

Choose a Transformer: Fourier or Galerkin

Shuhao Cao

Simon Storf

Summary

- The author combines 2 state-of-the-art methods:
 - Transformers
 - Fourier Neural Operators
- New variants of self-attention

Presentation Overview

- Prerequisites
 - Operator Learning
 - Transformers
- Paper
- Discussion

Prerequisites

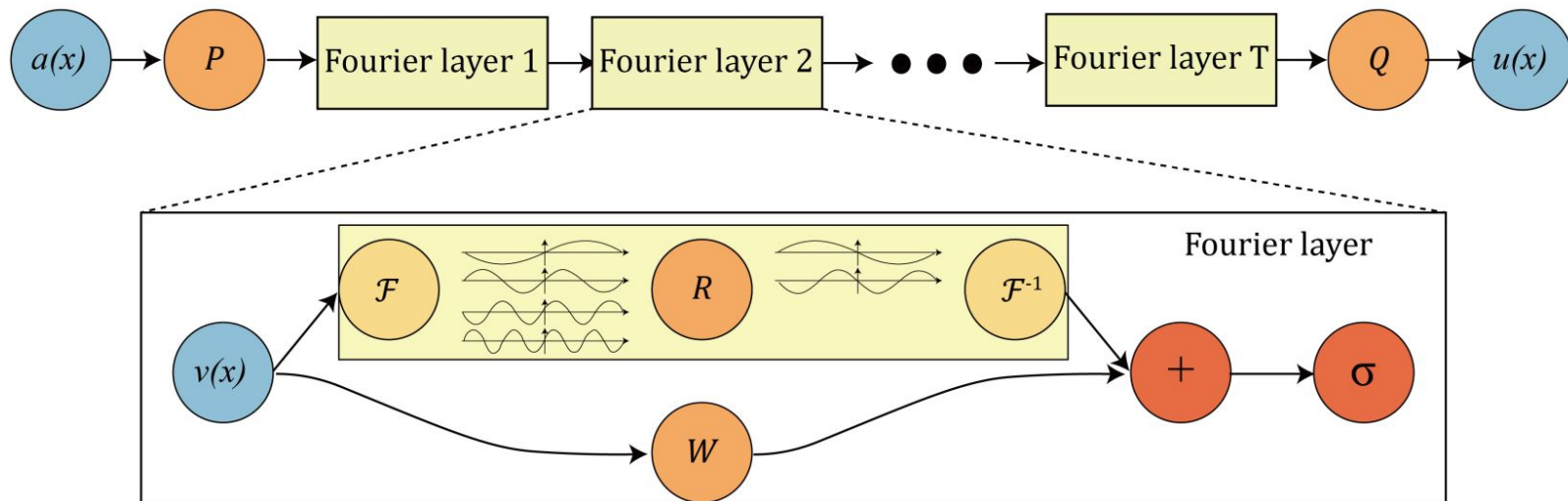
What is a Neural Operator?

Operator learning problem for parametric PDE

- Map from functions to functions
- Map from parameters + initial + boundary conditions to solution (or inverse)
- Approximate entire families of solutions for PDEs

Operator learning problem discretized

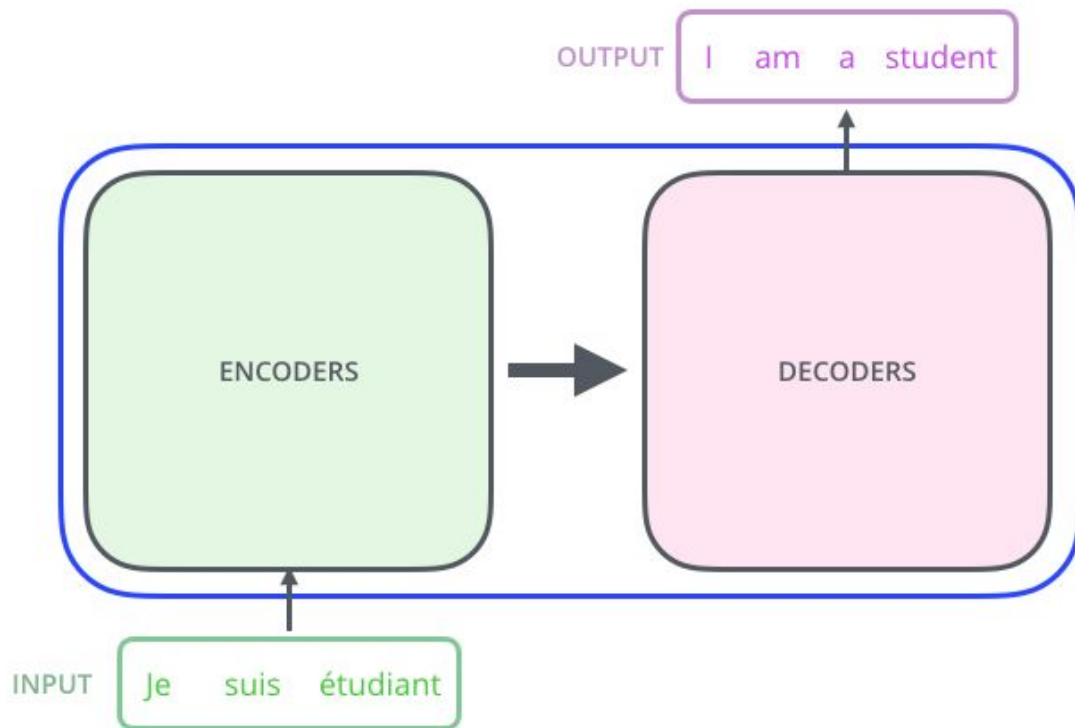
Fourier Neural Operator:



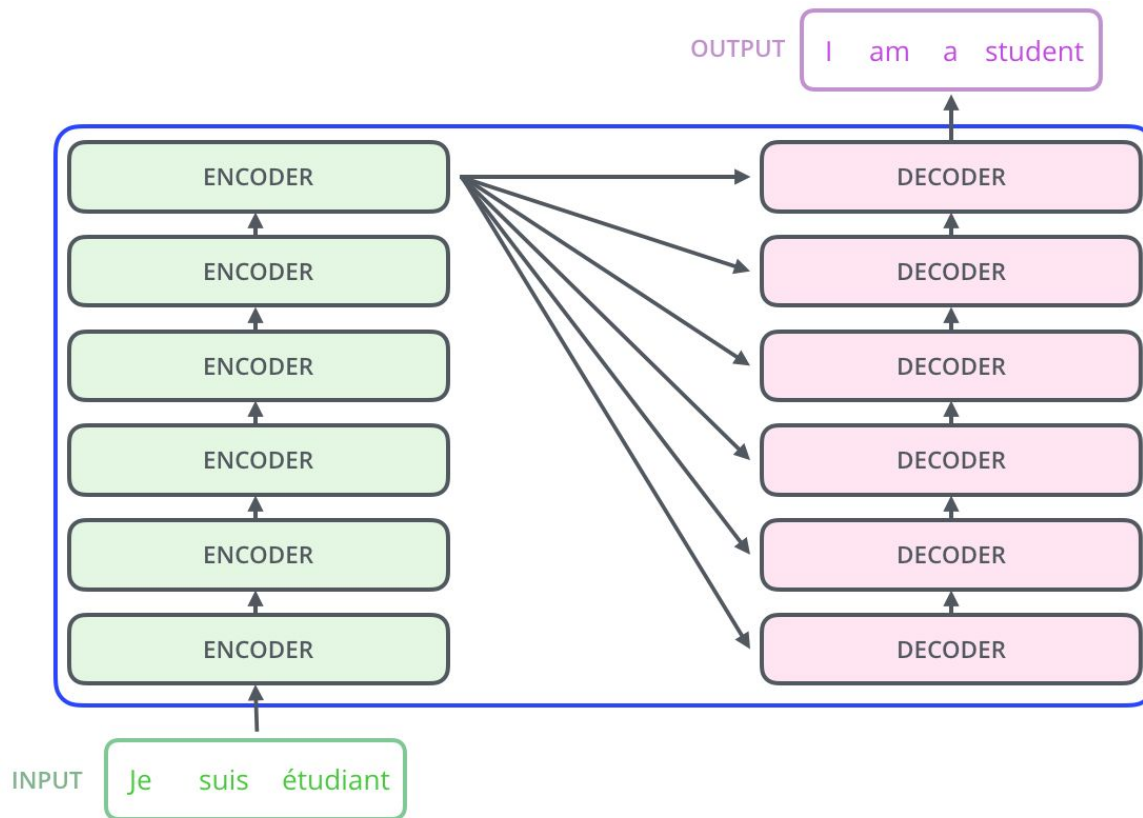
The discretized operator learning problem is a **seq2seq** problem

What is Self-Attention?

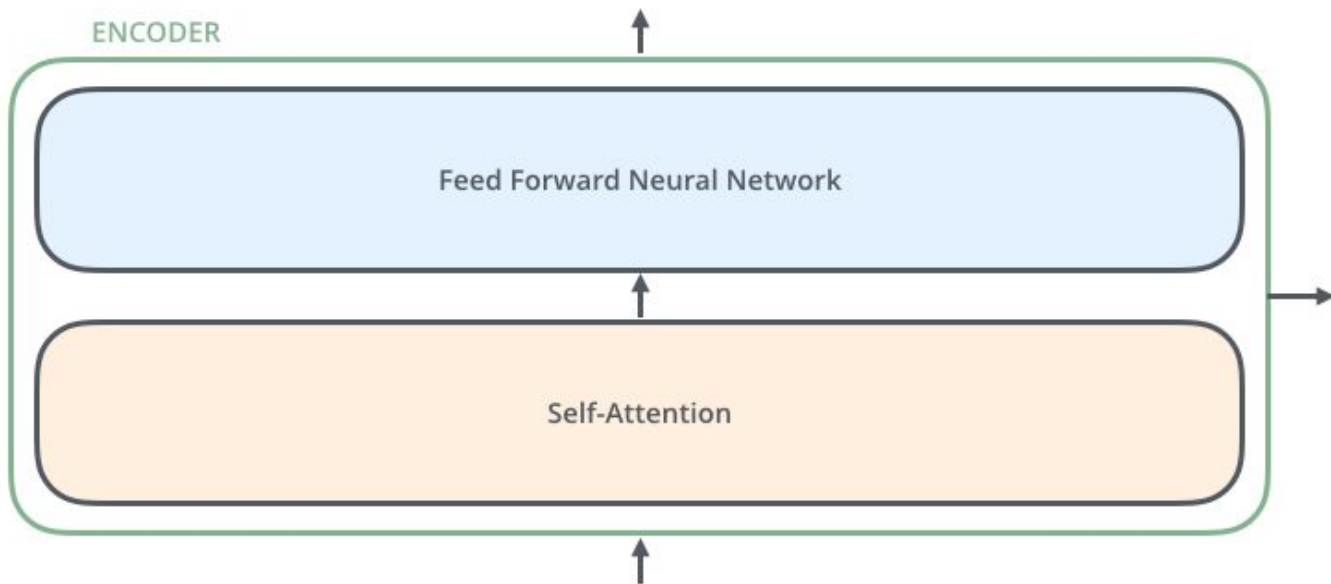
What is Self-Attention?



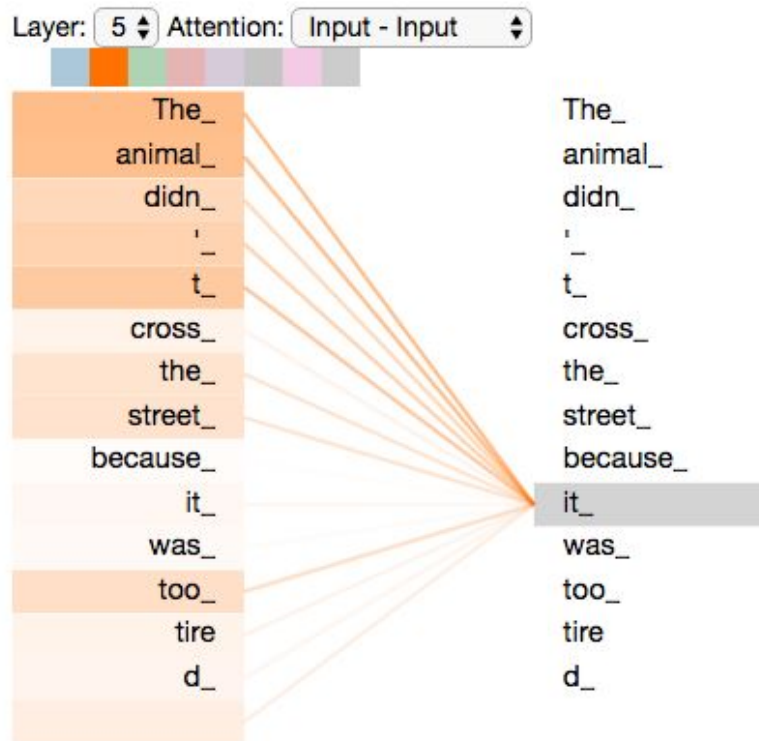
What is Self-Attention?



