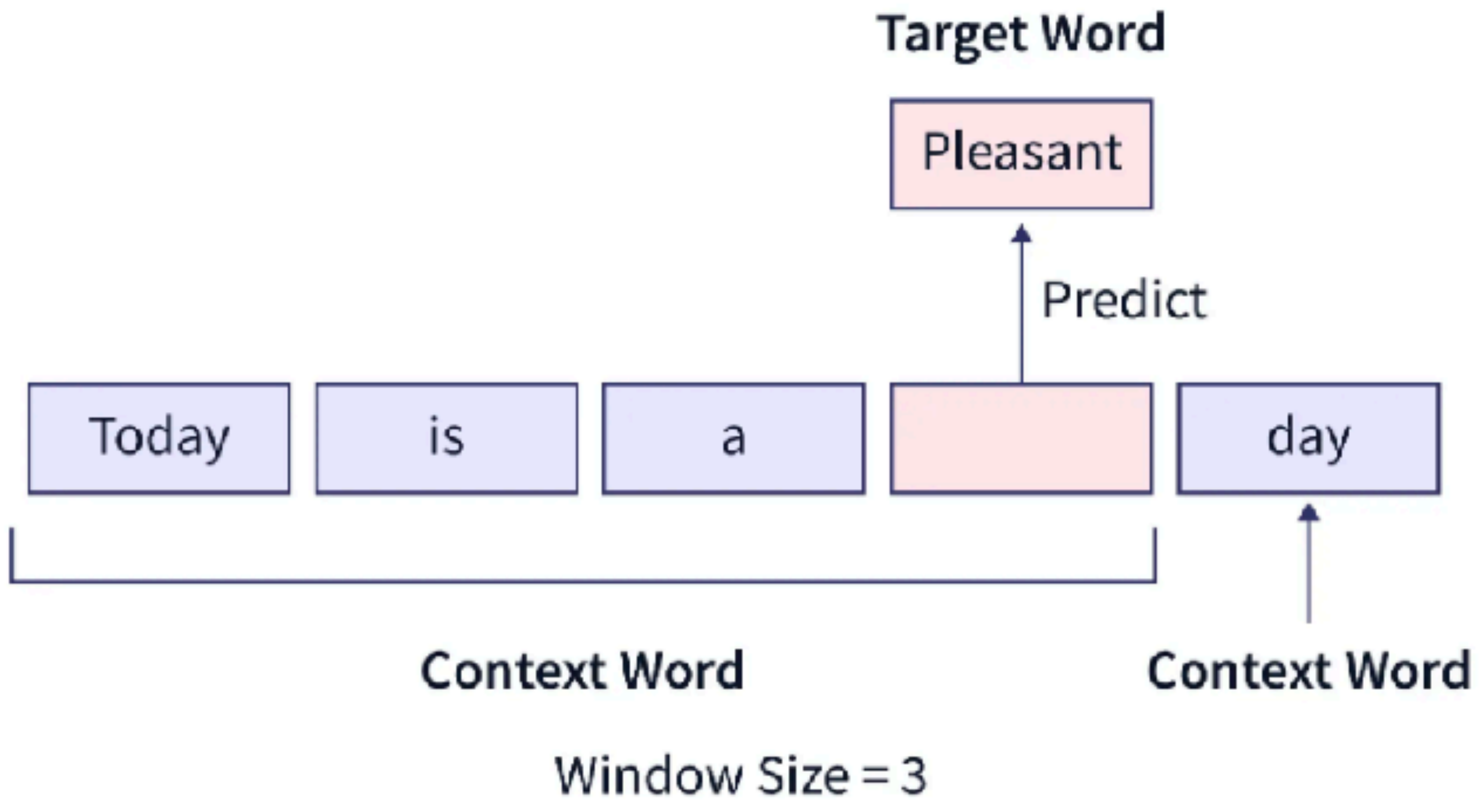
Input = *The capital of North* [MASK] *is Raleigh*

Target = Carolina

Masking strategy: mask one position at a time and aim to predict the original token

