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

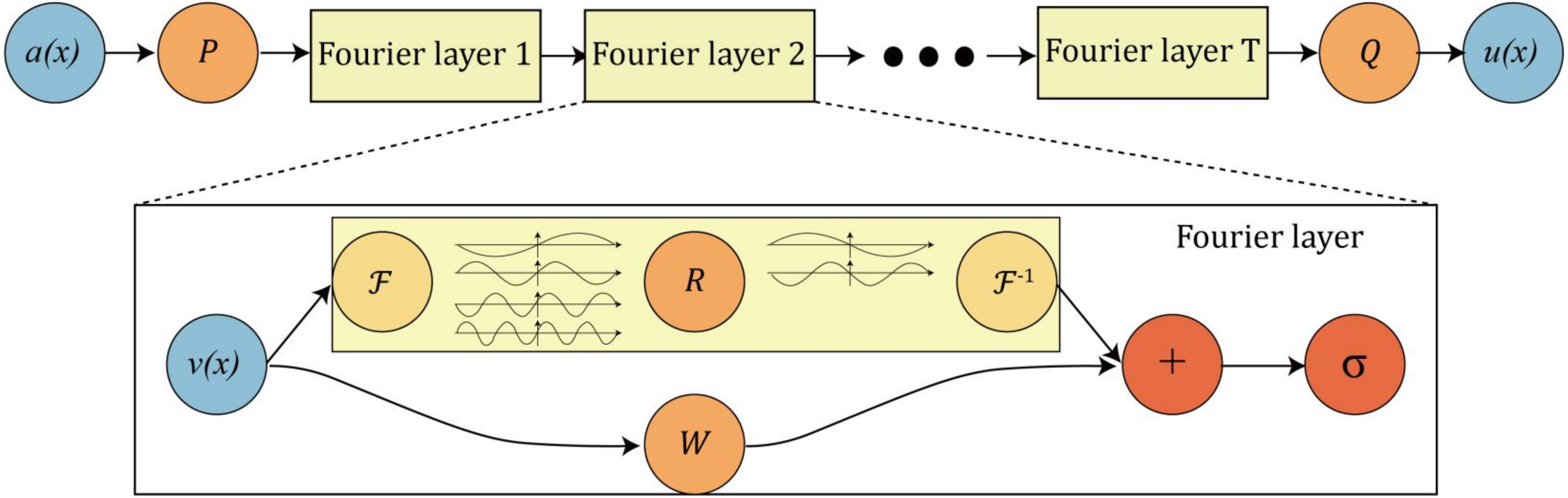
What is a Neural Operator?

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

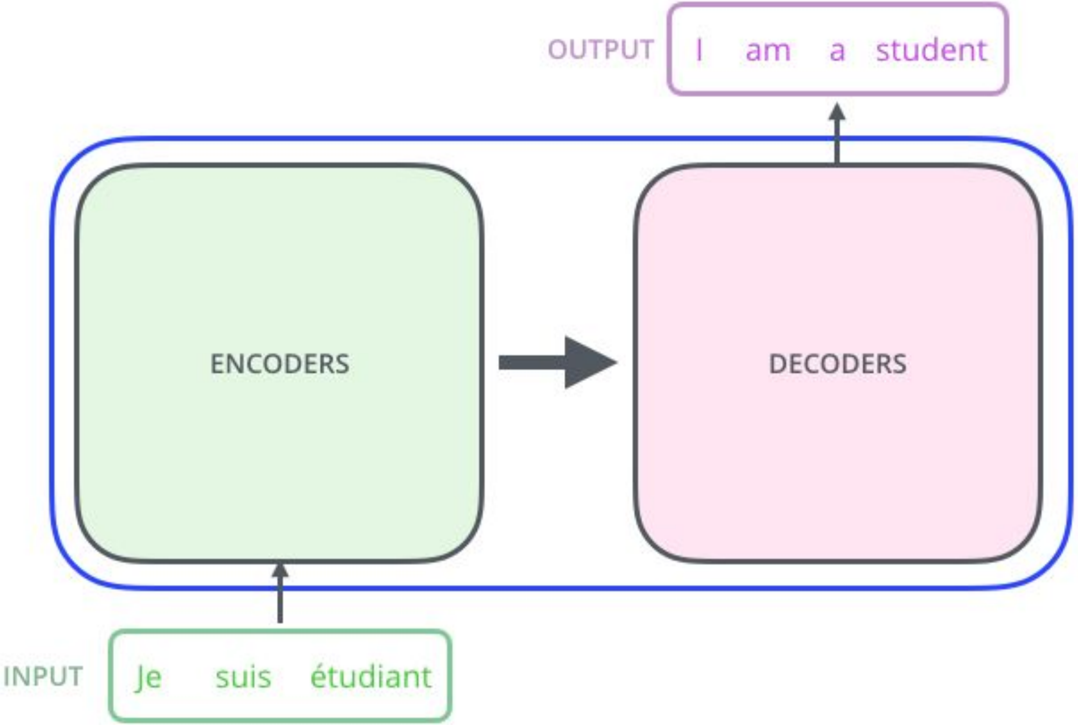
Fourier Neural Operator:



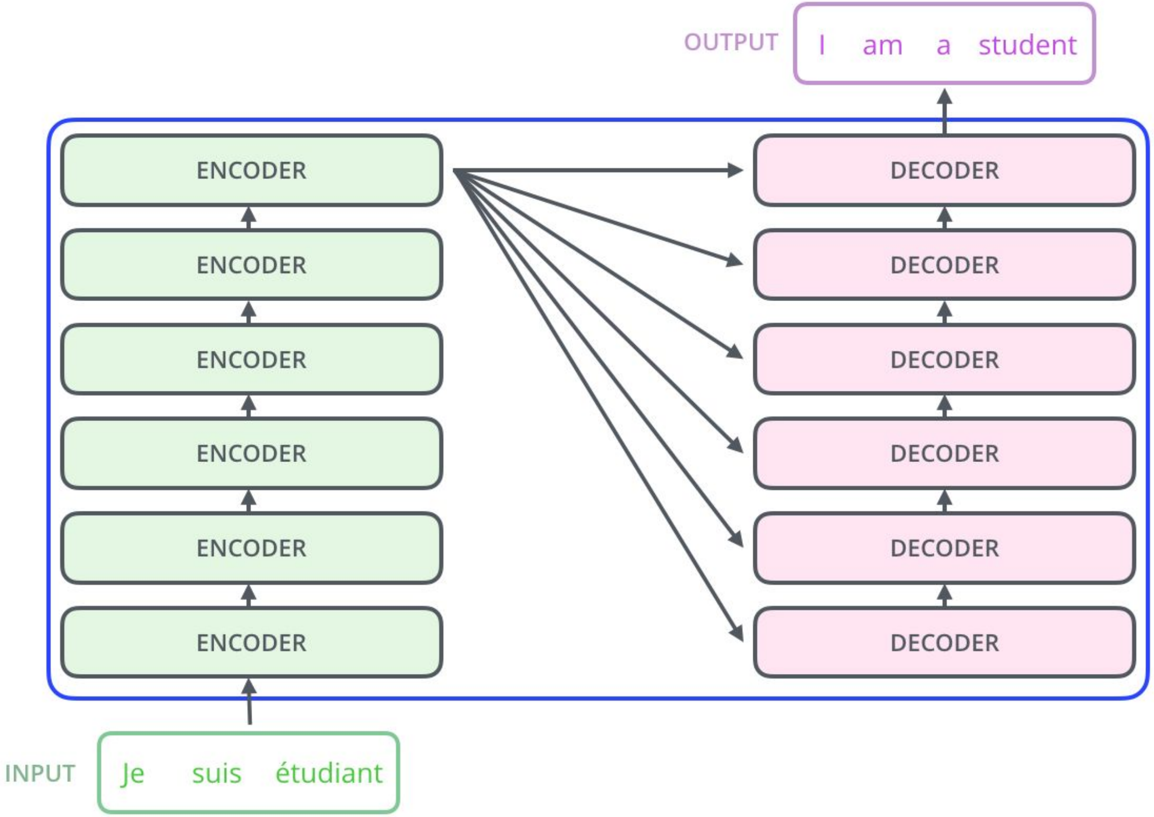
The discretized operator learning problem is a **seq2seq** problem

What is Self-Attention?

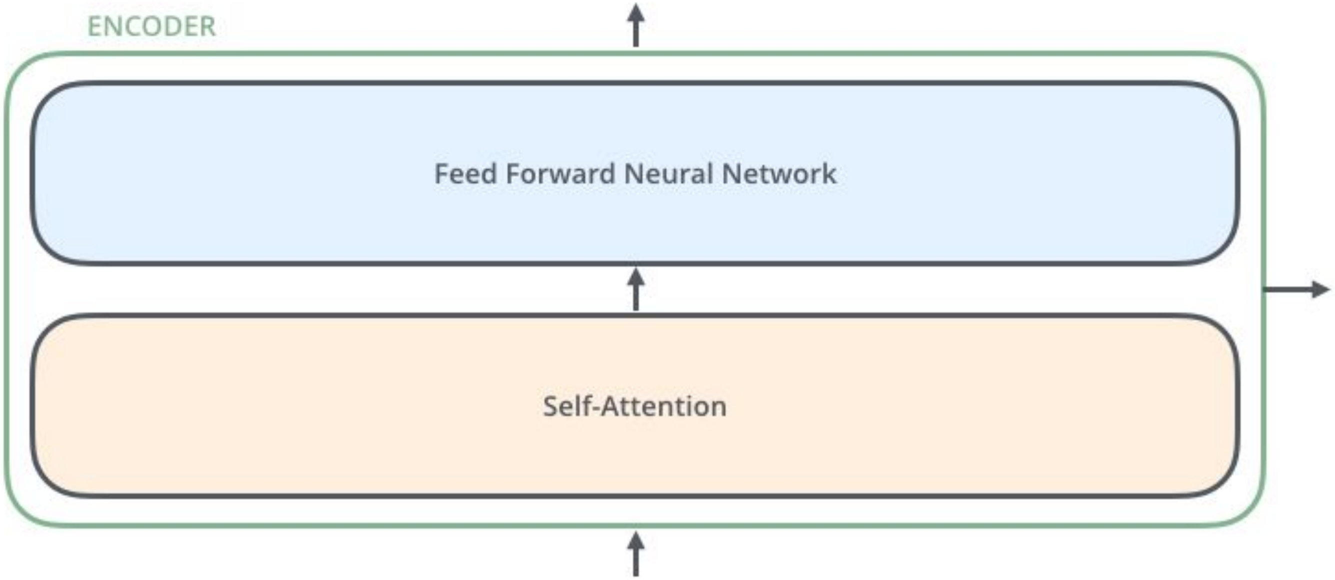
What is Self-Attention?



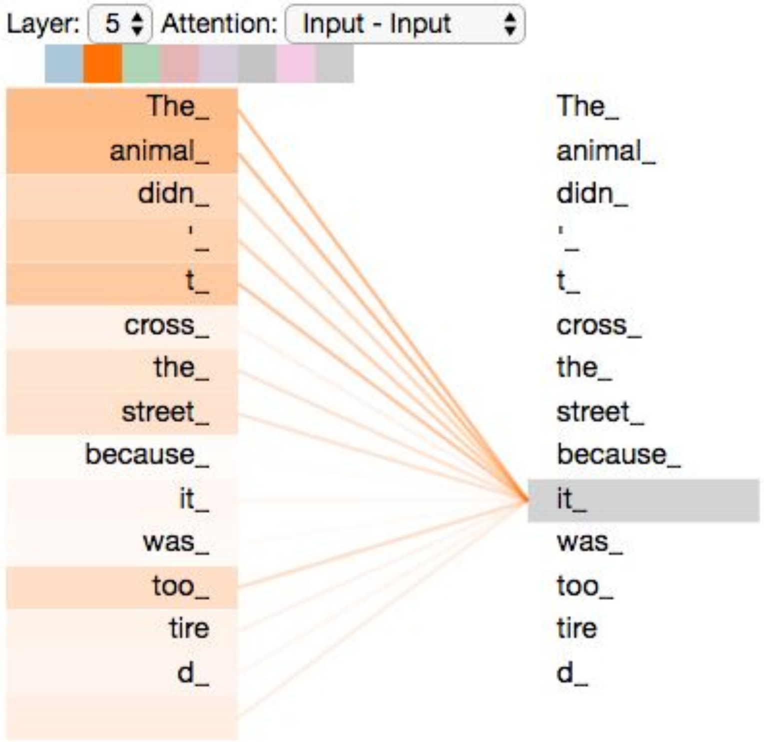
What is Self-Attention?



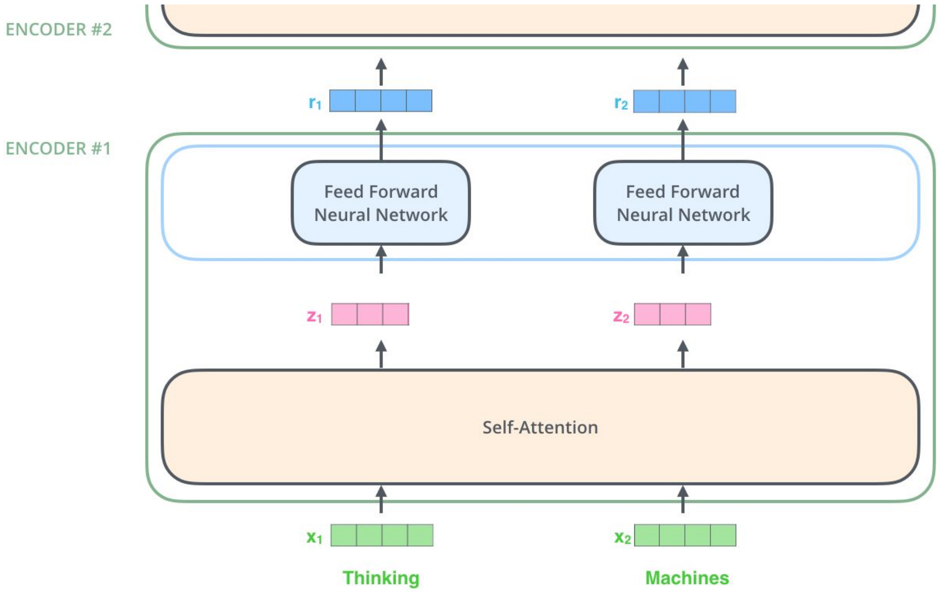
What is Self-Attention?



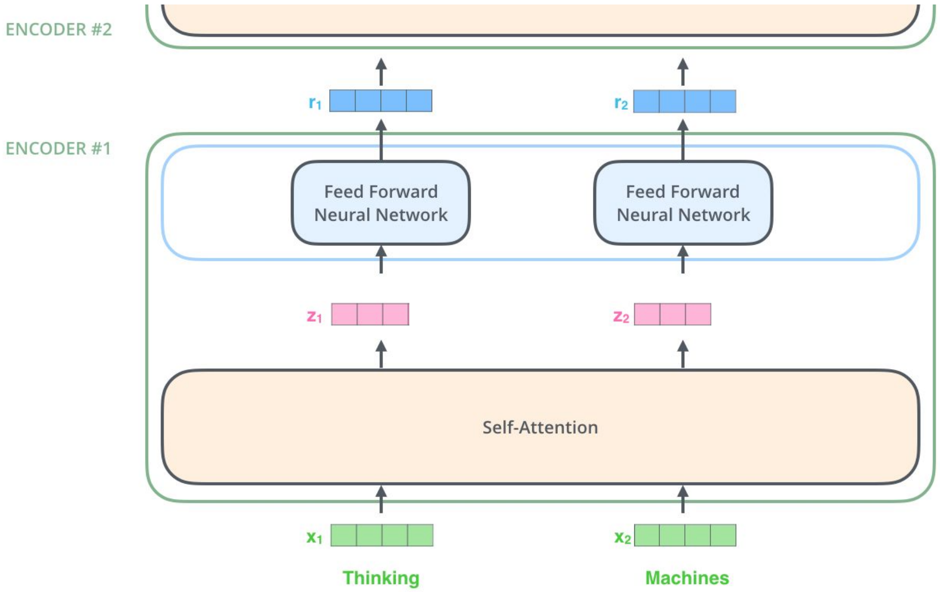
What is Self-Attention?