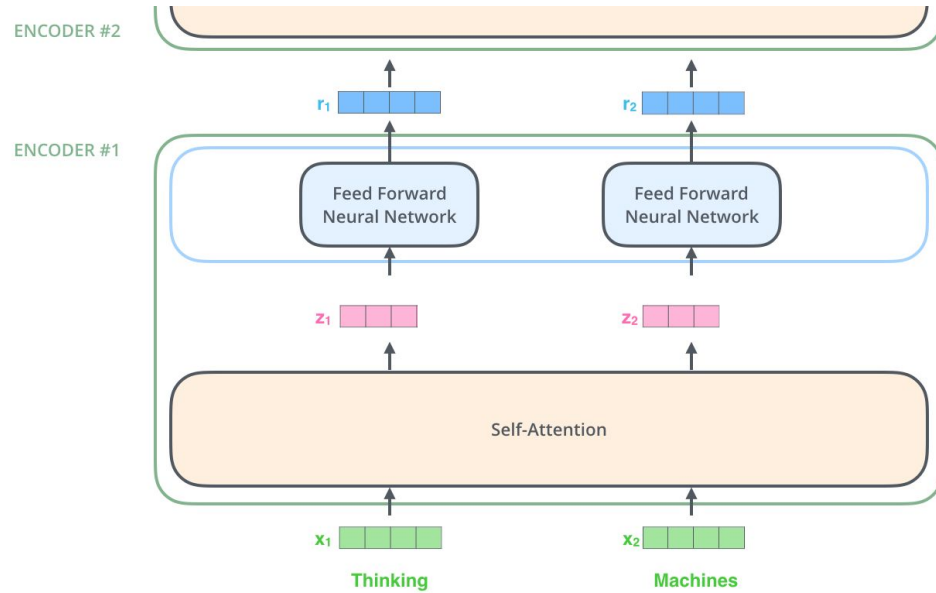
What is Self-Attention?



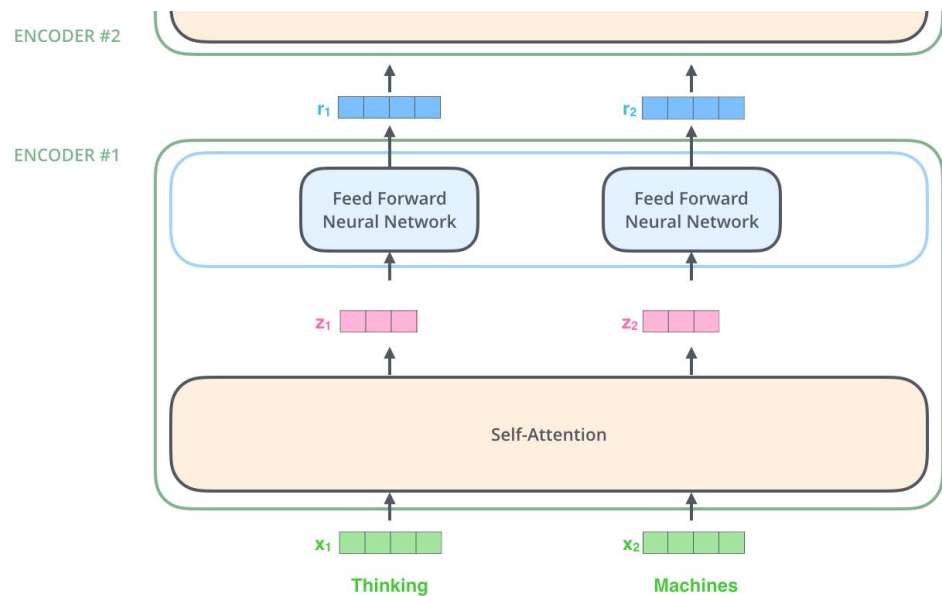
What is Self-Attention?



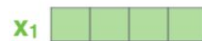
What is Self-Attention?



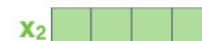
What is Self-Attention?



Thinking

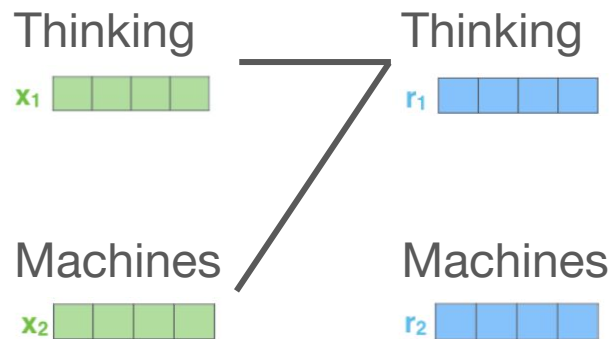
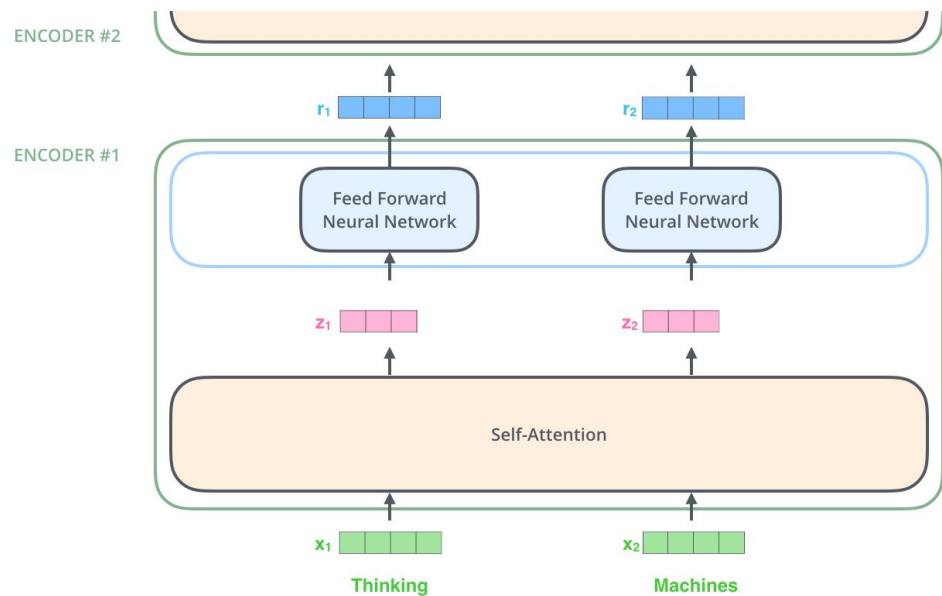


Machines

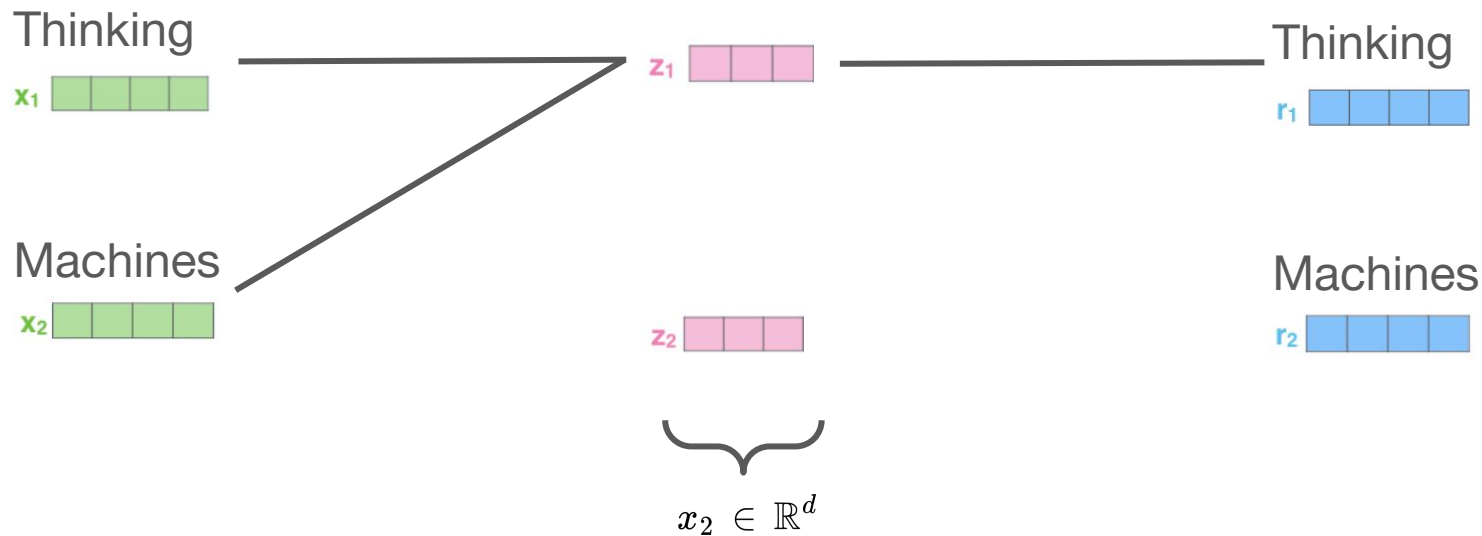


$$x_2 \in \mathbb{R}^l$$

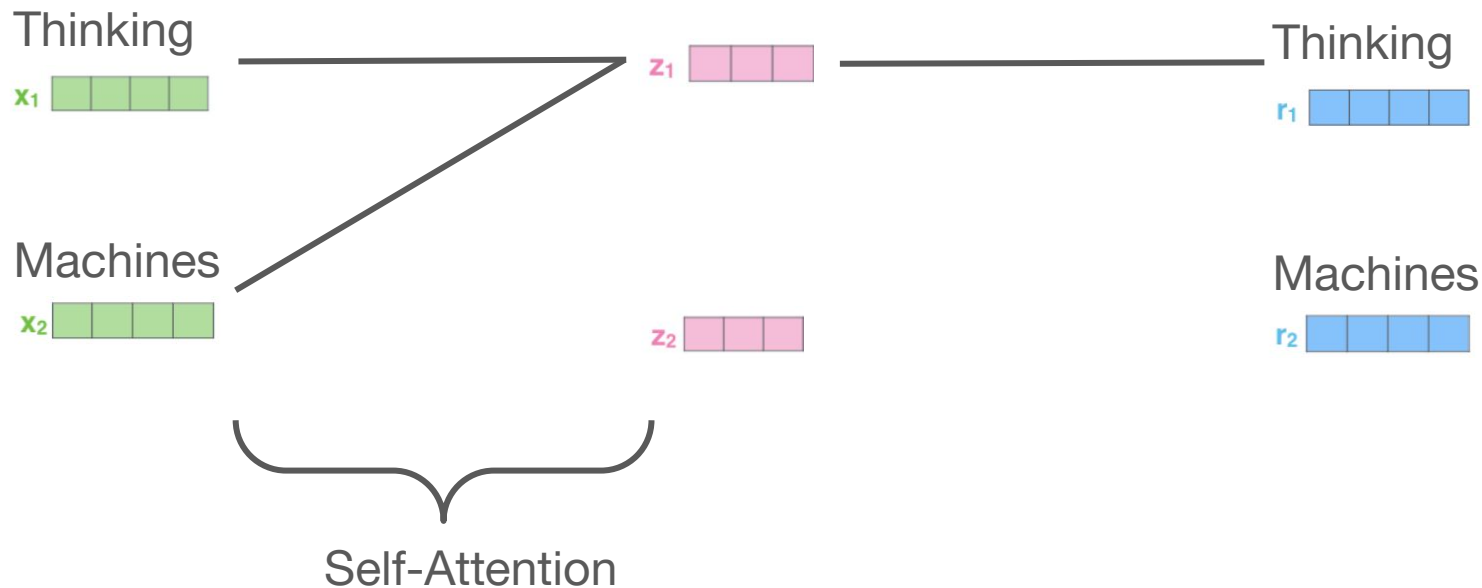
What is Self-Attention?



What happens mathematically inside Self-Attention?



What happens mathematically inside Self-Attention?



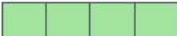
What happens mathematically inside Self-Attention?

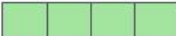
Input

Thinking

Machines

Embedding

X_1 

X_2 

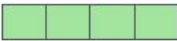
What happens mathematically inside Self-Attention?