Continuous Bag-of-words (CBOW) in word2vec

- Continuous Bag-of-words (CBOW) in word2vec
- Close to a latent factor model
- Given a sequence $x_1, x_2, \ldots, x_I$,
  - $p(x_i = k \mid \{x_j\}_{j \neq i}) \propto \exp(u_k^\top \sum_{j \neq i} v_{x_j})$
  - u_w, v_w are center and context embedding of word w
- **Each word in the context serves as a predictor for the missing word, then aggregate**



[Wibisono & W., UAI 2023]

Theorem (Informal): Bidirectional Attention = CBOW + MoE

Both objectives can be written as $\mathrm{crossentropy}\left(\bar{y}, \mathrm{softmax}\left(F(\bar{X})\right)\right)$, where $F(\bar{X}) = \sum_{j=1}^{S} \pi_j(\bar{X}) f_j(\bar{X})$

Mixture-of-experts, with each position serving as an expert

CBOW

MLM

$$\pi_j(\bar{X}) \propto \mathbf{1}\left(1 \leq |j - m| \leq w\right)$$

"weight"

$$\pi_j(\bar{X}) \propto \exp\left(\frac{e_j^\top (\bar{X}C + P) W^{KQ} \left(c_{|V|+1} + P^\top e_m\right)}{\sqrt{d_w}}\right)$$

$$f_j(\bar{X}) = W(\bar{X}C)^\top e_j$$

"similarity/co-occurrence"

$$f_j(\bar{X}) = W(\bar{X}C + P)^\top e_j + g + De_m$$

Notations: $(\bar{X}, \bar{y})$: input-output pairs (one-hot encoded), w: window size,
$m \in [S]$: masked position (S: sentence length), $b \in [|V|]$: masked token ($|V|$: vocabulary size)

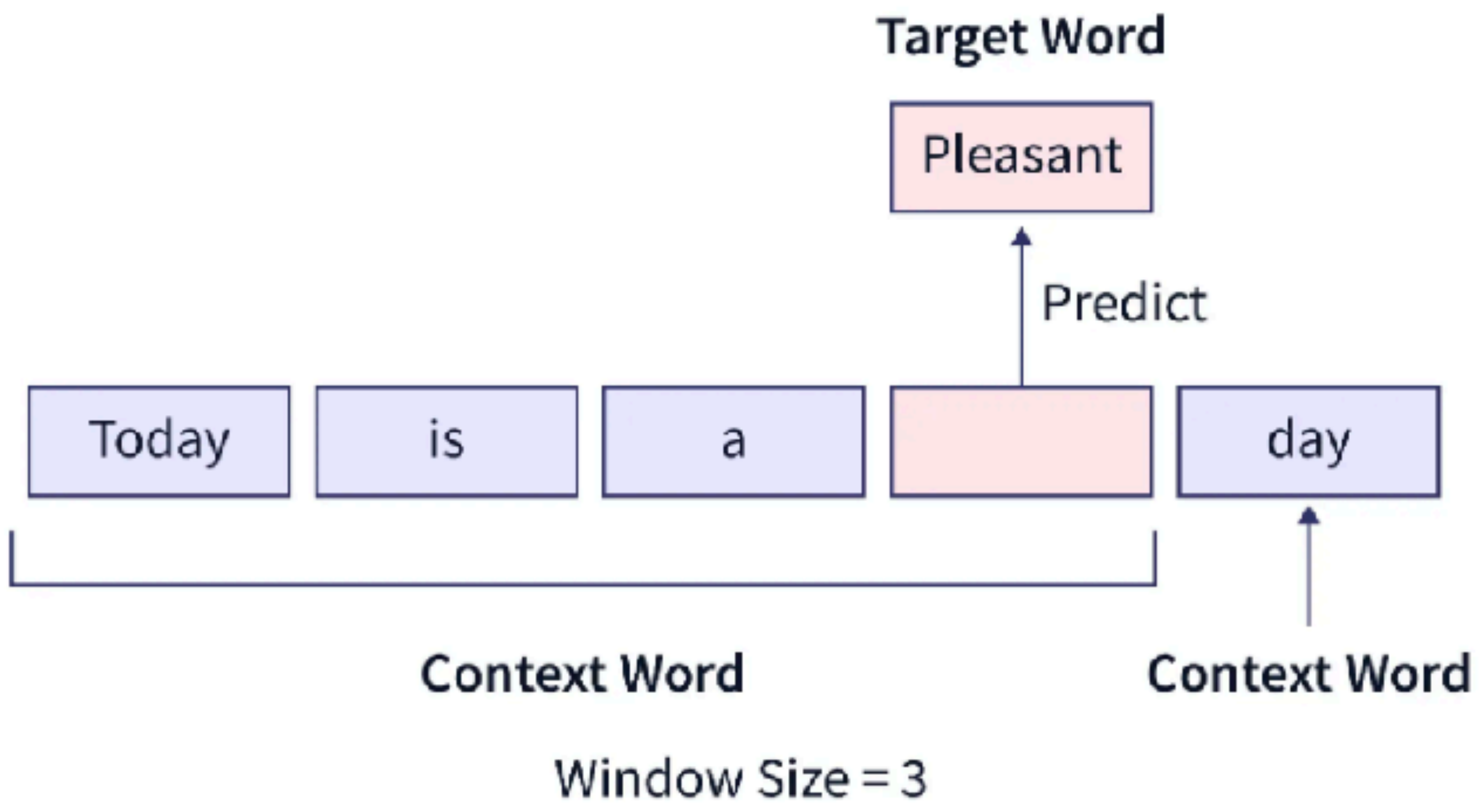
Focus on 1-head 1-layer bidirectional attention (BERT minus layer norm and MLP feed-forward layers)