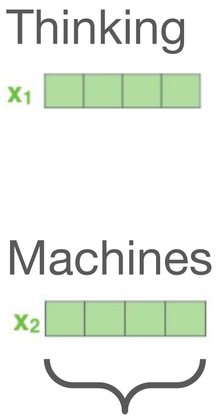


What is Self-Attention?



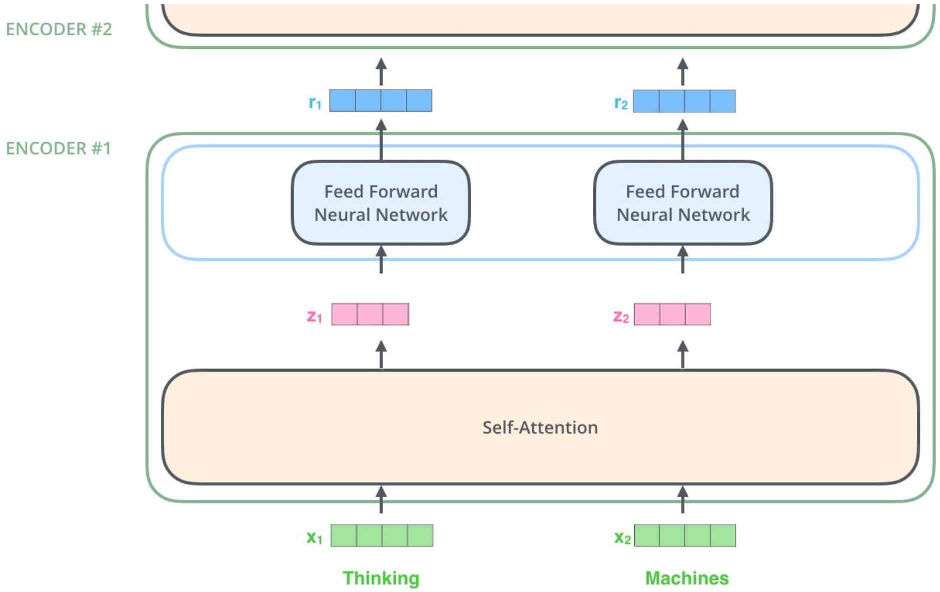
What is Self-Attention?

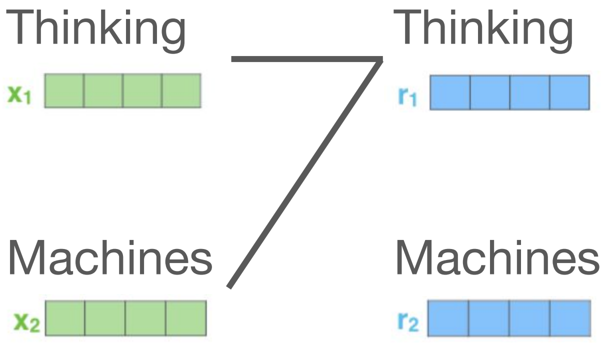




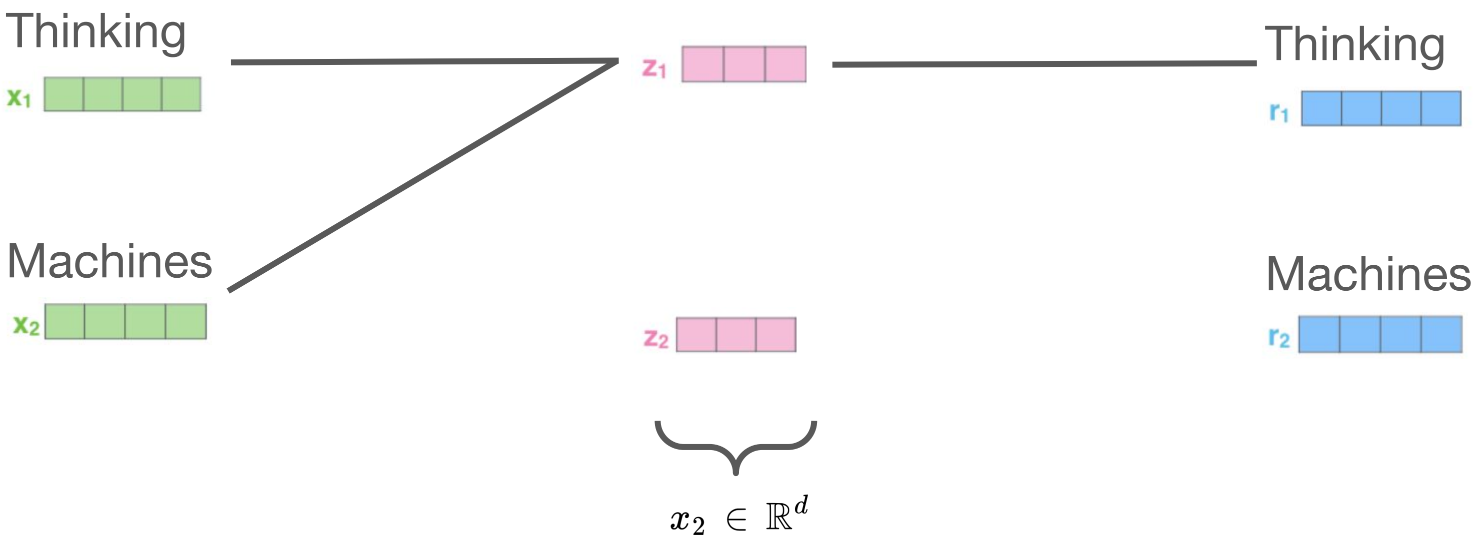
$$x_2 \in \mathbb{R}^l$$

What is Self-Attention?

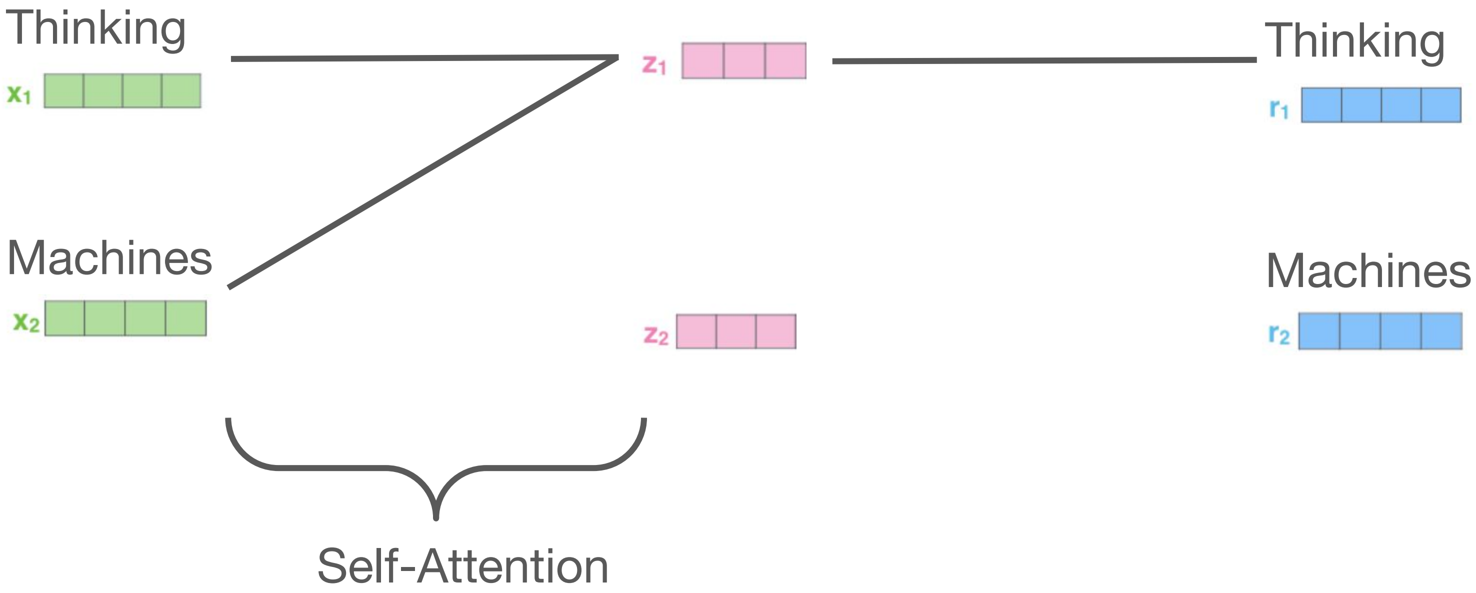




What happens mathematically inside Self-Attention?

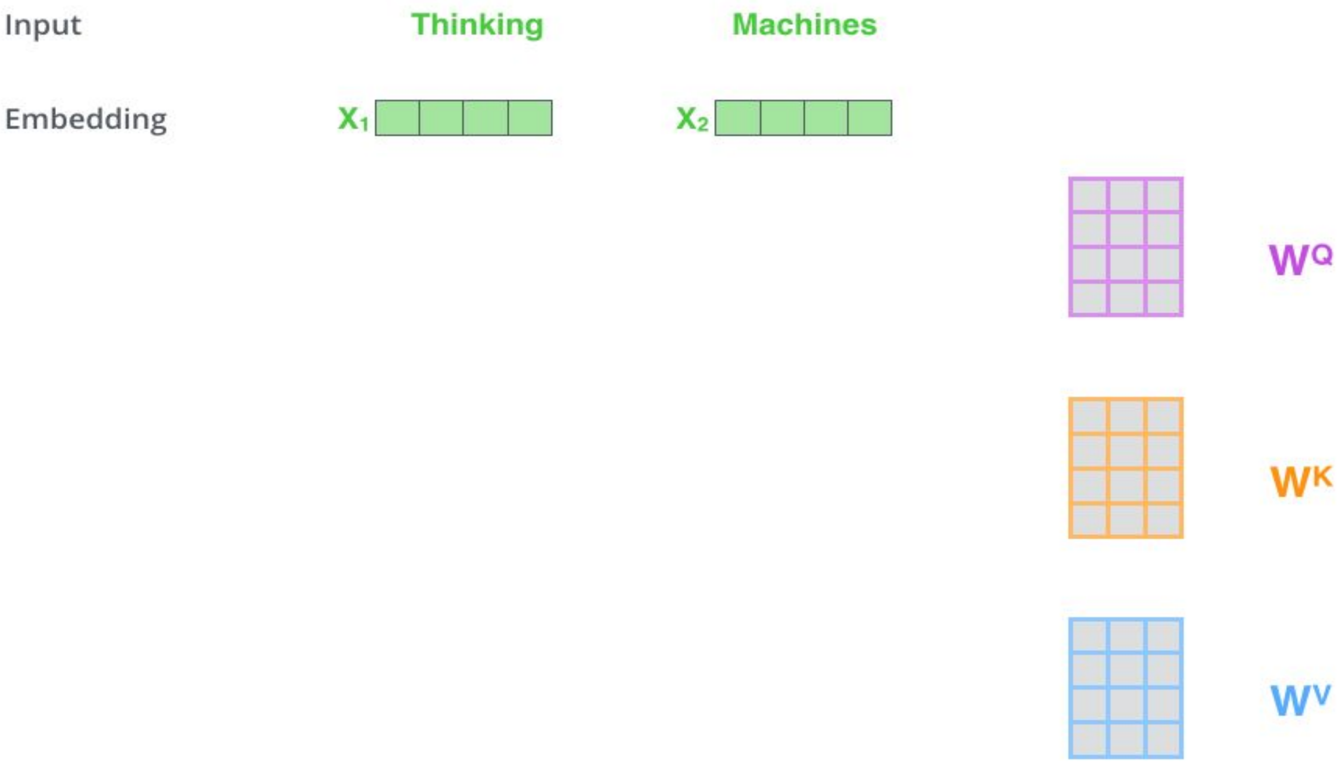


What happens mathematically inside Self-Attention?

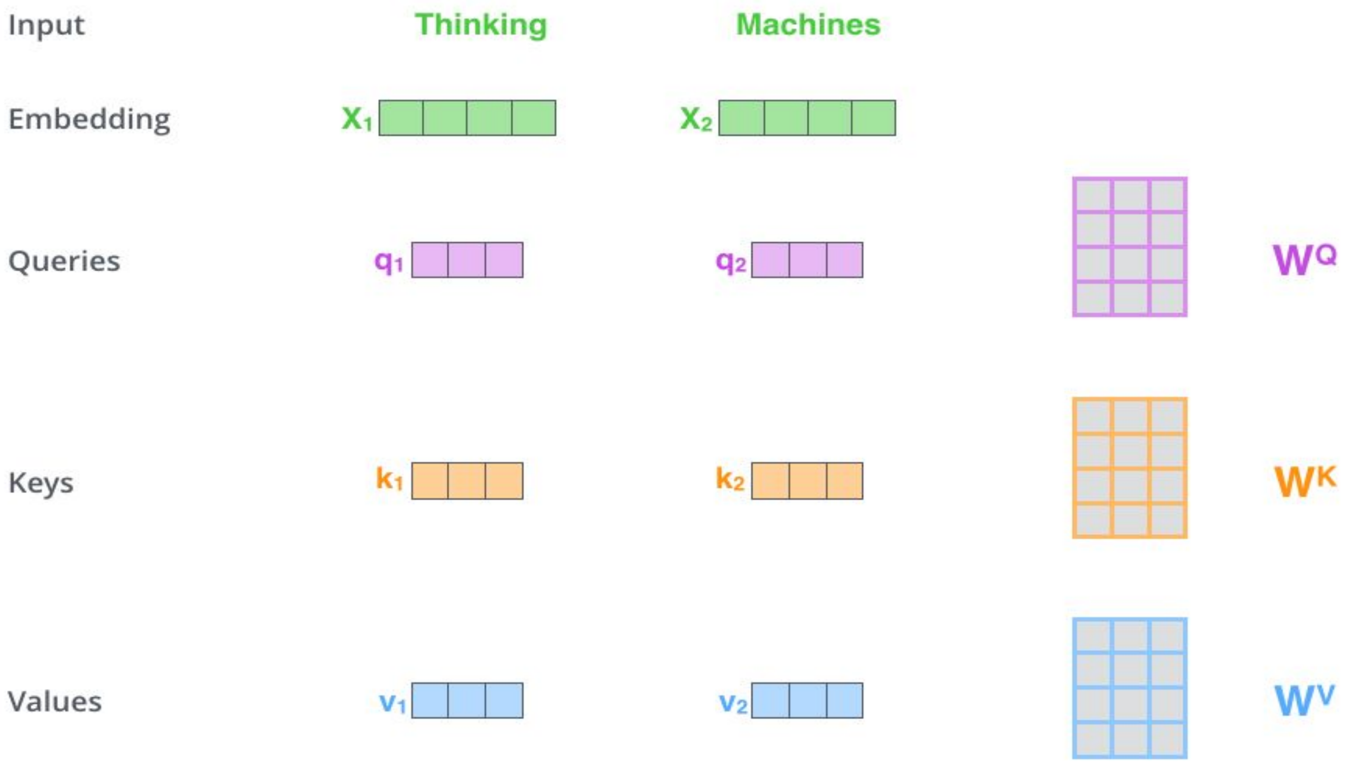


What happens mathematically inside Self-Attention?