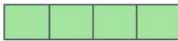
Input

Thinking

Machines

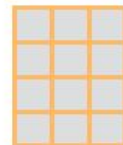
Embedding

X_1 

X_2 



W^Q

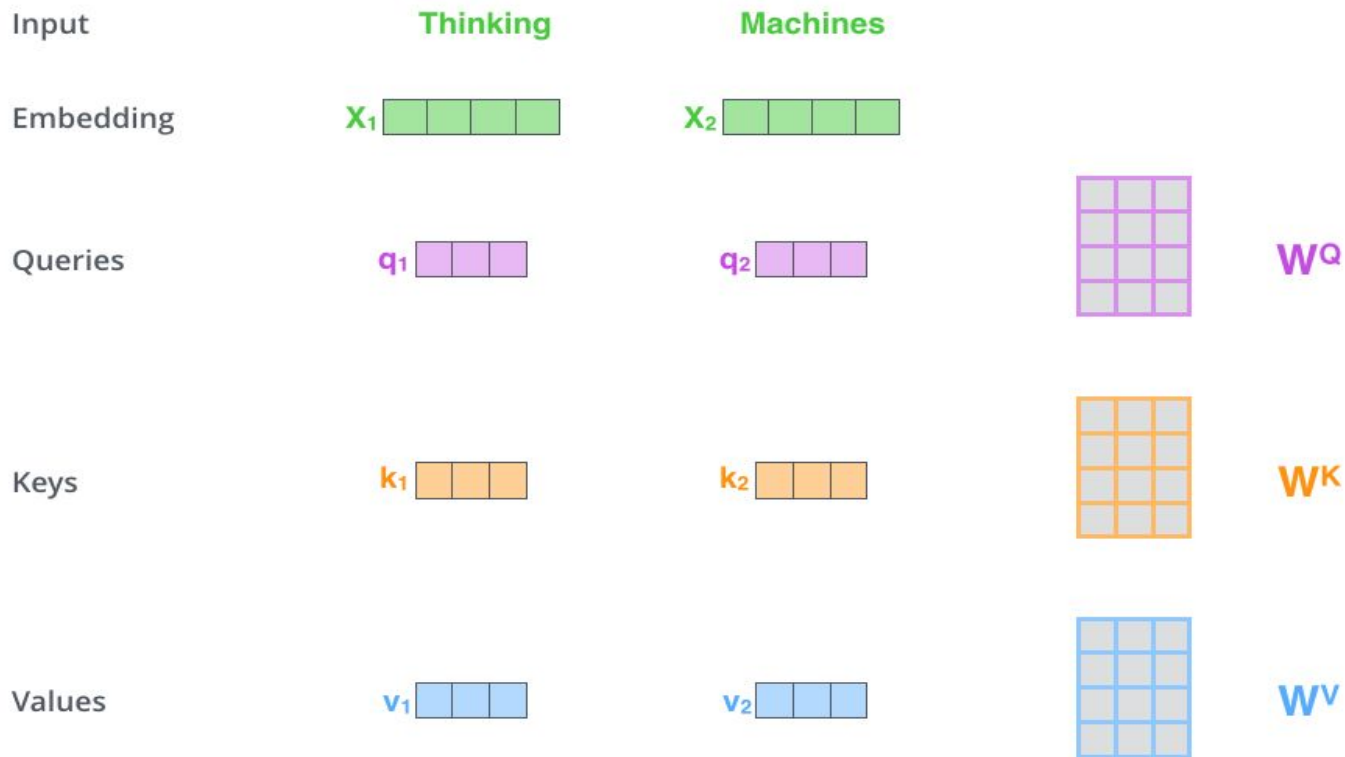


W^K



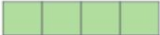
W^V


What happens mathematically inside Self-Attention?




What happens mathematically inside Self-Attention?


Thinking

x_1 

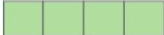
k_1 

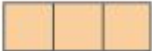
q_1 

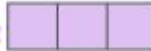
Thinking

z_1 

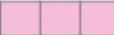
Machines

x_2 

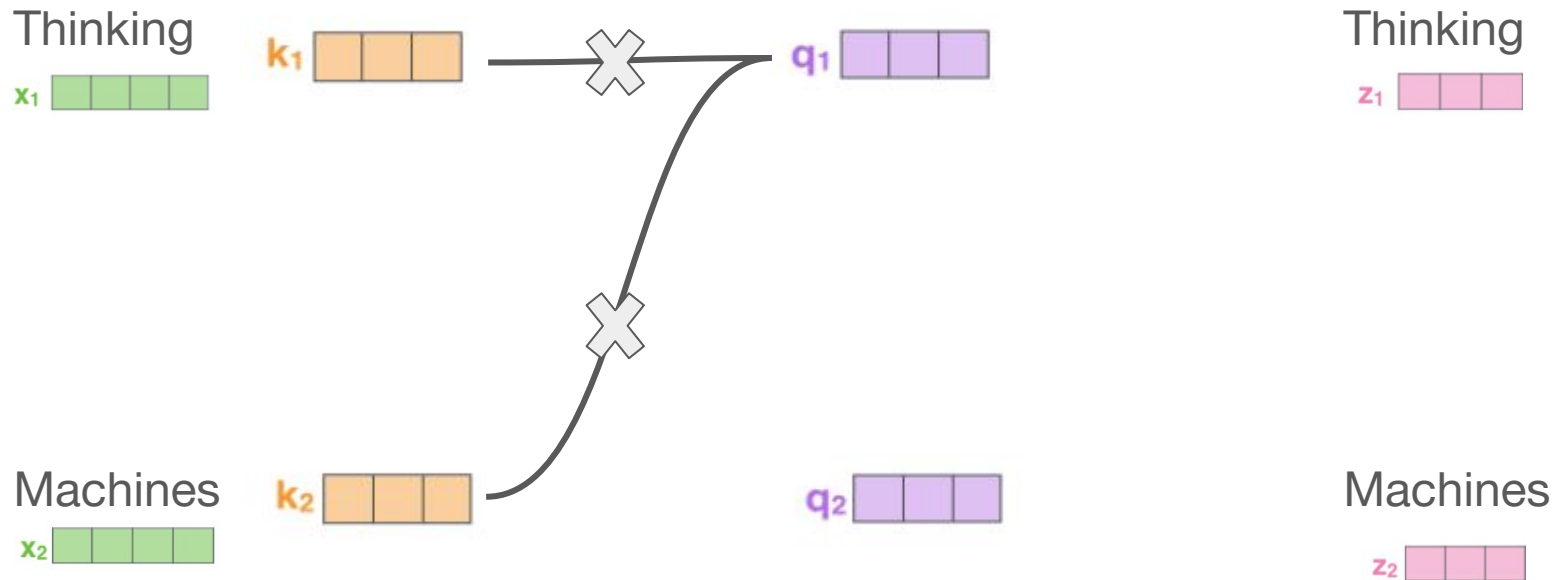
k_2 

q_2 

Machines

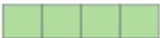
z_2 


What happens mathematically inside Self-Attention?




What happens mathematically inside Self-Attention?

Thinking

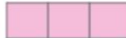
x_1 

k_1 



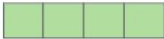
q_1 

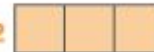
Thinking


z_1 



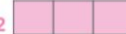
Machines

x_2 

k_2 

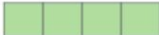
q_2 

Machines

z_2 

What happens mathematically inside Self-Attention?

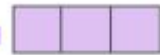
Thinking

x_1 


k_1



q_1

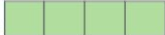


Thinking

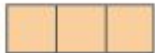
z_1 

Scale and Softmax

Machines

x_2 

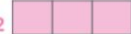
k_2



q_2

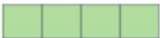



Machines

z_2 


What happens mathematically inside Self-Attention?