W = center embeddings
C = context embeddings

Each word in the context is an expert predictor

(Extensible to Unidirectional, Multi-layer, Multihead)

Theorem (Informal): Bidirectional Attention = CBOW + MoE

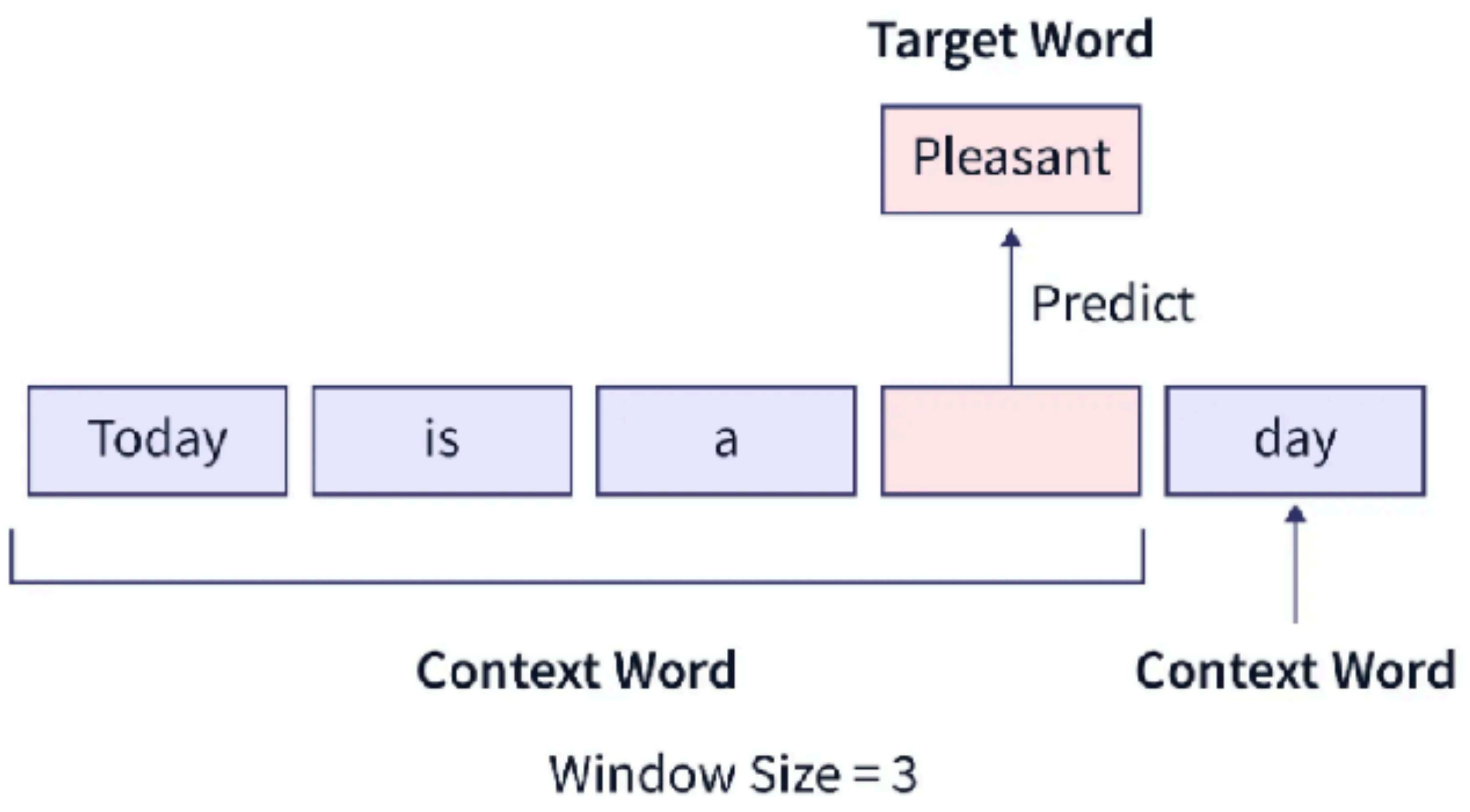


$$p(x_i = k \mid \{x_j\}_{j \neq i}) \propto \exp(u_k^\top \sum_{j \neq i} v_{x_j})$$

- **Sequence modeling:**
  - **Latent factor modeling** of **cooccurrence** between center word against each word in the neighborhood (e.g. left word, right word)
- **Attention** captures **heterogeneous** co-occurrence relationships
  - **Co-occurrence** between a center word and a context word depends on what **other words are in the context**

Theorem (Informal): Bidirectional Attention = CBOW + MoE

- Suggests an immediate extension to tabular and non-natural language sequential data



A tabular data extension

1	5	3	9
2	10	4	1
7	8	3	6
5	2	6	?

$C = 10;\ K = 3$

Classification problem
Features (K-dim) and response are C-class ordinal