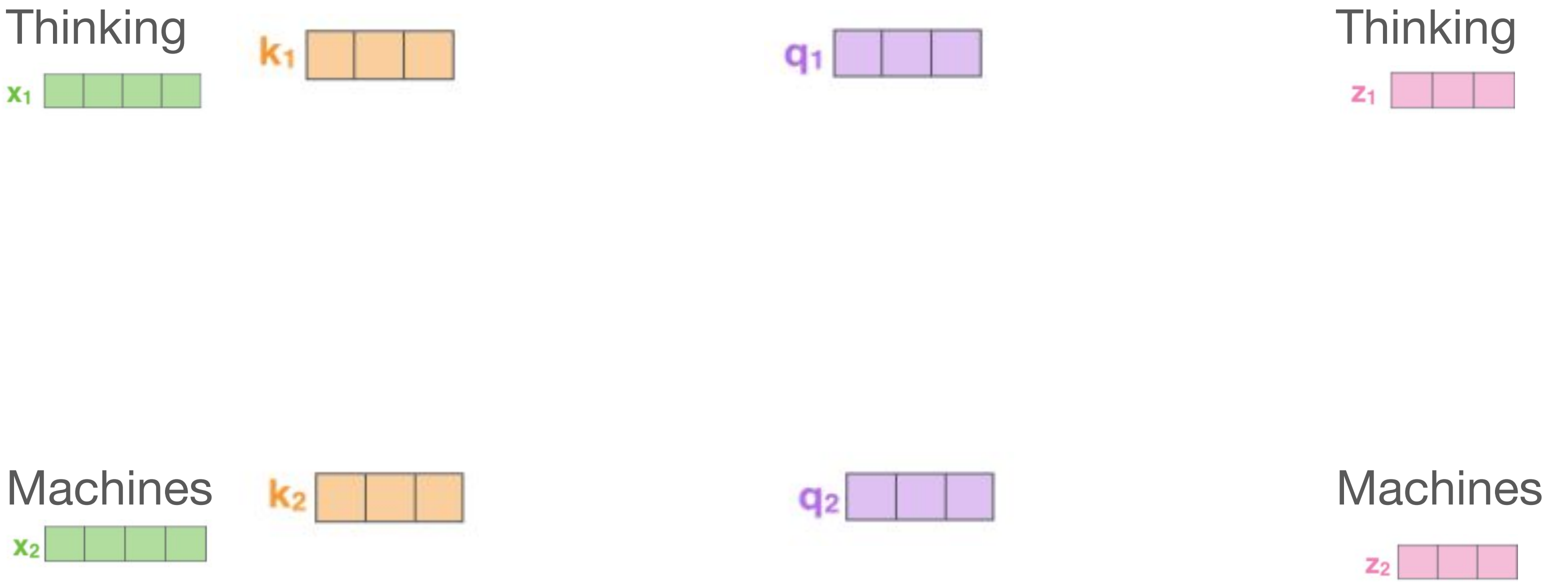
What happens mathematically inside Self-Attention?



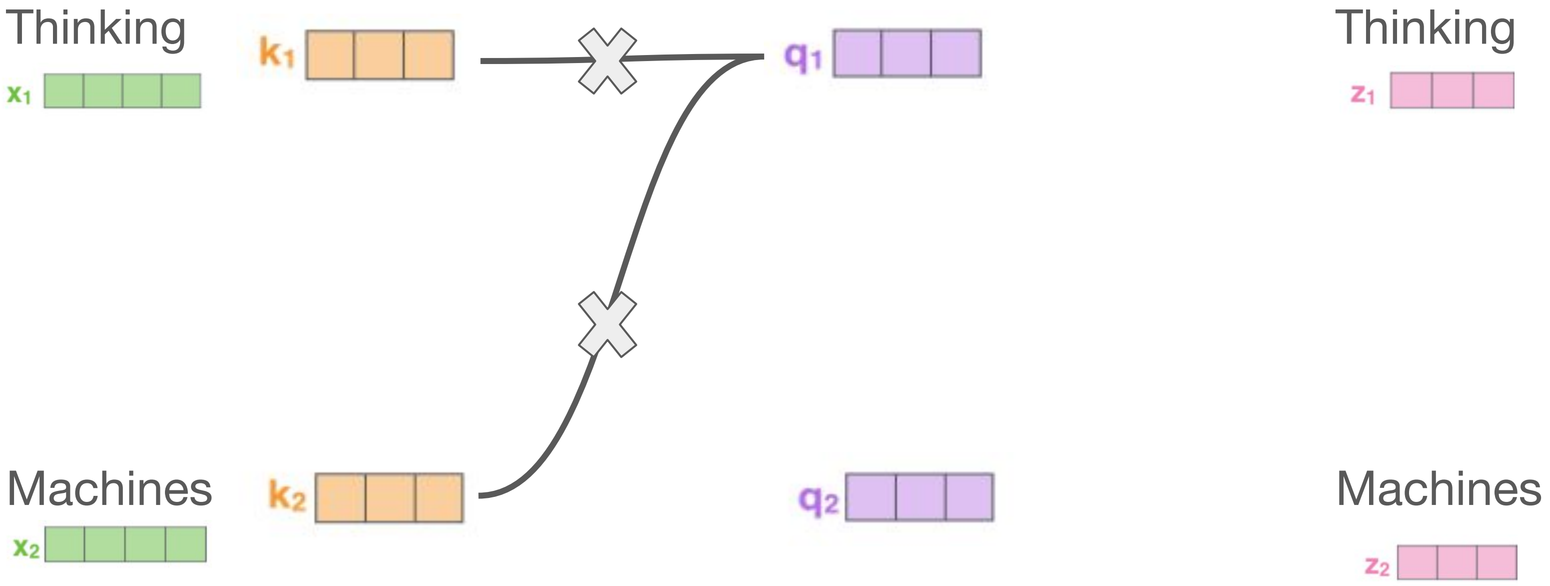
What happens mathematically inside Self-Attention?



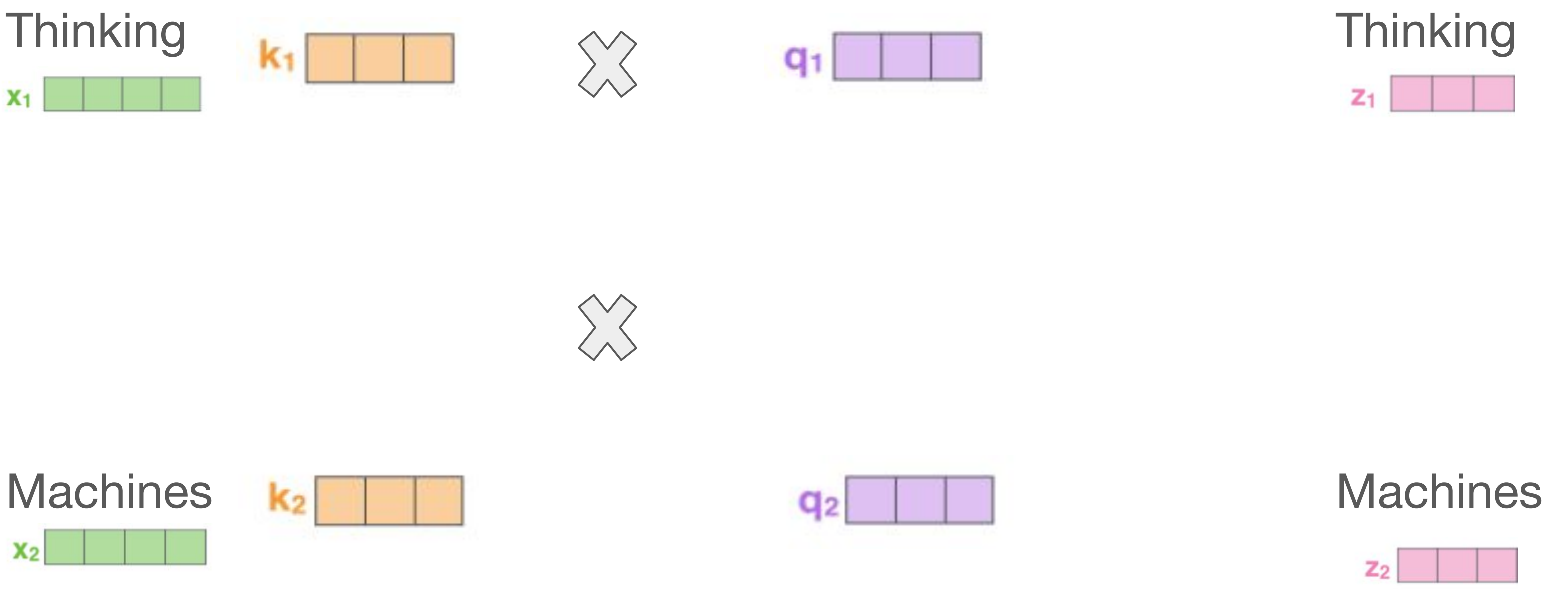
What happens mathematically inside Self-Attention?



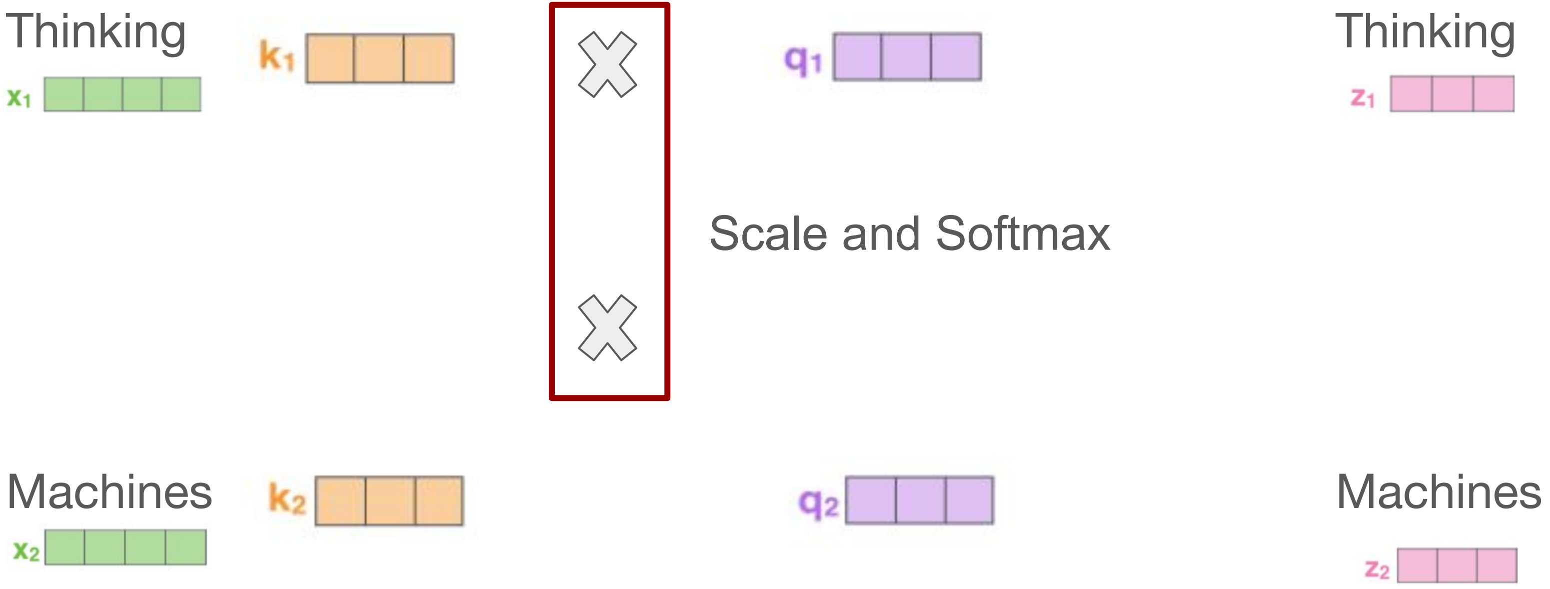
What happens mathematically inside Self-Attention?



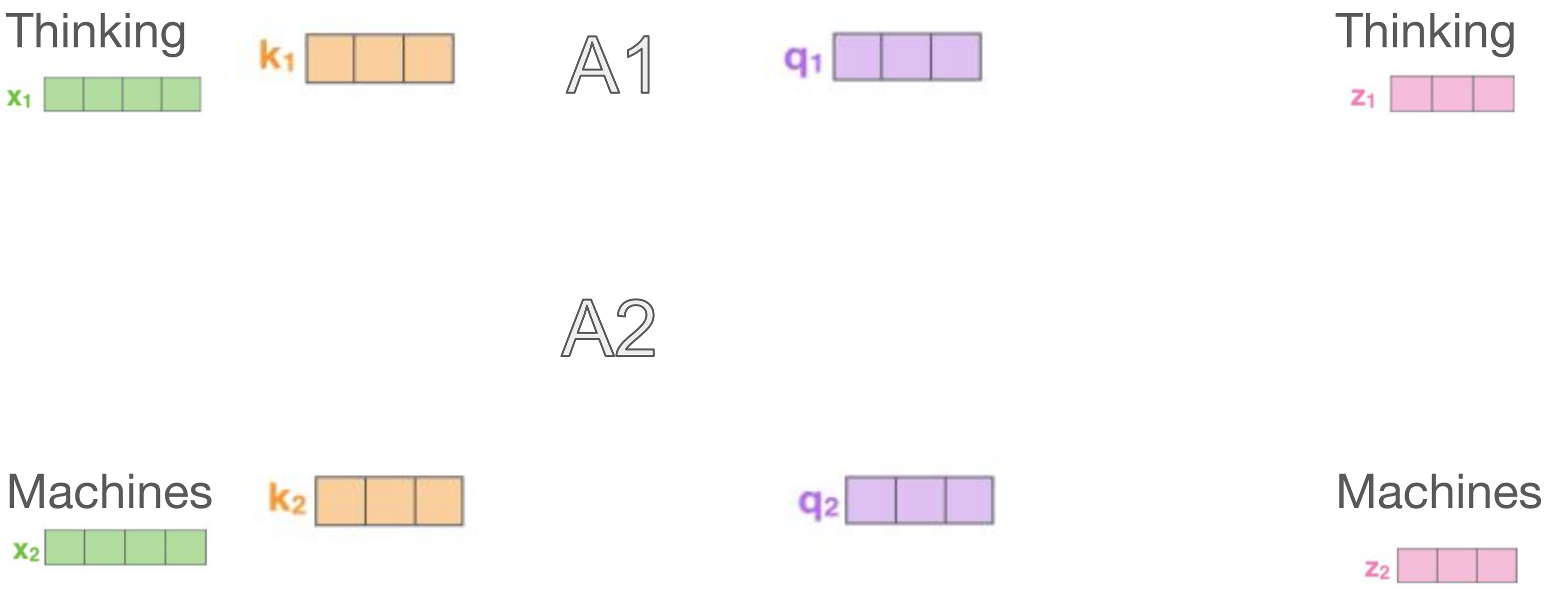
What happens mathematically inside Self-Attention?