Thinking

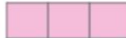
x_1 

k_1 

A_1

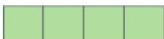
q_1 

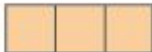
Thinking

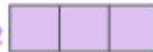
z_1 

A_2

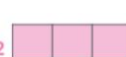
Machines

x_2 

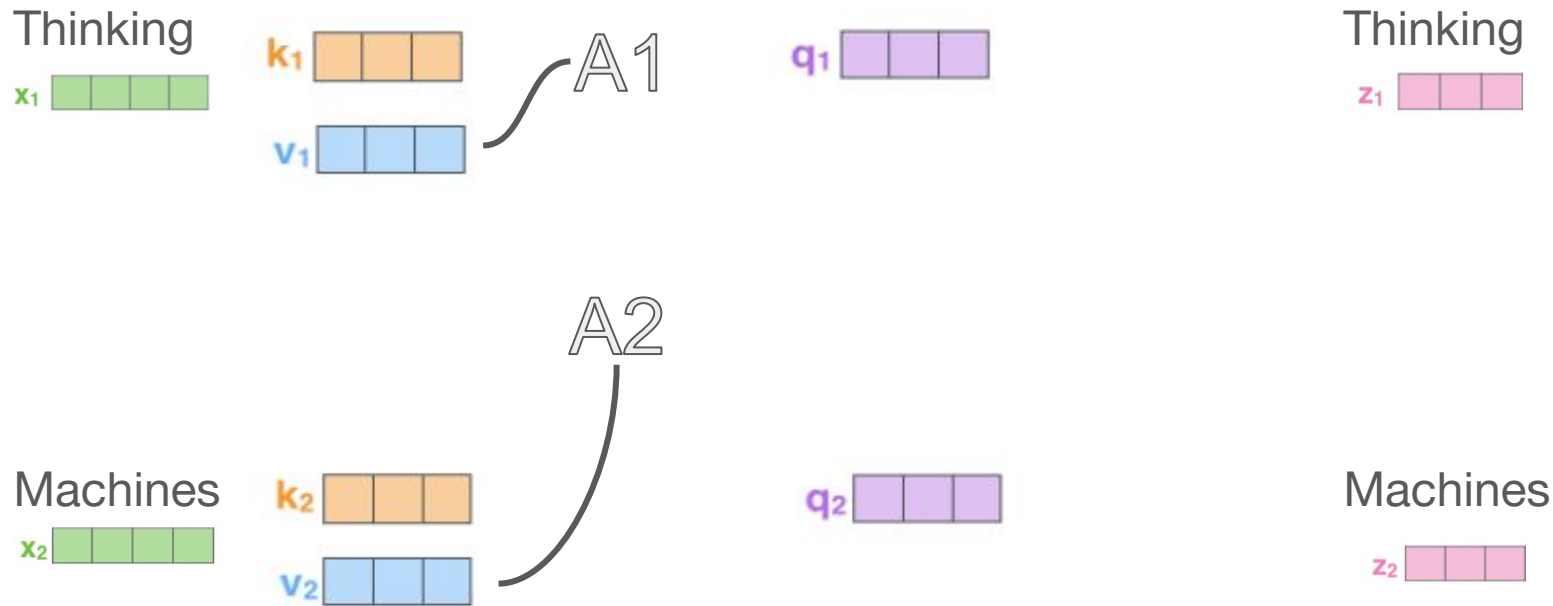
k_2 

q_2 

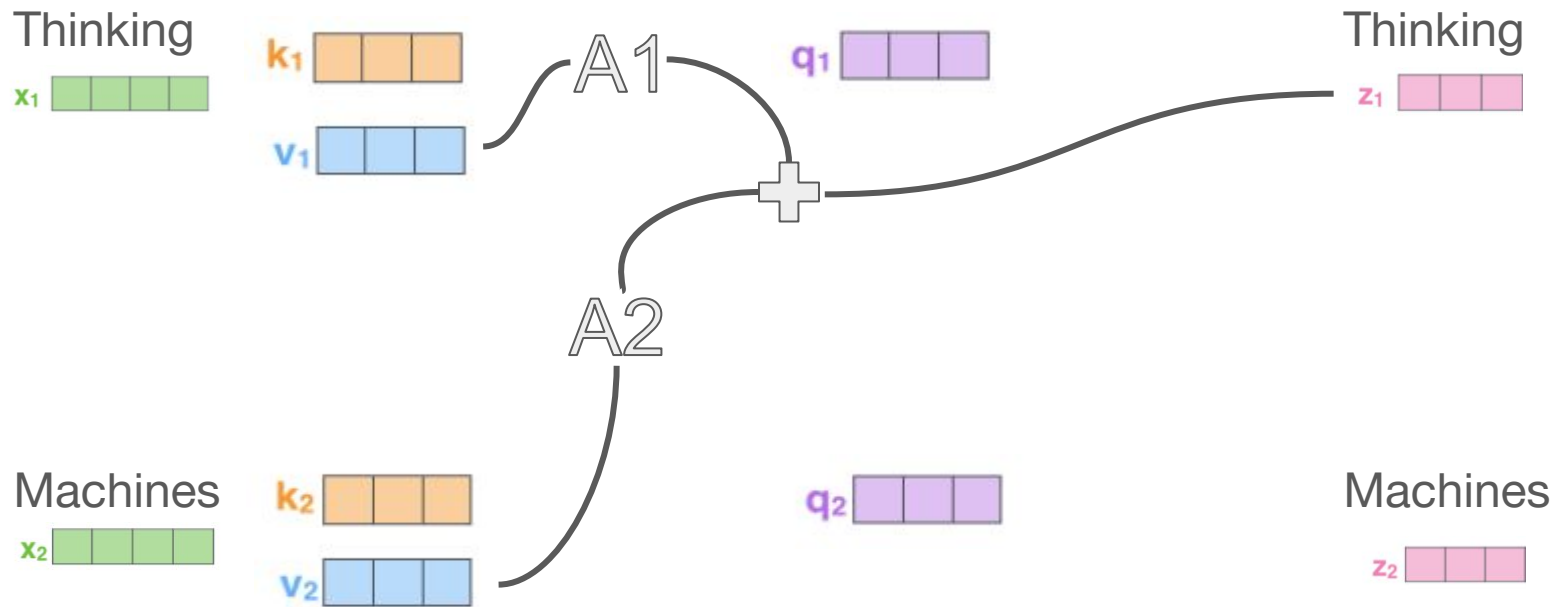
Machines

z_2 

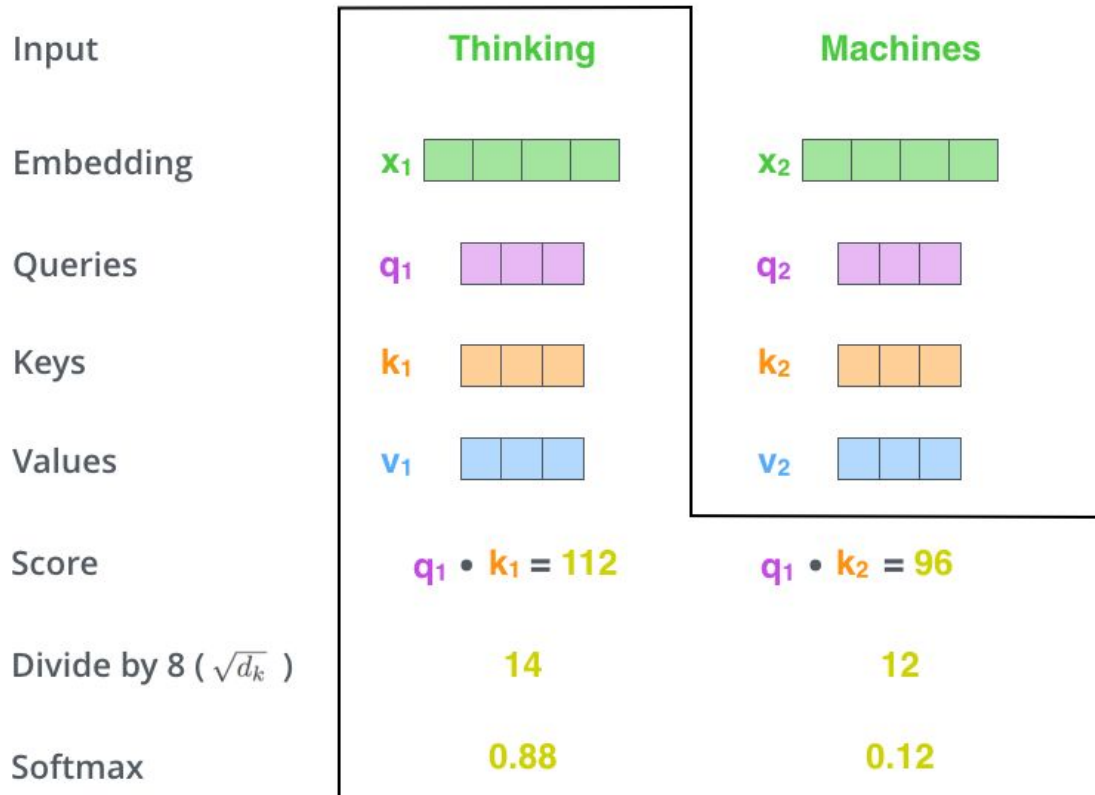
What happens mathematically inside Self-Attention?



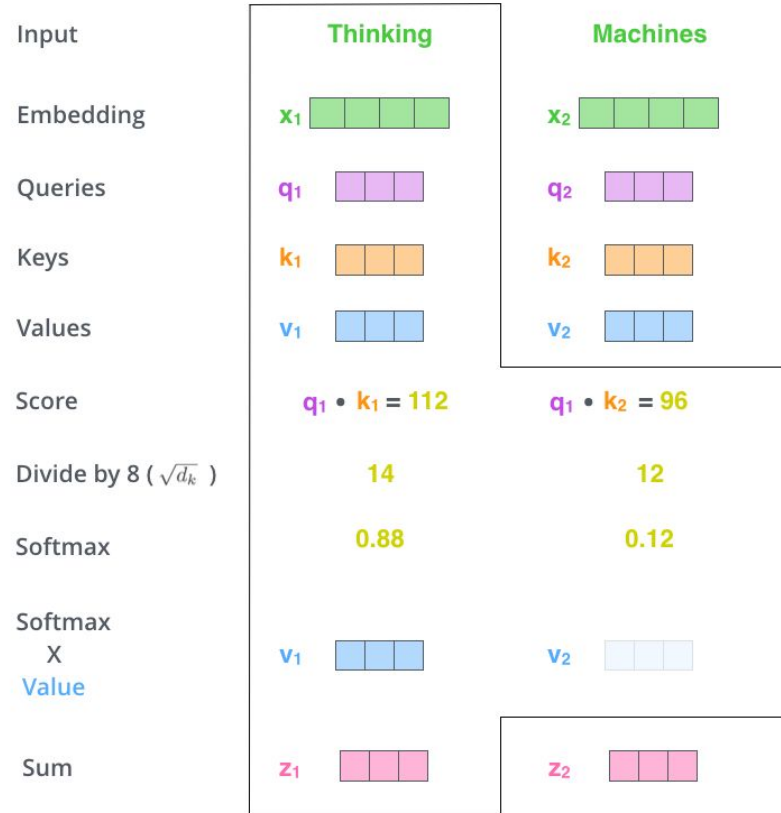
What happens mathematically inside Self-Attention?



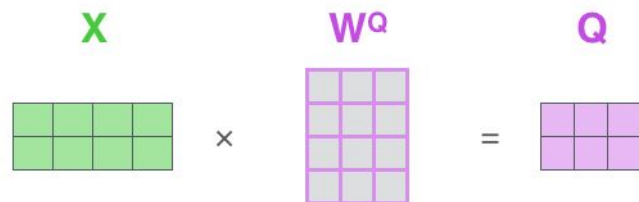
What happens mathematically inside Self-Attention?



What happens mathematically inside Self-Attention?



Self-Attention as Matrix Operations

$$\mathbf{X} \times \mathbf{W}^Q = \mathbf{Q}$$


A diagram illustrating the first matrix operation. On the left, a green 2x4 matrix labeled \mathbf{X} is multiplied by a purple 4x4 matrix labeled \mathbf{W}^Q . The result is a purple 2x4 matrix labeled \mathbf{Q} . The matrices are represented as grids of colored squares.

$$\mathbf{X} \times \mathbf{W}^K = \mathbf{K}$$

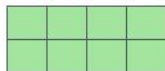

A diagram illustrating the second matrix operation. On the left, a green 2x4 matrix labeled \mathbf{X} is multiplied by an orange 4x4 matrix labeled \mathbf{W}^K . The result is an orange 2x4 matrix labeled \mathbf{K} . The matrices are represented as grids of colored squares.

$$\mathbf{X} \times \mathbf{W}^V = \mathbf{V}$$


A diagram illustrating the third matrix operation. On the left, a green 2x4 matrix labeled \mathbf{X} is multiplied by a blue 4x4 matrix labeled \mathbf{W}^V . The result is a blue 2x4 matrix labeled \mathbf{V} . The matrices are represented as grids of colored squares.

Self-Attention as Matrix Operations

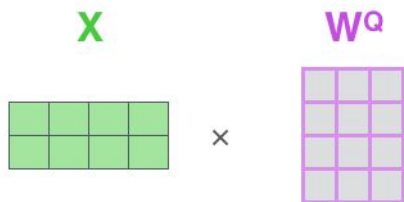
$$X \in \mathbb{R}^{n \times l}$$



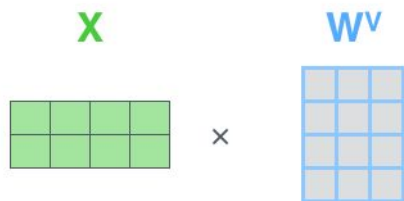
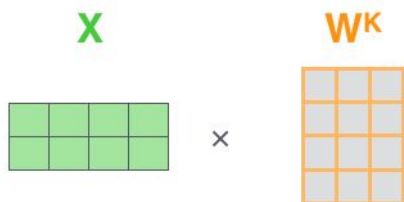
x

Self-Attention as Matrix Operations

$$X \in \mathbb{R}^{n \times l}$$

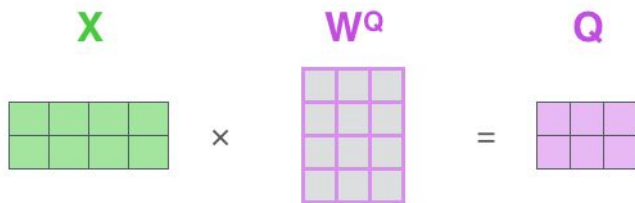


$$W^Q, W^K, W^V \in \mathbb{R}^{l \times d}$$



Self-Attention as Matrix Operations

$$X \in \mathbb{R}^{n \times l}$$



$$W^Q, W^K, W^V \in \mathbb{R}^{l \times d}$$



$$Q, K, V \in \mathbb{R}^{n \times d}$$



Self-Attention as Matrix Operations

$$\text{softmax} \left(\frac{\begin{matrix} \text{Q} \\ \begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \end{array} \end{matrix} \times \begin{matrix} \text{K}^T \\ \begin{array}{|c|} \hline \square \\ \hline \square \\ \hline \square \\ \hline \end{array} \end{matrix} \right) \begin{matrix} \text{V} \\ \begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \end{array} \end{matrix}$$
$$= \begin{matrix} \text{Z} \\ \begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \end{array} \end{matrix}$$

The diagram illustrates the self-attention mechanism as a sequence of matrix operations. It starts with a query matrix Q (purple, 2x3) and a key matrix K^T (orange, 3x3). These are multiplied together and the result is divided by the square root of the key dimension, √d_k. The result of this division is passed through a softmax function. The output of the softmax is then multiplied by the value matrix V (blue, 2x3) to produce the final attention matrix Z (pink, 2x3).

Self-Attention as Matrix Operations

$$\text{softmax} \left(\frac{\begin{matrix} \mathbf{Q} \\ \text{2x3 grid} \end{matrix} \times \begin{matrix} \mathbf{K}^T \\ \text{3x2 grid} \end{matrix}}{\sqrt{d_k}} \right) \begin{matrix} \mathbf{V} \\ \text{2x3 grid} \end{matrix}$$

$$= \begin{matrix} \mathbf{Z} \\ \text{2x3 grid} \end{matrix}$$

Input

Embedding

Queries

Keys

Values

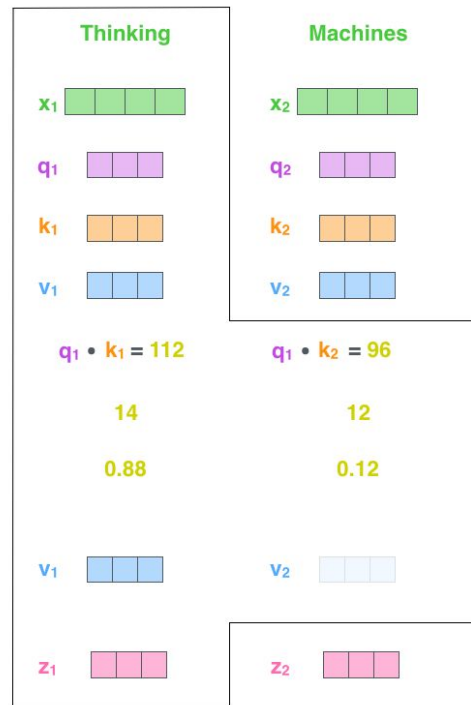
Score

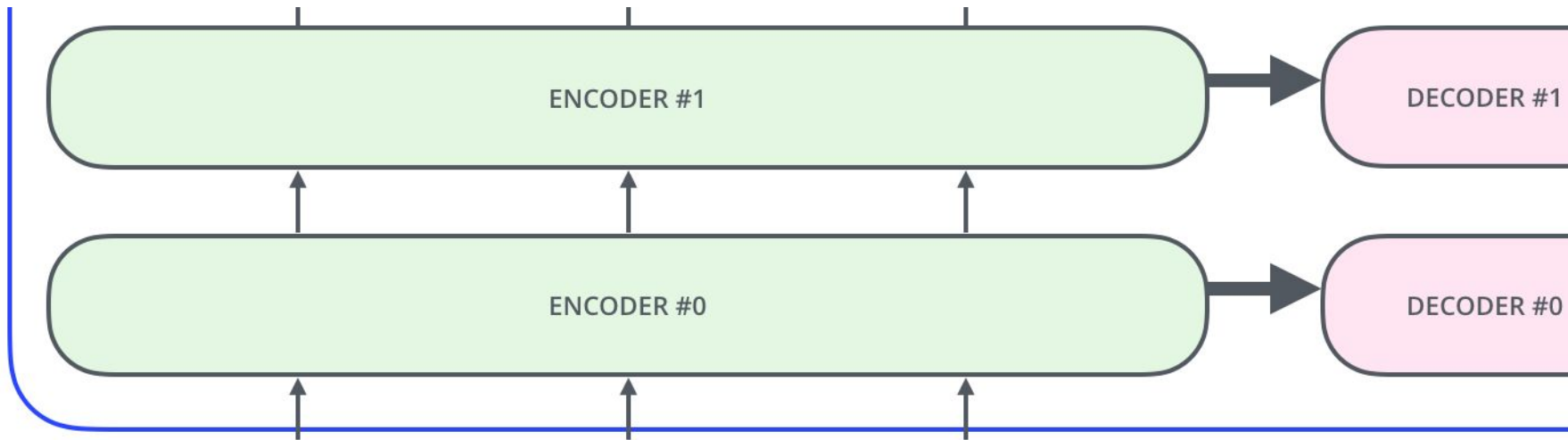
Divide by 8 ($\sqrt{d_k}$)

Softmax

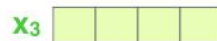
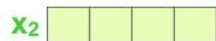
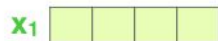
Softmax
X
Value

Sum





EMBEDDING
WITH TIME
SIGNAL

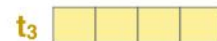
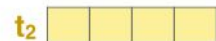
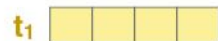


=

=

=

POSITIONAL
ENCODING

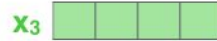
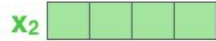
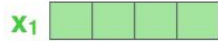


+

+

+

EMBEDDINGS



INPUT

Je

suis

étudiant

Paper

The Problem

$$\text{softmax} \left(\frac{\begin{matrix} \mathbf{Q} \\ \begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \end{array} \end{matrix} \times \begin{matrix} \mathbf{K}^T \\ \begin{array}{|c|c|} \hline \square & \square \\ \hline \square & \square \\ \hline \end{array} \end{matrix} \right) \mathbf{V}$$

The diagram illustrates a matrix operation. On the left, a purple 2x3 matrix labeled \mathbf{Q} is multiplied by an orange 3x2 matrix labeled \mathbf{K}^T . The result of this multiplication is divided by $\sqrt{d_k}$. This entire expression is enclosed in large parentheses, with the word "softmax" written to the left. To the right of the parentheses is a blue 2x2 matrix labeled \mathbf{V} .

