Galerkin Transformer

Shuhao Cao

supported in part by NSF grant DMS-1913080 and DMS-2136075

Washington University in St. Louis

https://arxiv.org/abs/2105.14995
https://github.com/scaomath/galerkin-transformer
https://scaomath.github.io/blog/galerkin-transformer/

How the story began

Prof. Dr. Long Chen (UC Irvine) shared with me a blog article in Chinese about a submission in ICLR 2021 back in October 2020[1].

无惧分辨率变化，顽强求解PDE家族：加州理工学院等提出傅里叶神经算子方法

机器之心 2020-10-23

机器之心报道

编辑：魔王、小舟

来自加州理工学院和普渡大学的研究者通过直接在傅里叶空间中对积分核进行参数化，构造了一种新的神经算子——傅里叶神经算子（FNO）。

这篇由加州理工学院 Zongyi Li、Anima Anandkumar，以及普渡大学（Purdue University）Kamyar Azizzadenesheli 等人提交的论文的审阅。

本文的作者之一 Anima Anandkumar 是加州理工学院教授，也是英伟达机器学习研究的负责人。

- Rough translation: The Caltech group proposed an awesome method called "Fourier Neural Operator" (FNO).
- Simple idea: learn a kernel function in frequency domain. Combining with nonlinearity in space to achieve the state-of-the-art accuracy and performance.

[1] Z. Li et al. In *International Conference on Learning Representations*, 2021.

Toy model: a simple viscid Burgers' equation benchmark

A nice benchmark problem with a periodic boundary condition in PDE dataset[2] for a viscosity $\nu = (2\pi)^{-1}0.1$

$$\begin{cases} \dfrac{\partial u}{\partial t} + u\dfrac{\partial u}{\partial x} = \nu\dfrac{\partial^2 u}{\partial x^2} & \text{for } (x,t) \in (0,1)\times(0,t_1], \\ u(x,0) = u_0(x) \text{ for } x \in (0,1). \end{cases}$$

$u_0 \in C^0_{per}([0,1]) \cap L^2([0,1])$, $u(\cdot, t_1) \in H^1_{per}((0,1))$.

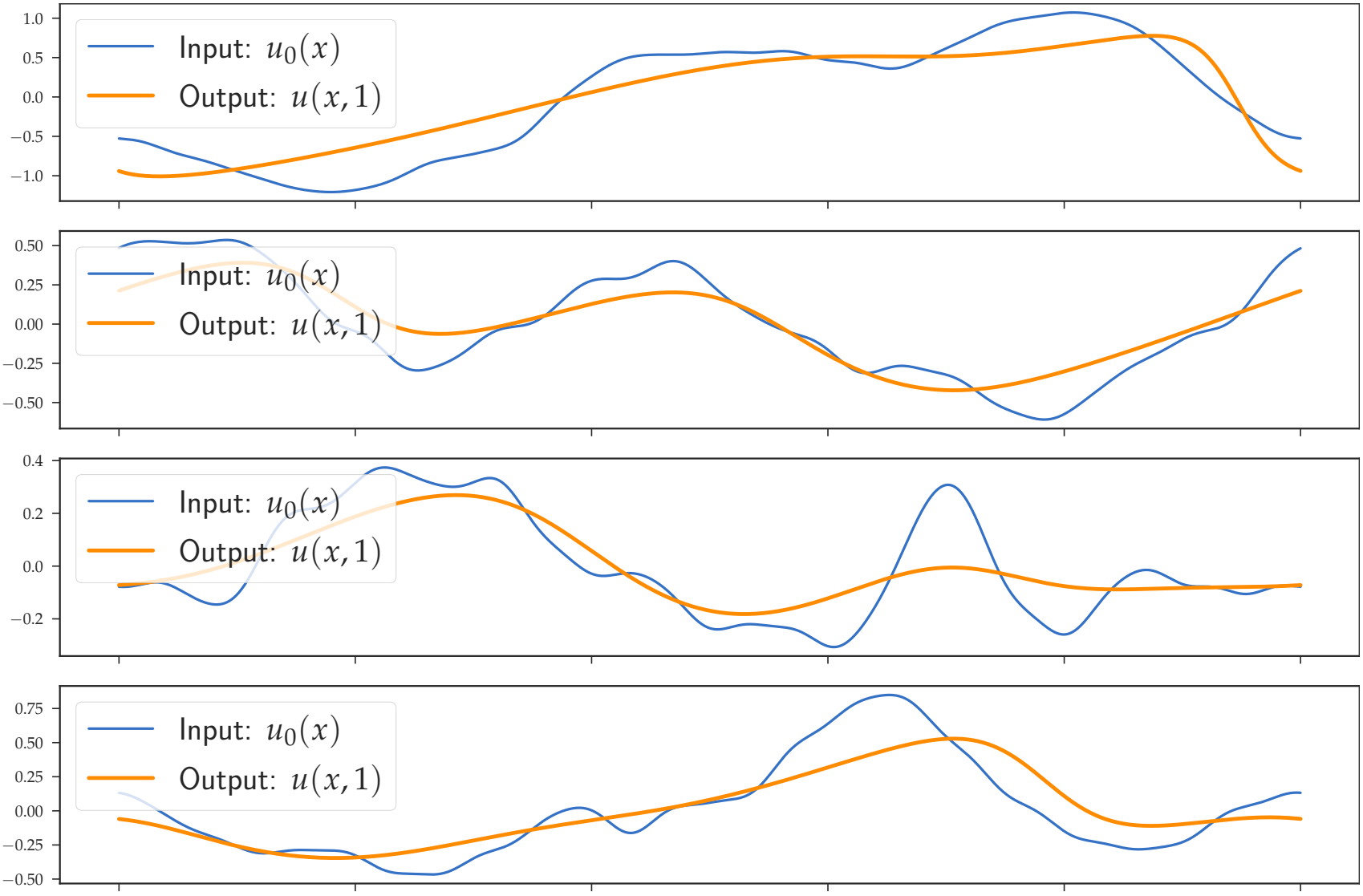
The operator to be learned is an approximation to $t_1 = 1 \gg \Delta t$:

$$T \text{ maps } u_0(\cdot) \text{ to } u(\cdot, 1).$$

- Directly inferring the solution at a time step that traditional integrator (usually in 10^{-4} to 10^{-5}range) can never dream of (Courant–Friedrichs–Lewy condition).

[2] Z. Li et al. In *International Conference on Learning Representations*, 2021.

What does this problem look like?



Operator learning problem related to PDE

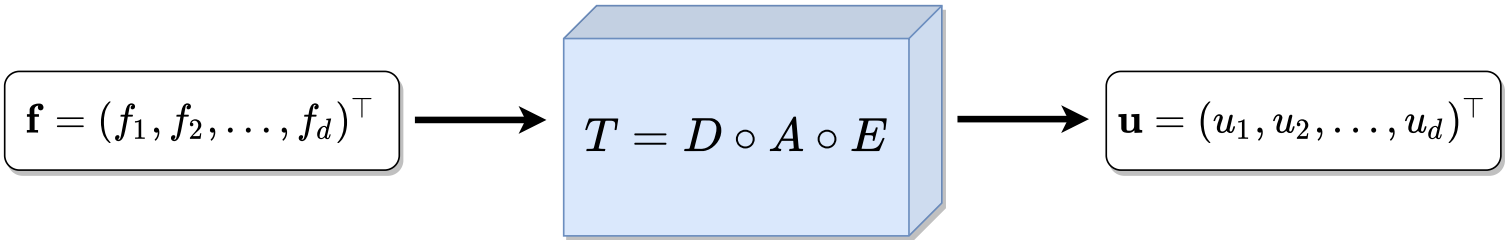
T to be learned are between infinite dimensional Hilbert (Banach) spaces

- Parametric PDEs:
  - The mapping between a varying coefficient to the solution.
  - The inverse mapping: solution + noise $\mapsto$ coeff.
- Nonlinear initial value problem for stiff systems:
  - Direct inference from the initial condition to the solution at a much later time.

Advantages:

- New sample, new boundary condition, new data, no retrain (unlike function learners).
- Train with few samples ($\sim$ `1e3`) unlike big language or vision models, fast inference.
- Highly nonlinear problems, problem with poor stability or well-posedness for traditional methods.

Operator learning problem discretized



- `seq2seq` in Neural Machine Translation is a discretized operator learning problem.
- Encoded message f $\in \mathbb{R}^{n \times d} \Longrightarrow$ Latent representation $\in \mathbb{R}^{n \times d} \Longrightarrow$ Decoded message u $\in \mathbb{R}^{n \times d}$.
- d: latent dimensions `dmodel` (usually fixed). n: length of the sequence (discretization size, can be arbitrary under the computational budget).
- The model can be trained on a lower resolution and evaluated at a (much) higher resolution.