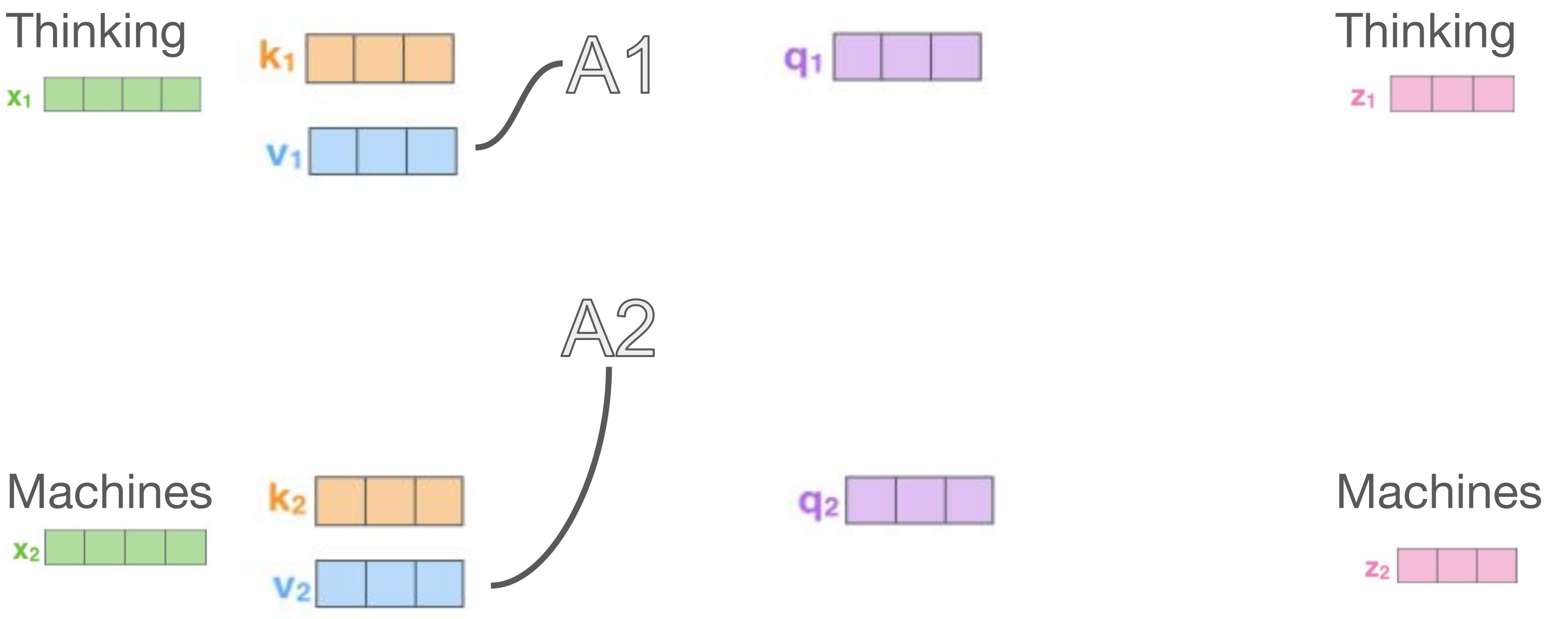
What happens mathematically inside Self-Attention?



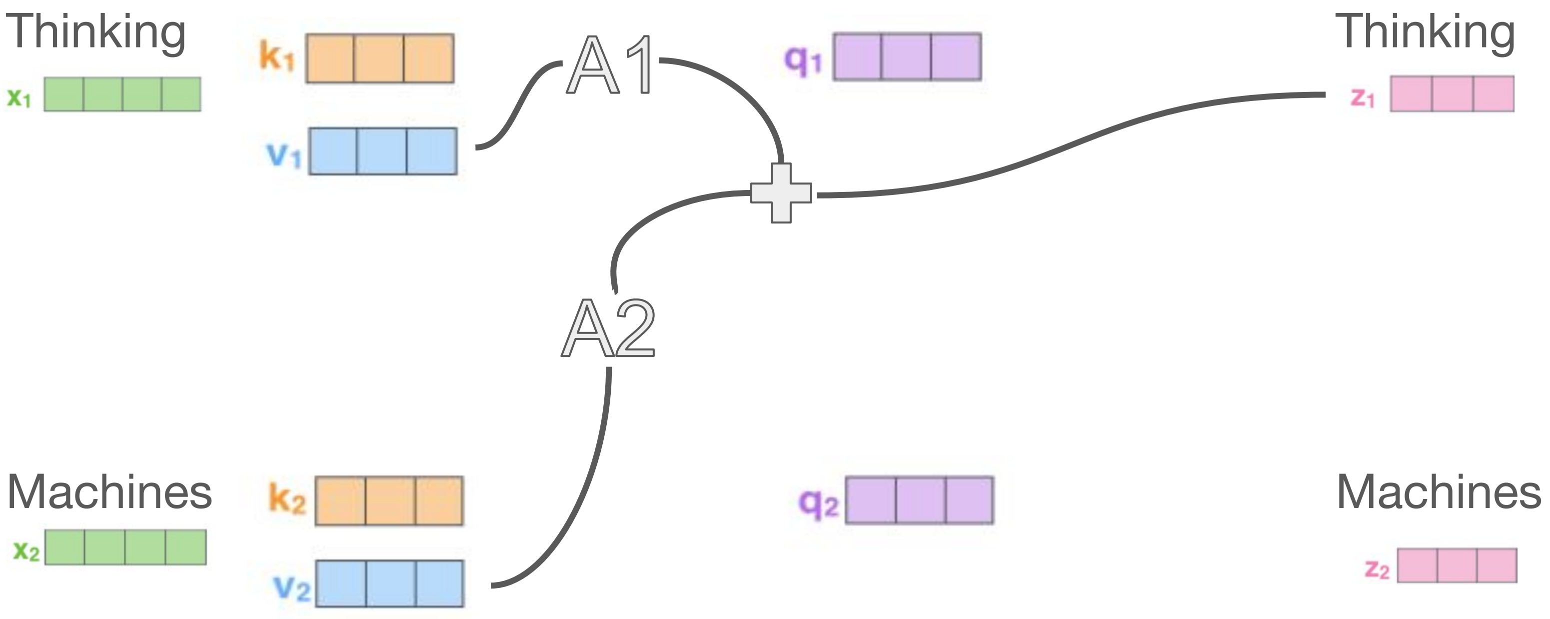
What happens mathematically inside Self-Attention?



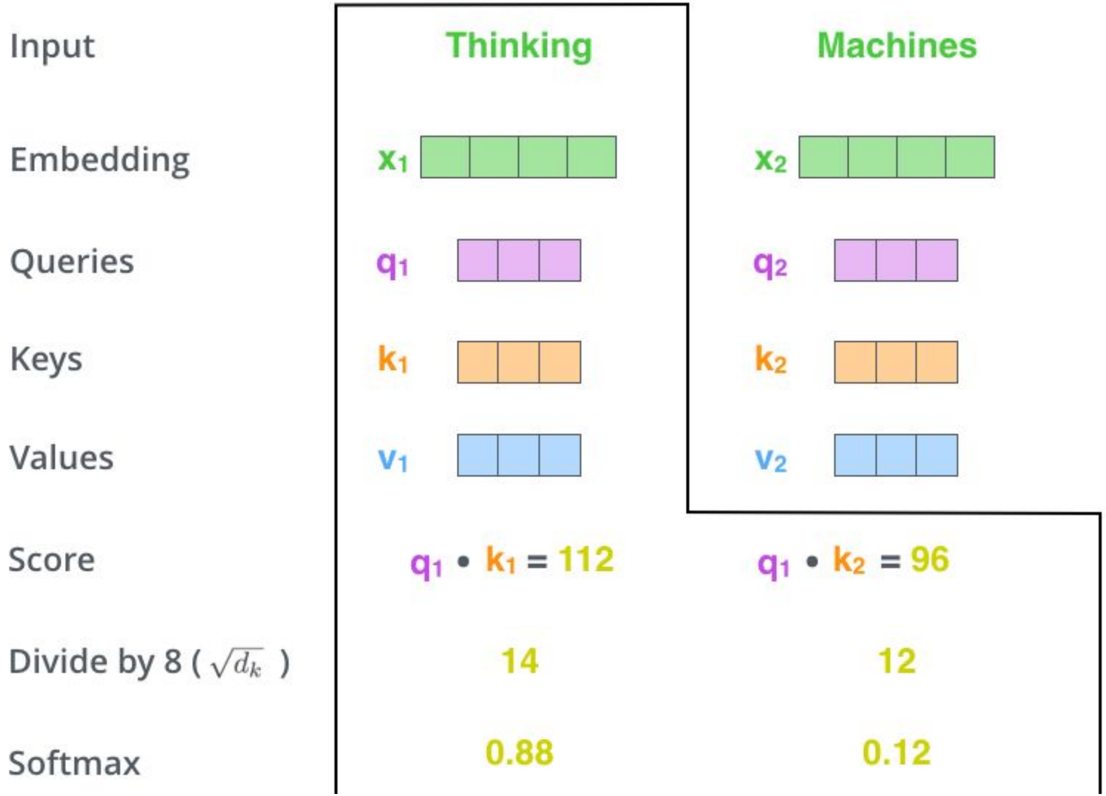
What happens mathematically inside Self-Attention?



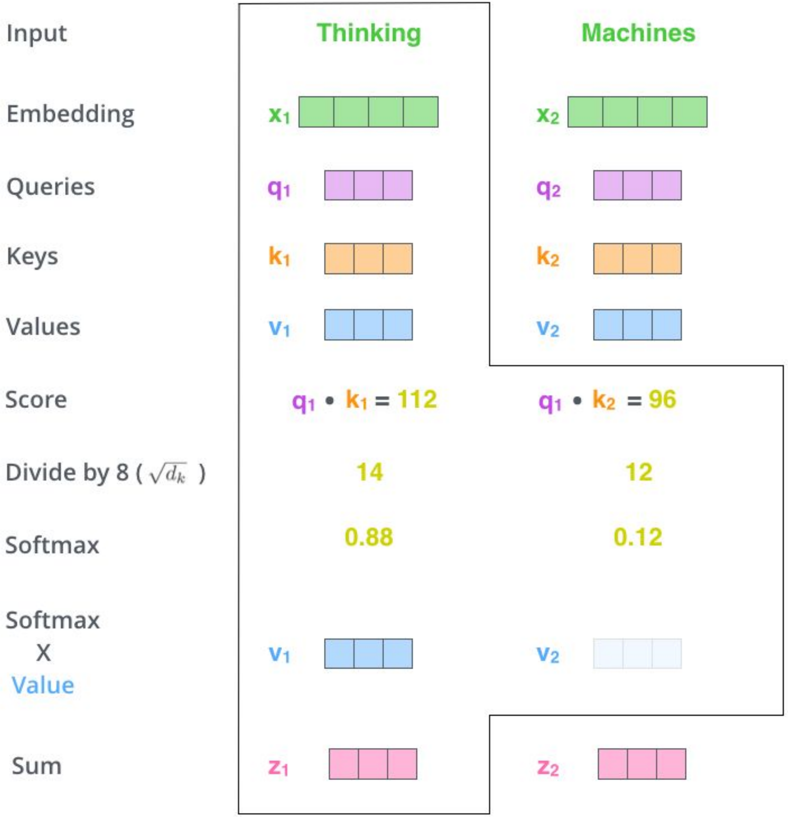
What happens mathematically inside Self-Attention?



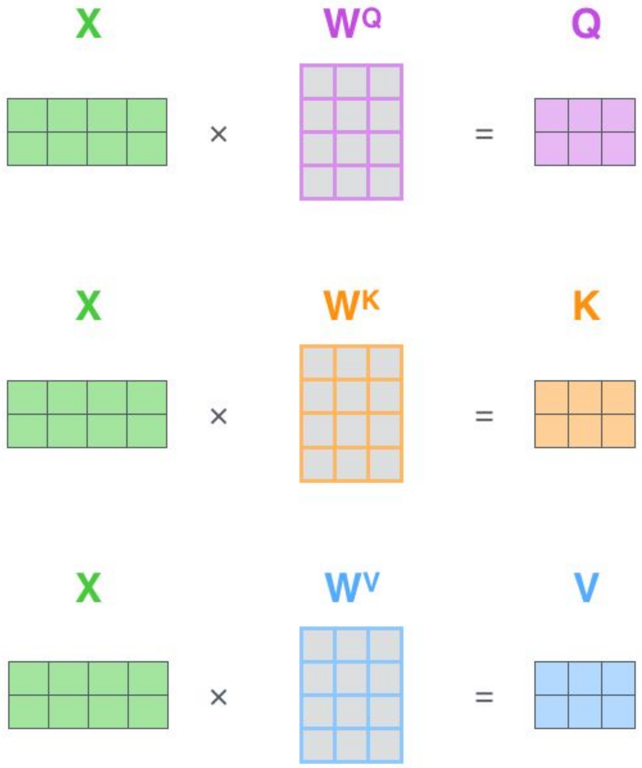
What happens mathematically inside Self-Attention?



What happens mathematically inside Self-Attention?