The Problem

$$\text{softmax} \left(\frac{\begin{matrix} \mathbf{Q} \\ \begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \end{array} \end{matrix} \times \begin{matrix} \mathbf{K}^T \\ \begin{array}{|c|c|} \hline \square & \square \\ \hline \square & \square \\ \hline \end{array} \end{matrix} \right) \begin{matrix} \mathbf{V} \\ \begin{array}{|c|c|} \hline \square & \square \\ \hline \square & \square \\ \hline \end{array} \end{matrix}$$

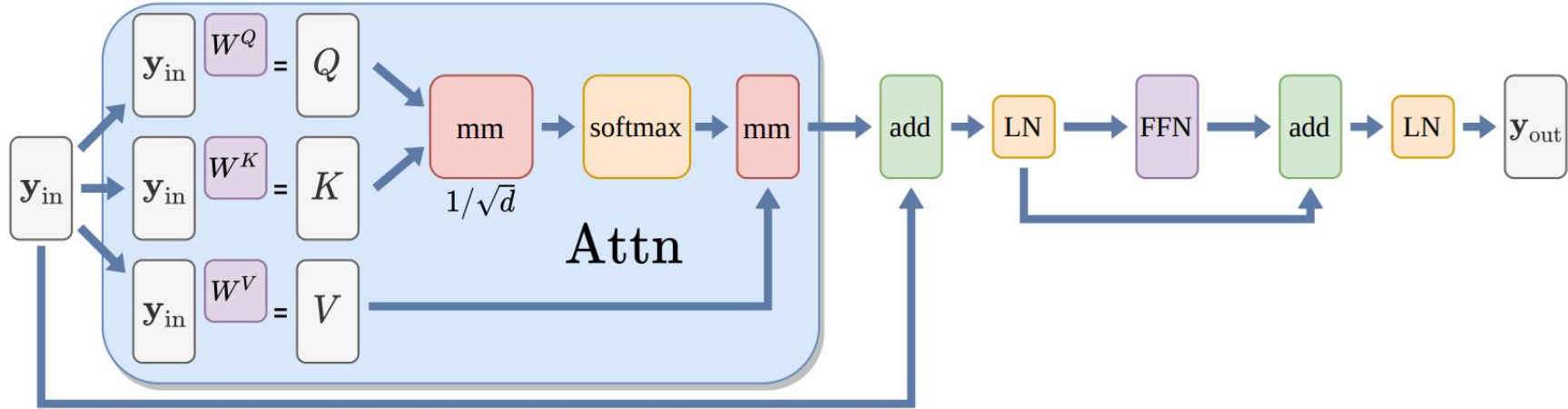
$$\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{n \times d}$$

The Problem

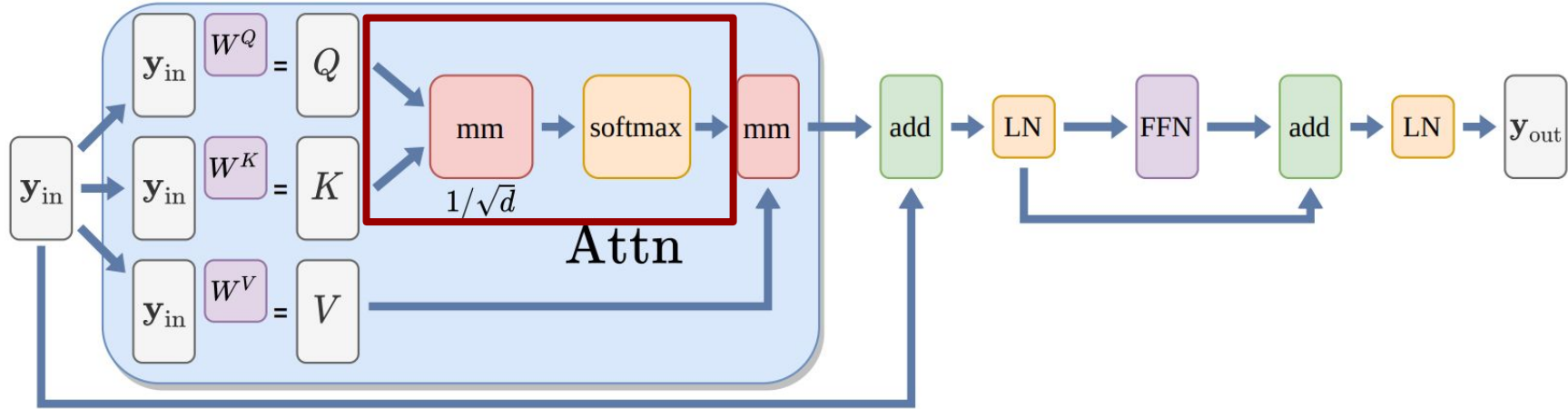
$$\text{softmax} \left(\frac{\begin{matrix} \text{Q} \\ \begin{matrix} \square & \square & \square \\ \square & \square & \square \end{matrix} \end{matrix} \times \begin{matrix} \text{K}^T \\ \begin{matrix} \square & \square \\ \square & \square \\ \square & \square \end{matrix} \end{matrix}}{\sqrt{d_k}} \right) \begin{matrix} \text{V} \\ \begin{matrix} \square & \square & \square \\ \square & \square & \square \end{matrix} \end{matrix}$$

$$Q, K, V \in \mathbb{R}^{n \times d} \quad O(n^2 d)$$

The Problem



The Problem



The Solution

$$\text{softmax} \left(\frac{\begin{matrix} \text{Q} \\ \begin{array}{|c|c|c|} \hline & & \\ \hline & & \\ \hline \end{array} \times \begin{matrix} \text{K}^T \\ \begin{array}{|c|c|} \hline & \\ \hline & \\ \hline & \\ \hline \end{array} \end{matrix}}{\sqrt{d_k}} \right) \begin{matrix} \text{V} \\ \begin{array}{|c|c|c|} \hline & & \\ \hline & & \\ \hline \end{array} \end{matrix}$$

The diagram illustrates the matrix multiplication step of the softmax function. It shows a purple 2x3 matrix labeled 'Q' multiplied by an orange 3x2 matrix labeled 'K^T'. The result is a blue 2x3 matrix labeled 'V'. The entire operation is enclosed in large parentheses, with a horizontal line and the square root of d_k below it, indicating a normalization step.

The Solution

$$\left(\begin{array}{c} \mathbf{Q} \\ \begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \end{array} \end{array} \right) \times \begin{array}{c} \mathbf{K}^T \\ \begin{array}{|c|c|} \hline \square & \square \\ \hline \square & \square \\ \hline \square & \square \\ \hline \end{array} \end{array} \right) \begin{array}{c} \mathbf{V} \\ \begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \end{array} \end{array}$$

The diagram illustrates a matrix multiplication operation. On the left, a purple 2x3 matrix labeled \mathbf{Q} is multiplied by an orange 3x2 matrix labeled \mathbf{K}^T . The result is a blue 2x3 matrix labeled \mathbf{V} . The matrices are represented by grids of squares, and the multiplication is indicated by a large right parenthesis on the left, a multiplication sign in the middle, and a large left parenthesis on the right.

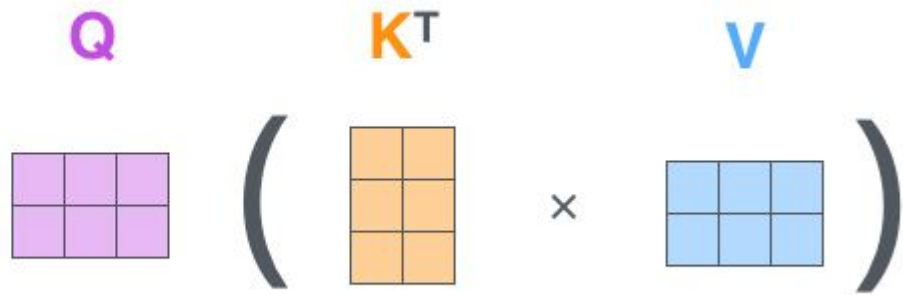
The Solution

$$Q = \left(K^T \times V \right)$$

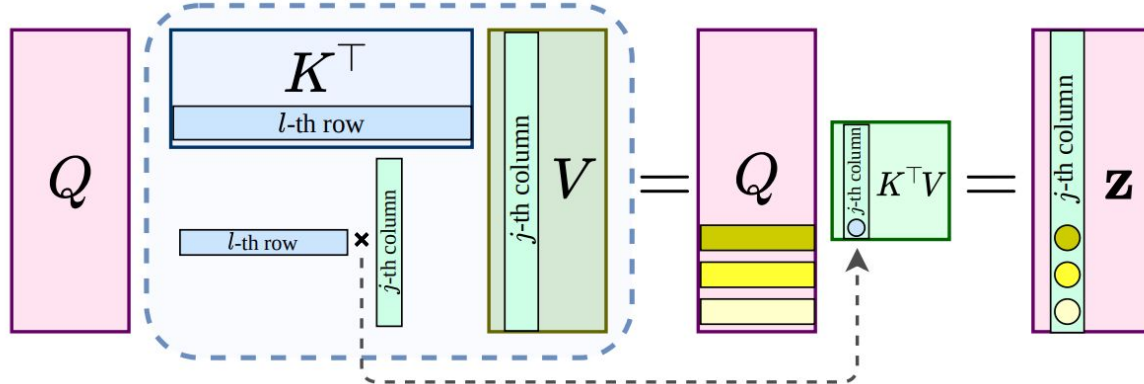
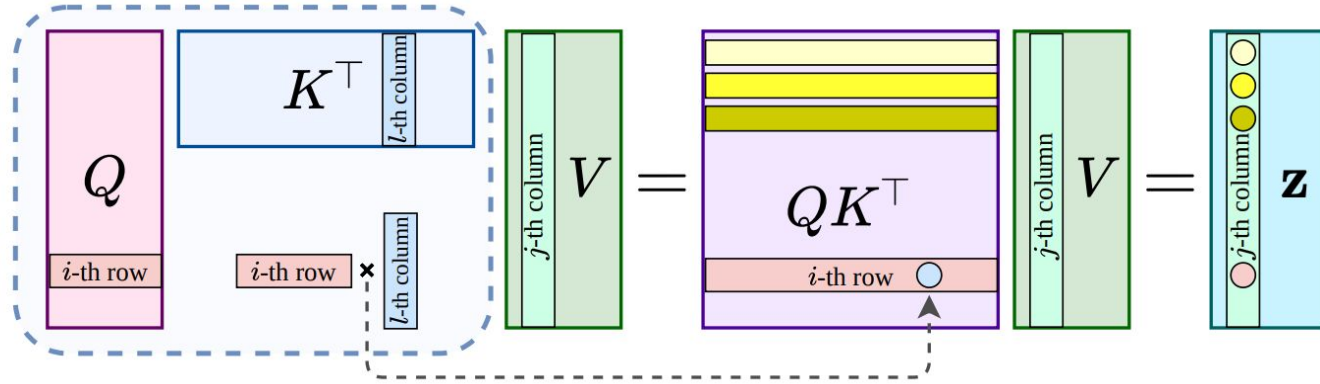
The diagram illustrates the matrix equation $Q = (K^T \times V)$. Matrix Q is a 2x3 purple grid. Matrix K^T is a 3x3 orange grid. Matrix V is a 3x2 blue grid. The multiplication is shown with a large right parenthesis around K^T and V , and a large left parenthesis around Q .

