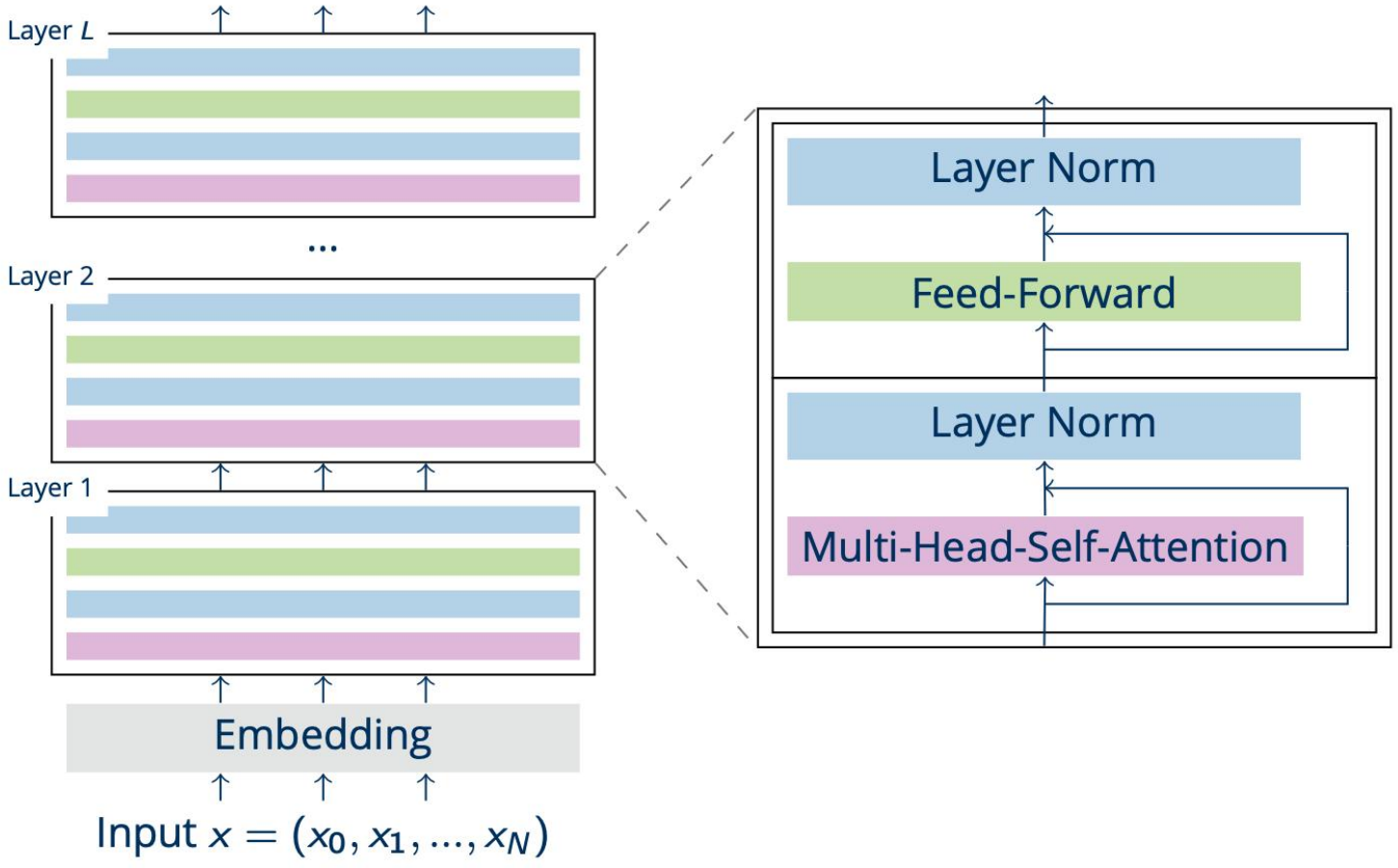
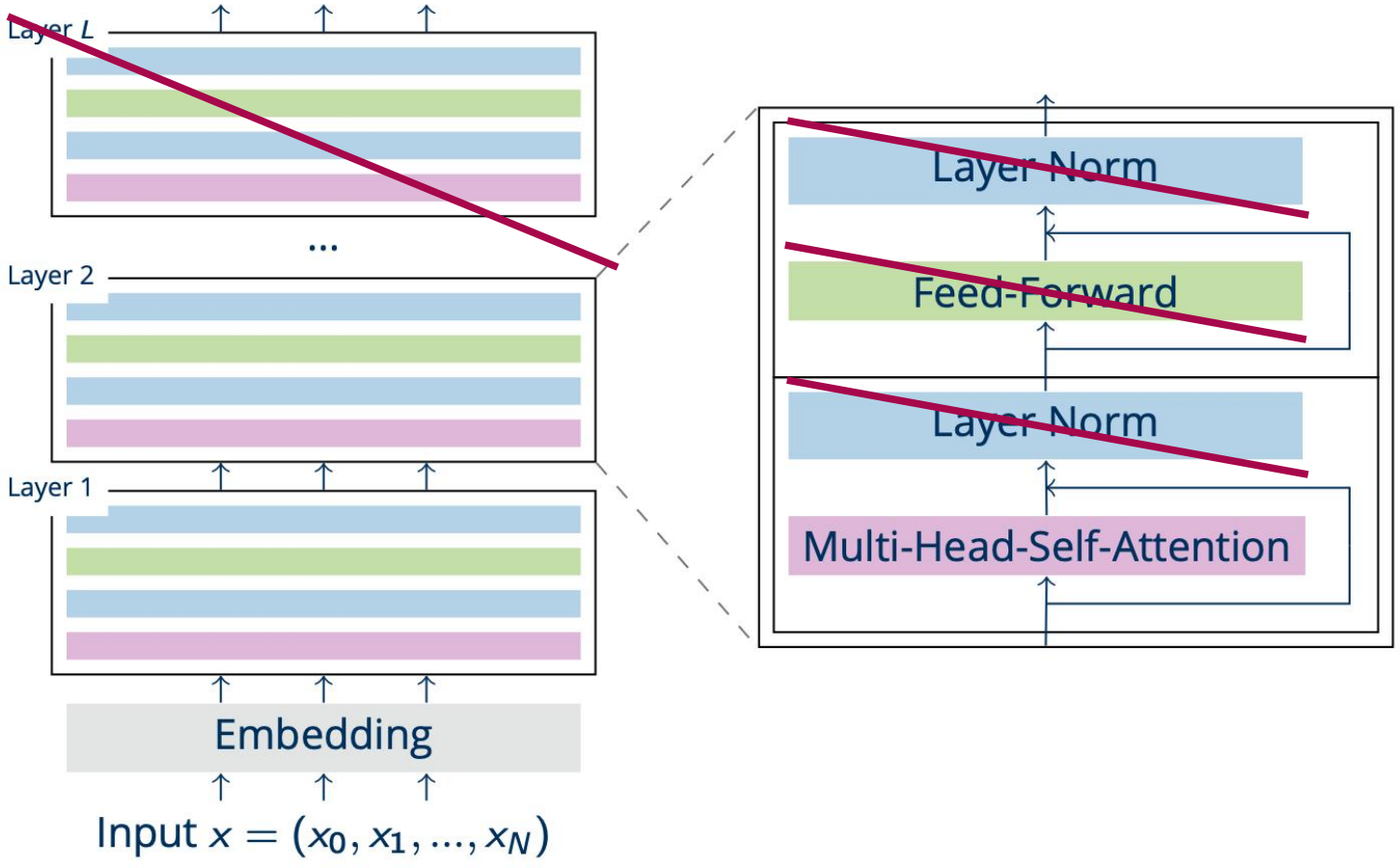
A Mathematical Framework for Transformer Circuits

ChatGPT Reading Group – 27.11.2023

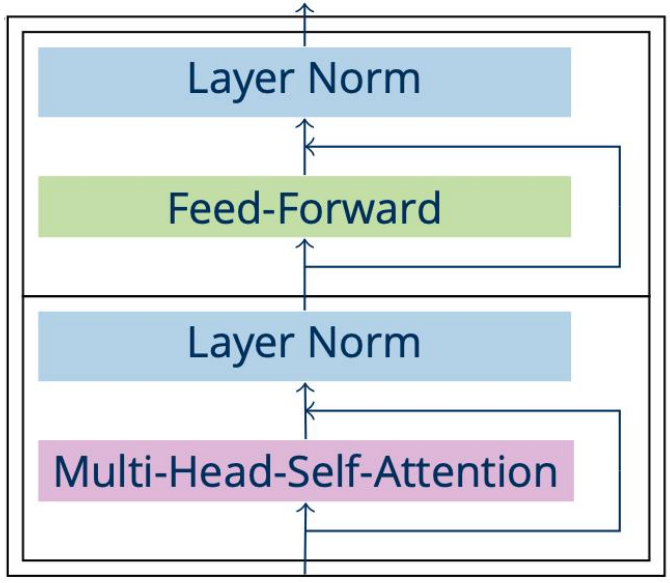
Transformer Architecture

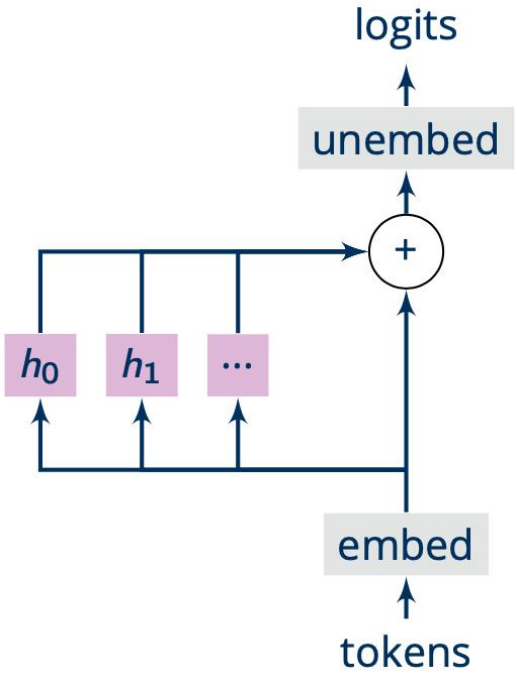


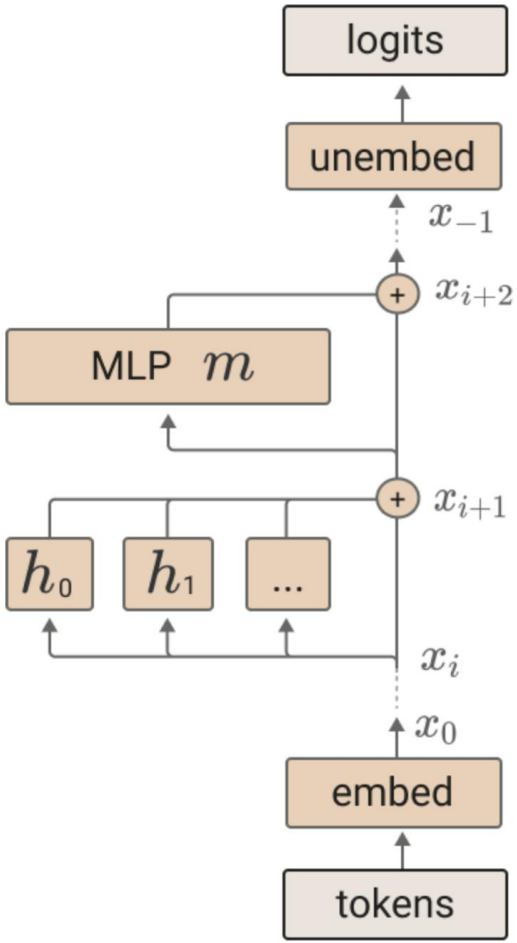
Transformer Architecture



Residual Stream







The final logits are produced by applying the unembedding.

$$T(t) = W_U x_{-1}$$

An MLP layer, m, is run and added to the residual stream.

$$x_{i+2} = x_{i+1} + m(x_{i+1})$$

Each attention head, h, is run and added to the residual stream.

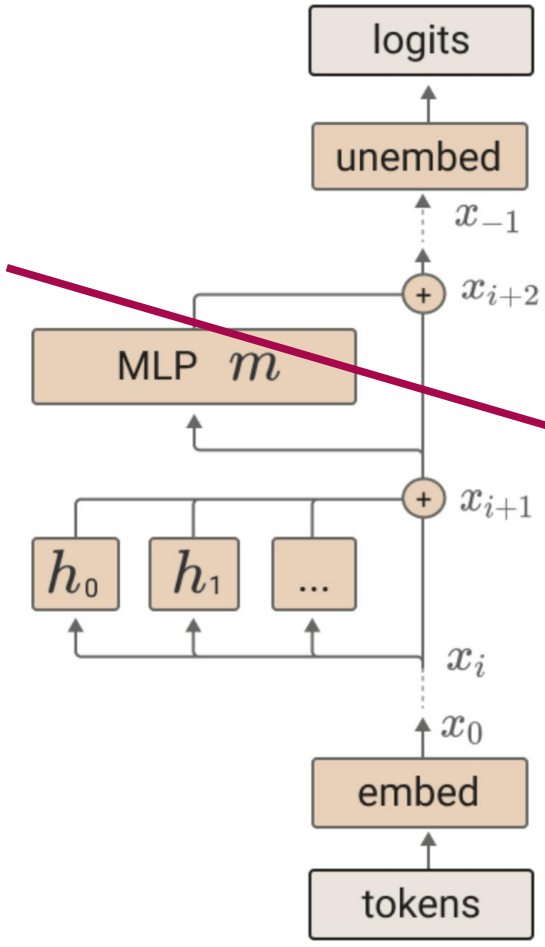
$$x_{i+1} = x_i + \sum_{h \in H_i} h(x_i)$$

One residual block

Token embedding.

$$x_0 = W_E t$$

0-Layer Transformer



The final logits are produced by applying the unembedding.

$$T(t) = W_U x_{-1}$$

An MLP layer, m, is run and added to the residual stream.

$$x_{i+2} = x_{i+1} + m(x_{i+1})$$

Each attention head, h, is run and added to the residual stream.

$$x_{i+1} = x_i + \sum_{h \in H_i} h(x_i)$$

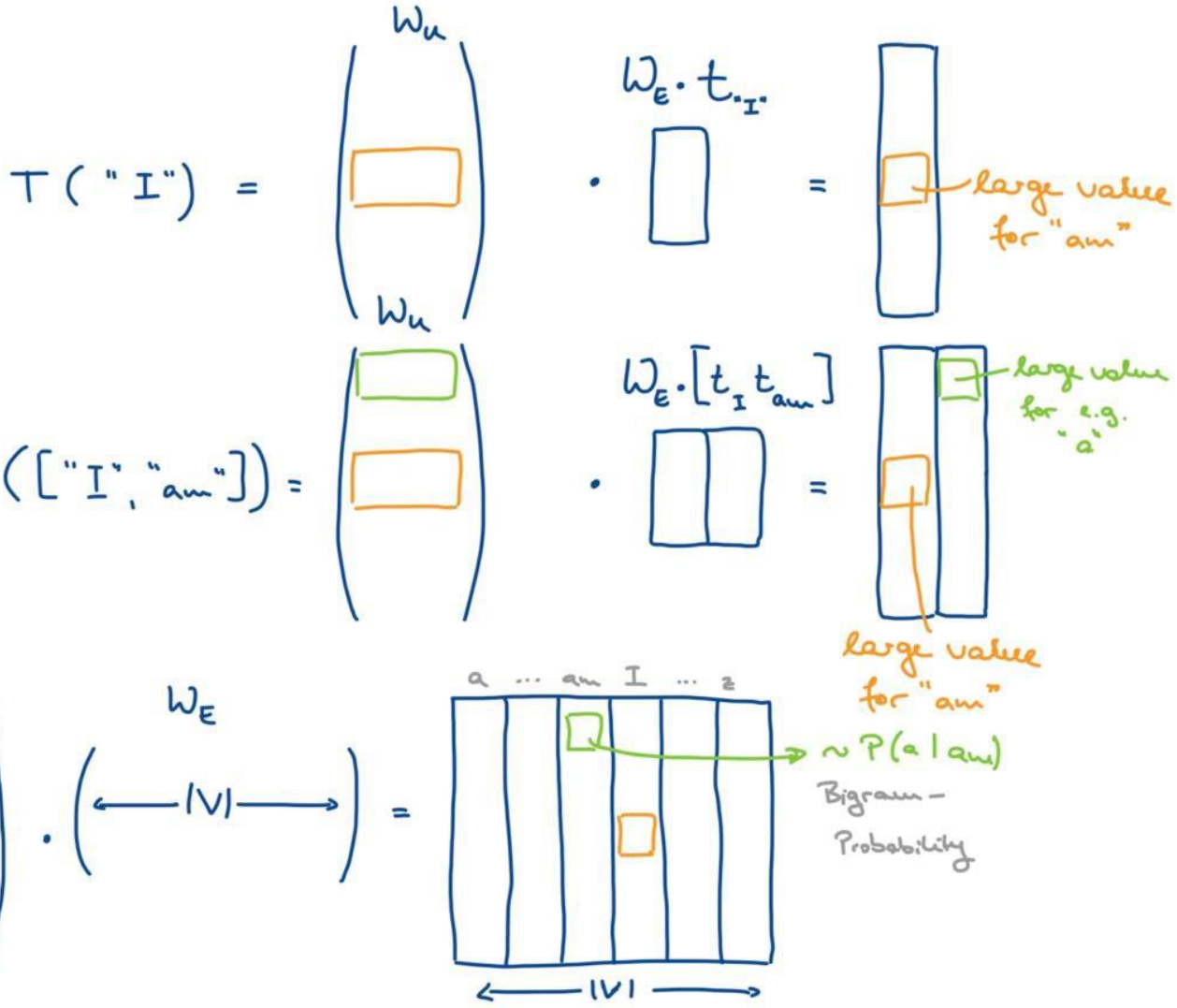
One residual block

Token embedding.