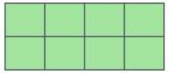
Self-Attention as Matrix Operations



Self-Attention as Matrix Operations

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

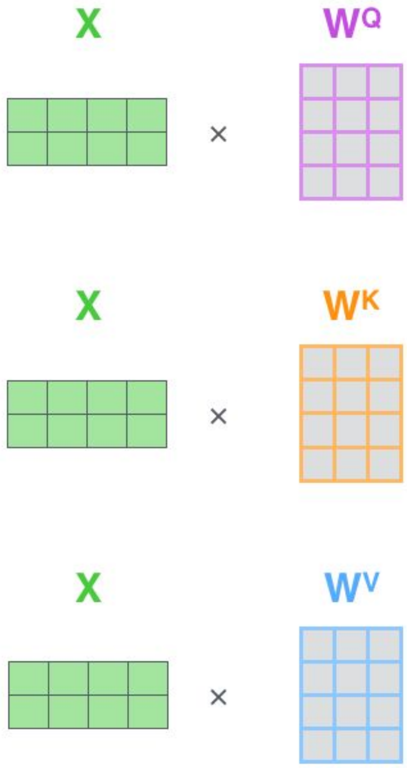
X



Self-Attention as Matrix Operations

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

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

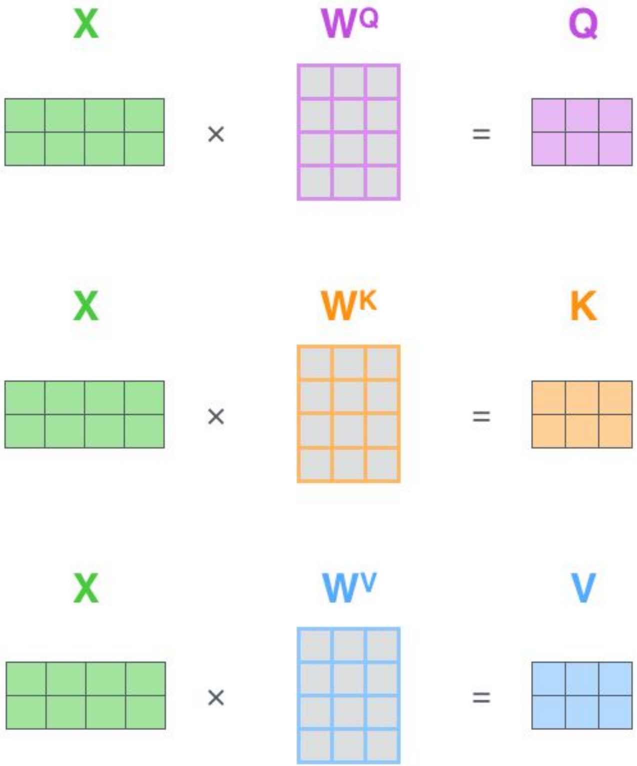


Self-Attention as Matrix Operations

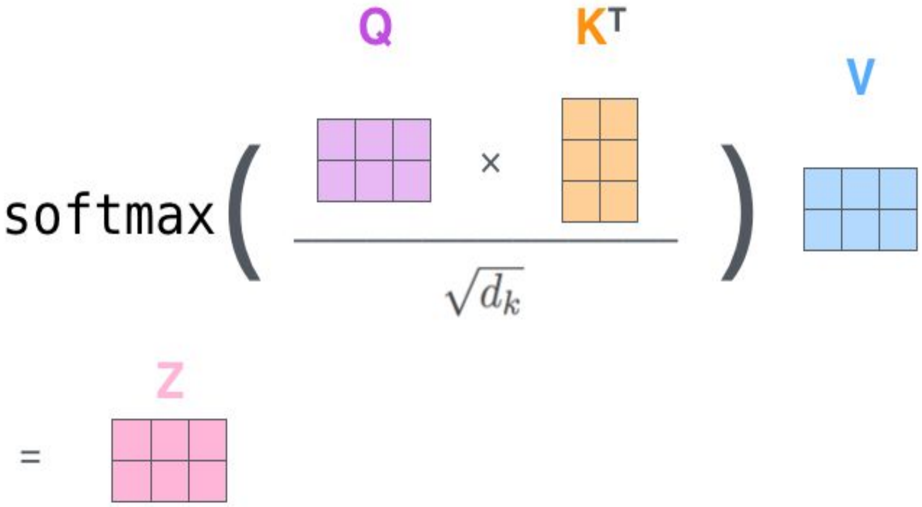
$$X \in \mathbb{R}^{n \times l}$$

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

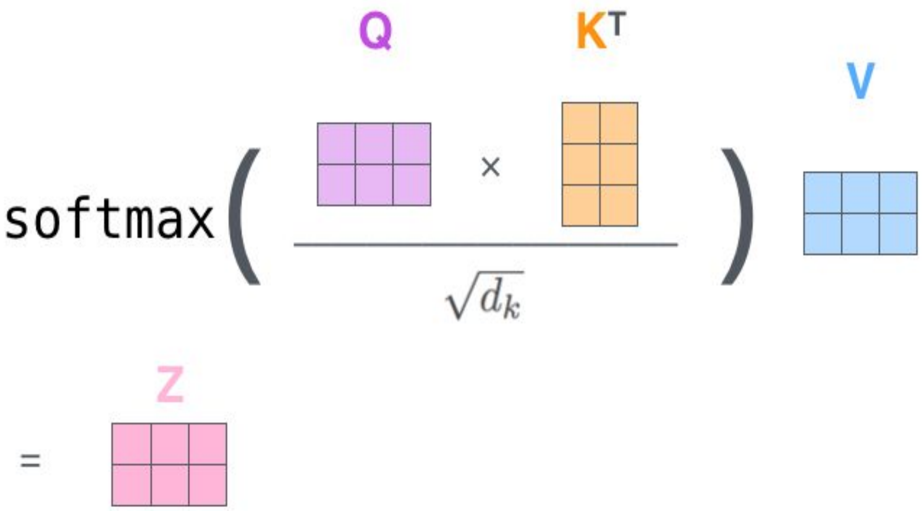
$$Q, K, V \in \mathbb{R}^{n \times d}$$

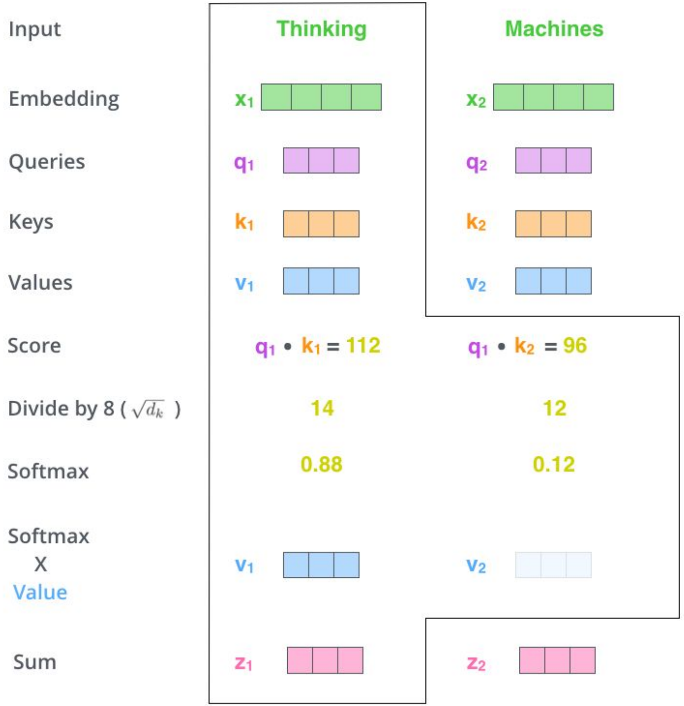


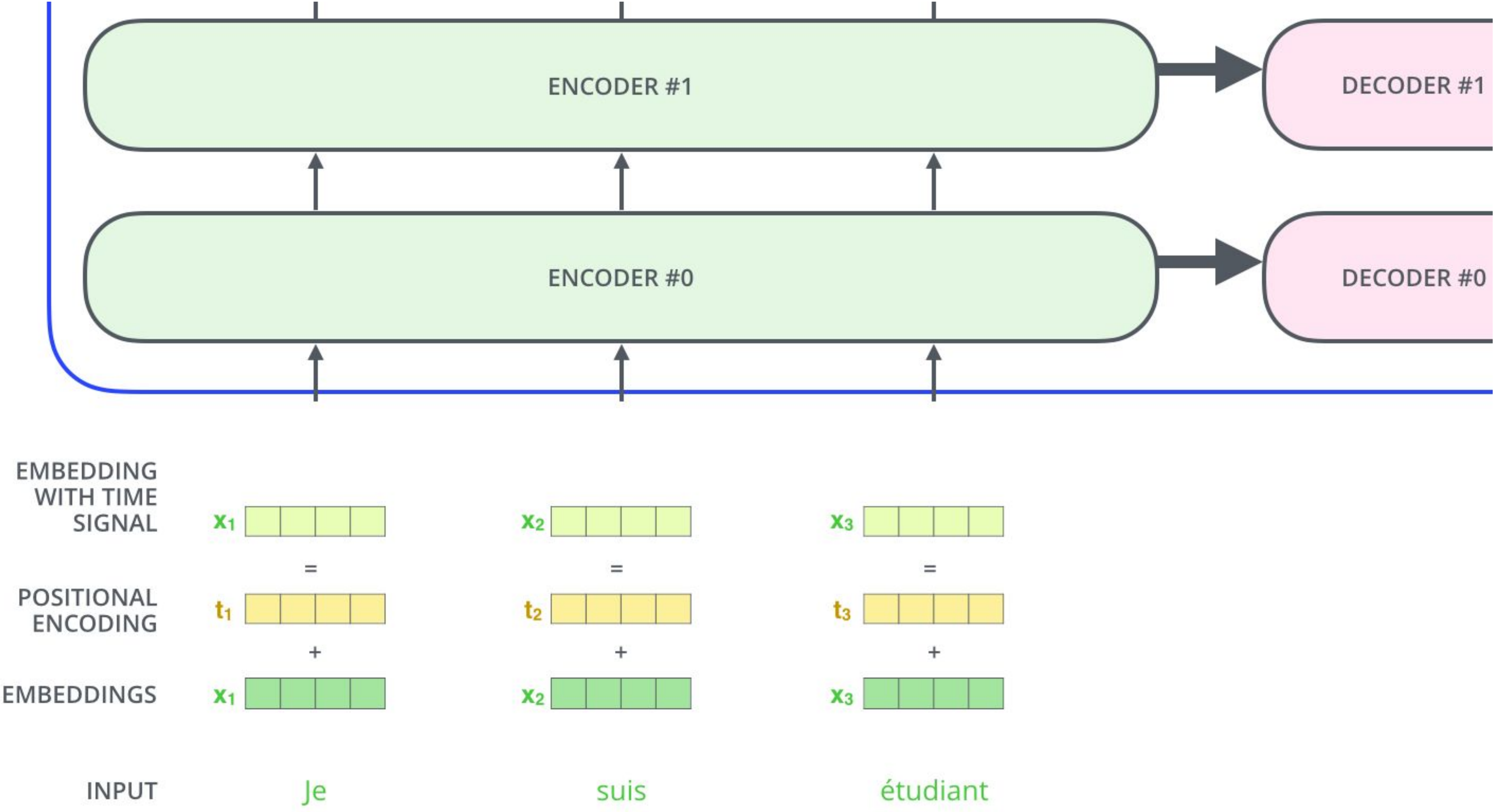
Self-Attention as Matrix Operations



Self-Attention as Matrix Operations



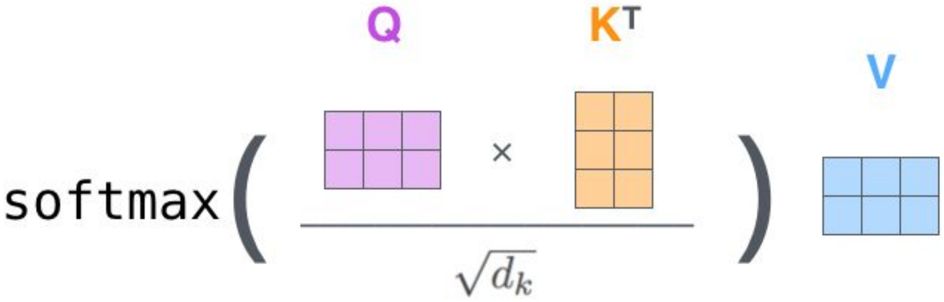




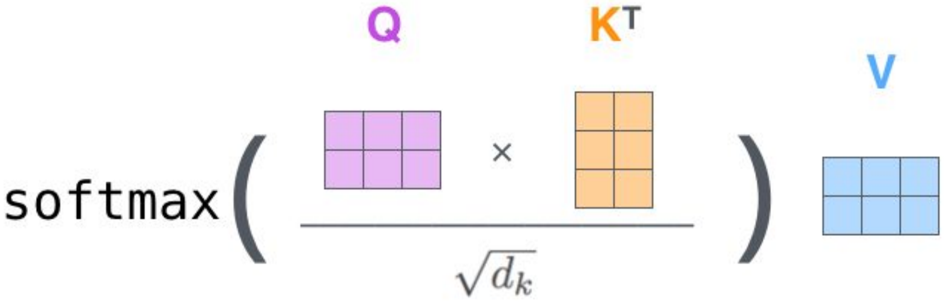
ENCODER #1
DECODER #1
ENCODER #0
DECODER #0
EMBEDDING WITH TIME SIGNAL
POSITIONAL ENCODING
EMBEDDINGS
INPUT
x_1
x_2
x_3
=
t_1
t_2
t_3
+
Je
suis
étudiant

Paper

The Problem

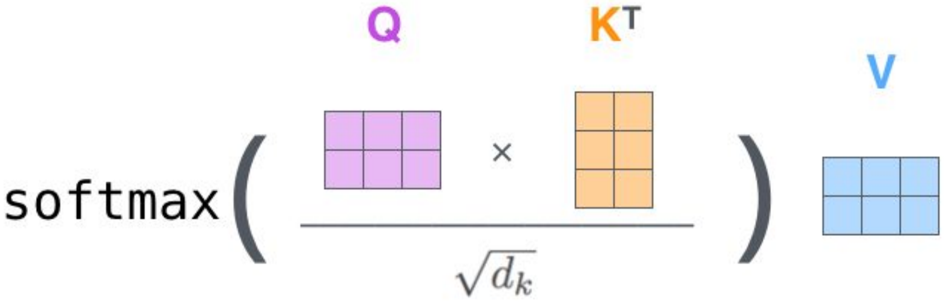


The Problem



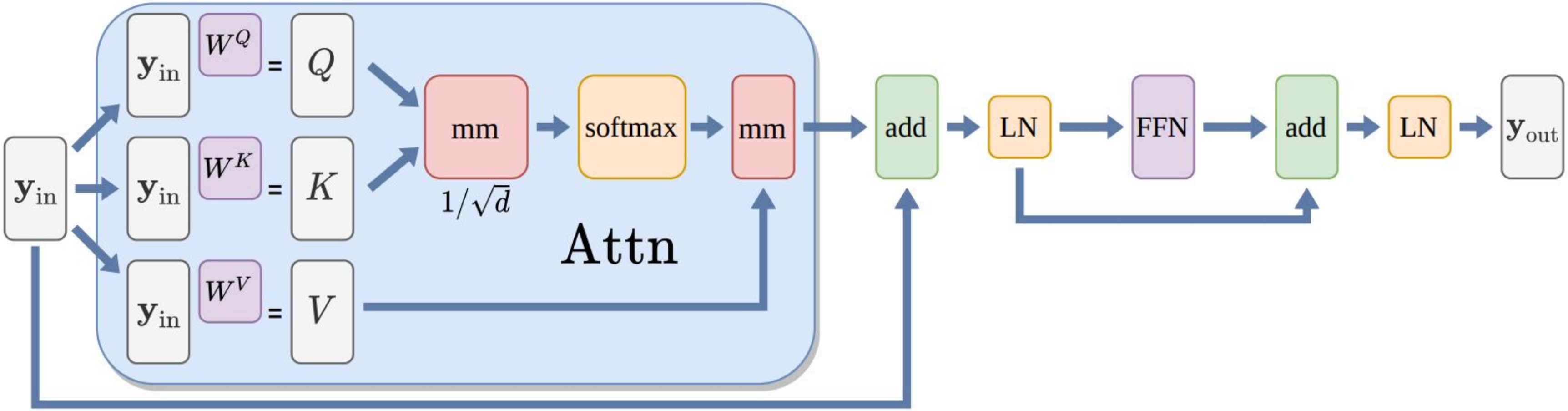
$$Q, K, V \in \mathbb{R}^{n \times d}$$

The Problem



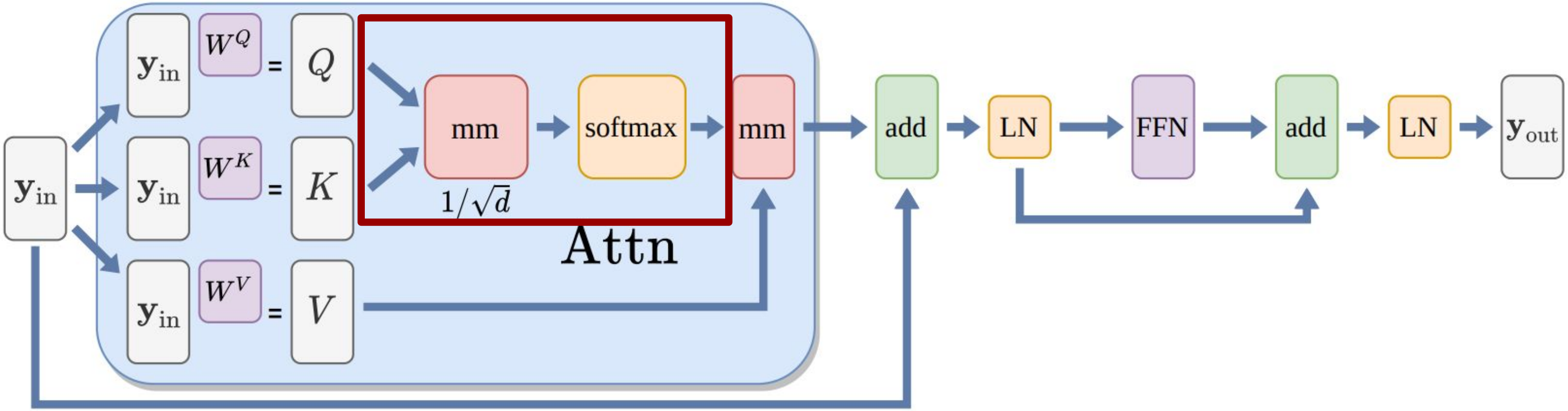
$$Q, K, V \in \mathbb{R}^{n \times d} \qquad O\left(n^2 d\right)$$