The Solution

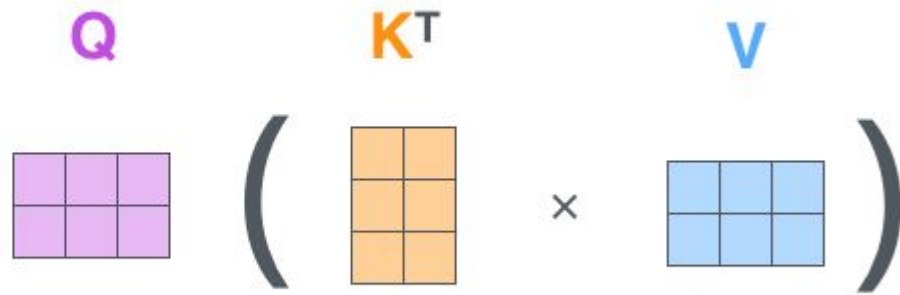
$O(nd^2)$



The Solution

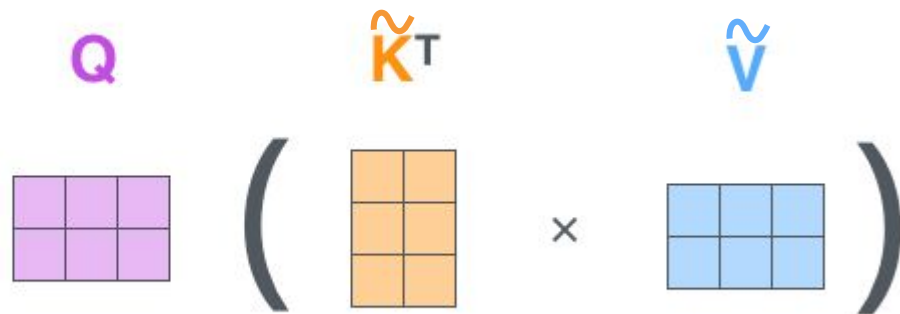


The Solution

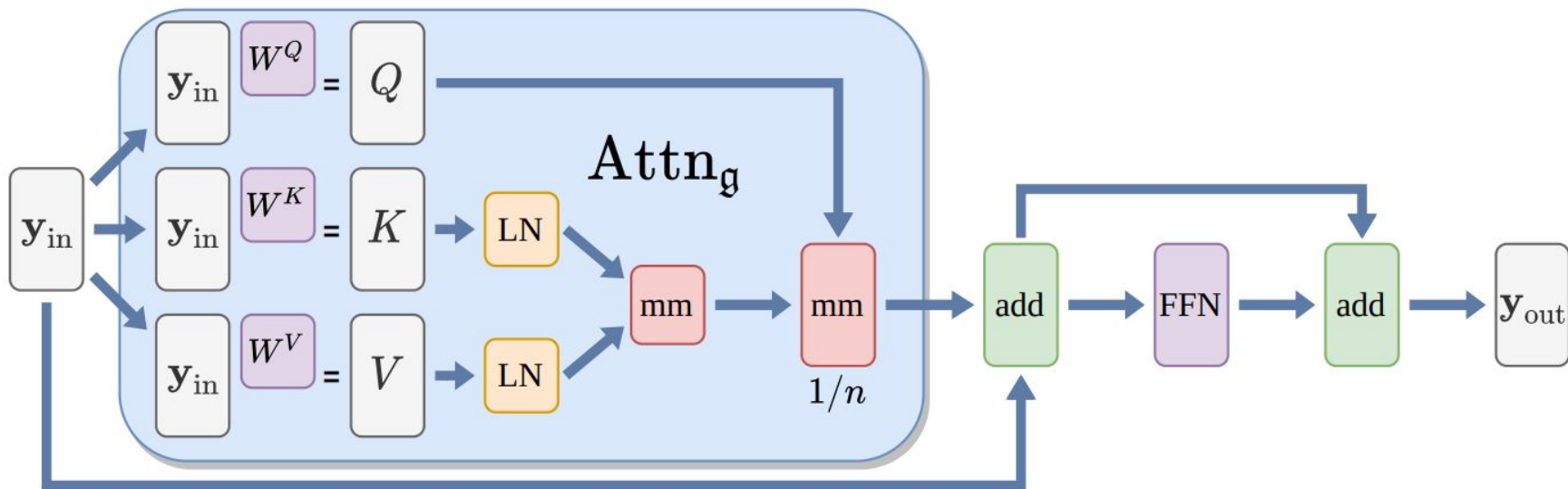
$$\mathbf{Q} \left(\mathbf{K}^T \times \mathbf{V} \right)$$


The diagram illustrates the matrix equation $\mathbf{Q} \left(\mathbf{K}^T \times \mathbf{V} \right)$. Matrix \mathbf{Q} is a 2x3 purple grid. Matrix \mathbf{K}^T is a 3x2 orange grid. Matrix \mathbf{V} is a 2x3 blue grid. The product $\mathbf{K}^T \times \mathbf{V}$ is enclosed in large parentheses.

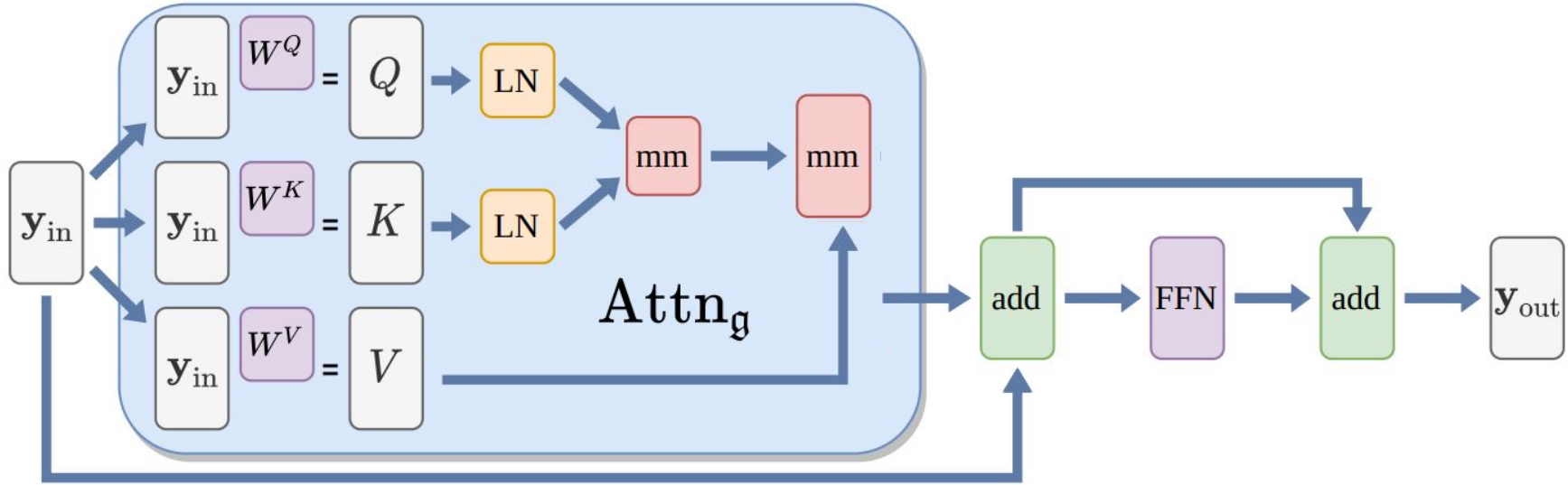
The Solution

$$\mathbf{Q} \left(\tilde{\mathbf{K}}^T \times \tilde{\mathbf{V}} \right)$$


The Galerkin Transformer



The Fourier Transformer



Mathematical Interpretation

$$z_j(x) := \sum_{l=1}^d \mathfrak{b}(k_l, v_j) q_l(x), \text{ for } j = 1, \dots, d, \text{ and } x \in \{x_i\}_{i=1}^n$$

Mathematical Interpretation

$$z_j(x) := \sum_{l=1}^d \mathfrak{b}(k_l, v_j) q_l(x), \text{ for } j = 1, \dots, d, \text{ and } x \in \{x_i\}_{i=1}^n$$

$$\mathbf{z}_j = \left(Q \left(\tilde{K}^T \tilde{V} \right) \right)_j = \sum_{i=1}^d \left(\tilde{K}_i \cdot \tilde{V}_j \right) Q_i$$

Mathematical Interpretation

$$z_j(x) := \sum_{l=1}^d \mathfrak{b}(k_l, v_j) q_l(x), \text{ for } j = 1, \dots, d, \text{ and } x \in \{x_i\}_{i=1}^n$$

$$\mathbf{z}_j = \left(Q \left(\tilde{K}^T \tilde{V} \right) \right)_j = \sum_{i=1}^d \left(\tilde{K}_i \cdot \tilde{V}_j \right) Q_i$$

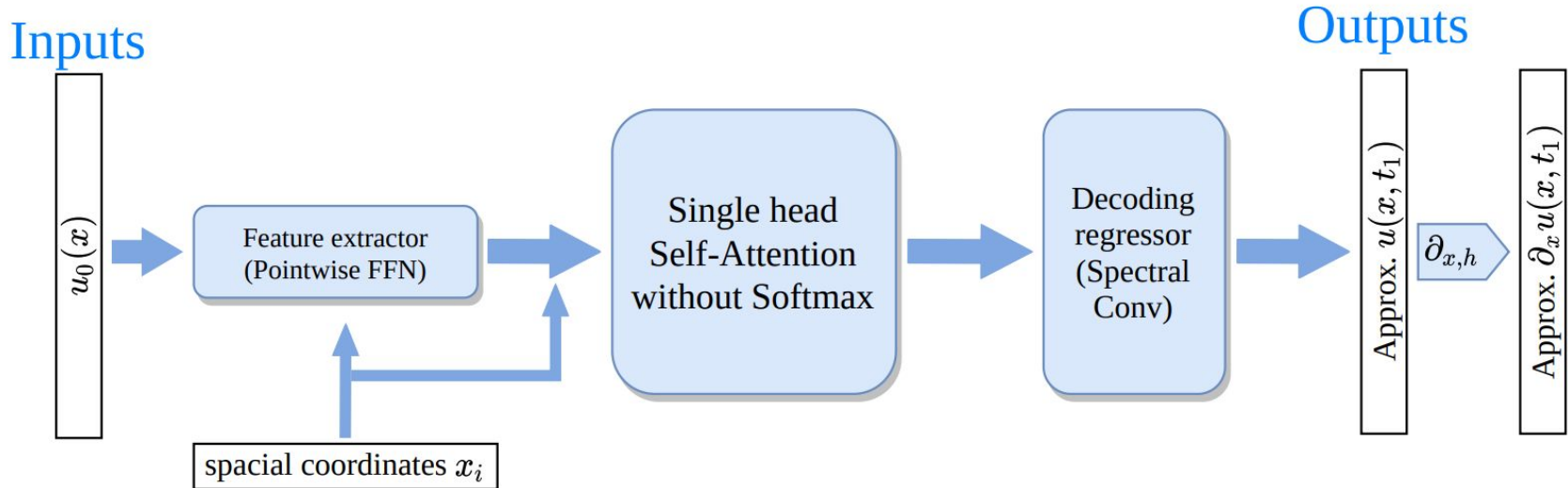
$$\mathfrak{b}(k_l, v_j) = \left(\tilde{K}_i \cdot \tilde{V}_j \right)$$

Example 1: viscous Burgers' equation

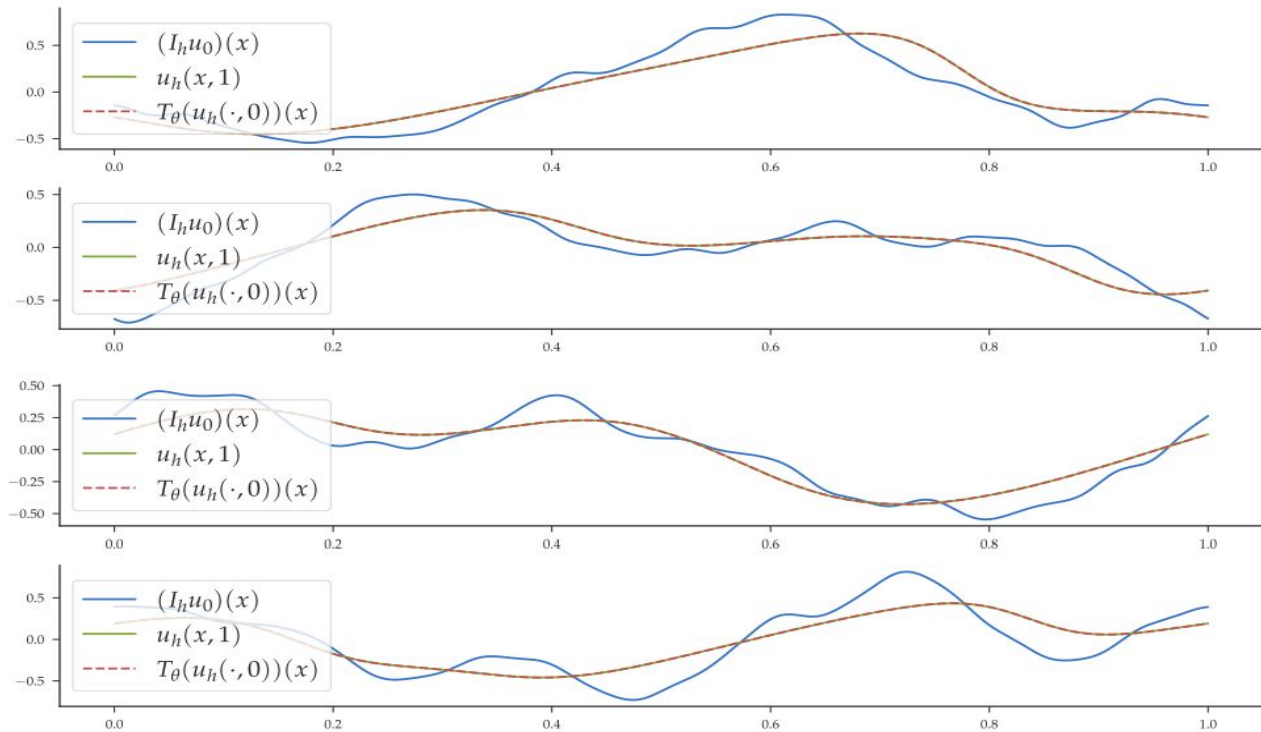
$$\begin{cases} \partial_t u + u \partial_x u = \nu \partial_{xx} u & \text{for } (x, t) \in (0, 1) \times (0, 1], \\ u(x, 0) = u_0(x) & \text{for } x \in (0, 1), \end{cases}$$

$$T : C_p^0(\Omega) \cap L^2(\Omega) \rightarrow C_p^0(\Omega) \cap H^1(\Omega), \quad u_0(\cdot) \mapsto u(\cdot, 1)$$