Task: Learn the joint distribution $p\left(x_{1:K}, y\right)$ — instead of $p\left(y \mid x_{1:K}\right)$ — via MLM

At training time,

1. 5 3 9 → $w_1 \;\; w_5 \;\; w_{\text{MASK}} \;\; w_9$

Given the masked sentence, the goal is to predict

$w_{\text{MASK}} = w_3$

$w_{1:10},\ w_{\text{MASK}}$: *word embeddings*

$p_{1:4}$: *positional encodings*

Tabular extension improves OOD generalization

Param. \ **Acc.**	LR	RF	GB	MLP	ATN
$(1,0,0.1)$	0.388	0.409	**0.413**	0.323	0.404
$(1,0,0.9)$	0.313	0.298	0.350	0.237	**0.389**
$(1,0.5,0.1)$	0.345	0.361	**0.366**	0.292	0.359
$(1,0.5,0.9)$	0.270	0.253	0.299	0.202	**0.306**
$(1,1.5,0.1)$	0.250	0.243	**0.253**	0.204	0.252
$(1,1.5,0.9)$	0.169	0.158	**0.172**	0.142	0.170
$(5,0,0.1)$	0.250	0.207	0.244	0.306	**0.419**
$(5,0,0.9)$	0.162	0.150	0.156	0.169	**0.392**
$(5,0.5,0.1)$	0.227	0.173	0.214	0.252	**0.318**
$(5,0.5,0.9)$	0.154	0.133	0.153	0.151	**0.269**
$(5,1.5,0.1)$	0.167	0.099	0.157	0.165	**0.171**
$(5,1.5,0.9)$	0.125	0.108	0.114	0.118	**0.133**