The Problem



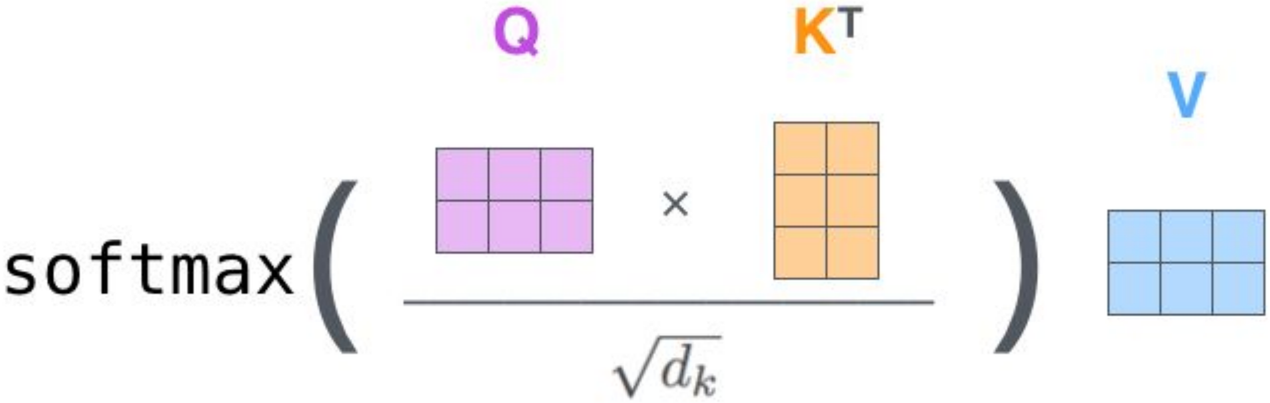
$\mathbf{y}_{\text{in}}$
$\mathbf{y}_{\text{in}}$ W^Q = Q
$\mathbf{y}_{\text{in}}$ W^K = K
$\mathbf{y}_{\text{in}}$ W^V = V
mm
$1/\sqrt{d}$
softmax
mm
Attn
add
LN
FFN
add
LN
$\mathbf{y}_{\text{out}}$

The Problem

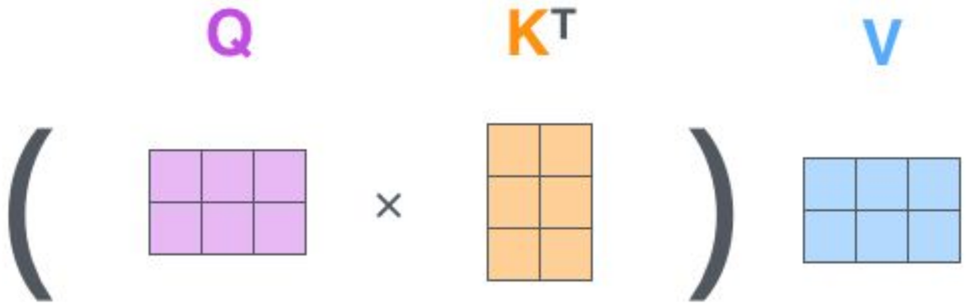


$\mathbf{y}_{\text{in}}$
$\mathbf{y}_{\text{in}}$
W^Q
=
Q
$\mathbf{y}_{\text{in}}$
W^K
=
K
$\mathbf{y}_{\text{in}}$
W^V
=
V
mm
$1/\sqrt{d}$
softmax
mm
Attn
add
LN
FFN
add
LN
$\mathbf{y}_{\text{out}}$