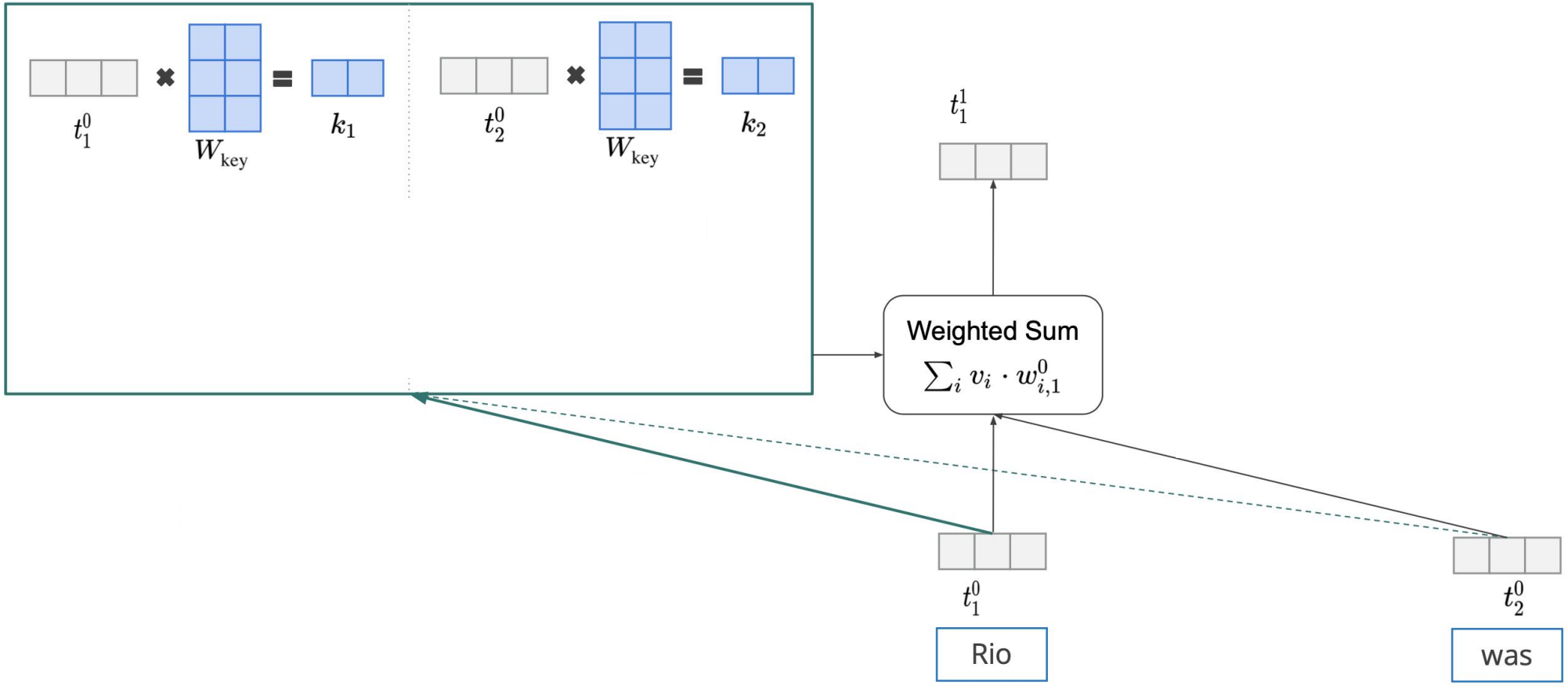
$$x_0 = W_E t$$

0-Layer

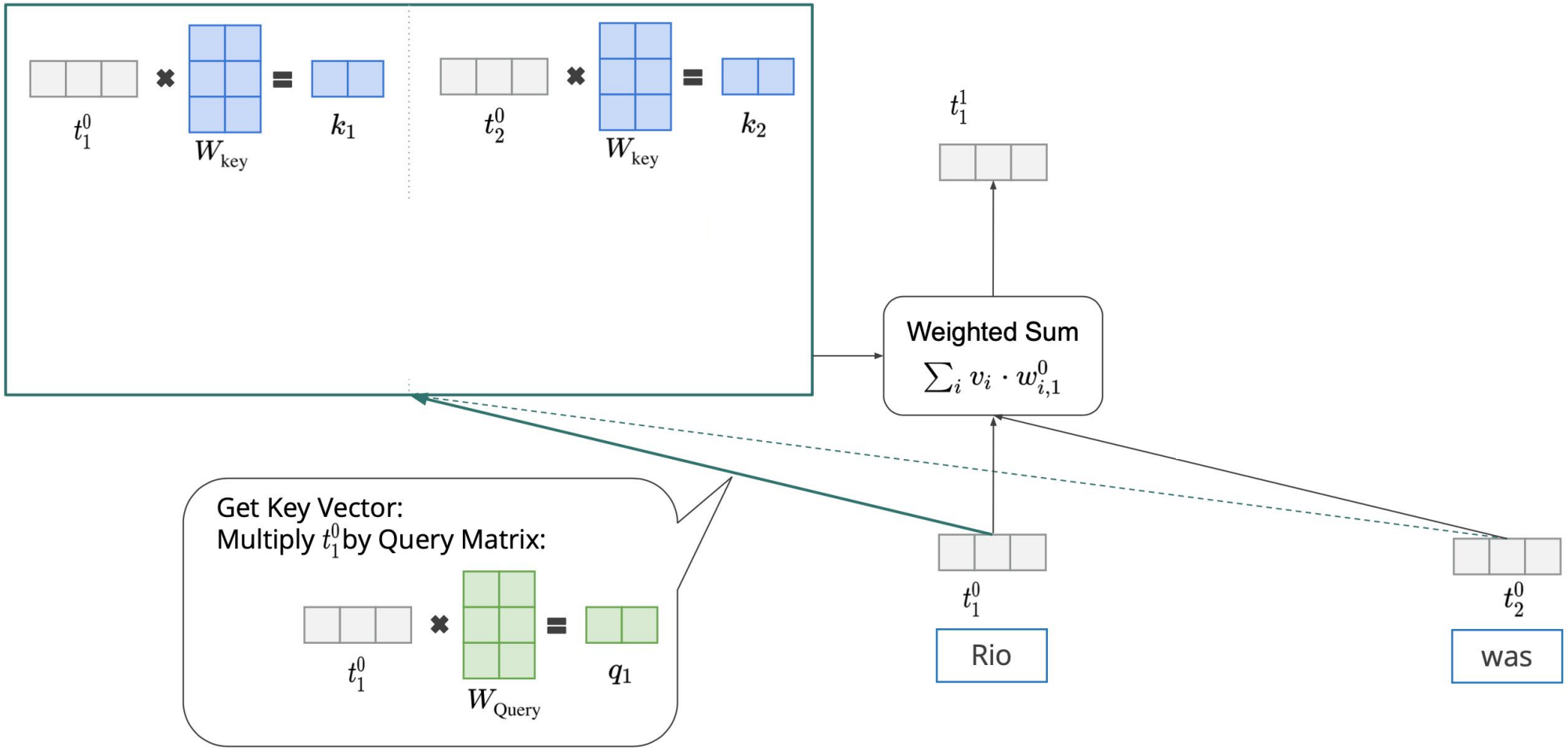
$$x_0 = W_E \cdot t \qquad T(t) = W_u x_0 \quad \Rightarrow \quad T(t) = W_u W_E \cdot t$$



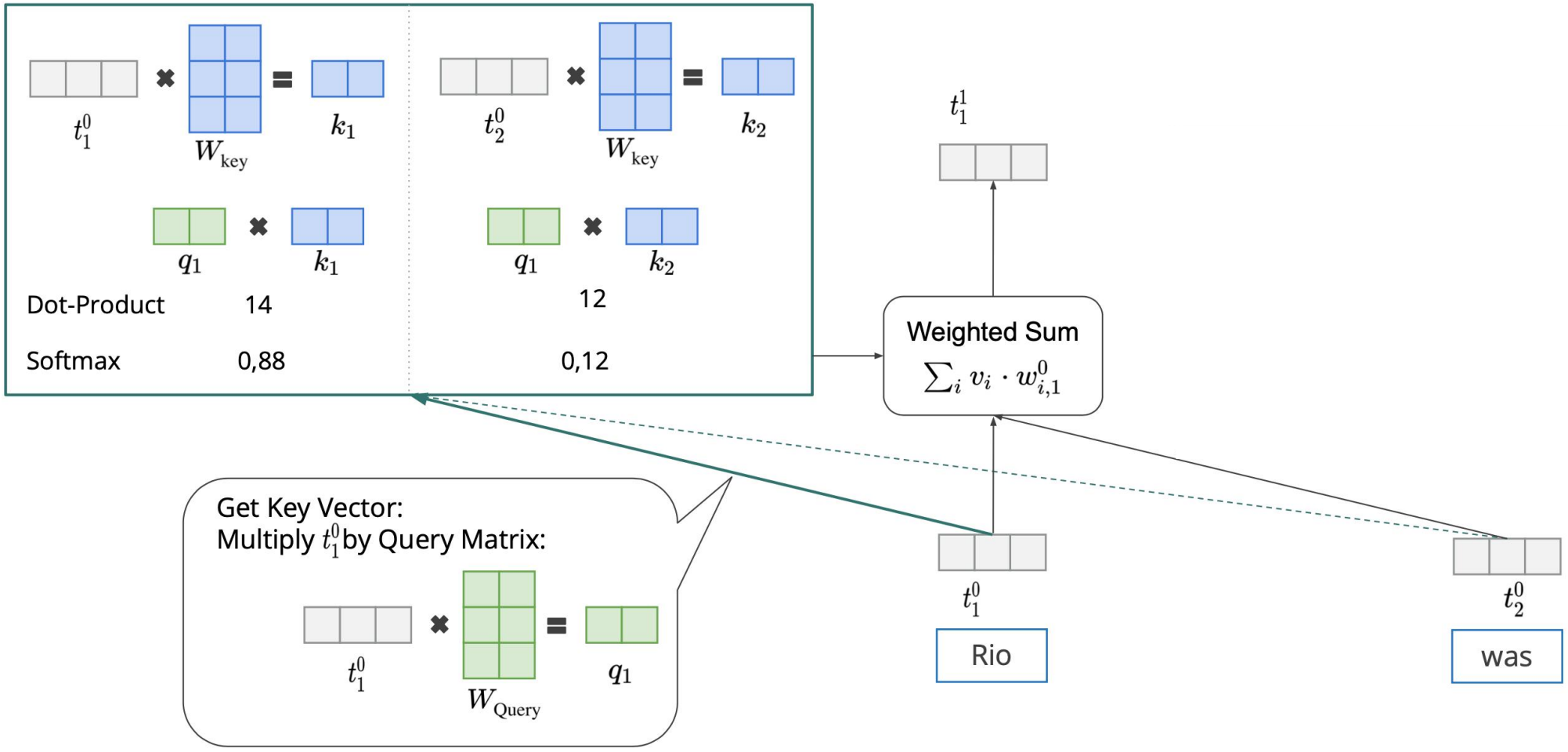
Self-Attention



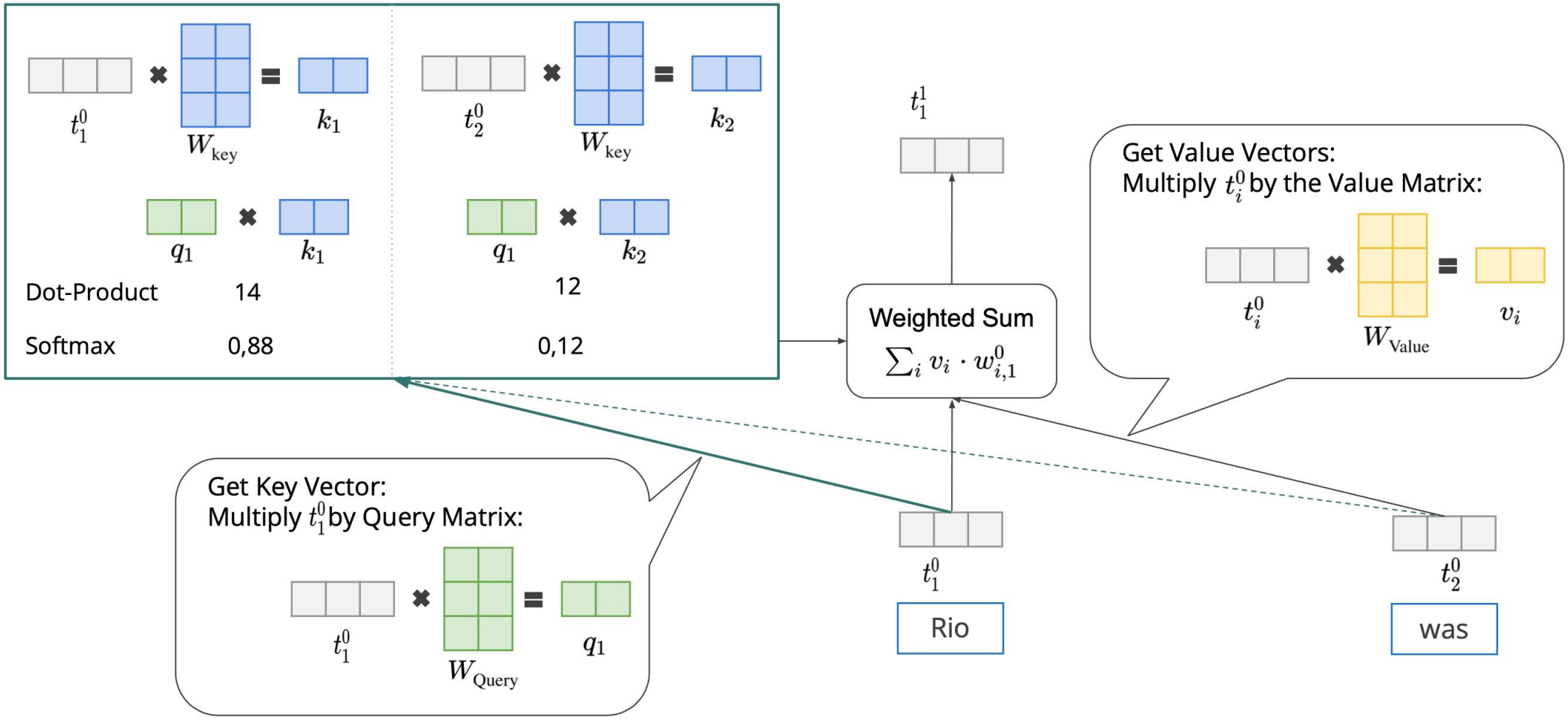
Self-Attention



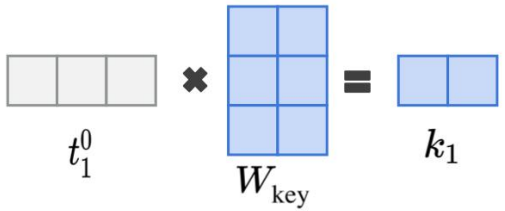
Self-Attention

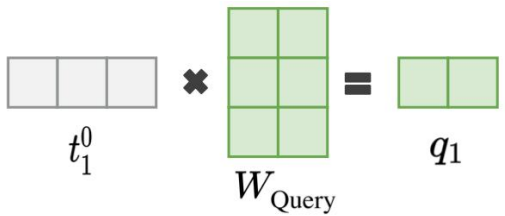


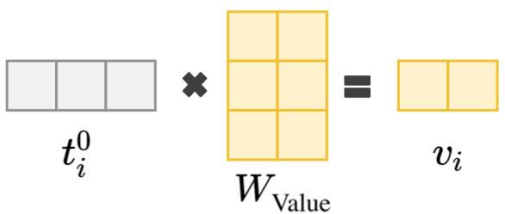
Self-Attention



Self-Attention







$$\alpha_{ij} = \frac{\exp e_{ij}}{\sum_{k=1}^{N} \exp e_{ik}}, e_{ij} = \frac{q_i \cdot k_j^T}{\sqrt{d_k}}.$$

$$c_i = \sum_{j=1}^{N} \alpha_{ij} \cdot v_j$$

$$q_i = x_i \cdot W_q, k_i = x_i \cdot W_k, v_i = x_i \cdot W_v$$

Multi-Head-Self-Attention

$$\text{Head}_1 = (W_q^{(1)}, W_k^{(1)}, W_v^{(1)})$$
$$\text{Head}_2 = (W_q^{(2)}, W_k^{(2)}, W_v^{(2)})$$
$$\dots$$
$$\text{Head}_H = (W_q^{(H)}, W_k^{(H)}, W_v^{(H)})$$