(a) Accuracy

Param. \ **MSE**	LR	RF	GB	MLP	ATN
$(1,0,0.1)$	3.015	**2.694**	2.730	4.059	2.941
$(1,0,0.9)$	5.163	9.331	4.855	7.911	**3.078**
$(1,0.5,0.1)$	3.416	3.201	**3.123**	4.704	3.281
$(1,0.5,0.9)$	5.955	10.106	6.070	8.123	**4.465**
$(1,1.5,0.1)$	5.725	5.685	**5.415**	7.199	5.594
$(1,1.5,0.9)$	8.942	12.340	9.837	9.874	**7.339**
$(5,0,0.1)$	5.333	8.498	5.967	2.814	**1.521**
$(5,0,0.9)$	5.674	10.101	8.858	7.842	**1.633**
$(5,0.5,0.1)$	6.021	10.236	6.844	4.056	**2.605**
$(5,0.5,0.9)$	6.118	10.427	8.283	7.884	**2.355**
$(5,1.5,0.1)$	9.159	16.154	9.538	**8.313**	8.316
$(5,1.5,0.9)$	8.410	10.409	10.110	9.966	**6.501**

(b) MSE

- Simulated data sets (scores are averaged across 20 different seeds)
- (•, •, 0.9) corresponds to a more extreme case of covariate shift
- Significant improvement over MLP; more robust to covariate shifts

Tabular extension improves OOD generalization

	LR	RF	GB	MLP	CE	FT	TT	AI	TN	ATN (ours)
Accuracy	0.657	0.721	0.657	0.700	0.764	0.707	0.707	0.364	0.600	**0.793**
MSE	0.343	0.279	0.343	0.300	0.236	0.293	0.293	0.636	0.486	**0.207**

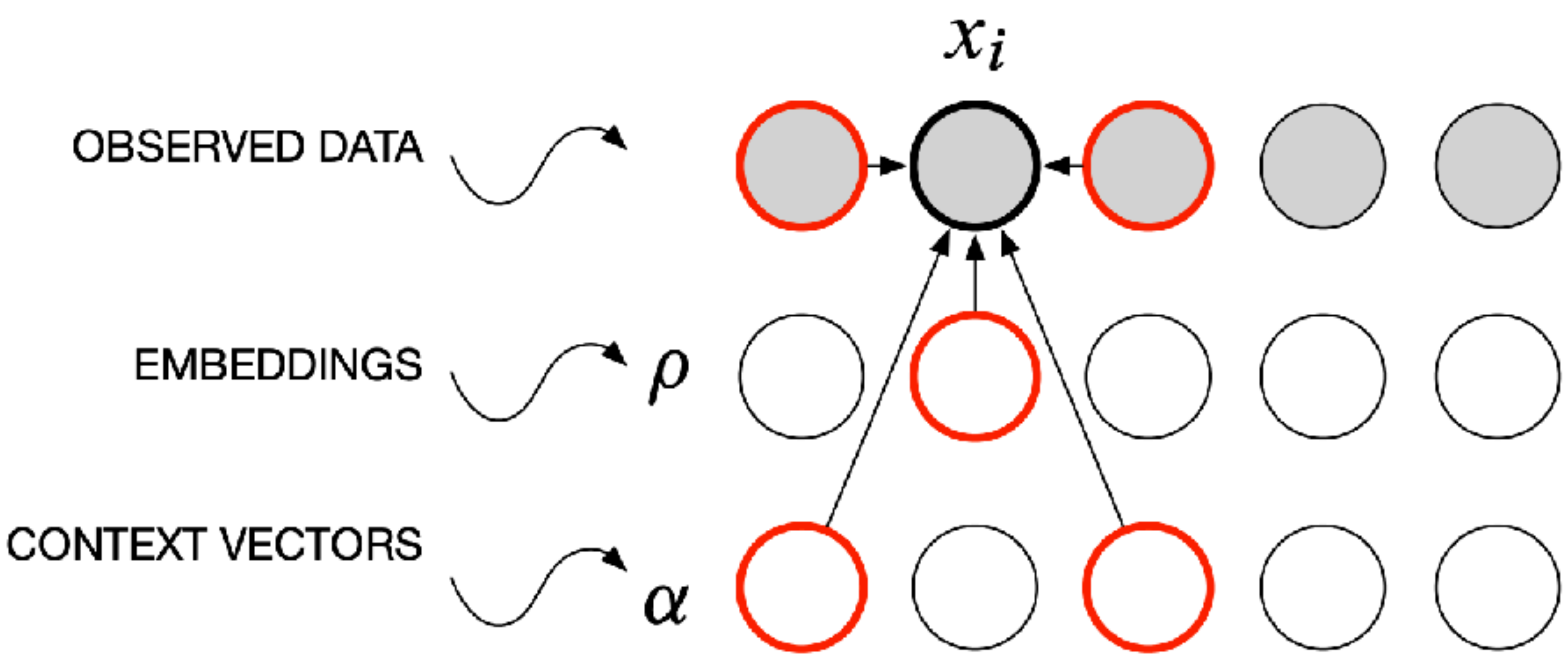
- *Goal:* predict mpg (*auto-mpg* data set)
- Each feature and response is split into 3 quantile-based categories
- *Covariate shift:* assign origin 1 to training set, origins 2 and 3 to test set [Sugiyama and Storkey, 2006]