Example 1: viscous Burgers' equation



Example 1: viscous Burgers' equation



Example 1: viscous Burgers' equation

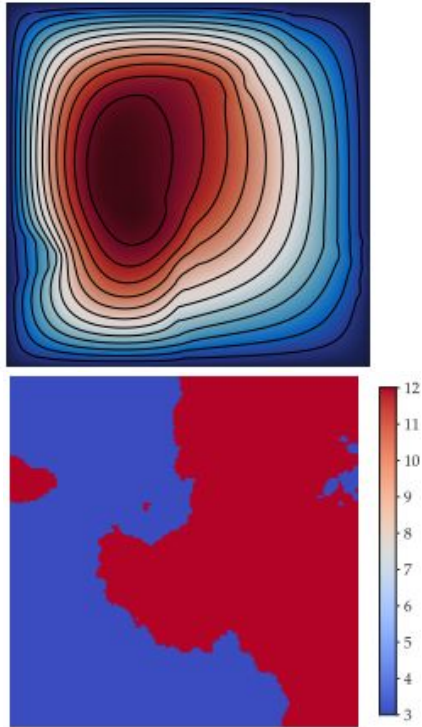
	$n = 512$	$n = 2048$	$n = 8192$
FNO1d [57]	15.8	14.6	13.9
FNO1d 1cycle	4.373	4.126	4.151
FT regular Ln	1.400	1.477	1.172
GT regular Ln	2.181	1.512	2.747
ST regular Ln	1.927	2.307	1.981
LT regular Ln	1.813	1.770	1.617
FT Ln on Q, K	1.135	1.123	1.071
GT Ln on K, V	1.203	1.150	1.025
ST Ln on Q, K	1.271	1.266	1.330
LT Ln on K, V	1.139	1.149	1.221

Example 2: Darcy flow

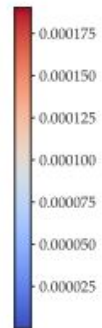
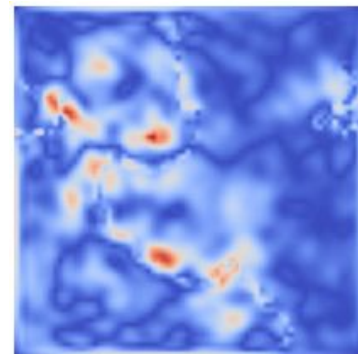
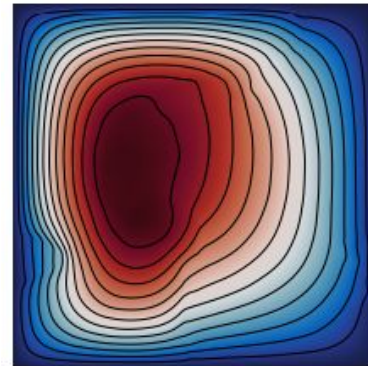
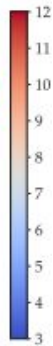
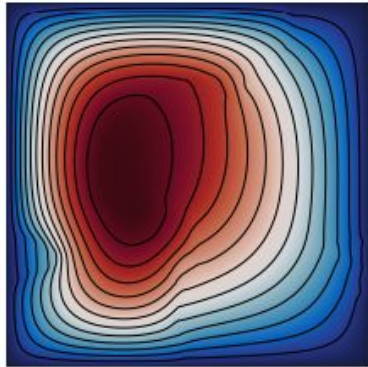
$$\begin{cases} -\nabla \cdot (a \nabla u) = f & \text{in } \Omega, \\ u = 0 & \text{on } \partial\Omega \end{cases}$$

$$T : L^\infty(\Omega) \rightarrow H_0^1(\Omega), \quad a \mapsto u$$

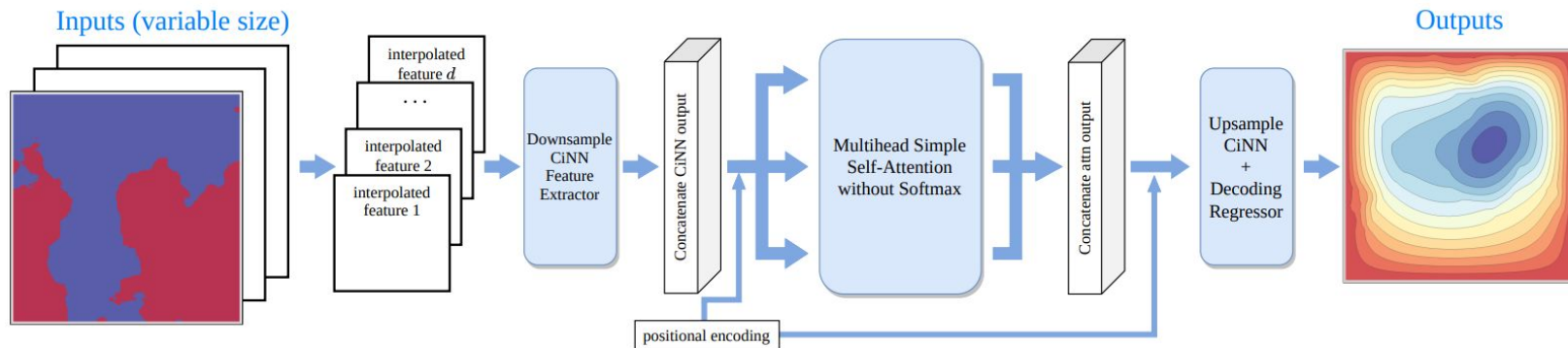
Example 2: Darcy flow



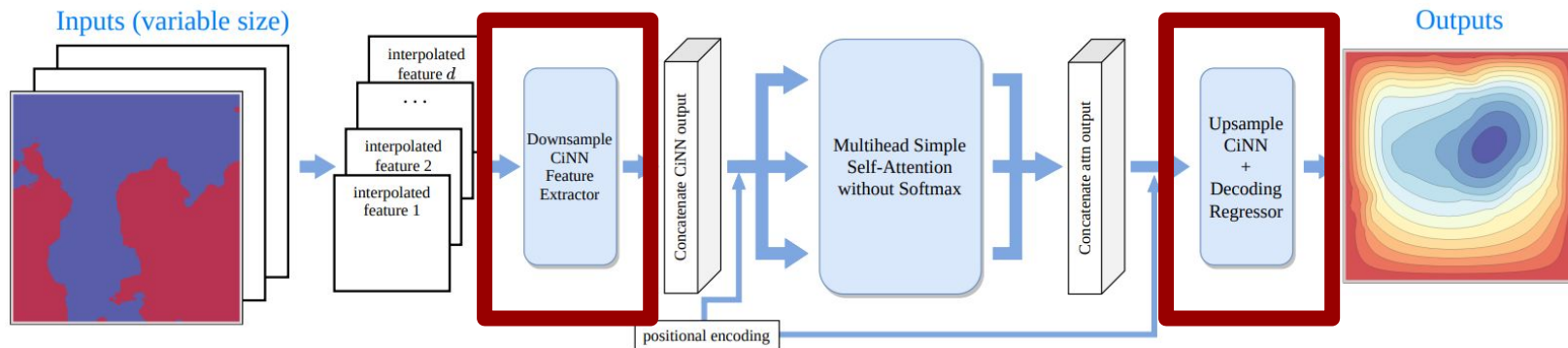
Example 2: Darcy flow



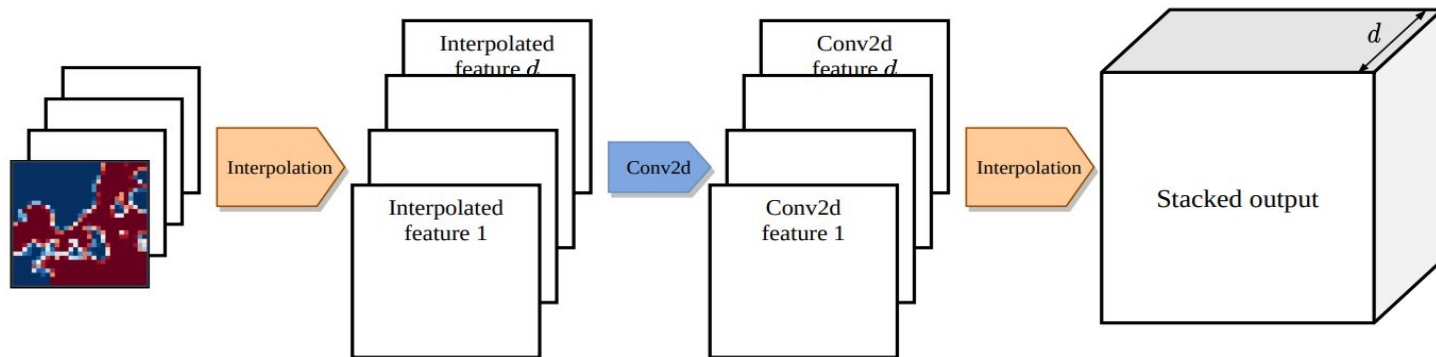
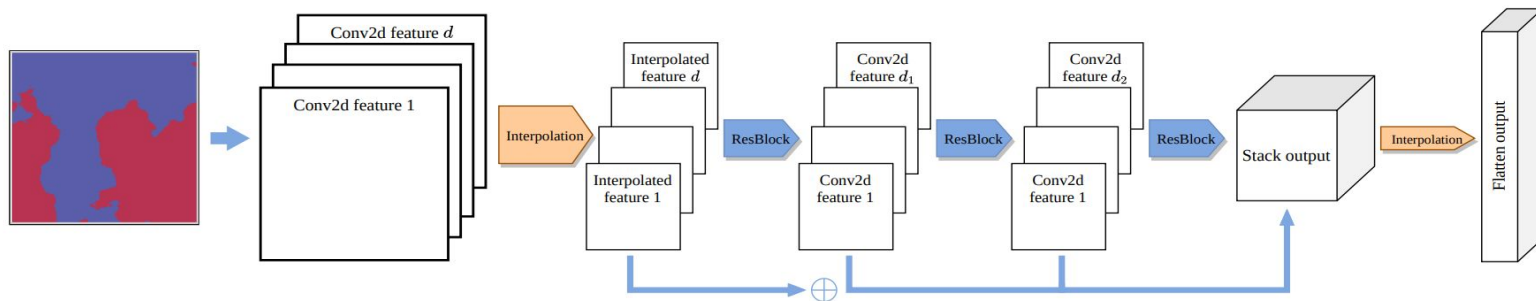
Example 2: Darcy flow



Example 2: Darcy flow



Example 2: Darcy flow



Example 2: Darcy flow

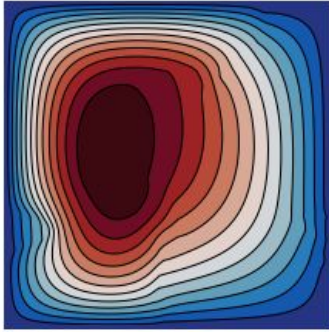
	$n_f, n_c = 141, 43$	$n_f, n_c = 211, 61$
FNO2d [57]	1.09	1.09
FNO2d 1cycle	1.419	1.424
FT regular Ln	0.838	0.847
GT regular Ln	0.894	0.856
ST regular Ln	1.075	1.131
LT regular Ln	1.024	1.130
FT Ln on Q, K	0.873	0.921
GT Ln on K, V	0.839	0.844
ST Ln on Q, K	0.946	0.959
LT Ln on K, V	0.875	0.970