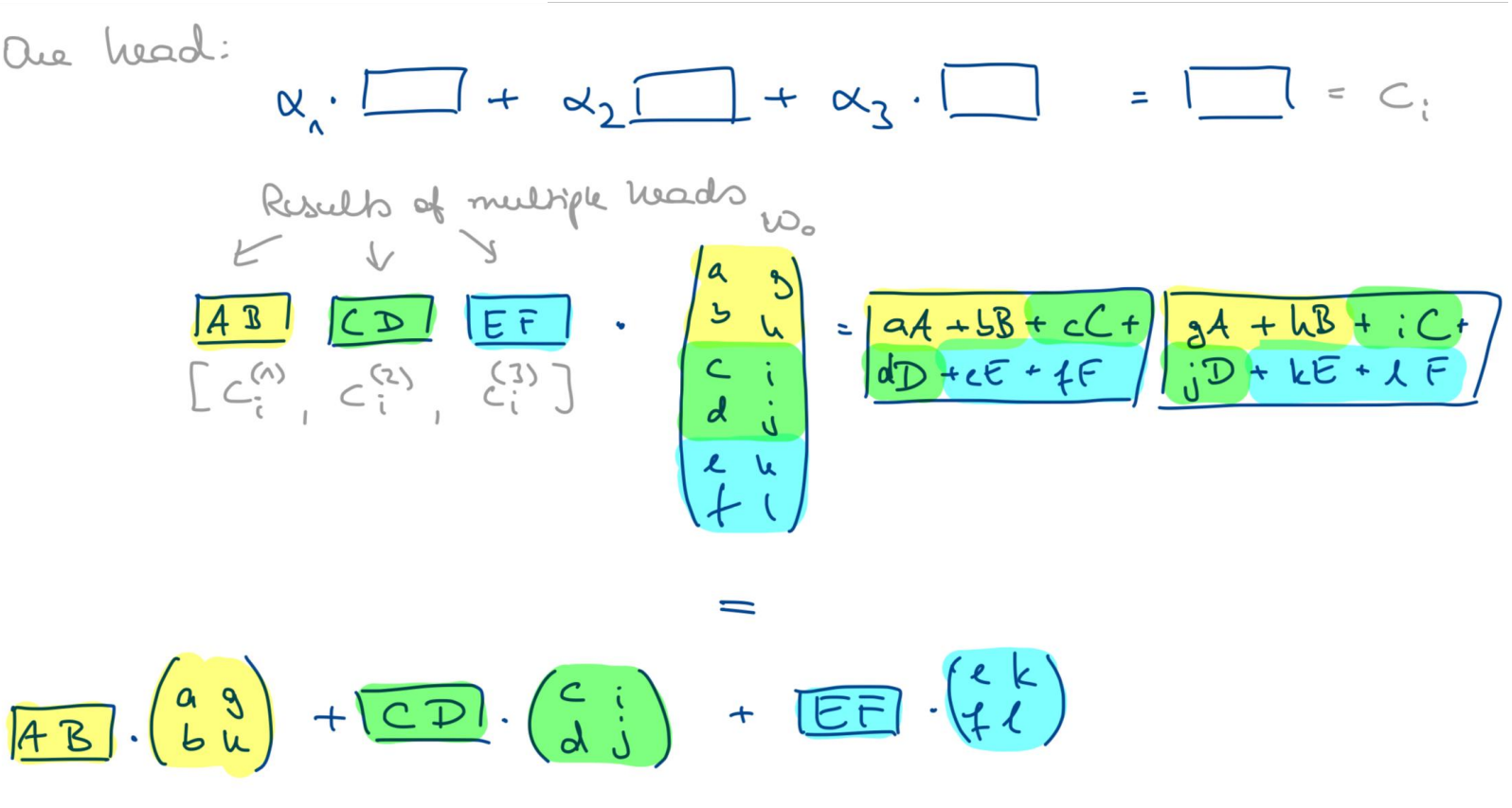
$$\text{Multi-Head-Self-Attention}(x) = [c_i^{(1)}, c_i^{(2)}, \dots, c_i^{(H)}] \cdot W_o,$$

Moving W_O into each head

$$W_O^H \begin{bmatrix} r^{h_1} \\ r^{h_2} \\ \dots \end{bmatrix} = \left[W_O^{h_1}, W_O^{h_2}, \dots \right] \cdot \begin{bmatrix} r^{h_1} \\ r^{h_2} \\ \dots \end{bmatrix} = \sum_i W_O^{h_i} r^{h_i}$$

Multi-Head-Self-Attention

$$W_O^H \begin{bmatrix} r^{h_1} \\ r^{h_2} \\ \dots \end{bmatrix} = \left[W_O^{h_1}, W_O^{h_2}, \dots\right] \cdot \begin{bmatrix} r^{h_1} \\ r^{h_2} \\ \dots \end{bmatrix} = \sum_i W_O^{h_i} r^{h_i}$$



Attention Calculation

1. Compute the value vector for each token from the residual stream ($v_i = W_V x_i$).
2. Compute the "result vector" by linearly combining value vectors according to the attention pattern ($r_i = \sum_j A_{i,j} v_j$).
3. Finally, compute the output vector of the head for each token ($h(x)_i = W_O r_i$).[8]

$$h(x) = (A \otimes W_O W_V) \cdot x$$

A mixes across tokens while $W_O W_V$ acts on each vector independently.

$$(A \otimes W_o W_v)\, x = A \cdot x \cdot (W_o W_v)^T$$
$$= A \cdot x \cdot W_v^T W_o^T$$

Tensor Product

There are several completely equivalent ways to interpret these products. If the symbol is unfamiliar, you can pick whichever you feel most comfortable with:

- **Left-right multiplying:** Multiplying x by a tensor product $A \otimes W$ is equivalent to simultaneously left and right multiplying: $(A \otimes W)x = AxW^T$. When we add them, it is equivalent to adding the results of this multiplication: $(A_1 \otimes W_1 + A_2 \otimes W_2)x = A_1 x W_1^T + A_2 x W_2^T$.
- **Kronecker product:** The operations we want to perform are linear transformations on a flattened ("vectorized") version of the activation matrix x. But flattening gives us a huge vector, and we need to map our matrices to a much larger block matrix that performs operations like "multiply the elements which previously corresponded to a vector by this matrix". The correct operation to do this is the Kronecker product. So we can interpret $\otimes$ as a Kronecker product acting on the vectorization of x, and everything works out equivalently.
- **Tensor Product:** $A \otimes W$ can be interpreted as a tensor product turning the matrices A and W into a 4D tensor. In NumPy notation, it is equivalent to `A[:,:,None,None] * W[None, None, :, :]` (although one wouldn't computationally represent them in that form). More formally, A and W are "type $(1,1)$" tensors (matrices mapping vectors to vectors), and $A \otimes W$ is a "type $(2,2)$" tensor (which can map matrices to matrices).

Tensor Product

$$(A \otimes W)\,x = A \times W^T$$

in x: every row is one token vector!

$$A = \begin{pmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \end{pmatrix} \qquad x = \begin{pmatrix} x_1 & x_4 \\ x_2 & x_5 \\ x_3 & x_6 \end{pmatrix}$$

$$W = \begin{pmatrix} w_1 & w_2 \\ w_3 & w_4 \end{pmatrix} \qquad W^T = \begin{pmatrix} w_1 & w_3 \\ w_2 & w_4 \end{pmatrix}$$

linear comb. of first components ... of second comp.

$$A \cdot x = \begin{pmatrix} (a_1x_1 + a_2x_2 + a_3x_3) & a_1x_4 + a_2x_5 + a_3x_6 \\ a_4x_1 + a_5x_2 + a_6x_3 & a_4x_4 + a_5x_5 + a_6x_6 \end{pmatrix}$$

Tensor Product

$$(A \otimes W)\,x \quad = \quad A\,x\,W^T$$

in x: every row is one token vector!

$$A = \begin{pmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \end{pmatrix} \qquad x = \begin{pmatrix} x_1 & x_4 \\ x_2 & x_5 \\ x_3 & x_6 \end{pmatrix}$$

$$W = \begin{pmatrix} w_1 & w_2 \\ w_3 & w_4 \end{pmatrix} \qquad W^T = \begin{pmatrix} w_1 & w_3 \\ w_2 & w_4 \end{pmatrix}$$

linear comb. of first components … of second comp.

$$A \cdot x = \begin{pmatrix} (a_1x_1 + a_2x_2 + a_3x_3) & a_1x_4 + a_2x_5 + a_3x_6 \\ a_4x_1 + a_5x_2 + a_6x_3 & a_4x_4 + a_5x_5 + a_6x_6 \end{pmatrix}$$

linear combination of first row vector!

$$Ax \cdot W^T = \begin{matrix} (a_1x_1+a_2x_2+a_3x_3)w_1 + (a_1x_4+a_2x_5+a_3x_6)w_2 & (a_1x_1+a_2x_2+a_3x_3)w_3 + (a_1x_4+a_2x_5+a_3x_6)w_4 \\ (a_4x_1+a_5x_2+a_6x_3)w_1 + (a_4x_4+a_5x_5+a_6x_6)w_2 & (a_4x_1+a_5x_2+a_6x_3)w_3 + (a_4x_4+a_5x_5+a_6x_6)w_4 \end{matrix}$$

Attention Calculation

1. Compute the value vector for each token from the residual stream ($v_i = W_V x_i$).
2. Compute the "result vector" by linearly combining value vectors according to the attention pattern ($r_i = \sum_j A_{i,j} v_j$).
3. Finally, compute the output vector of the head for each token ($h(x)_i = W_O r_i$).[8]

$$h(x) = (A \otimes W_O W_V) \cdot x$$

A mixes across tokens while $W_O W_V$ acts on each vector independently.

$$(A \otimes W_o W_v)\, x = A \cdot x \cdot (W_o W_v)^T$$
$$= A \cdot x \cdot W_v^T W_o^T$$

Attention Calculation

$$k_i = W_K x_i$$

$$q_i = W_Q x_i$$

$$A = \text{softmax}(q^T k)$$

$$A = \text{softmax}(x^T W_Q^T W_K x)$$

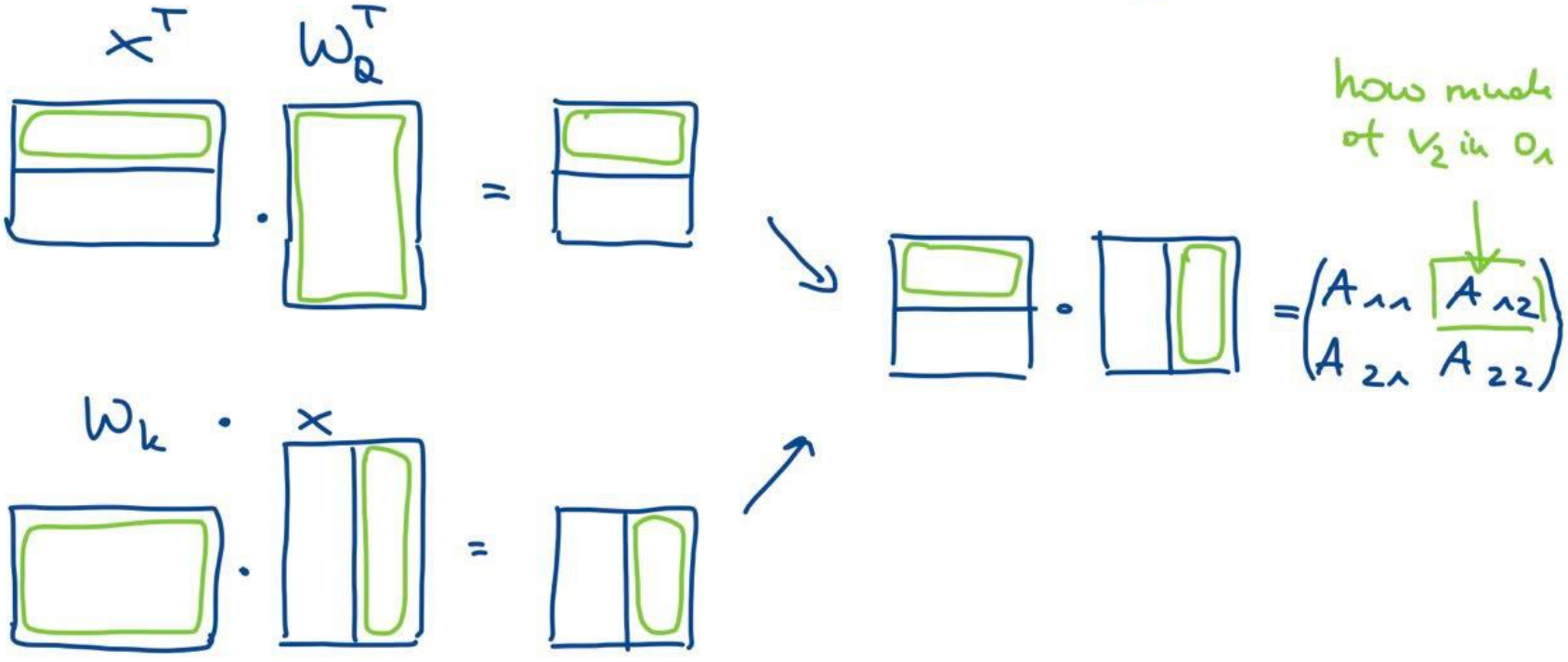
Attention Calculation

$$h(x) = (A \otimes W_O W_V) \cdot x \qquad A = \text{softmax}(x^T W_Q^T W_K x)$$

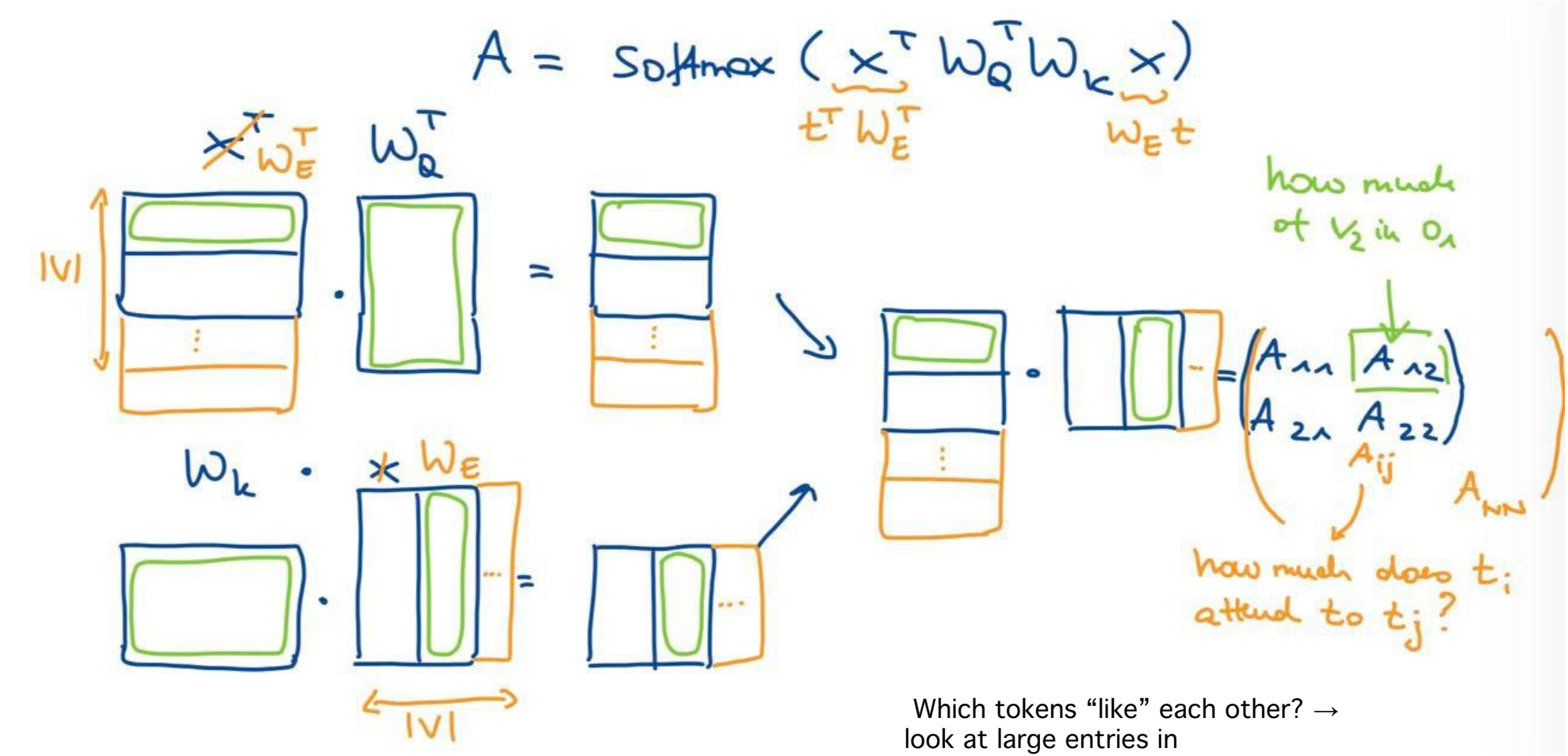
Because $W_O W_V$ and $W_Q W_K$ always operate together, we like to define variables representing these combined matrices, $W_{OV} = W_O W_V$ and $W_{QK} = W_Q^T W_K$.