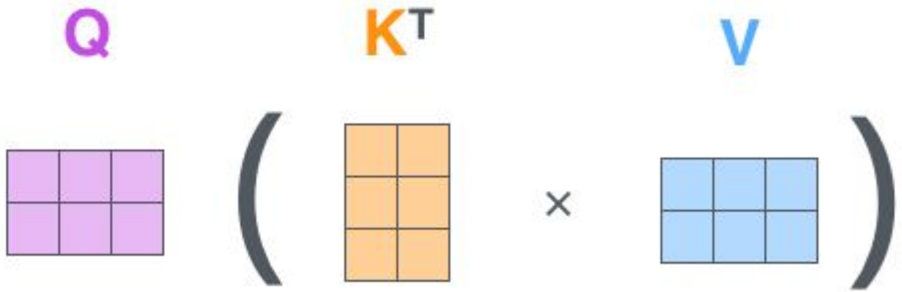
The Solution



The Solution

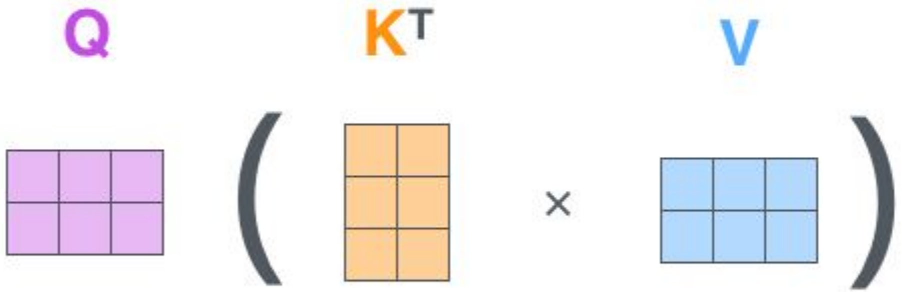


The Solution

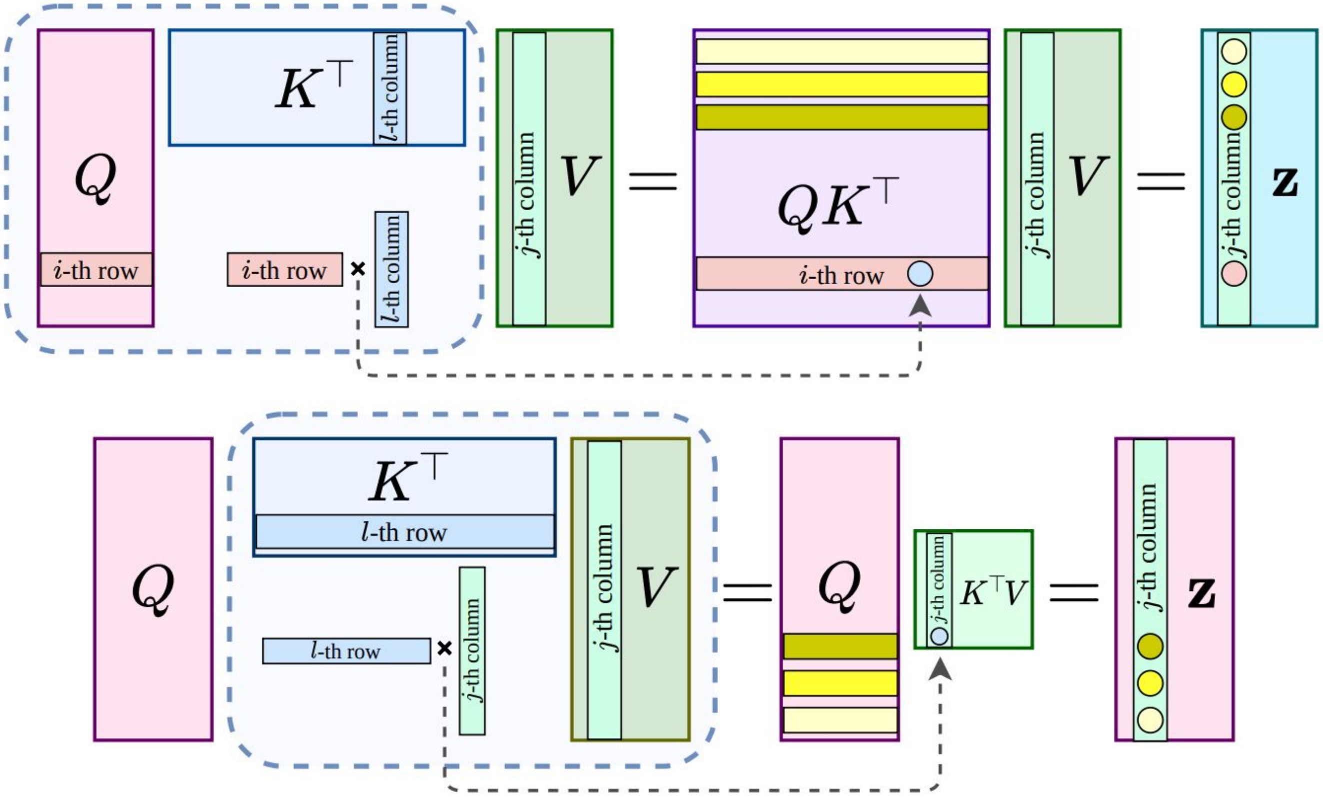


The Solution

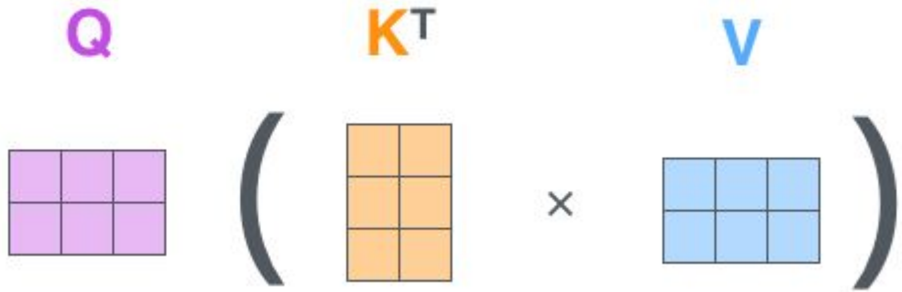
$$O\left(nd^2\right)$$



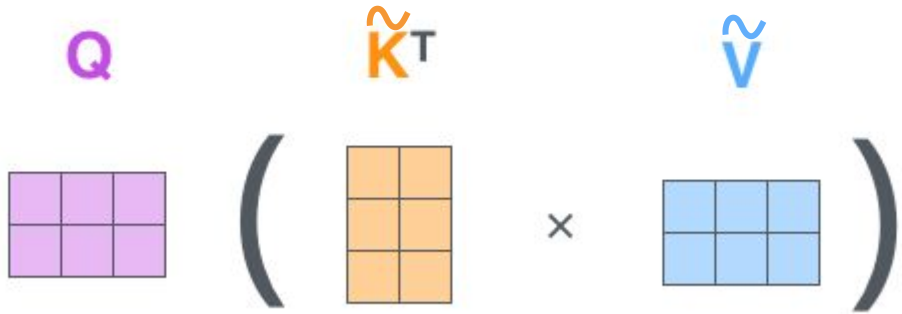
The Solution



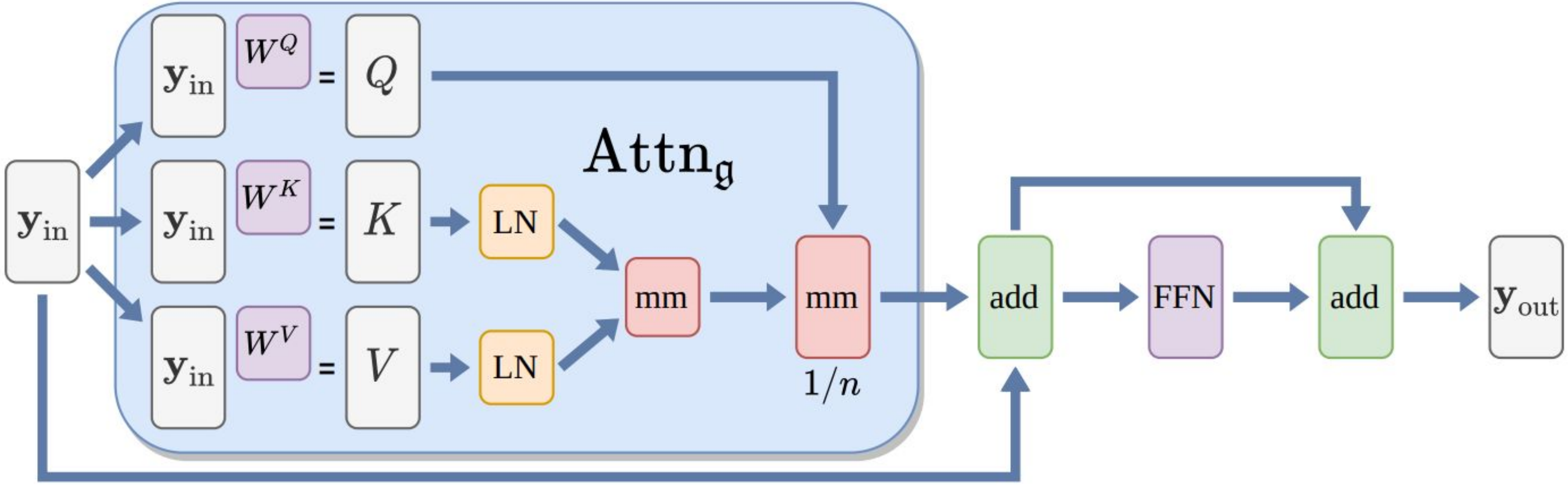
The Solution



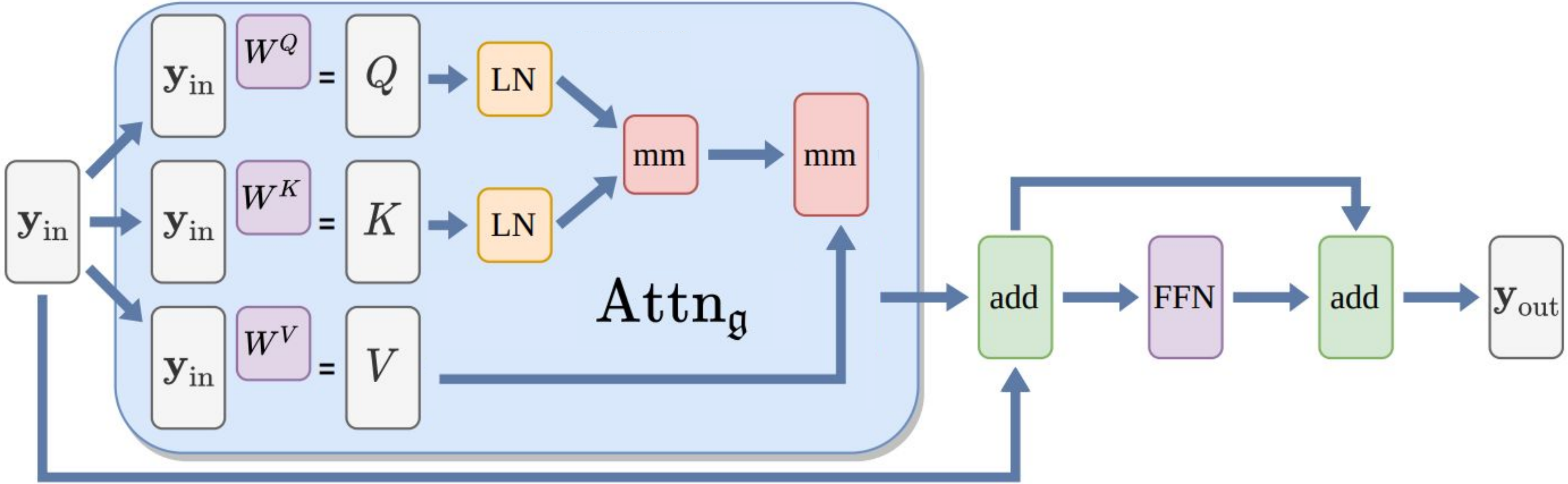
The Solution



The Galerkin Transformer



The Fourier Transformer



Mathematical Interpretation

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

Mathematical Interpretation

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

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

Mathematical Interpretation

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

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

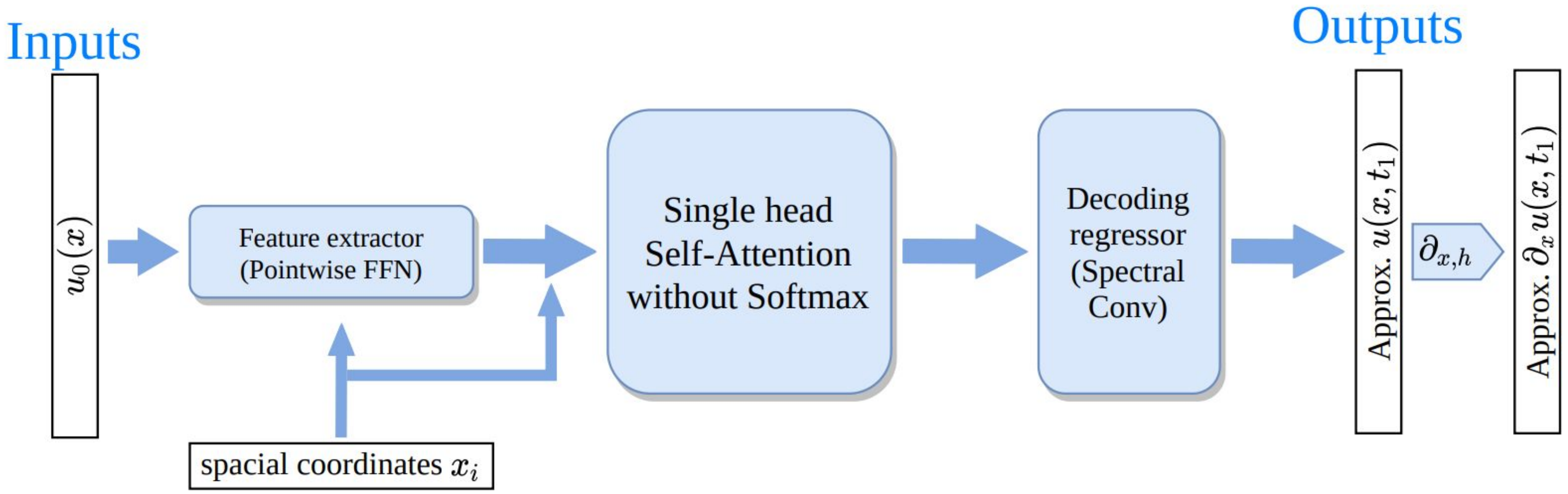
$$\mathfrak{b}\left(k_l, v_j\right) = \left(\widetilde{K}_i \cdot \widetilde{V}_j\right)$$

Example 1: viscous Burgers' equation

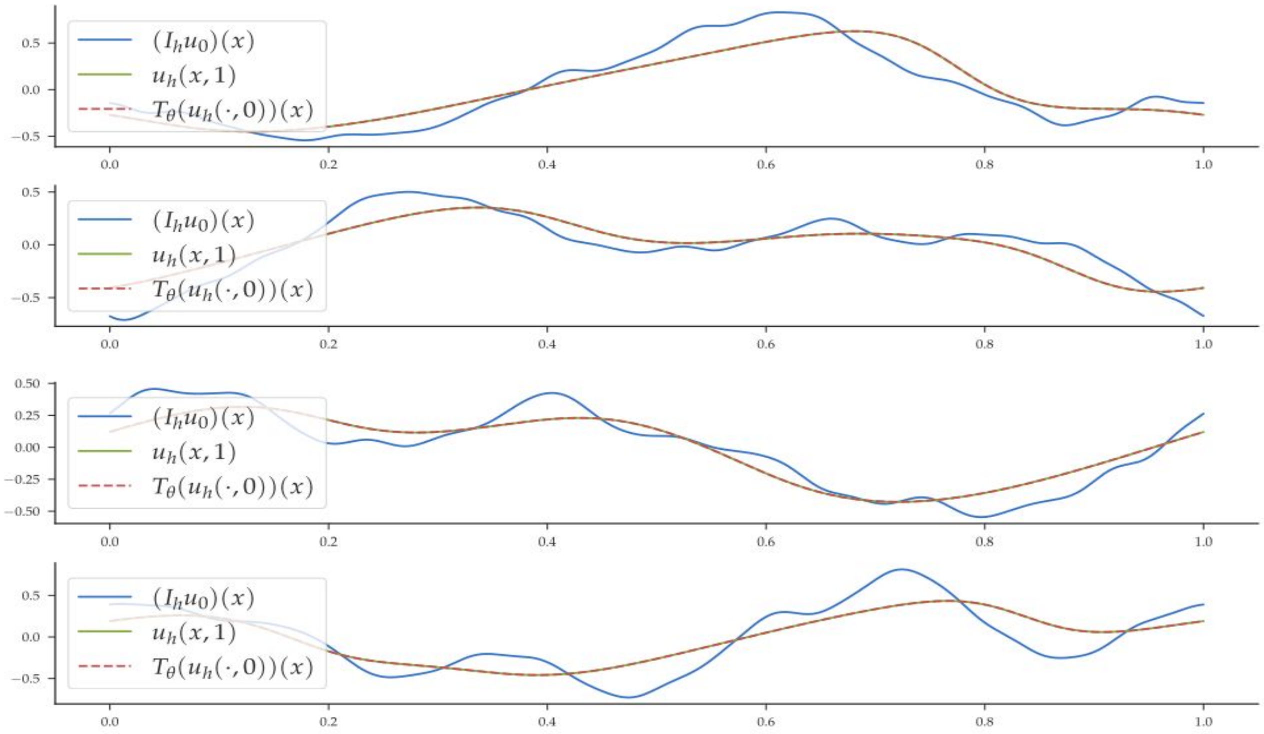
$$\begin{cases} \partial_t u + u\partial_x u = \nu\partial_{xx} u \quad \text{for } (x,t) \in (0,1)\times(0,1], \\ u(x,0) = u_0(x) \text{ for } x \in (0,1), \end{cases}$$

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

Example 1: viscous Burgers' equation



Example 1: viscous Burgers' equation



Example 1: viscous Burgers' equation

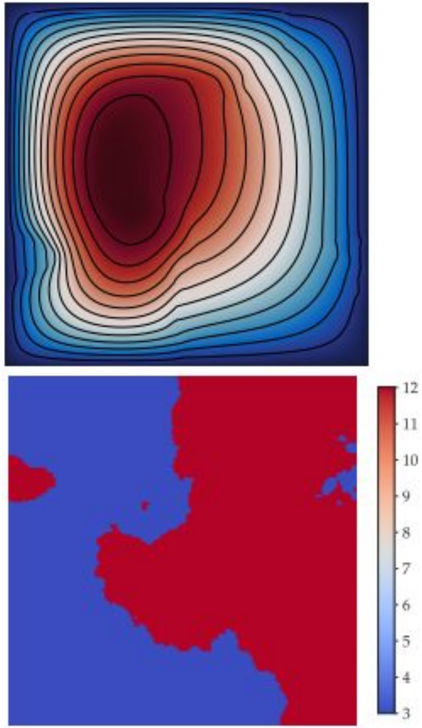
	$n = 512$	$n = 2048$	$n = 8192$
FNO1d [57]	15.8	14.6	13.9
FNO1d `1cycle`	4.373	4.126	4.151
FT regular Ln	1.400	1.477	1.172
GT regular Ln	2.181	1.512	2.747
ST regular Ln	1.927	2.307	1.981
LT regular Ln	1.813	1.770	1.617
FT Ln on Q, K	**1.135**	**1.123**	**1.071**
GT Ln on K, V	**1.203**	**1.150**	**1.025**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

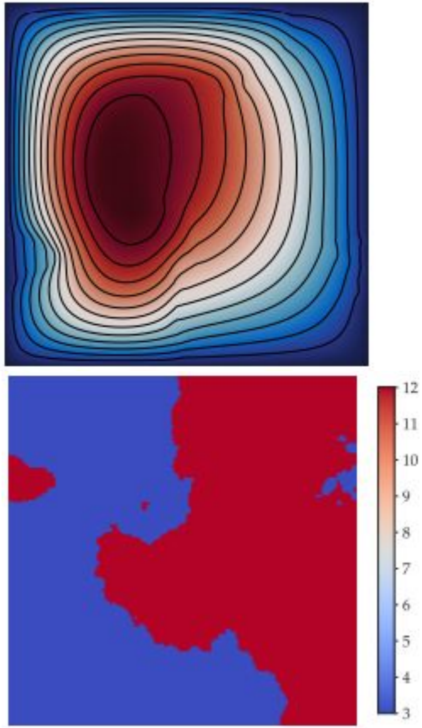
$$\begin{cases} -\nabla \cdot (a \nabla u) = f \text{ in } \Omega, \\ u = 0 \text{ on } \partial\Omega \end{cases}$$

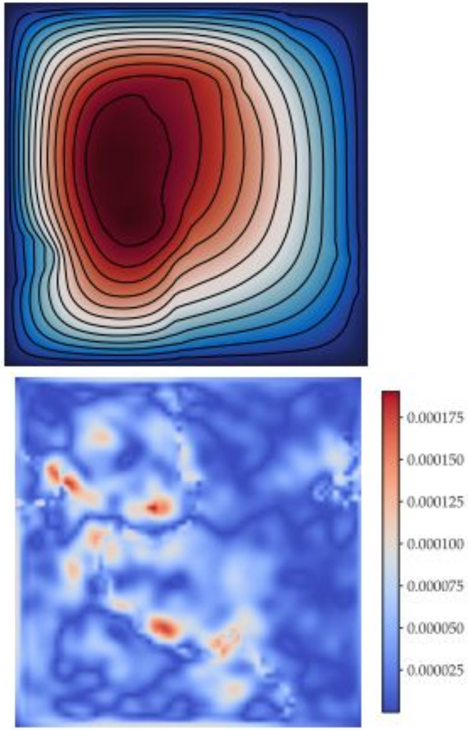
$$T : L^\infty(\Omega) \to H_0^1(\Omega), \quad a \mapsto u$$