CE = CategoryEmbedding (Joseph, 2021); FT = FTTransformer (Gorishniy et al., 2021); TT = TabTransformer (Huang et al., 2020); AI = AutoInt (Song et al., 2019); TN = TabNet (Arik and Pfister, 2021)

Differences: our approach models the joint data distribution and shares word and positional embeddings

Exponential family attention (EFA)

- Generalizes attention to other types of data via exponential families
  - *Text:* x_i = word in i^{th} position
  - *Market basket:* $x_{(n,t)}$ = number of items n purchased in basket t
  - *Item recommendation:* $x_{(n,t)}$ = rating of item n by user t



Each data point is modeled based on its context

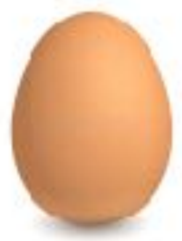
$$x_i \mid x_{c_i} \sim \mathrm{ExpFam}(\eta_i\left(x_{c_i}\right), t(x_i))$$

Natural parameter:

$$\eta_i\left(x_{c_i}\right) = f_i\left(\rho[i]^\top \sum_{j \in c_i} \alpha[j] x_j\right)$$

Modeling market basket data

N = number of items, T = number of baskets

	B_1	B_2	B_3
	3	1	10
	0	1	0
	2	1	0
	0	0	1

$N = 4, \;\; T = 3$

Attention yields better item embeddings

- *Subsetted Instacart data:* ~26.5K baskets, 63 most popular items
- Each basket contains at least 10 items
- Test set pseudo-likelihood:
  - Vanilla Bernoulli model: -0.402
  - Plus positional encoding: -0.363
  - Plus positional encoding and self-attention: -0.047

Takeaways

- In-context learning is **not specific to transformer**.
  - Can replicate it with **continuous-bag-of-words (CBOW)** in word2vec (latent-factor modeling)
- **Transformer = CBOW + (stacked) mixture-of-experts**
- It helps statistical modeling
  - Extend to **non-natural-language sequence** data and **tabular** data

Thank you!

- K.C. Wibisono and Y. Wang
  From Unstructured Data to In-Context Learning: Exploring What Tasks Can Be Learned and When
  NeurIPS 2024
  https://github.com/yixinw-lab/icl-unstructured

- K.C. Wibisono and Y. Wang
  Bidirectional Attention as a Mixture of Continuous Word Experts
  Uncertainty in Artificial Intelligence, 2023
  https://github.com/yixinw-lab/attention-uai

- K.C. Wibisono and Y. Wang
  On the Role of Unstructured Training Data in Transformers' In-Context Learning Capabilities
  NeurIPS 2023 Workshop on Mathematics of Modern Machine Learning