What happens when we compute A?

$$A = \text{Softmax}\left(x^T W_Q^T W_k x\right)$$

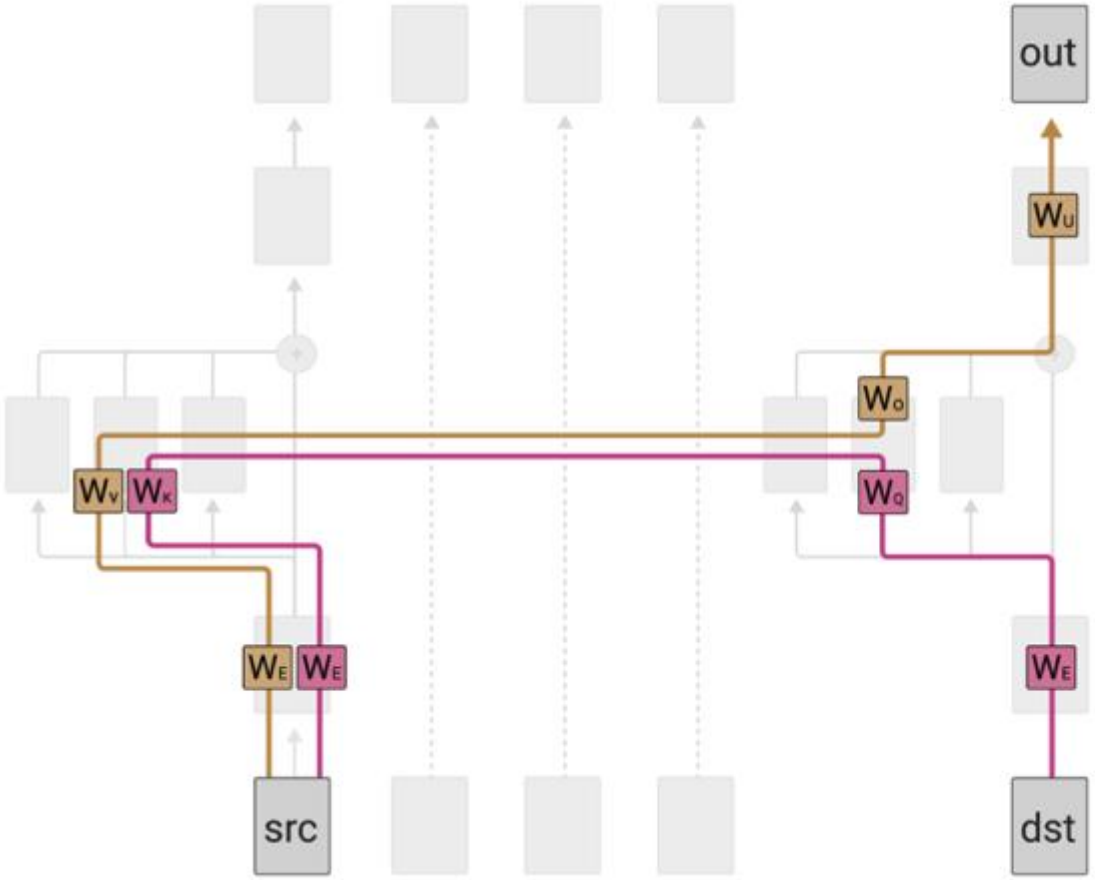


What happens when we compute A?



Which tokens “like” each other? → look at large entries in (W_E^T W_Q^T W_K W_E)

1-Layer Transformer



The **OV ("output-value") circuit** determines how attending to a given token affects the logits.

$$W_U W_O W_V W_E$$

The **QK ("query-key") circuit** controls which tokens the head prefers to attend to.

$$W_E^T W_Q^T W_K W_E$$

Interpretation?

The **QK ("query-key") circuit** controls which tokens the head prefers to attend to.

$$W_E^T W_Q^T W_K W_E$$

destination token

source token

The **OV ("output-value") circuit** determines how attending to a given token affects the logits.

$$W_U W_O W_V W_E$$

out token

source token

In the above example, we fix a given source token and look at the largest corresponding QK entries (the destination token) and largest corresponding OV entries (the out token). The source token is selected to show interesting behavior, but the destination and out token are the top entries unless entries are explicitly skipped with an ellipsis; they are colored by the intensity of their value in the matrix.

Examples

In the above example, we fix a given source token and look at the largest corresponding QK entries (the destination token) and largest corresponding OV entries (the out token). The source token is selected to show interesting behavior, but the destination and out token are the top entries unless entries are explicitly skipped with an ellipsis; they are colored by the intensity of their value in the matrix.

Some examples of large entries QK/OV circuit

Source Token	Destination Token	Out Token	Example Skip Tri-grams
" perfect"	" are", " looks", " is", " provides"	" perfect", " super", " absolute", " pure"	" perfect... are perfect", " perfect... looks super"
" large"	" contains", " using", " specify", " contain"	" large", " small", " very", " huge"	" large... using large", " large... contains small"
" two"	" One", "\n ", " has", "\r\n ", "One"	" two", " three", " four", " five", " one"	" two... One two", " two... has three"
"lambda"	" $\\", "}{\\", "+\\", "(\\", " ${\\"	"lambda", "sorted", " lambda", "operator"	"lambda... $\\lambda", "lambda... +\\lambda"
"nbsp"	"&", " \"&", "}&", ">&", "=&"	"nbsp", "01", "gt", "00012", "nbs", "quot"	"nbsp... ", "nbsp... > "
"Great"	"The", " The", " the", " contains", " /"	" Great", " great", " poor", " Every"	"Great... The Great", "Great... the great"

More Examples

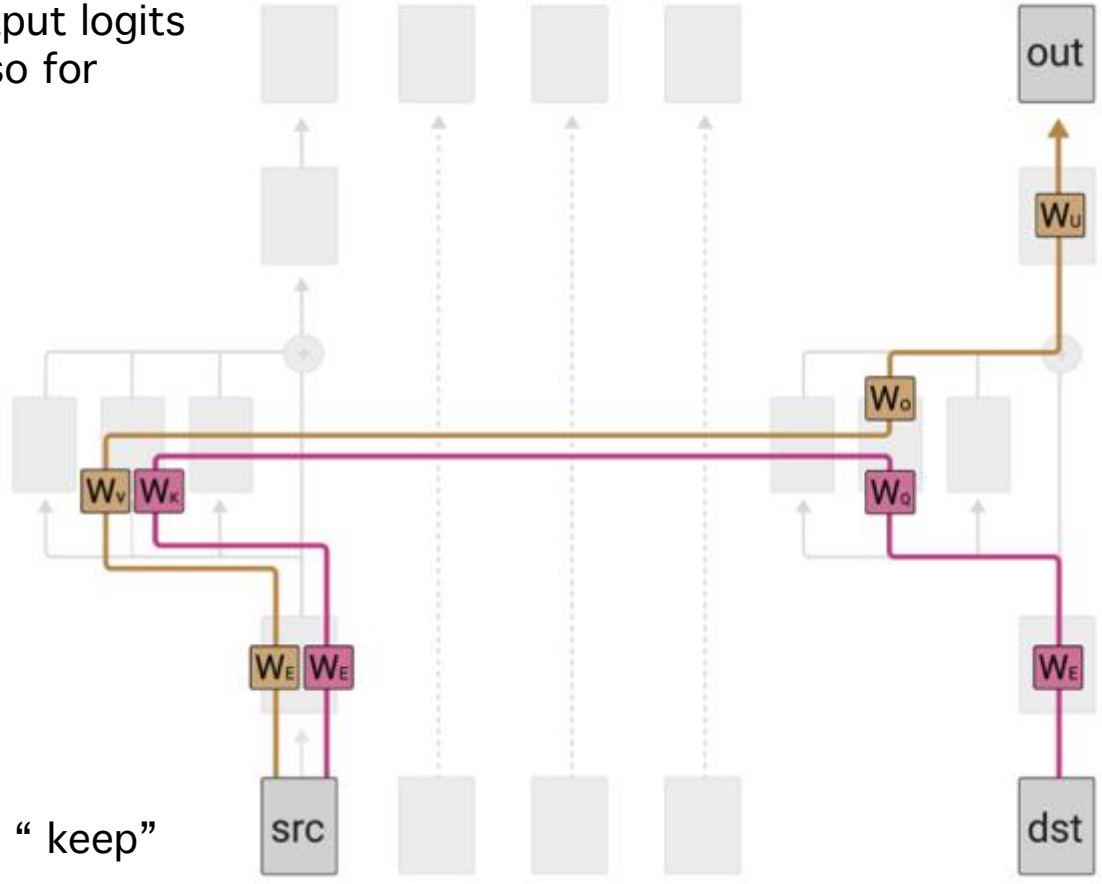
More examples of large entries QK/OV circuit

Source Token	Destination Token	Out Token	Example Skip Tri-grams
"indy"	" C", "C", " V", "V", " R", " c"	"indy", "obby", "INDY", "loyd"	"indy... Cindy", "indy... CINDY"
" Pike"	" P", "P", "V", "Sp", " V", "R"	"ike", "ikes", "ishing", "owler"	" Pike... Pike", " Pike... Spikes"
" Ralph"	"R", " R", " P", "P", "V", " r"	"alph", "ALPH", "obby", "erald"	" Ralph... Ralph", " Ralph... RALPH"
" Lloyd"	"L", " L", " P", "P", "R", " C"	"loyd", "alph", "\n ", "acman", ... "atherine"	" Lloyd... Lloyd", " Lloyd... Catherine"
" Pixmap"	" P", " Q", "P", " p", " U"	"ixmap", "Canvas", "Embed", "grade"	" Pixmap... Pixmap", " Pixmap... QCanvas"

Interpretation

Attending to src = keep will increase the output logits for mind, but also for bay and wraps

out = mind



The **OV ("output-value") circuit** determines how attending to a given token affects the logits.

$$W_U W_O W_V W_E$$

The **QK ("query-key") circuit** controls which tokens the head prefers to attend to.

$$W_E^T W_Q^T W_K W_E$$

Src = " keep"

Dst = current token = "in"

“Bugs”

Limited Expressivity Can Create Bugs which Seem Strange from the Outside

Source Token	Destination Token	Out Token	“Correct” Skip Tri-grams	“Bug” Skip Tri-grams
“ Pixmap”	“ P”, “ Q”, “P”, “ p”, “ U”	“ixmap”, “Canvas”, “Embed”, “grade”	“ Pixmap... Pixmap”, “ Pixmap... QCanvas”	“ Pixmap... PCanvas”

Source Token	Destination Token	Out Token	“Correct” Skip Tri-grams	“Bug” Skip Tri-grams
“ Lloyd”	“L”, “ L”, “ P”, “P”, “R”, “ C”	“loyd”, “alph”, “\n ”, “acman”, ... “atherine”	“ Lloyd... Lloyd”, “ Lloyd... Catherine”	“ Lloyd... Cloyd”, “ Lloyd... Latherine”

Source Token	Destination Token	Out Token	“Correct” Skip Tri-grams	“Bug” Skip Tri-grams
“ keep”	“ in”, “ at”, “ out”, “ under”, “ off”	“ bay”, ... “ mind”, ... “ wraps”	“ keep... in mind”, “ keep... at bay”, “ keep... under wraps”	“ keep... in bay”, “ keep... at wraps”, “ keep... under mind”

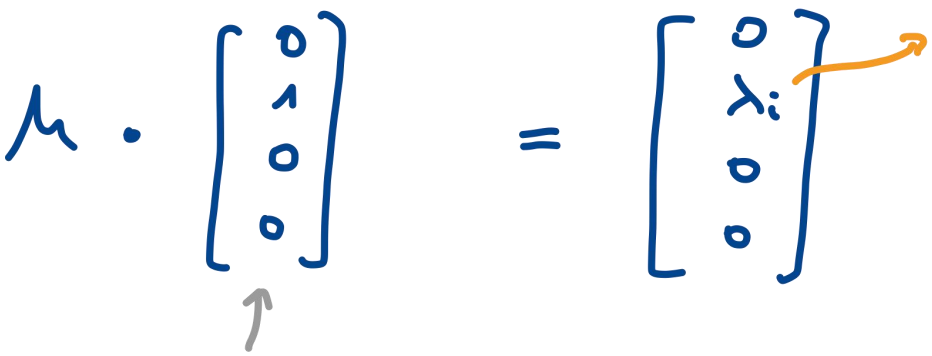
How to find copying heads?

Look at OV-Circuit:

$$Mv_i = \lambda_i v_i \text{ with } M = W_U W_{OV} W_E$$

The **OV ("output-value") circuit** determines how attending to a given token affects the logits.

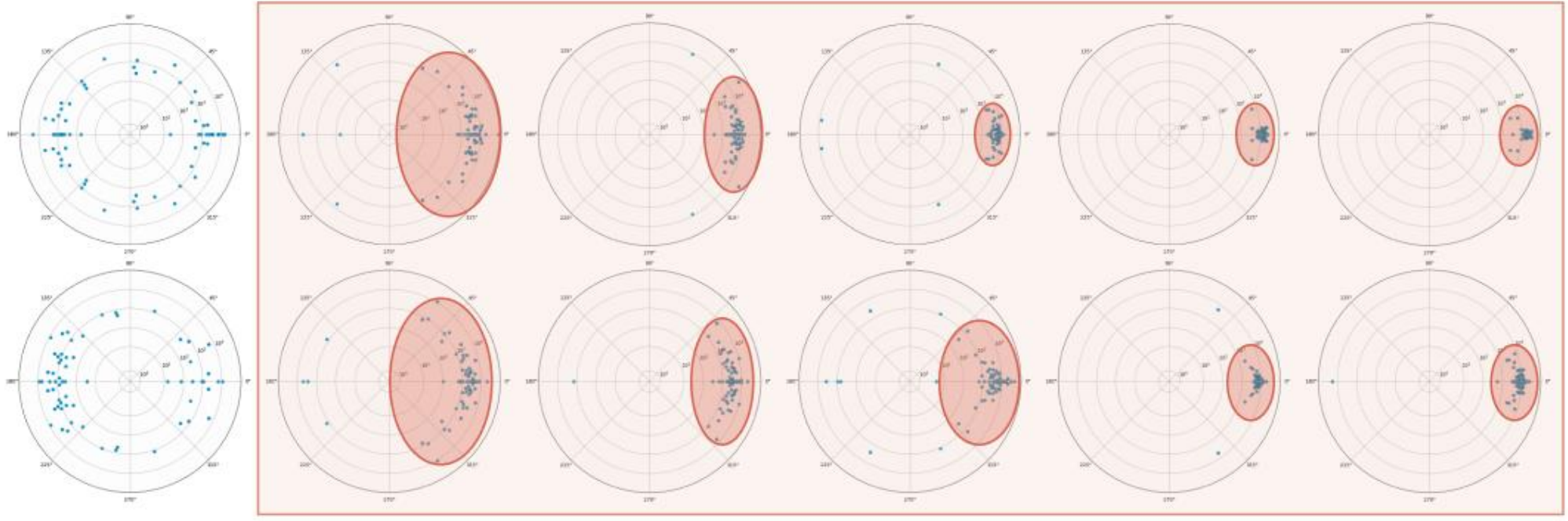
$W_U W_O W_V W_E$



How to find copying heads?

Eigenvalue analysis of **first layer** attention head OV circuits

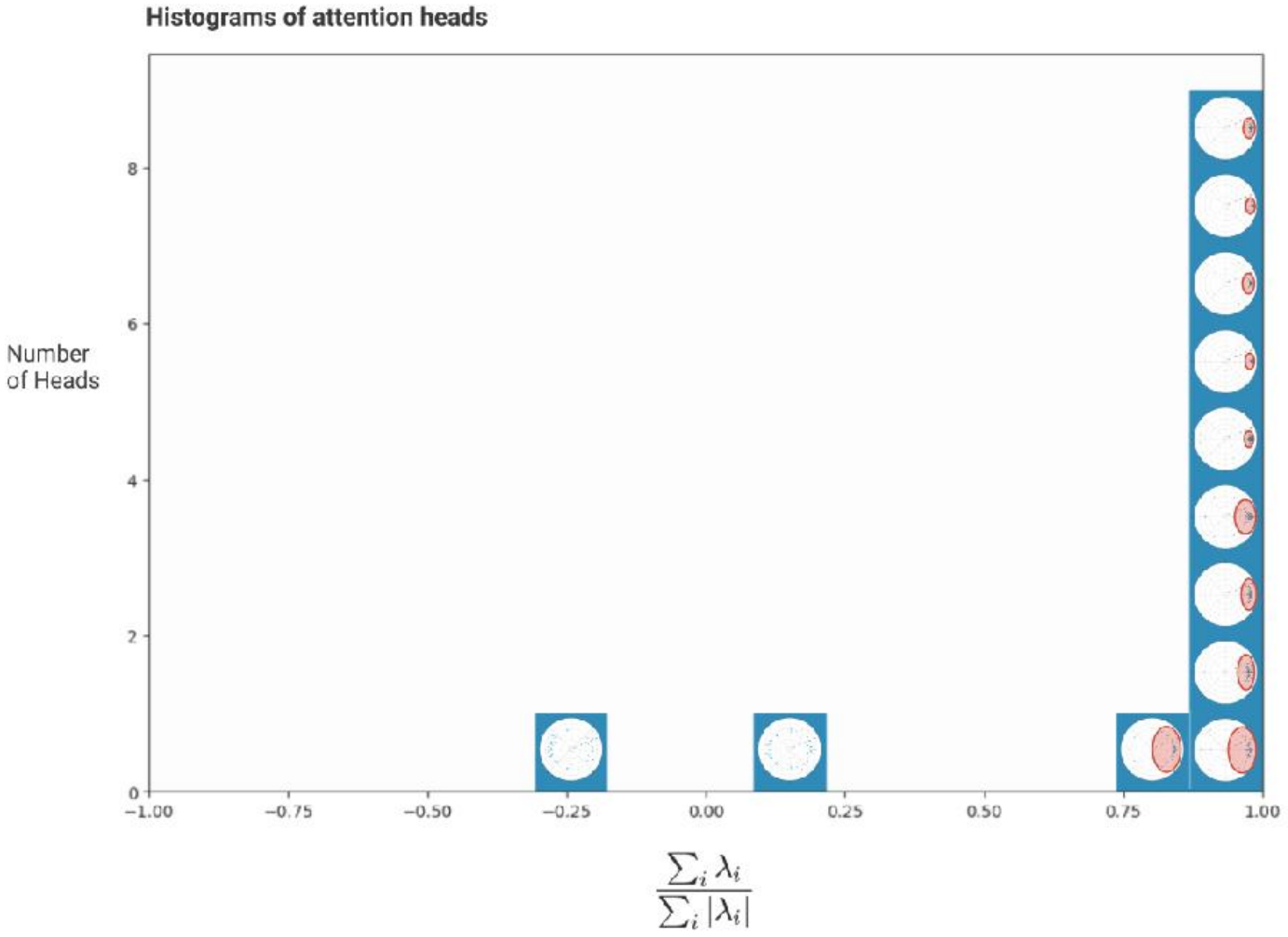
10/12 of layer 1 heads have mostly positive OV eigenvalues,
and appear to significantly perform copying



non-positive eigenvalues
not copying heads?

positive eigenvalues
copying heads?