Example 2: Darcy flow

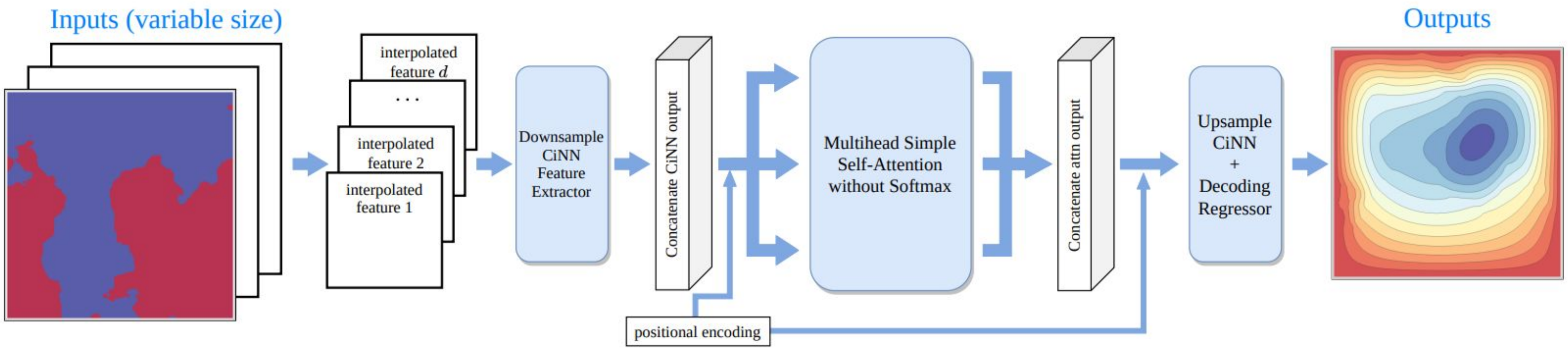


Example 2: Darcy flow

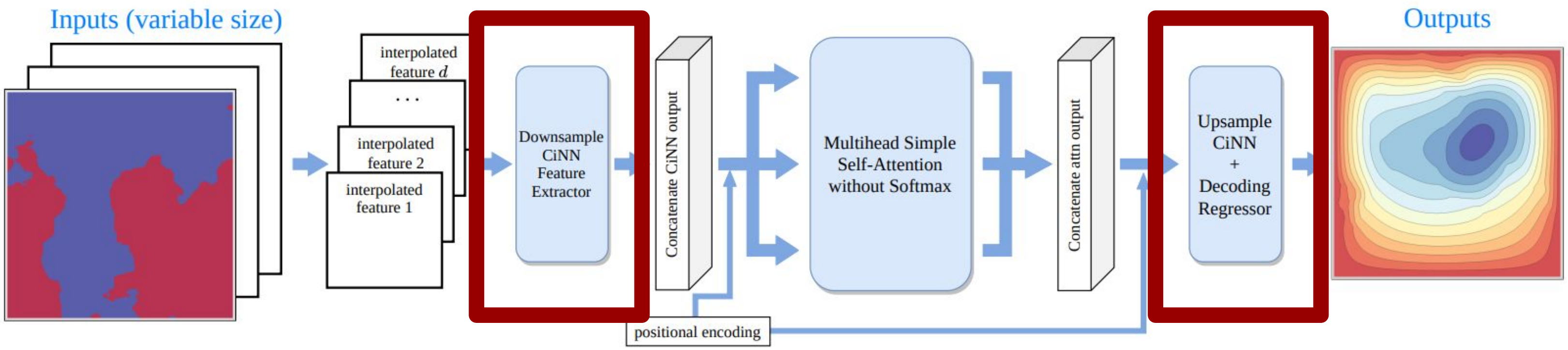




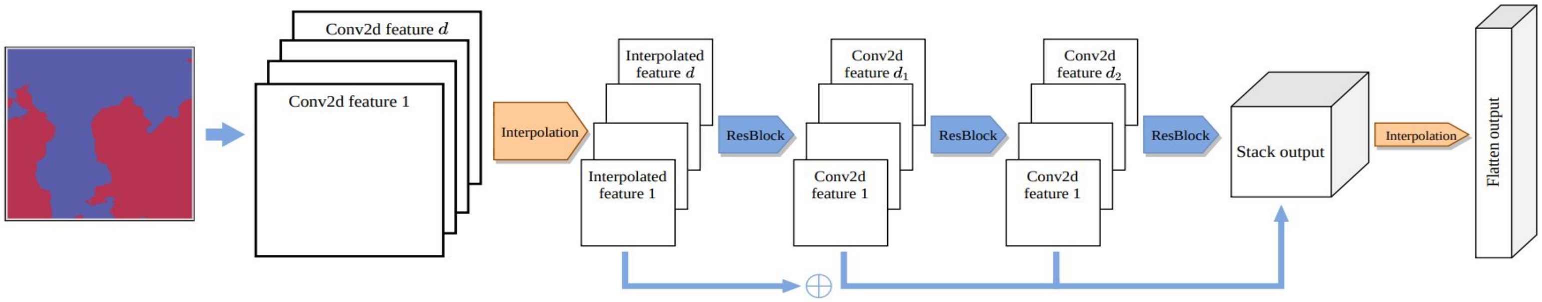
Example 2: Darcy flow

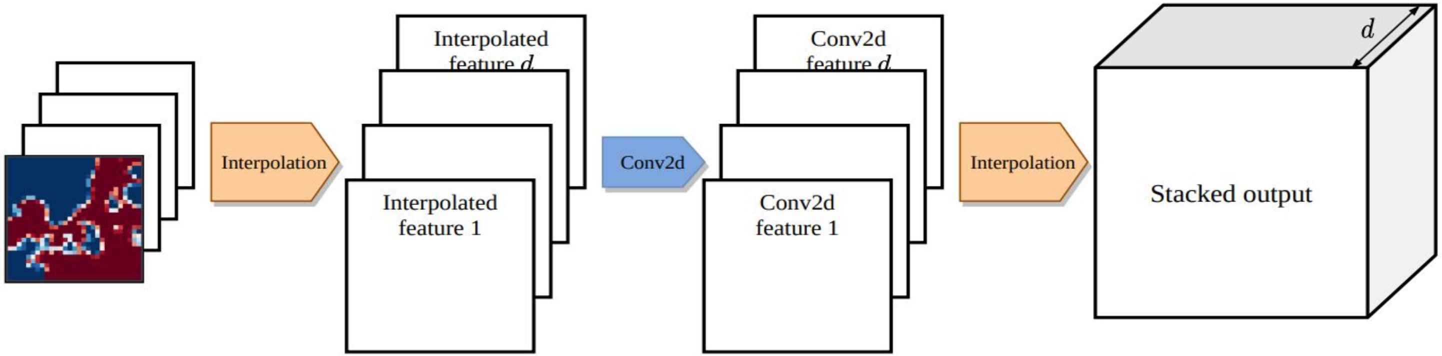


Example 2: Darcy flow



Example 2: Darcy flow





Example 2: Darcy flow

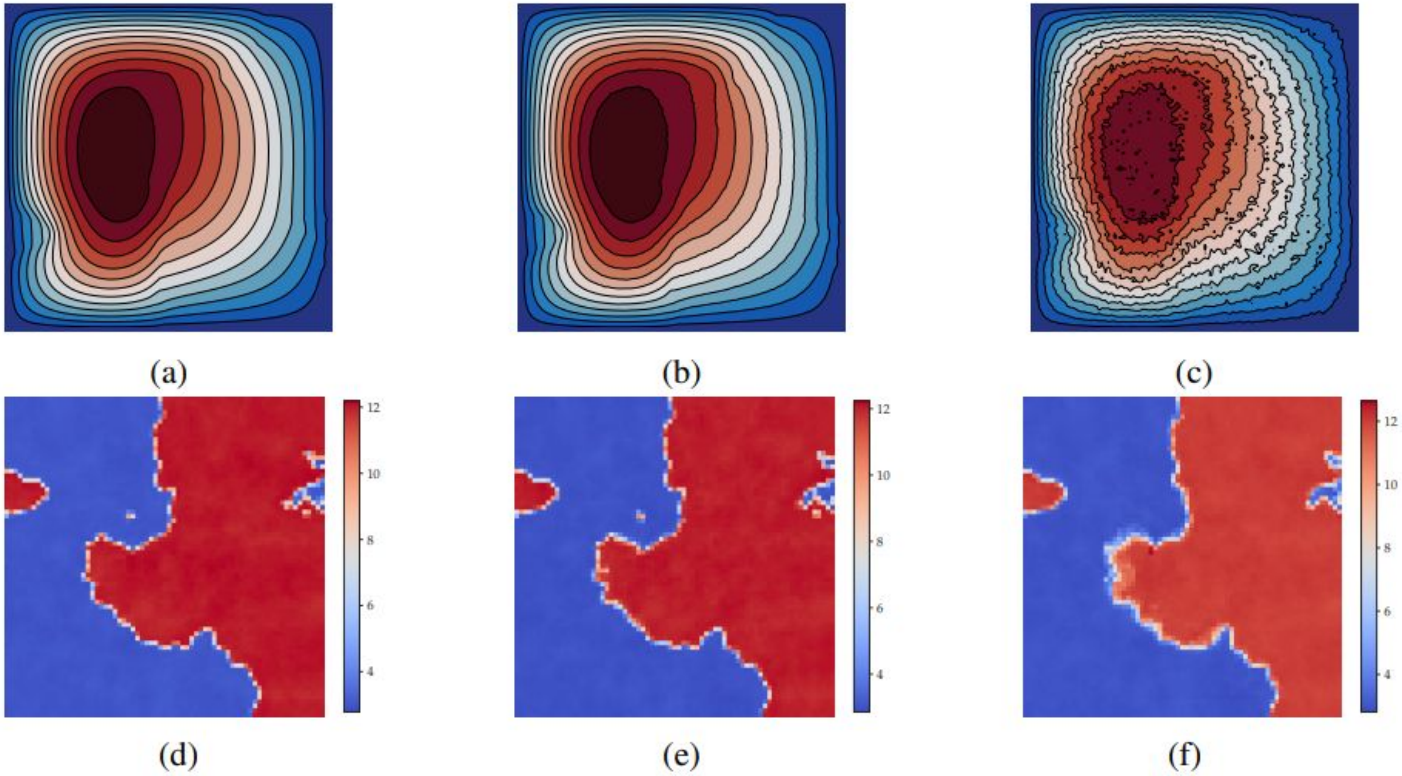
	$n_f, n_c = 141, 43$	$n_f, n_c = 211, 61$
FNO2d [57]	1.09	1.09
FNO2d `1cycle`	1.419	1.424
FT regular Ln	**0.838**	**0.847**
GT regular Ln	0.894	0.856
ST regular Ln	1.075	1.131
LT regular Ln	1.024	1.130
FT Ln on Q, K	0.873	0.921
GT Ln on K, V	**0.839**	**0.844**
ST Ln on Q, K	0.946	0.959
LT Ln on K, V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

Example 3: inverse coefficient identification for Darcy flow

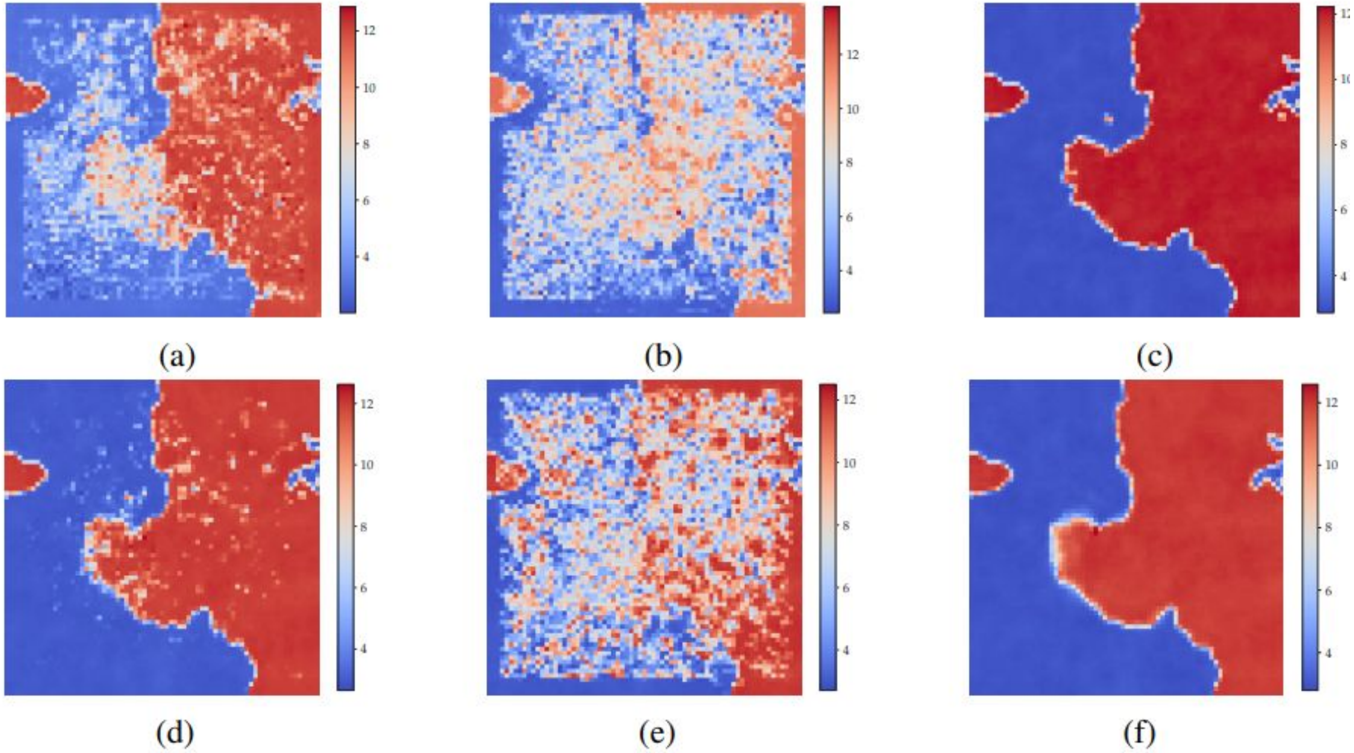
$$\begin{cases} -\nabla \cdot (a\nabla u) = f \text{ in } \ \Omega, \\ \qquad\qquad\quad u = 0 \text{ on } \partial\Omega \end{cases}$$

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

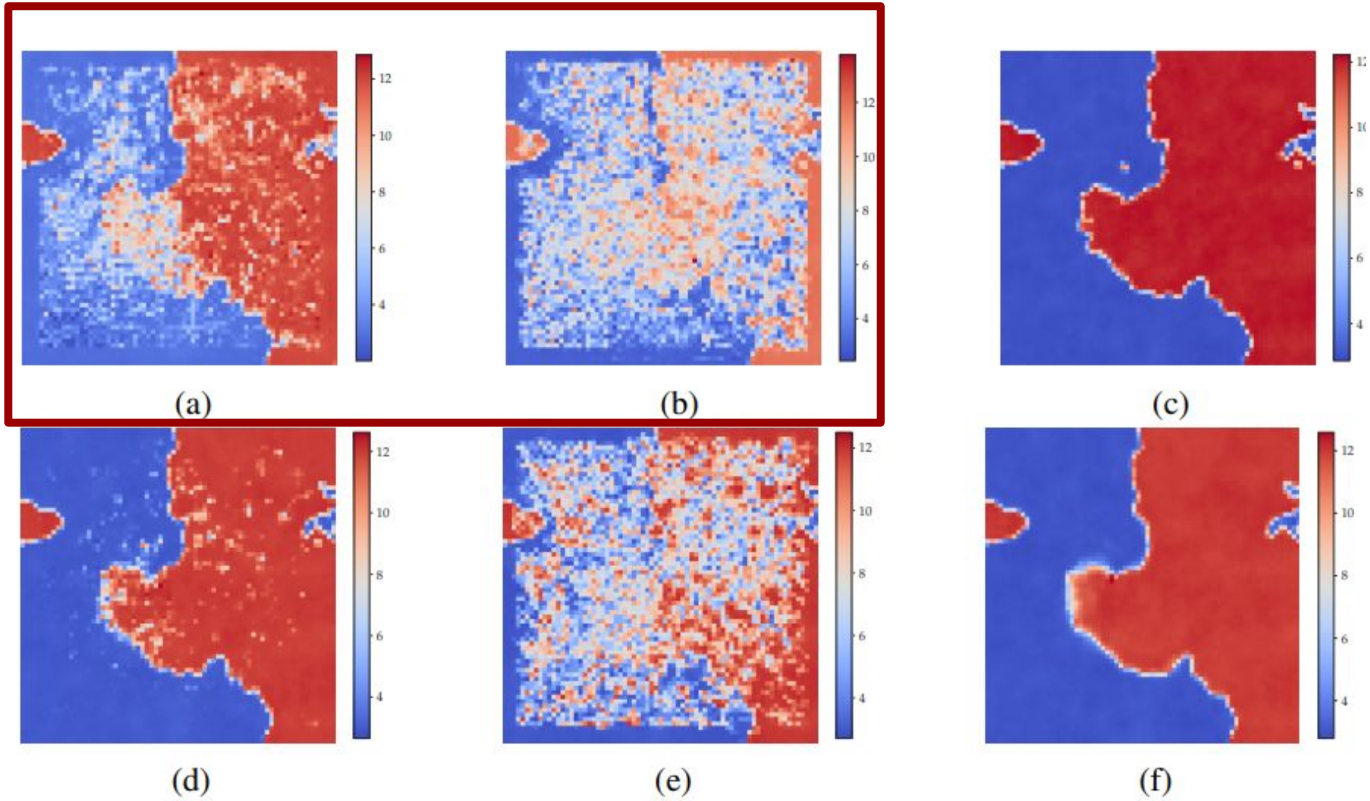
Example 3: inverse coefficient identification for Darcy flow



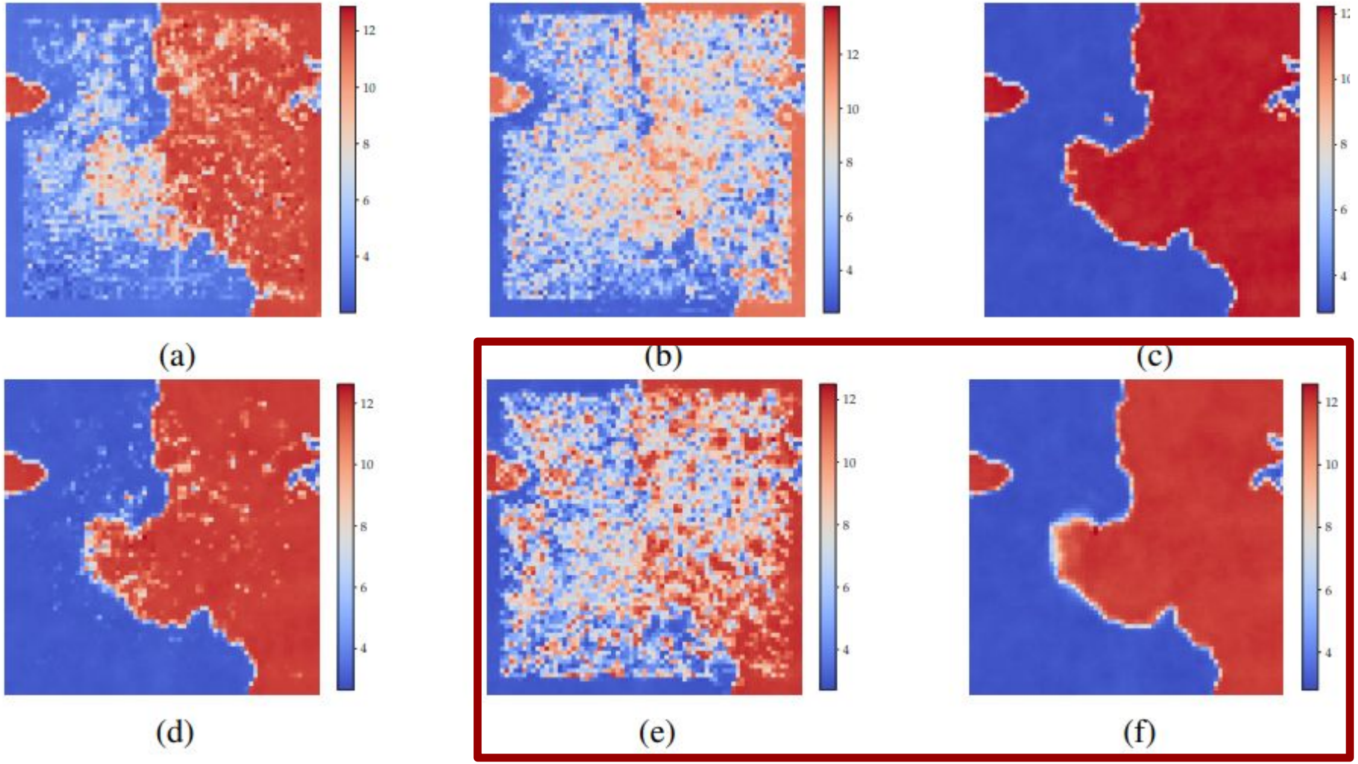
Example 3: inverse coefficient identification for Darcy flow



Example 3: inverse coefficient identification for Darcy flow



Example 3: inverse coefficient identification for Darcy flow



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.