Example 3: inverse coefficient identification for Darcy flow

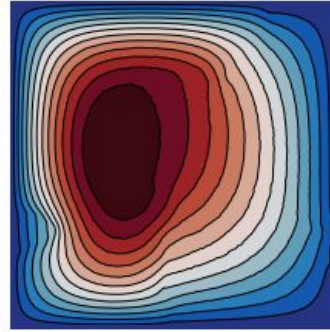
$$\begin{cases} -\nabla \cdot (a \nabla u) = f & \text{in } \Omega, \\ u = 0 & \text{on } \partial\Omega \end{cases}$$

$$T : H_0^1(\Omega) \rightarrow L^\infty(\Omega), u + \epsilon N_\nu(u) \mapsto a$$

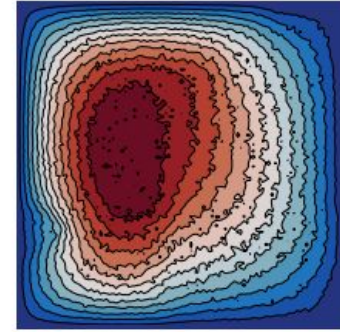
Example 3: inverse coefficient identification for Darcy flow



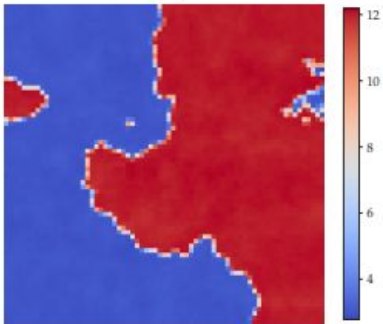
(a)



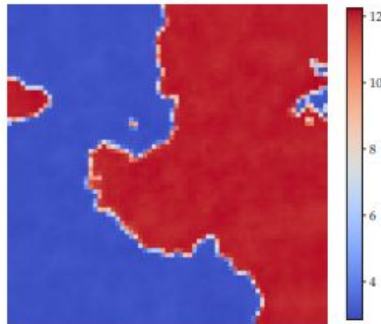
(b)



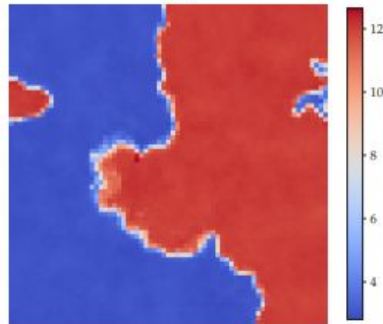
(c)



(d)

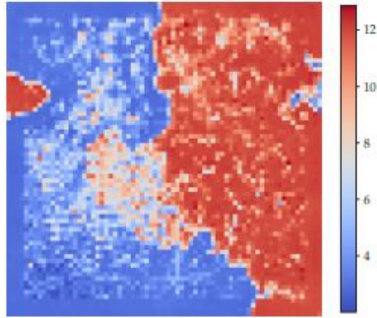


(e)

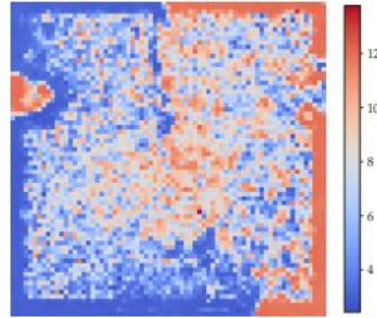


(f)

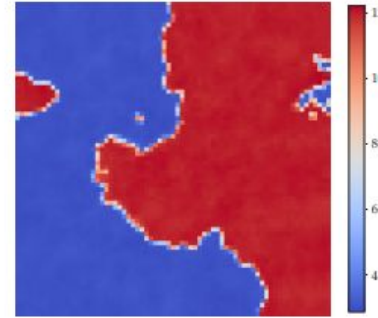
Example 3: inverse coefficient identification for Darcy flow



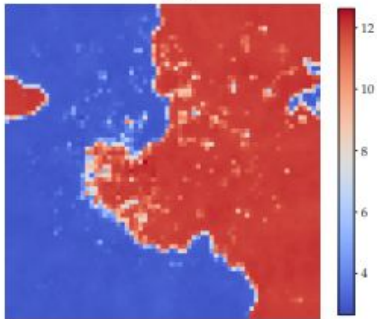
(a)



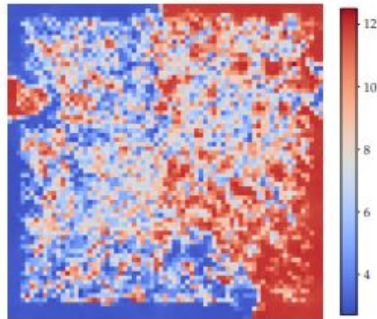
(b)



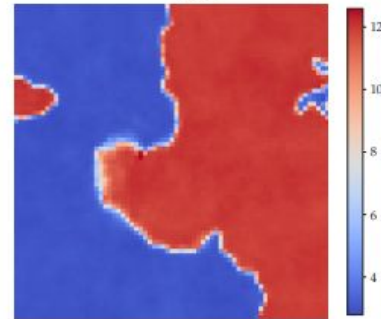
(c)



(d)

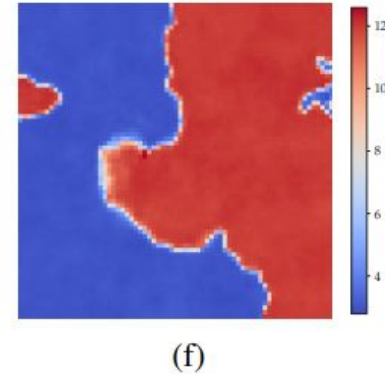
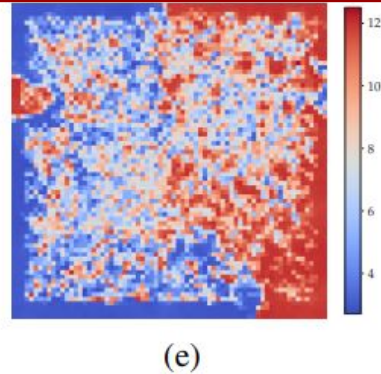
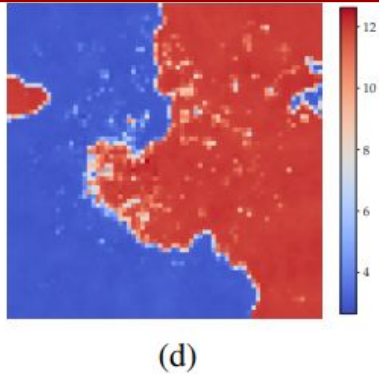
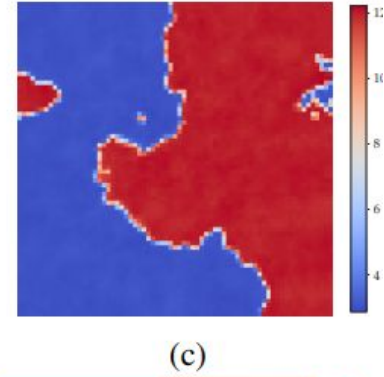
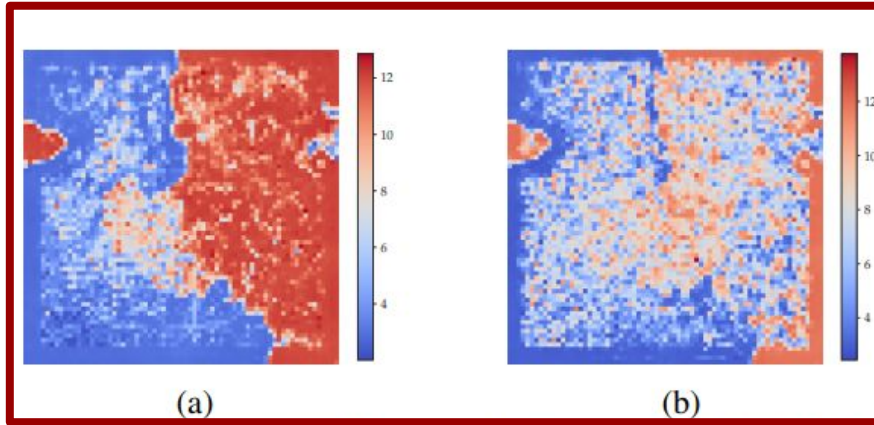


(e)

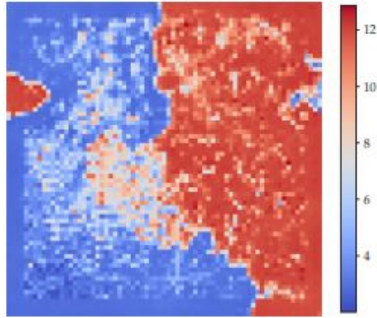


(f)

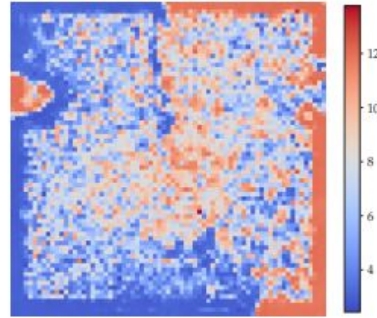
Example 3: inverse coefficient identification for Darcy flow



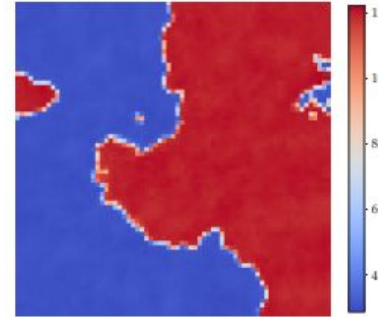
Example 3: inverse coefficient identification for Darcy flow



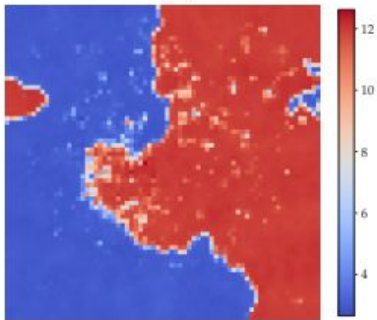
(a)



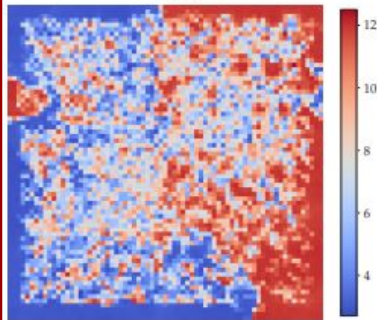
(b)



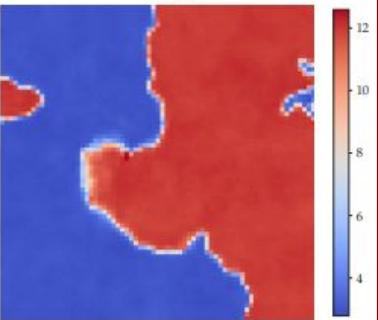
(c)



(d)



(e)



(f)

Example 3: inverse coefficient identification for Darcy flow

	$n_f, n_c = 141, 36$			$n_f, n_c = 211, 71$		
	$\epsilon = 0$	$\epsilon = 0.01$	$\epsilon = 0.1$	$\epsilon = 0$	$\epsilon = 0.01$	$\epsilon = 0.1$
FNO2d (only n_f)	13.71	13.78	15.12	13.93	13.96	15.04
FNO2d (only n_c)	14.17	14.31	17.30	13.60	13.69	16.04
FT regular Ln	1.799	2.467	6.814	1.563	2.704	8.110
GT regular Ln	2.026	2.536	6.659	1.732	2.775	8.024
ST regular Ln	2.434	3.106	7.431	2.069	3.365	8.918
LT regular Ln	2.254	3.194	9.056	2.063	3.544	9.874
FT Ln on Q, K	1.921	2.717	6.725	1.523	2.691	8.286
GT Ln on K, V	1.944	2.552	6.689	1.651	2.729	7.903
ST Ln on Q, K	2.160	2.807	6.995	1.889	3.123	8.788
LT Ln on K, V	2.360	3.196	8.656	2.136	3.539	9.622

Discussion

Strengths

- Insightful contribution to “linearizing” transformers
- Well designed Experiments and solid results

Strengths

- Insightful contribution to “linearizing” transformers
- Well designed Experiments and solid results

Weaknesses

- Operator must exhibit low dimensional attributes
- Not efficient to apply at full resolution
- Encoder only

Conclusion

- The paper proposes a very versatile transformer variant
- Improves the state of the art operator learner
- Speed ups in geoscience, medical imaging and NLP

Thank you

Sources

<https://jalammar.github.io/illustrated-transformer/>

<https://openreview.net/forum?id=ssohLcmn4-r>

<https://github.com/scaomath/galerkin-transformer>

<https://towardsdatascience.com/galerkin-transformer-a-one-shot-experiment-at-neurips-2021-96efcbaefd3e>

<https://scaomath.github.io/blog/galerkin-transformer/>

Sources

<https://medium.com/@bogdan.raonke/operator-learning-convolutional-neural-operators-for-robust-and-accurate-learning-of-pdes-ebbc43b57434#:~:text=Once%20trained%2C%20Neural%20Operators%20solely,distinguishing%20them%20from%20traditional%20solvers.>