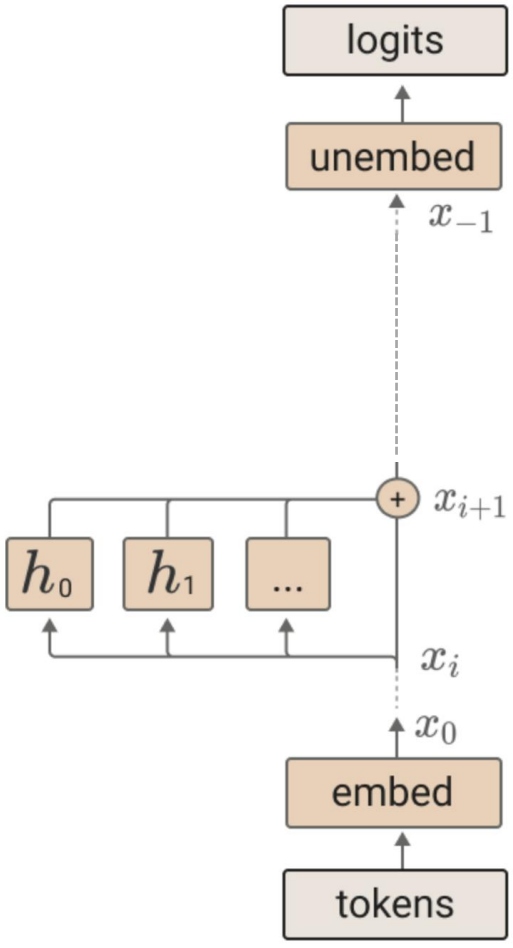
How to find copying heads?



Another way to find copying heads?

- One might try to formalize "copying matrices" in other ways. One possibility is to look at the diagonal of a matrix, which describes how each token affects its own probability. As expected, entries on the diagonal are very positive-leaning.

Review Last Week



The final logits are produced by applying the unembedding.

$$T(t) = W_U x_{-1}$$

One residual block

Each attention head, h, is run and added to the residual stream.

$$x_{i+1} = x_i + \sum_{h \in H_i} h(x_i)$$

Token embedding.

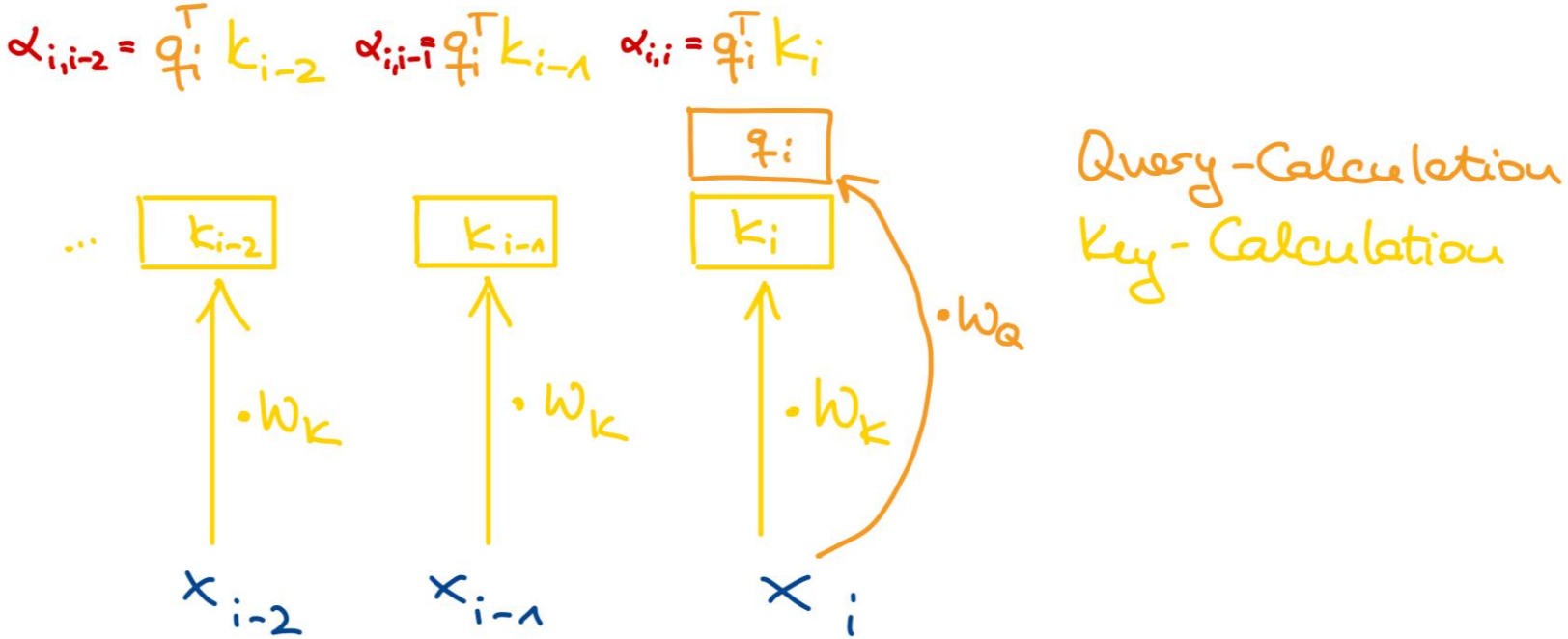
$$x_0 = W_E t$$

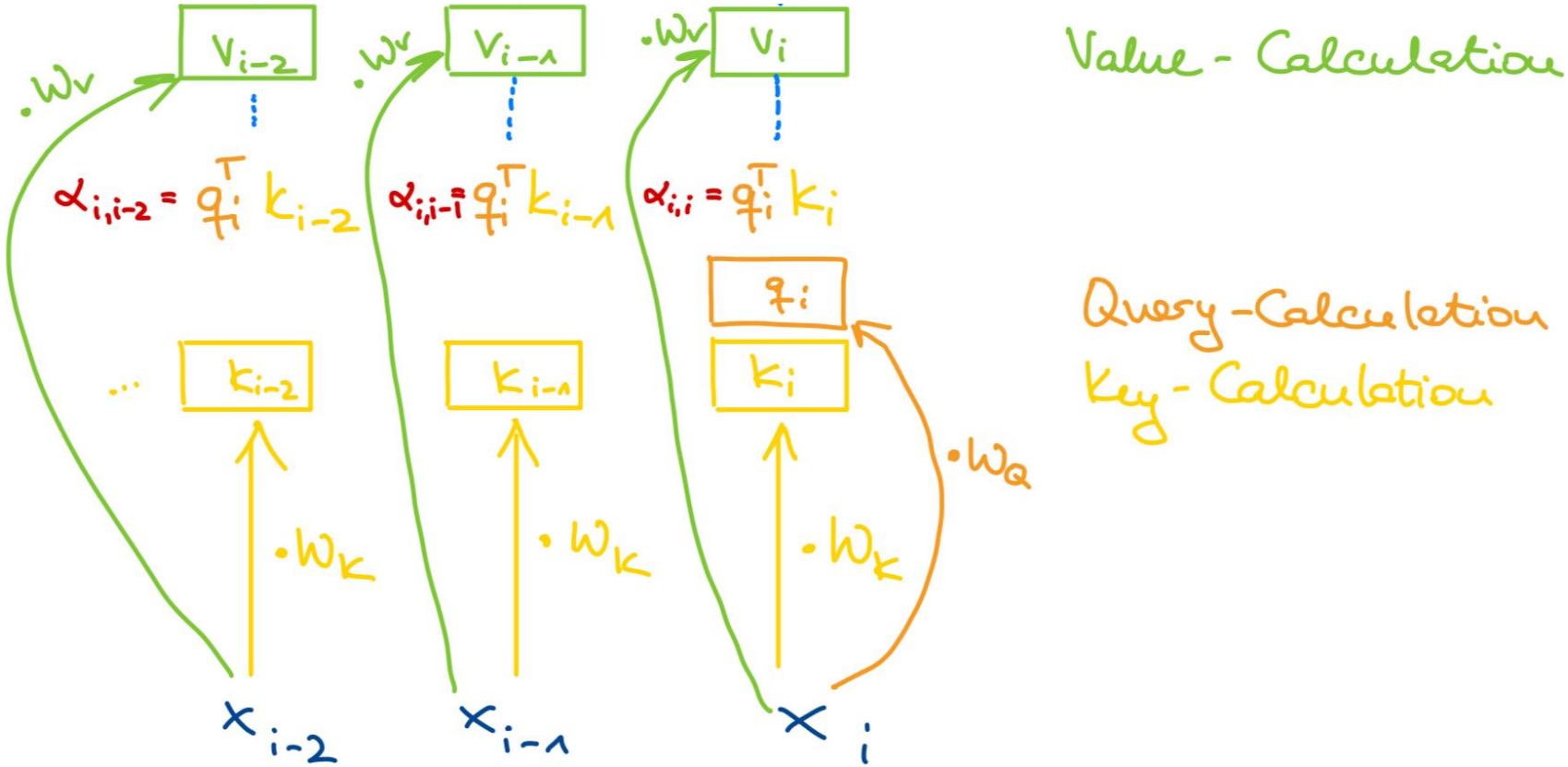
Attention Calculation

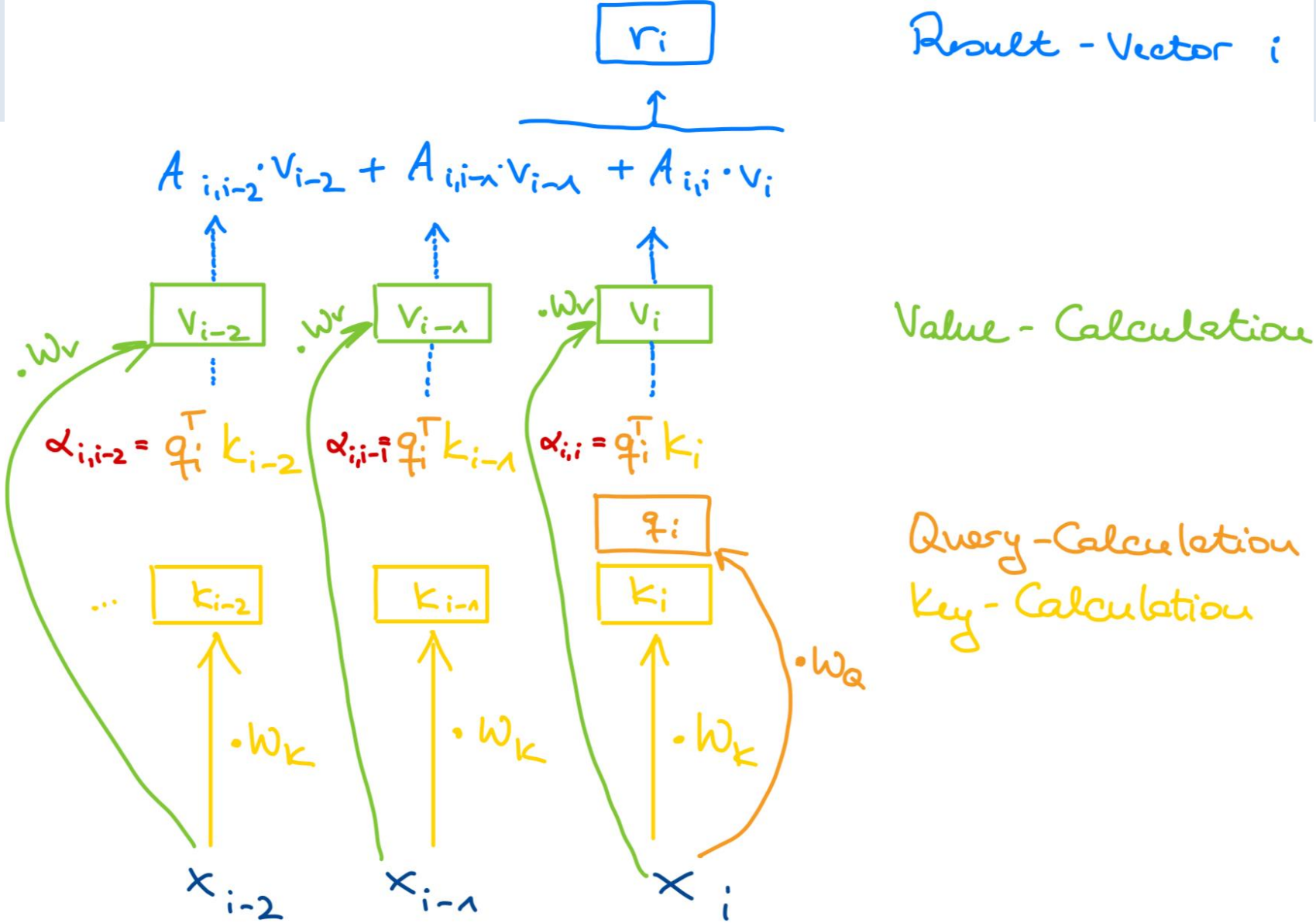
1. Compute the value vector for each token from the residual stream ($v_i = W_V x_i$).
2. Compute the "result vector" by linearly combining value vectors according to the attention pattern ($r_i = \sum_j A_{i,j} v_j$).
3. Finally, compute the output vector of the head for each token ($h(x)_i = W_O r_i$).[8]

$$h(x) = (A \otimes W_O W_V) \cdot x$$

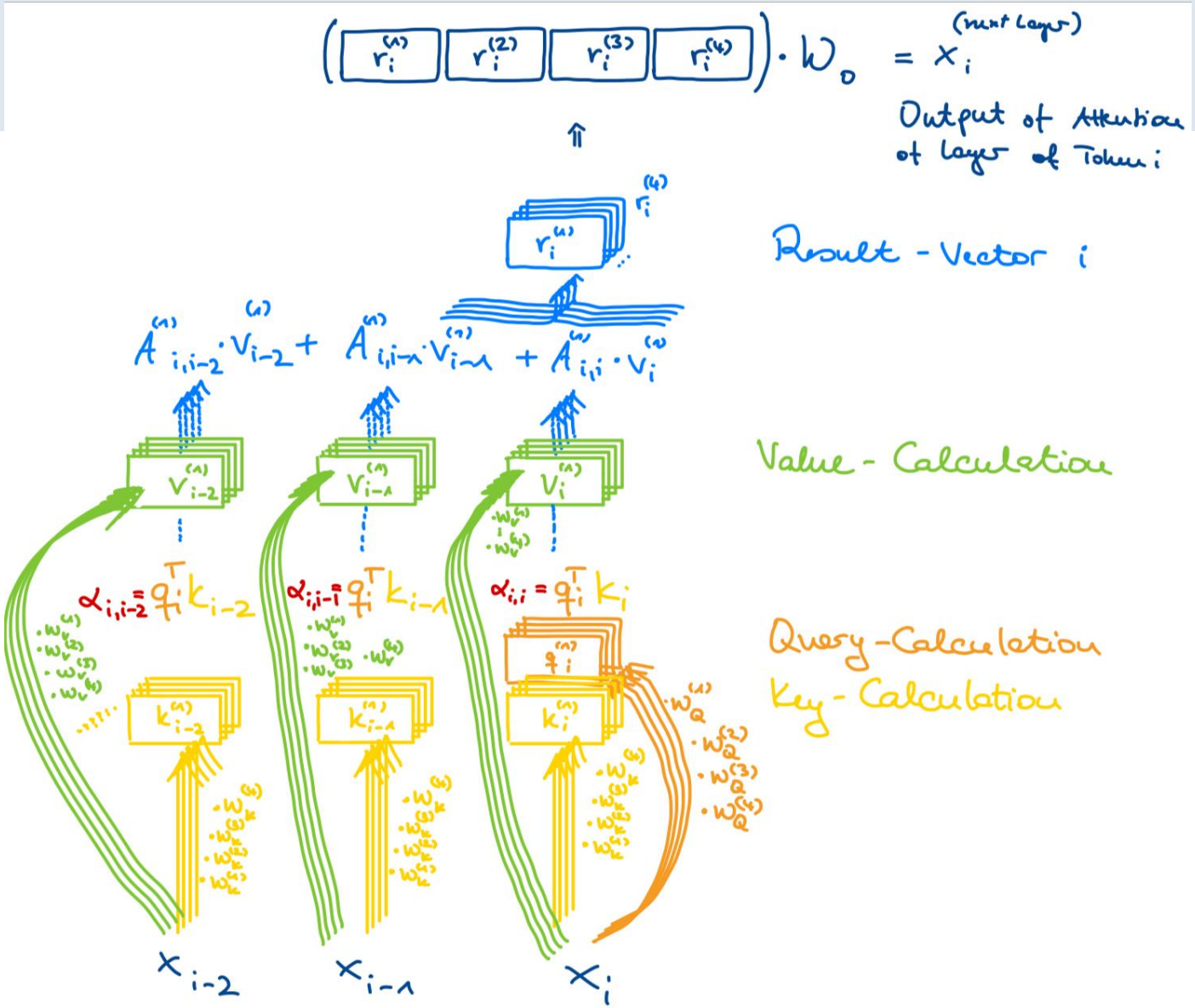
$$\left.\begin{aligned} k_i &= W_K x_i \\ q_i &= W_Q x_i \\ A &= \text{softmax}(q^T k) \end{aligned}\right\} \quad A = \text{softmax}(x^T W_Q^T W_K x)$$


$\alpha_{i,i-2} = q_i^T k_{i-2}$
$\alpha_{i,i-1} = q_i^T k_{i-1}$
$\alpha_{i,i} = q_i^T k_i$
q_i
k_{i-2}
k_{i-1}
k_i
$\cdot W_Q$
$\cdot W_K$
$\cdot W_K$
$\cdot W_K$
x_{i-2}
x_{i-1}
x_i
Query-Calculation
Key-Calculation


v_{i-2}
v_{i-1}
v_i
$\cdot W_V$
$\alpha_{i,i-2} = q_i^T k_{i-2}$
$\alpha_{i,i-1} = q_i^T k_{i-1}$
$\alpha_{i,i} = q_i^T k_i$
q_i
k_{i-2}
k_{i-1}
k_i
$\cdot W_K$
$\cdot W_Q$
x_{i-2}
x_{i-1}
x_i
Value - Calculation
Query - Calculation
Key - Calculation


r_i
Result - Vector i
$A_{i,i-2} \cdot v_{i-2} + A_{i,i-1} \cdot v_{i-1} + A_{i,i} \cdot v_i$
v_{i-2}
v_{i-1}
v_i
Value - Calculation
$\cdot W_V$
$\alpha_{i,i-2} = q_i^T k_{i-2}$
$\alpha_{i,i-1} = q_i^T k_{i-1}$
$\alpha_{i,i} = q_i^T k_i$
q_i
k_{i-2}
k_{i-1}
k_i
Query-Calculation
Key-Calculation
$\cdot W_K$
$\cdot W_Q$
x_{i-2}
x_{i-1}
x_i

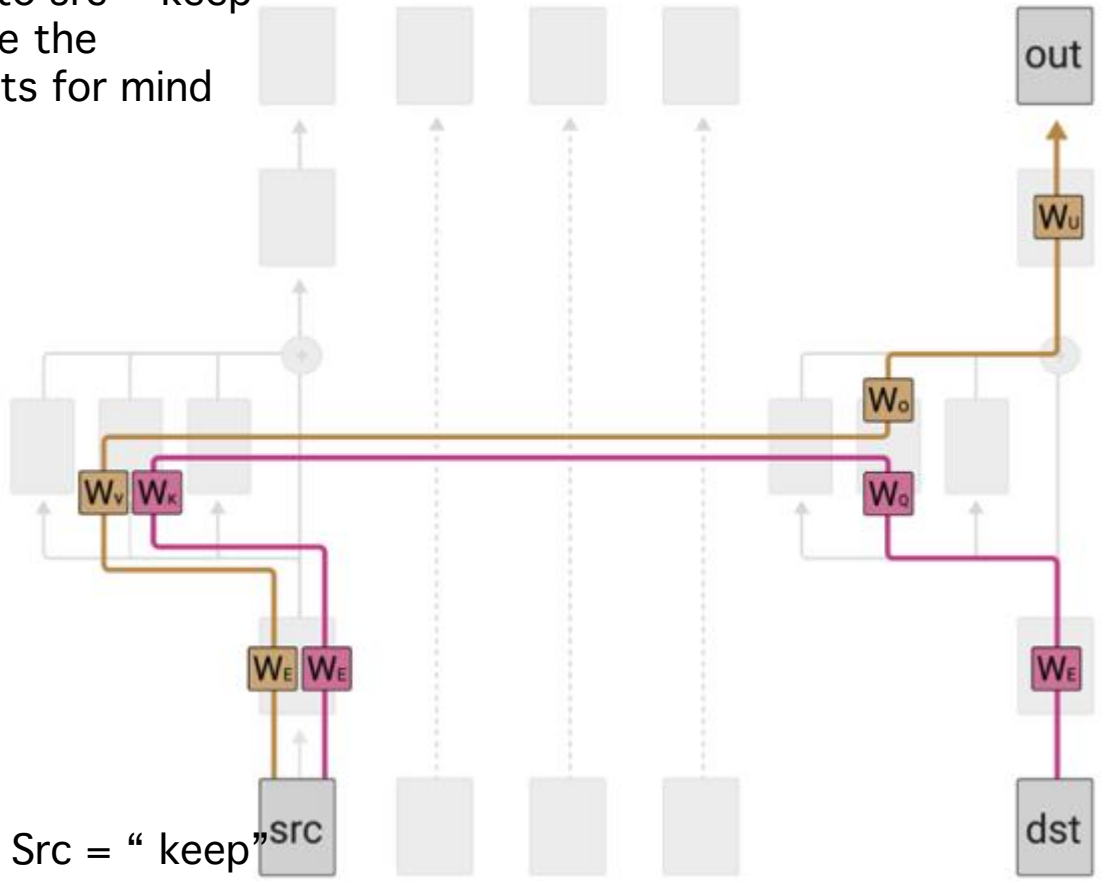
Multiple Heads parallel



Review Layer 1

Attending to src = keep will increase the output logits for mind

out = mind



Src = " keep"

Dst = current token = "in"

The **QK ("query-key") circuit** controls which tokens the head prefers to attend to.

$$W_E^T W_Q^T W_K W_E$$

destination token source token

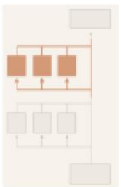
The **OV ("output-value") circuit** determines how attending to a given token affects the logits.

$$W_U W_O W_V W_E$$

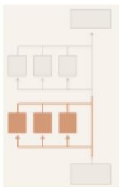
out token source token

2-Layers Transformer

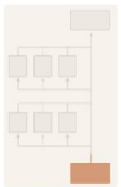
$$T = \underbrace{\mathrm{Id} \otimes W_U}_{\text{Unembed}} \cdot \underbrace{\left(Id + \sum_{h \in H_2} A^h \otimes W_{OV}^h\right)}_{\text{The second attention layer has multiple attention heads which add into the residual stream}} \cdot \underbrace{\left(Id + \sum_{h \in H_1} A^h \otimes W_{OV}^h\right)}_{\text{The first attention layer has multiple attention heads which add into the residual stream}} \cdot \underbrace{\mathrm{Id} \otimes W_E}_{\text{Embed}}$$



The second **attention layer** has multiple attention heads which add into the residual stream



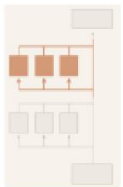
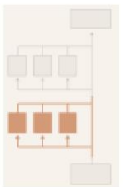
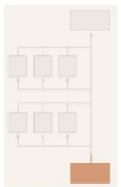
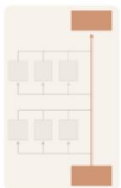
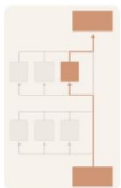
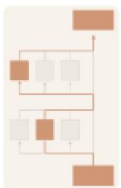
The first **attention layer** has multiple attention heads which add into the residual stream



2-Layers Transformer

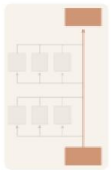
$$T = \underbrace{\mathrm{Id} \otimes W_U}_{\text{Id}} \cdot \underbrace{\left(Id + \sum_{h \in H_2} A^h \otimes W_{OV}^h\right)}_{\text{The second attention layer has multiple attention heads which add into the residual stream}} \cdot \underbrace{\left(Id + \sum_{h \in H_1} A^h \otimes W_{OV}^h\right)}_{\text{The first attention layer has multiple attention heads which add into the residual stream}} \cdot \underbrace{\mathrm{Id} \otimes W_E}_{\text{Id}}$$

$$= \underbrace{\mathrm{Id} \otimes W_U W_E}_{\text{"Direct path" term contributes to bigram statistics.}} + \underbrace{\sum_{h \in H_1 \cup H_2} A^h \otimes (W_U W_{OV}^h W_E)}_{\text{The individual attention head terms describe the effects of individual attention heads in linking input tokens to logits, similar to those we saw in the one layer model.}} + \underbrace{\sum_{h_2 \in H_2} \sum_{h_1 \in H_1} (A^{h_2} A^{h_1}) \otimes (W_U W_{OV}^{h_2} W_{OV}^{h_1} W_E)}_{\text{The virtual attention head terms correspond to V-composition of attention heads. They function a lot like individual attention heads, with their own attention patterns (the compositon of the heads patterns) and own OV matrix.}}$$

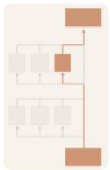








2-Layers Transformer

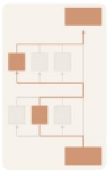
$$= \underbrace{\mathrm{Id} \otimes W_U W_E}_{\text{“Direct path” term contributes to bigram statistics.}} + \underbrace{\sum_{h \in H_1 \cup H_2} A^h \otimes (W_U W_{OV}^h W_E)}_{\text{The individual attention head terms describe the effects of individual attention heads in linking input tokens to logits, similar to those we saw in the one layer model.}} + \underbrace{\sum_{h_2 \in H_2} \sum_{h_1 \in H_1} (A^{h_2} A^{h_1}) \otimes (W_U W_{OV}^{h_2} W_{OV}^{h_1} W_E)}_{\text{The virtual attention head terms correspond to V-composition of attention heads. They function a lot like individual attention heads, with their own attention patterns (the compositon of the heads patterns) and own OV matrix.}}$$



“Direct path” term contributes to bigram statistics.



The **individual attention head** terms describe the effects of individual attention heads in linking input tokens to logits, similar to those we saw in the one layer model.



The **virtual attention head** terms correspond to V-composition of attention heads. They function a lot like individual attention heads, with their own attention patterns (the compositon of the heads patterns) and own OV matrix.

Same as for 0 layer

Same as for 1 layer

New!

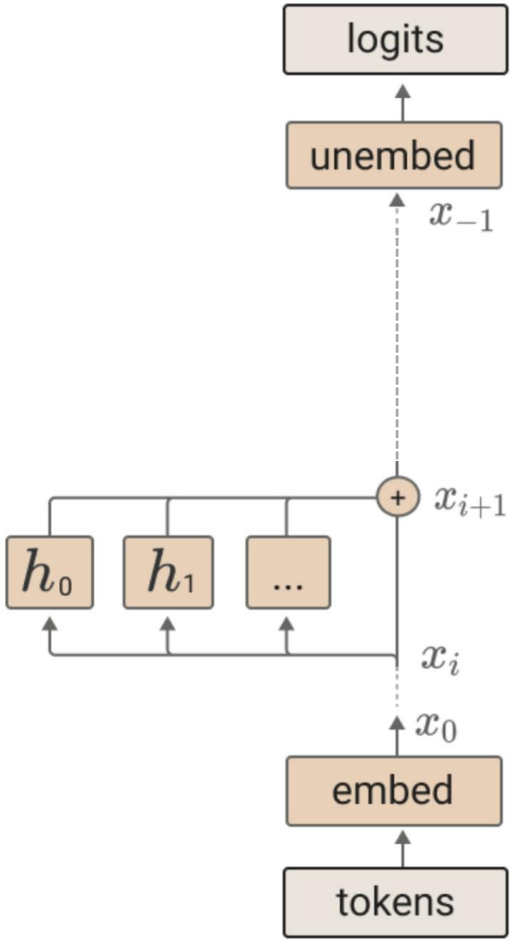
Calculating Attention

$$A^h = \text{softmax}^*\left(t^T \cdot C^h_{QK} t\right)$$

$$C^{h\in H_1}_{QK} = x_0^T W^h_{QK} x_0 = W_E^T W^h_{QK} W_E$$

$$C^{h\in H_2}_{QK} = x_1^T W^h_{QK} x_1$$

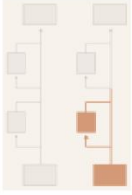
x_1 is residual stream after layer 1 attention



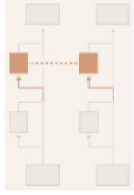
Calculating Attention Pattern in 2nd Layer

$$C_{QK}^{h\in H_2} = x_1^T W_{QK}^h x_1 \qquad h(x) = (A \otimes W_O W_V) \cdot x \qquad A = \text{softmax}(x^T W_Q^T W_K x)$$

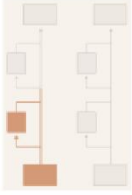
$$= \left(\text{Id}\otimes\text{Id}\otimes W_E^T + \sum_{h_q\in H_1} A^{h_q}\otimes\text{Id}\otimes(W_{OV}^{h_q}W_E)^T\right) \cdot \text{Id}\otimes\text{Id}\otimes W_{QK}^h \cdot \left(\text{Id}\otimes\text{Id}\otimes W_E + \sum_{h_k\in H_1}\text{Id}\otimes A^{h_k}\otimes W_{OV}^{h_k}W_E\right)$$



The **"query side" residual stream** at the start of the second layer contains both the layer 1 direct path and layer 1 attention heads. All terms are of the form $\ldots\otimes\text{Id}\otimes\ldots$ because they don't move key information.



W_{QK} of the second layer head combines both sides into attention scores.



The **"key side" residual stream** at the start of the second layer contains both the layer 1 direct path and attention heads. All terms are of the form $\text{Id}\otimes\ldots$ because they don't move query information.

$$C_{QK}^{h\in H_2} = x_1^T W_{QK}^h x_1$$

$$x_1 = x_0 + \sum_{h\in H_1} (A^h \otimes W_O W_V) x_0 = W_E \cdot t + \sum_{h\in H_1} (A^h \otimes W_O W_V) W_E \cdot t$$

$$= \underbrace{\mathrm{Id} \otimes \mathrm{Id} \otimes \left(W_E^T W_{QK}^h W_E\right)}_{\text{The no composition term. Both first layer follows the direct path on both the key and query side.}} + \underbrace{\sum_{h_q\in H_1} A^{h_q} \otimes \mathrm{Id} \otimes \left(W_E^T W_{OV}^{h_q\,T} W_{QK}^h W_E\right)}_{\text{These terms correspond to pure } \textbf{Q-composition}\text{. A previous attention head is used to generate the query side, but the key side is the first layer direct path.}}$$

$$+ \underbrace{\sum_{h_k\in H_1} \mathrm{Id} \otimes A^{h_k} \otimes \left(W_E^T W_{QK}^h W_{OV}^{h_k} W_E\right)}_{\text{These terms correspond to pure } \textbf{K-composition}\text{. A previous attention head is used to generate part of the key, but the query side is the first layer direct path.}} + \underbrace{\sum_{h_q\in H_1} \sum_{h_k\in H_1} A^{h_q} \otimes A^{h_k} \otimes \left(W_E^T W_{OV}^{h_q\,T} W_{QK}^h W_{OV}^{h_k} W_E\right)}_{\text{These terms are interactions between both } \textbf{Q-composition} \text{ and } \textbf{K-composition}\text{. A previous attention head is used to generate the query and key sides.}}$$

Which L2 head uses information of which previous L1 Head?

The above diagram shows Q-, K-, and V-Composition between attention heads in the first and second layer. That is, how much does the query, key or value vector of a second layer head read in information from a given first layer head? This is measured by looking at the Frobenius norm of the product of the relevant matrices, divided by the norms of the individual matrices. For Q-Composition, $||W_{QK}^{h_2 T} W_{OV}^{h_1}||_F / (||W_{QK}^{h_2 T}||_F ||W_{OV}^{h_1}||_F)$, for K-Composition $||W_{QK}^{h_2} W_{OV}^{h_1}||_F / (||W_{QK}^{h_2}||_F ||W_{OV}^{h_1}||_F)$, for V-Composition $||W_{OV}^{h_2} W_{OV}^{h_1}||_F / (||W_{OV}^{h_2}||_F ||W_{OV}^{h_1}||_F)$. By default, we subtract off the empirical expected amount for random matrices of the same shapes (most attention heads have a much smaller composition than random matrices). In practice, for this model, there is only significant K-composition, and only with one layer 0 head.

Q-Composition

$$\frac{||W_{QK}^{h_2 T} W_{OV}^{h_1}||_F}{||W_{QK}^{h_2 T}||_F ||W_{OV}^{h_1}||_F}$$

K-Composition

$$\frac{||W_{QK}^{h_2} W_{OV}^{h_1}||_F}{||W_{QK}^{h_2}||_F ||W_{OV}^{h_1}||_F}$$

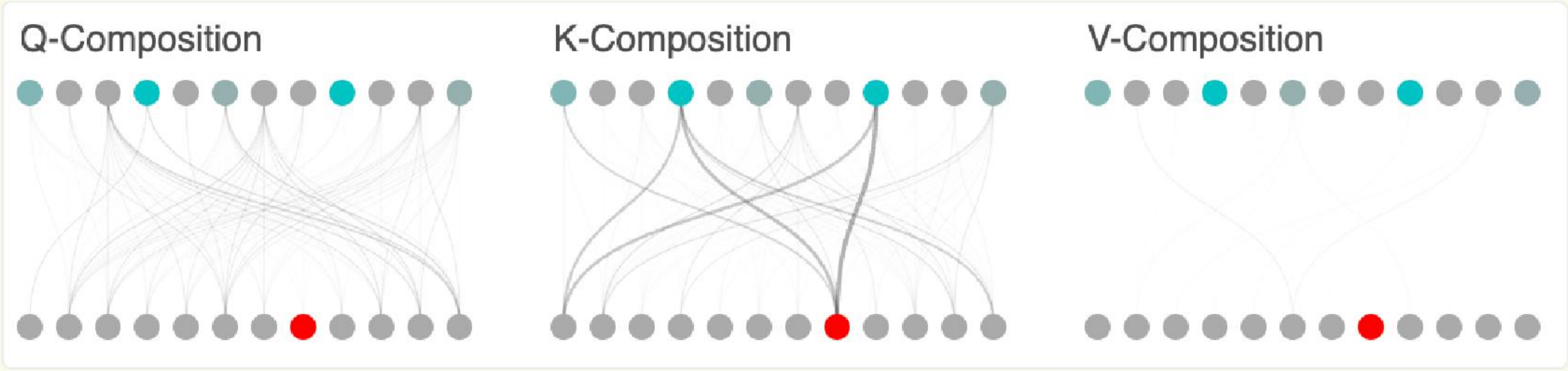
V-Composition

$$\frac{||W_{OV}^{h_2} W_{OV}^{h_1}||_F}{||W_{OV}^{h_2}||_F ||W_{OV}^{h_1}||_F}$$

$$\|A\|_{\mathrm{F}} = \sqrt{\sum_{i}^{m} \sum_{j}^{n} |a_{ij}|^2}$$

Which L2 head uses information of which previous L1 Head?

CORRECTED PLOT (BASELINE SUBTRACTED)



Induction Heads

Previous Token Heads

Note: Line widths between original and corrected plots are not directly comparable.

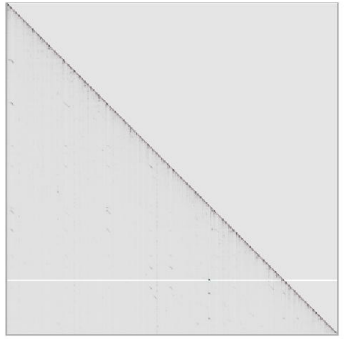
Which L2 head uses information of which previous L1 Head?

The above diagram shows the *value-weighted attention pattern* for various attention heads; that is, the attention patterns with attention weights scaled by the norm of the value vector at the source position $||v^h_{src}||$. You can think of the value-weighted attention pattern as showing "how big a vector is moved from each position." (This approach was also recently introduced by Kobayashi *et al.* [16].) This is especially useful because attention heads will sometimes use certain tokens as a kind of default or resting position when there isn't a token that matches what they're looking for; the value vector at these default positions will be small, and so the value weighted pattern is more informative.

The interface allows one to isolate attention heads, shows the overall attention pattern, and allows one to explore the attention for individual tokens. Attention heads involved in K-composition are colored using the same scheme as above. We suggest trying to isolate these heads.

When destination = “Pot”, which tokens are the source?

Attention Pattern



Attention Heads (hover to focus, click to lock)

0:0 0:1 0:2 0:3 0:4 0:5 0:6 0:7 0:8 0:9 0:10 0:11

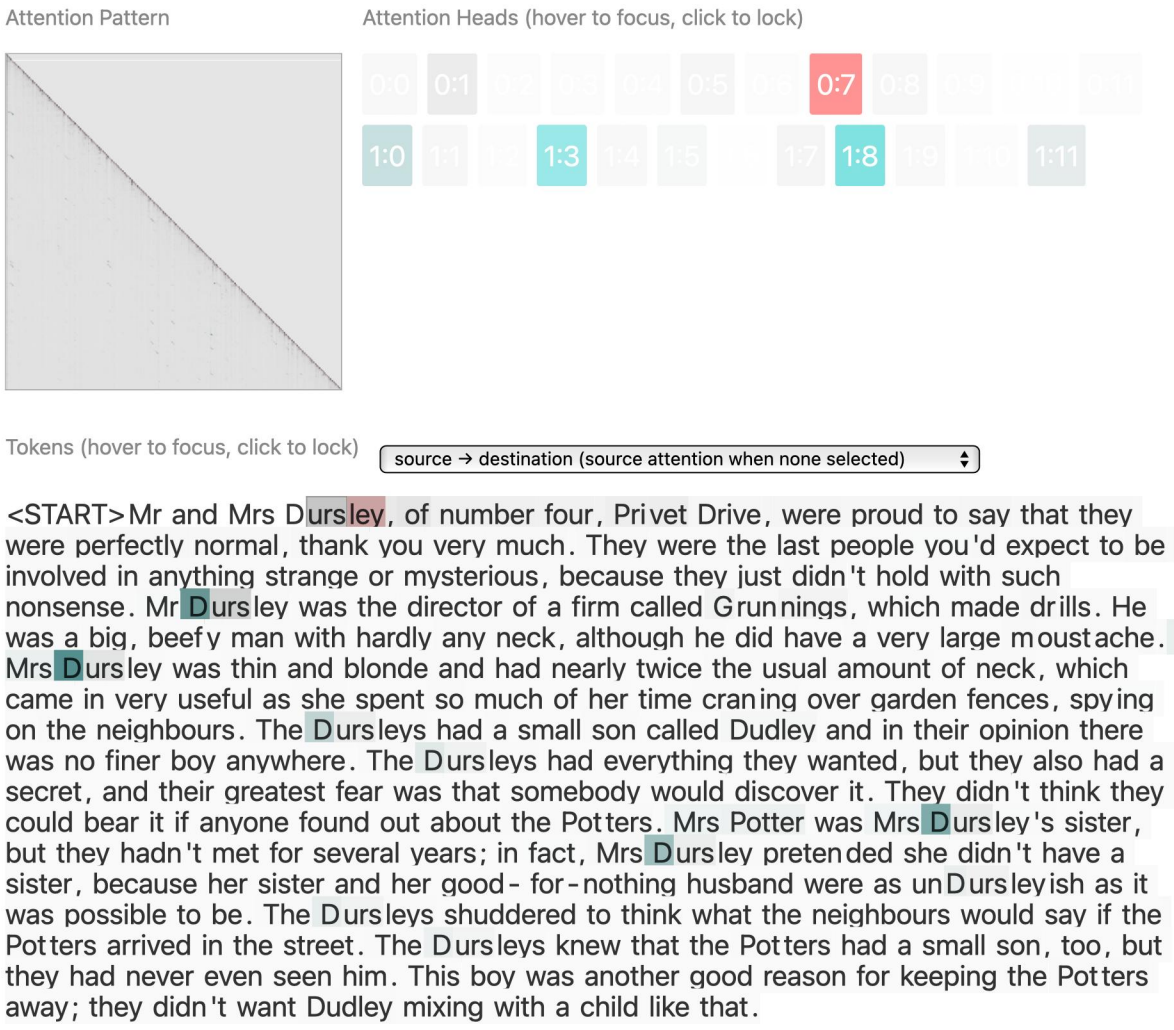
1:0 1:1 1:2 1:3 1:4 1:5 1:6 1:7 1:8 1:9 1:10 1:11

Tokens (hover to focus, click to lock)

source ← destination (destination attention when none selected)

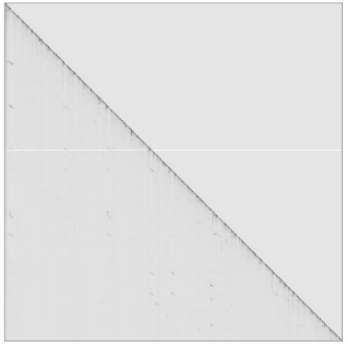
<START>Mr and Mrs Dursley, of number four, Privet Drive, were proud to say that they were perfectly normal, thank you very much. They were the last people you'd expect to be involved in anything strange or mysterious, because they just didn't hold with such nonsense. Mr Dursley was the director of a firm called Grunnings, which made drills. He was a big, beefy man with hardly any neck, although he did have a very large moustache. Mrs Dursley was thin and blonde and had nearly twice the usual amount of neck, which came in very useful as she spent so much of her time craning over garden fences, spying on the neighbours. The Dursleys had a small son called Dudley and in their opinion there was no finer boy anywhere. The Dursleys had everything they wanted, but they also had a secret, and their greatest fear was that somebody would discover it. They didn't think they could bear it if anyone found out about the Potters. Mrs Potter was Mrs Dursley's sister, but they hadn't met for several years; in fact, Mrs Dursley pretended she didn't have a sister, because her sister and her good-for-nothing husband were as unDursleyish as it was possible to be. The Dursleys shuddered to think what the neighbours would say if the Potters arrived in the street. The Dursleys knew that the Potters had a small son, too, but they had never even seen him. This boy was another good reason for keeping the Potters away; they didn't want Dudley mixing with a child like that.

When reading which token does the model attend to ‘urs’ -> everytime ‘D’ is the current tokens,but also when predicting the following token of ‘urs’ (but here we use more Head 0:7)

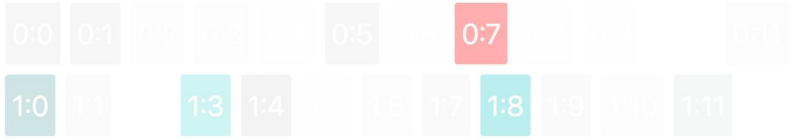


“When is “urs” the source token?

Attention Pattern



Attention Heads (hover to focus, click to lock)

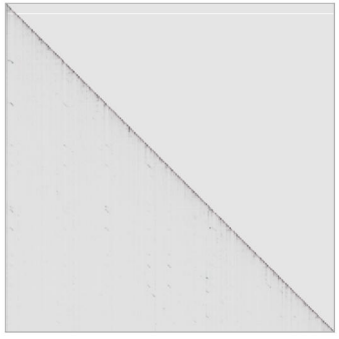


Tokens (hover to focus, click to lock)

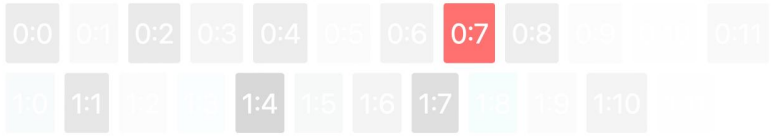
source → destination (source attention when none selected)

<START>Mr and Mrs Dursley, of number four, Privet Drive, were proud to say that they were perfectly normal, thank you very much. They were the last people you'd expect to be involved in anything strange or mysterious, because they just didn't hold with such nonsense. Mr Dursley was the director of a firm called Grunnings, which made drills. He was a big, beefy man with hardly any neck, although he did have a very large moustache. Mrs Dursley was thin and blonde and had nearly twice the usual amount of neck, which came in very useful as she spent so much of her time craning over garden fences, spying on the neighbours. The Dursleys had a small son called Dudley and in their opinion there was no finer boy anywhere. The Dursleys had everything they wanted, but they also had a secret, and their greatest fear was that somebody would discover it. They didn't think they could bear it if anyone found out about the Potters. Mrs Potter was Mrs Dursley's sister, but they hadn't met for several years; in fact, Mrs Dursley pretended she didn't have a sister, because her sister and her good-for-nothing husband were as unDursleyish as it was possible to be. The Dursleys shuddered to think what the neighbours would say if the Potters arrived in the street. The Dursleys knew that the Potters had a small son, too, but they had never even seen him. This boy was another good reason for keeping the Potters away; they didn't want Dudley mixing with a child like that.

Attention Pattern



Attention Heads (hover to focus, click to lock)

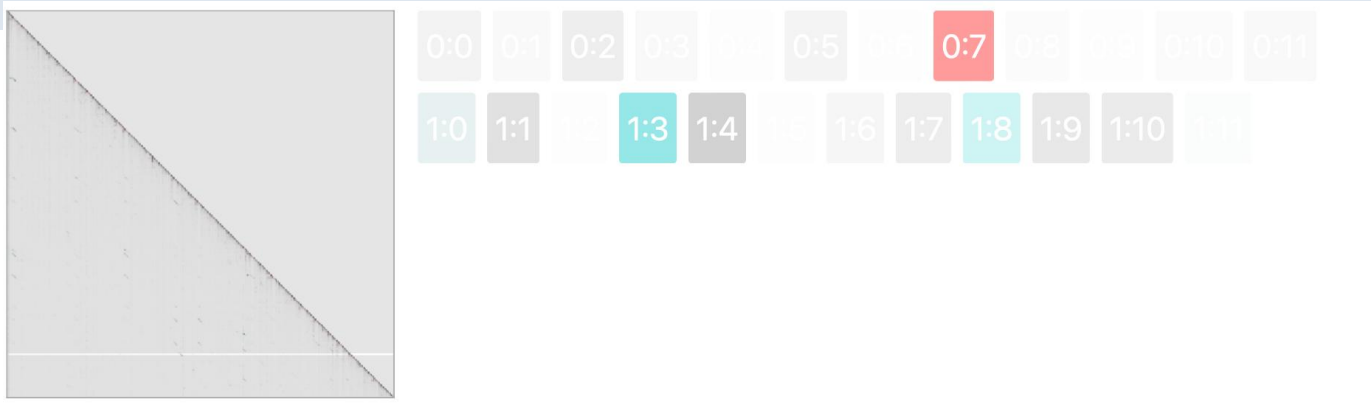


Tokens (hover to focus, click to lock)

source → destination (source attention when none selected)

<START>Mr and Mrs Dursley, of number four, Privet Drive, were proud to say that they were perfectly normal, thank you very much. They were the last people you'd expect to be involved in anything strange or mysterious, because they just didn't hold with such nonsense. Mr Dursley was the director of a firm called Grunnings, which made drills. He was a big, beefy man with hardly any neck, although he did have a very large moustache. Mrs Dursley was thin and blonde and had nearly twice the usual amount of neck, which came in very useful as she spent so much of her time craning over garden fences, spying on the neighbours. The Dursleys had a small son called Dudley and in their opinion there was no finer boy anywhere. The Dursleys had everything they wanted, but they also had a secret, and their greatest fear was that somebody would discover it. They didn't think they could bear it if anyone found out about the Potters. Mrs Potter was Mrs Dursley's sister, but they hadn't met for several years; in fact, Mrs Dursley pretended she didn't have a sister, because her sister and her good-for-nothing husband were as unDursleyish as it was possible to be. The Dursleys shuddered to think what the neighbours would say if the Potters arrived in the street. The Dursleys knew that the Potters had a small son, too, but they had never even seen him. This boy was another good reason for keeping the Potters away; they didn't want Dudley mixing with a child like that.

If you look carefully, you'll notice that the aqua colored "induction heads" often attend back to previous instances of the token which will come next. We'll investigate this more in the next section.



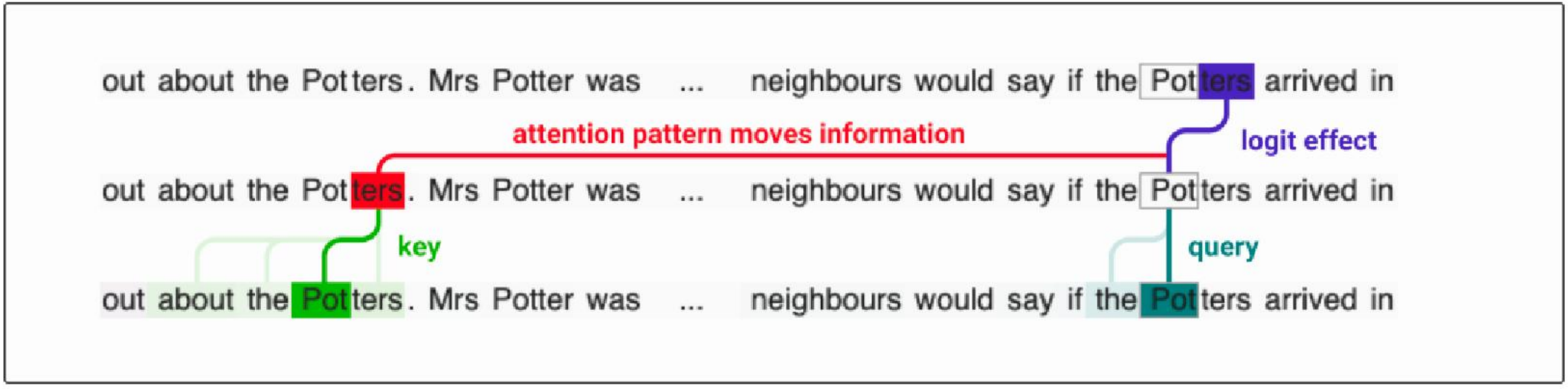
Tokens (hover to focus, click to lock) source ← destination (destination attention when none selected)

<START>Mr and Mrs Dursley, of number four, Privet Drive, were proud to say that they were perfectly normal, thank you very much. They were the last people you'd expect to be involved in anything strange or mysterious, because they just didn't hold with such nonsense. Mr Dursley was the director of a firm called Grunnings, which made drills. He was a big, beefy man with hardly any neck, although he did have a very large moustache. Mrs Dursley was thin and blonde and had nearly twice the usual amount of neck, which came in very useful as she spent so much of her time craning over garden fences, spying on the neighbours. The Dursleys had a small son called Dudley and in their opinion there was no finer boy anywhere. The Dursleys had everything they wanted, but they also had a secret, and their greatest fear was that somebody would discover it. They didn't think they could bear it if anyone found out about the Potters. Mrs Potter was Mrs Dursley's sister, but they hadn't met for several years; in fact, Mrs Dursley pretended she didn't have a sister, because her sister and her good-for-nothing husband were as unDursleyish as it was possible to be. The Dursleys shuddered to think what the neighbours would say if the Potters arrived in the street. The Dursleys knew that the Potters had a small son, too, but

Induction Heads

HOW INDUCTION HEADS WORK

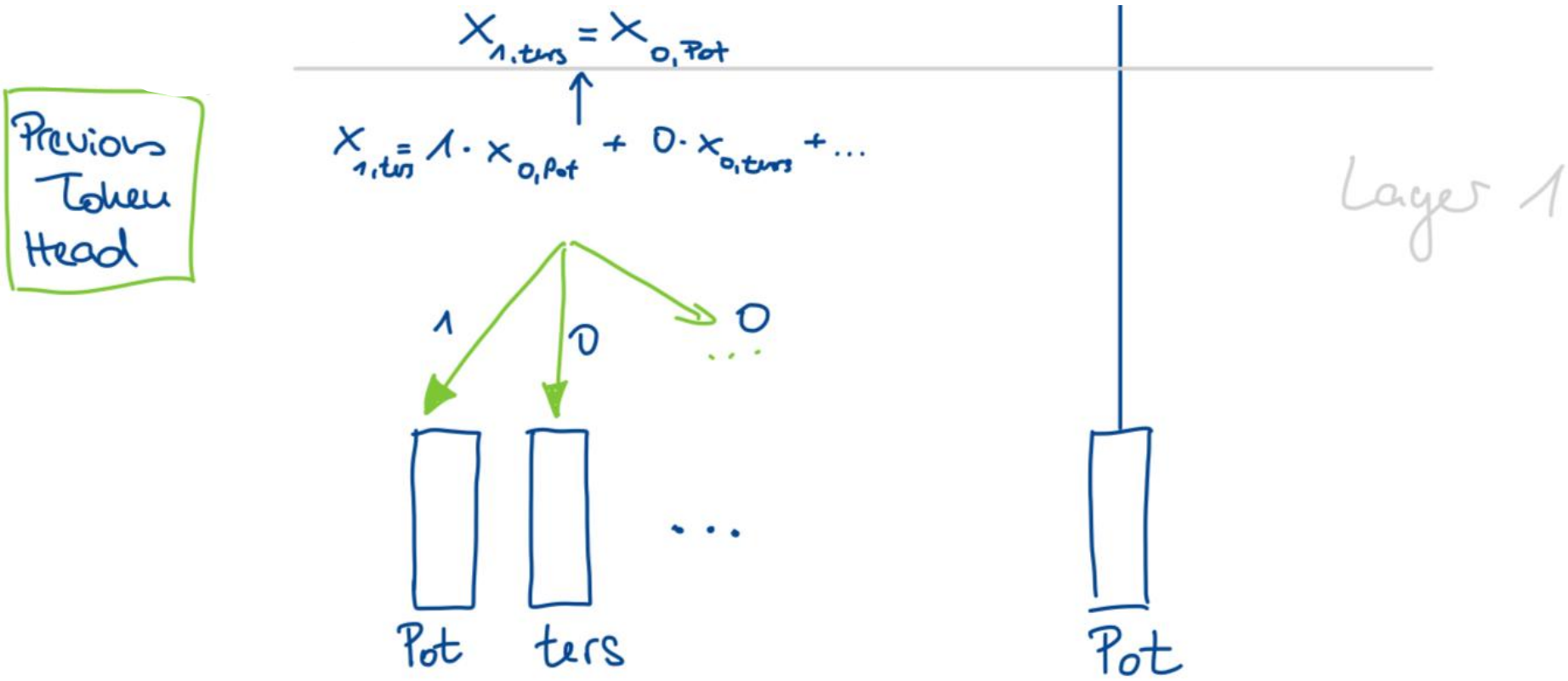
The central trick to induction heads is that the key is computed from tokens shifted one token back.[20] The query searches for "similar" key vectors, but because keys are shifted, finds the next token.



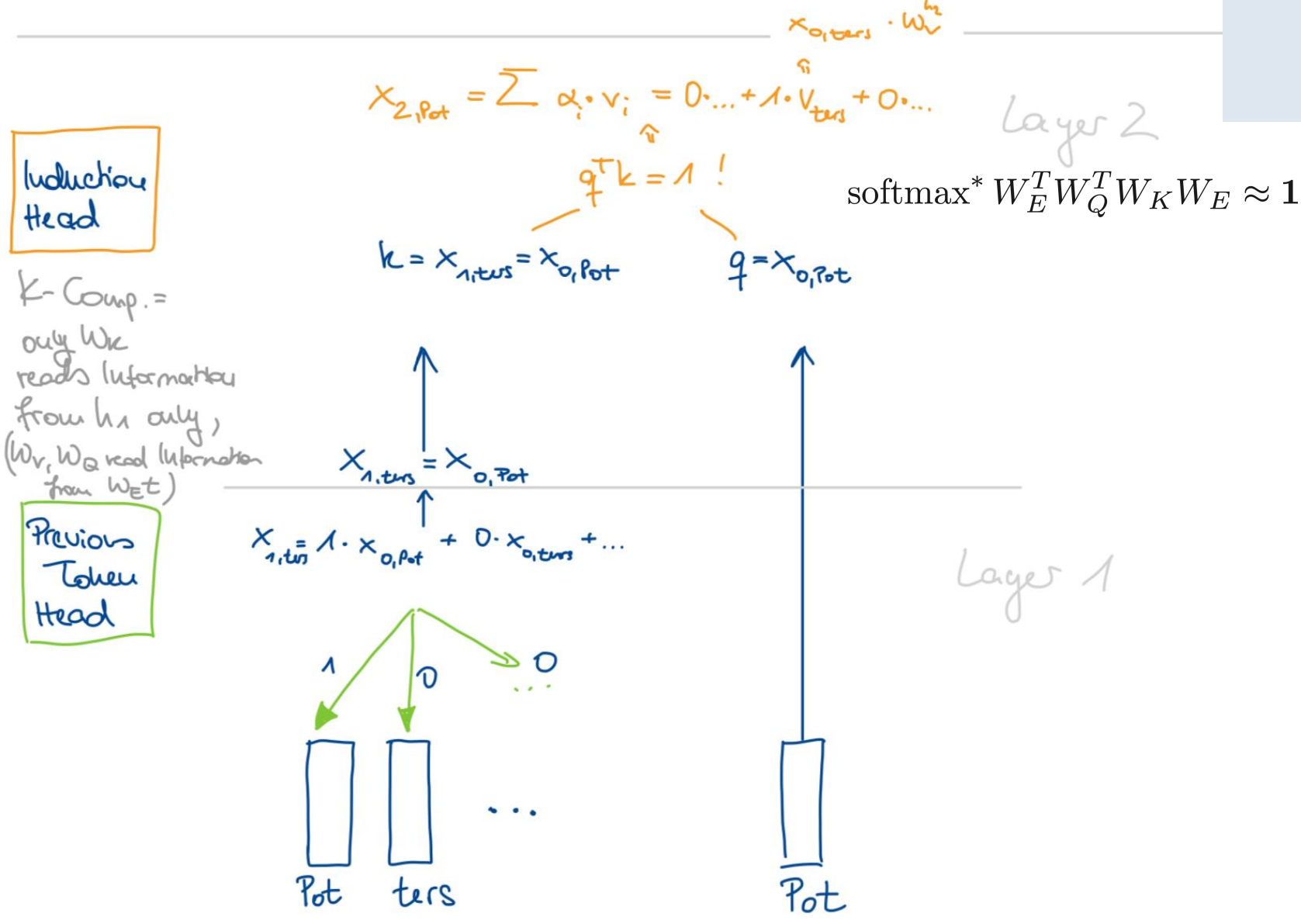
Induction Heads

The minimal way to create an induction head is to use K-composition with a previous token head to shift the key vector forward one token. This creates a term of the form $\mathrm{Id} \otimes A^{h-1} \otimes W$ in the QK-circuit (where A^{h-1} denotes an attention pattern attending to the previous token). If W matches cases where the tokens are the same — the QK version of a "copying matrix" — then this term will increase attention scores when the previous token before the source position is the same as the destination token.

Induction Heads



Induction Heads



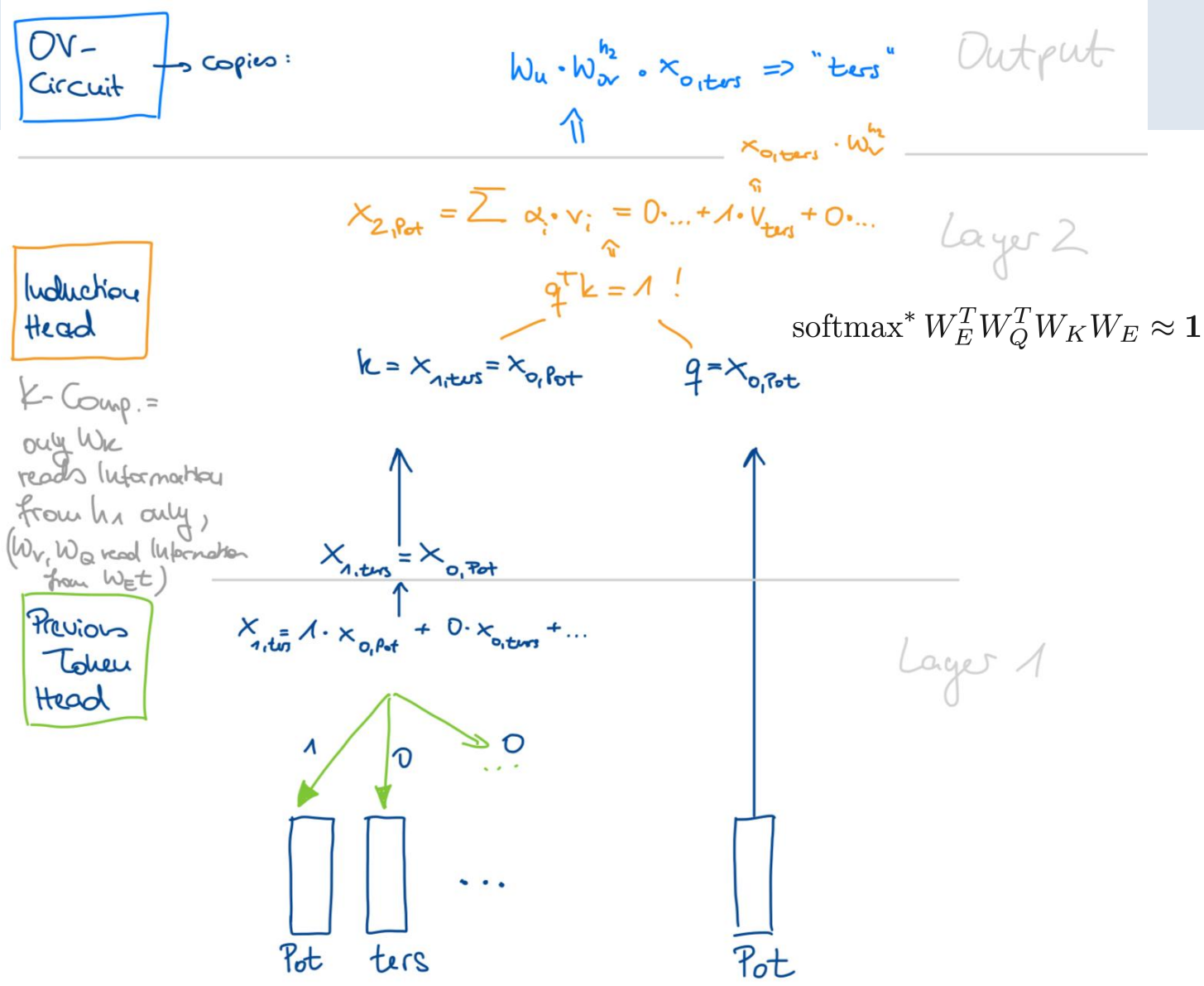
$$\text{softmax}^* \, W_E^T W_Q^T W_K W_E \approx \mathbf{1}$$

Induction Heads

Copy OV-Circuit:

$$W_U \cdot W_{OV}^{h_2} \cdot W_E \approx \mathbf{1}$$

--> simply enhances the logit of the token that “goes in”

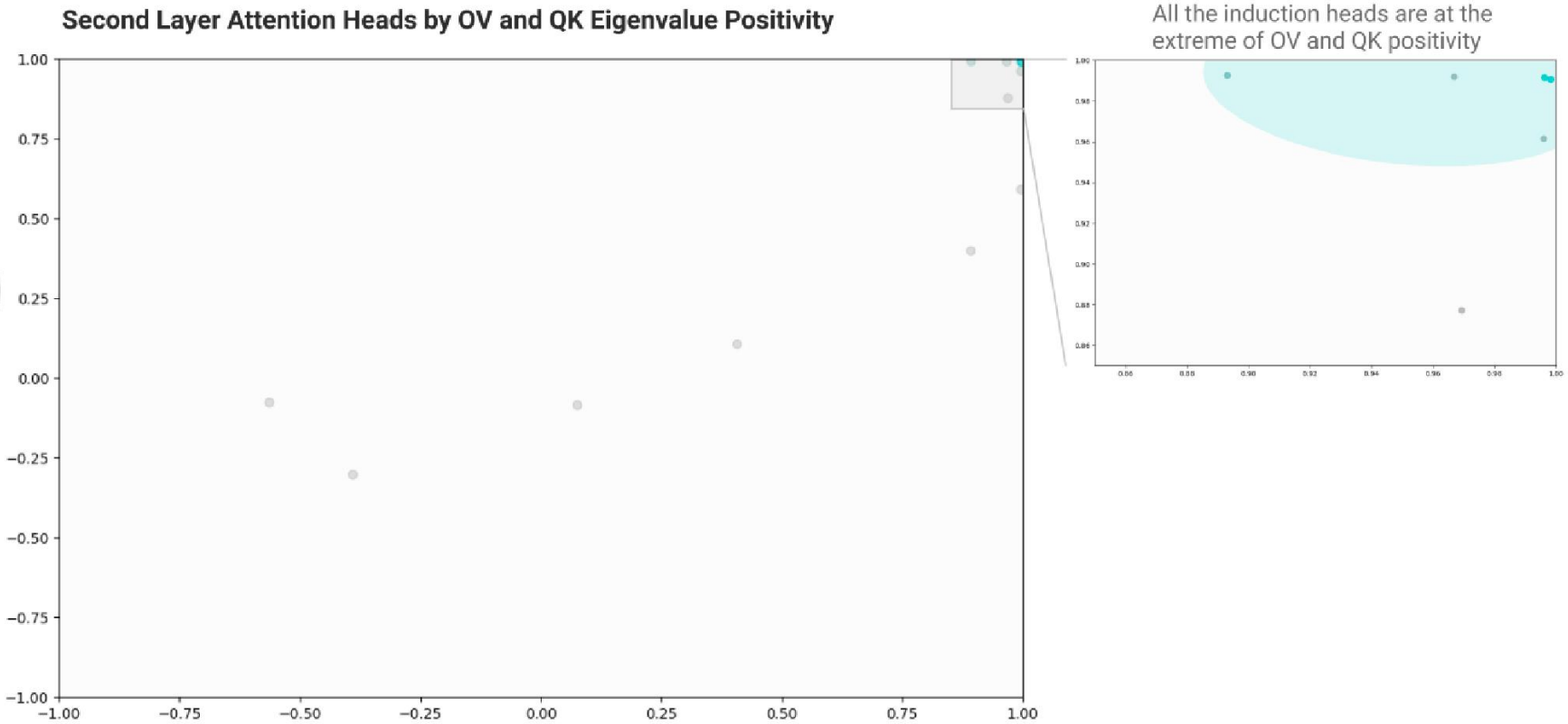


$$\text{softmax}^* \, W_E^T W_Q^T W_K W_E \approx \mathbf{1}$$

Find these Heads

Eigenvalue positivity of W in the QK term $\mathrm{Id} \otimes A^{h-1} \otimes W$

$$\left(\frac{\sum \lambda_i}{\sum |\lambda_i|} \quad \text{of} \quad W_E^T W_{QK}^h W_{OV}^{h-1} W_E \right)$$



Eigenvalue positivity of the OV circuit $\left(\frac{\sum \lambda_i}{\sum |\lambda_i|} \quad \text{of} \quad W_U W_{OV}^h W_E \right)$

Which terms are actually important?

$$= \underbrace{\mathrm{Id} \otimes W_U W_E}_{\text{"Direct path" term contributes to bigram statistics.}} + \underbrace{\sum_{h \in H_1 \cup H_2} A^h \otimes (W_U W_{OV}^h W_E)}_{\text{The individual attention head terms describe the effects of individual attention heads in linking input tokens to logits, similar to those we saw in the one layer model.}} + \underbrace{\sum_{h_2 \in H_2} \sum_{h_1 \in H_1} (A^{h_2} A^{h_1}) \otimes (W_U W_{OV}^{h_2} W_{OV}^{h_1} W_E)}_{\text{The virtual attention head terms correspond to V-composition of attention heads. They function a lot like individual attention heads, with their own attention patterns (the compositon of the heads patterns) and own OV matrix.}}$$

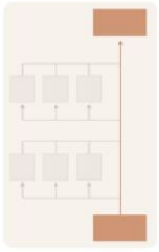
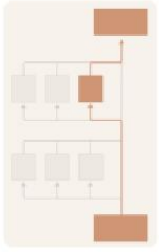
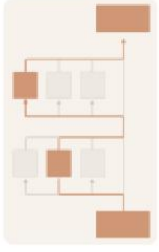
ALGORITHM FOR MEASURING MARGINAL LOSS REDUCTION OF NTH ORDER TERMS

Step 1: Run model, save all attention patterns.

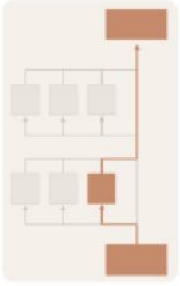
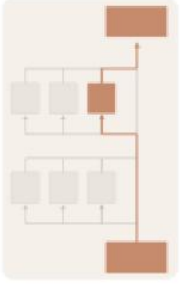
Step 2: Run model, forcing all attention patterns to be the version you recorded, and instead of adding attention head outputs to residual stream, save the output, and then replace it with a zero tensor of the same shape. Record resulting loss.

Step n: Run model, forcing all attention patterns to be the version you recorded, and instead of adding attention head outputs to residual stream, save the output, and replace it with the value you saved for this head last time. Record resulting loss.

Which terms are actually important?

Type	Example	Equation	Marginal Loss Reduction
direct path order 0		$W_U W_E$	**- 1.8 nats** relative to uniform predictions -1.8 nats/term (- 1.8 nats / 1 term)
individual attention head order 1		$A^h \otimes (W_U W_{OV}^h W_E)$	**- 5.2 nats** relative to only using direct path -0.2 nats/term (5.2 nats / 24 terms)
virtual attention head order 2		$(A^{h_2} A^{h_1}) \otimes (W_U W_{OV}^{h_2} W_{OV}^{h_1} W_E)$	**- 0.3 nats** relative to only using above -0.002 nats/term (0.3 nats / 144 terms)

Individual Attention Heads

Type	Example	Marginal Loss Reduction	Notes
Layer 1 Attention Heads		**- 0.05 nats** -0.004 nats/head relative to direct path + layer 2 **- 1.3 nats** -0.1 nats/head relative to direct path only	Relatively small effect, but keep in mind these heads also contribute to layer 2 QK circuits.
Layer 2 Attention Heads		**- 4.0 nats** -0.3 nats/head relative to direct path + layer 1 **- 5.2 nats** -0.4 nats/head relative to direct path only	**We'll focus on these.** Much larger effect. These heads are a lot more sophisticated than the layer 1 heads, since they can use layer 1 heads in their QK circuits.