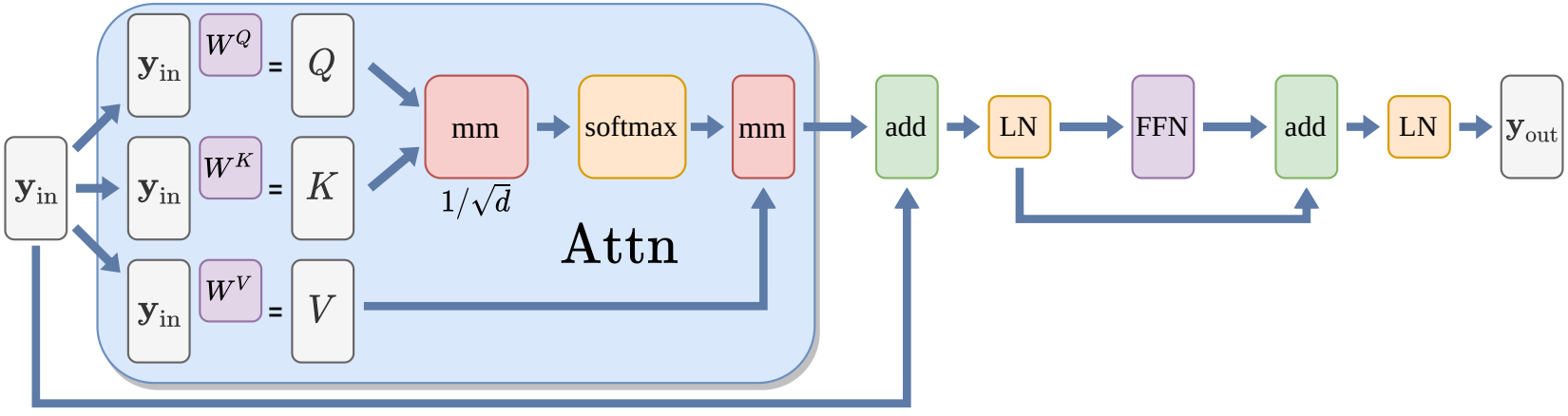
Can we apply the attention mechanism in Transformer which is all we need to the operator learning?

[PDF] Attention is all you need

A Vaswani, N Shazeer, N Parmar… - Advances in neural …, 2017 - papers.nips.cc

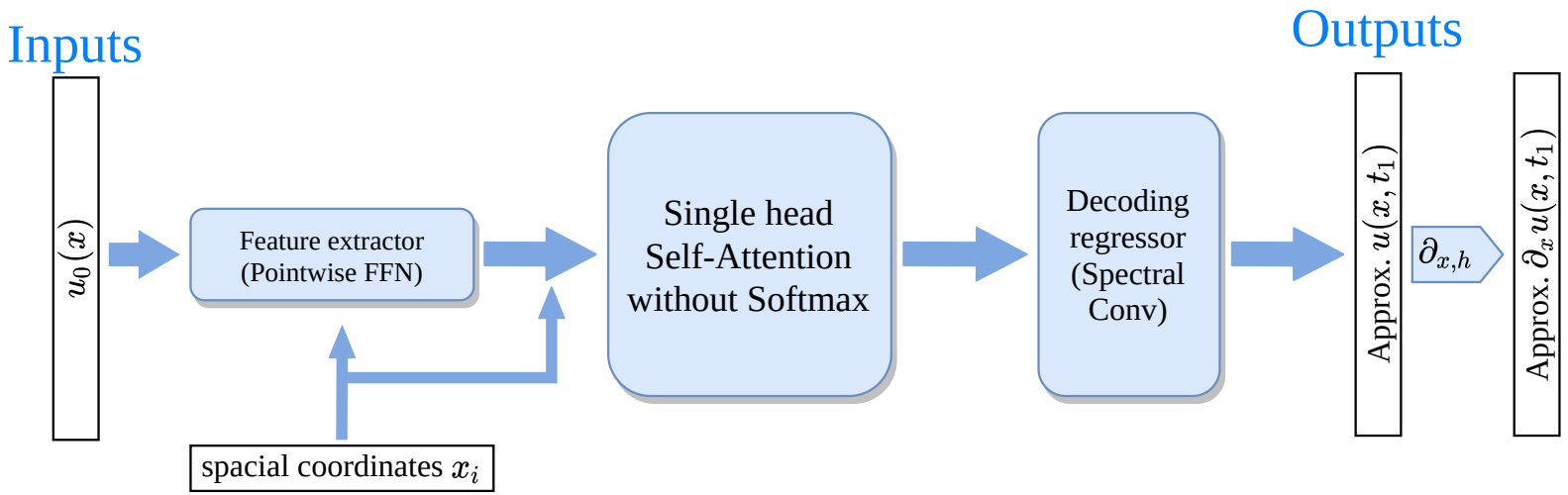
The dominant sequence transduction models are based on complex recurrent orconvolutional neural networks in an encoder and decoder configuration. The best performing such models also connect the encoder and decoder through an attentionm …

☆ 99 Cited by 25287 Related articles All 28 versions »



Self-attention mechanism in the classical Transformer (figure reproduced for a single attention head from *Attention is All You Need*).

A simple encoder-decoder operator learner



A simple attention-based "encoder-decoder" mimicking the ones in NLP.

- Feature extractor: Feedforward, CNN, RNN, etc.
- Encoder: a stack of identical attention-based encoder layers.
- Decoder: a regressor mapping the latent representation to the target dimension, problem dependent.
- Positional encodings: problem dependent.

Burgers' benchmark: first shot

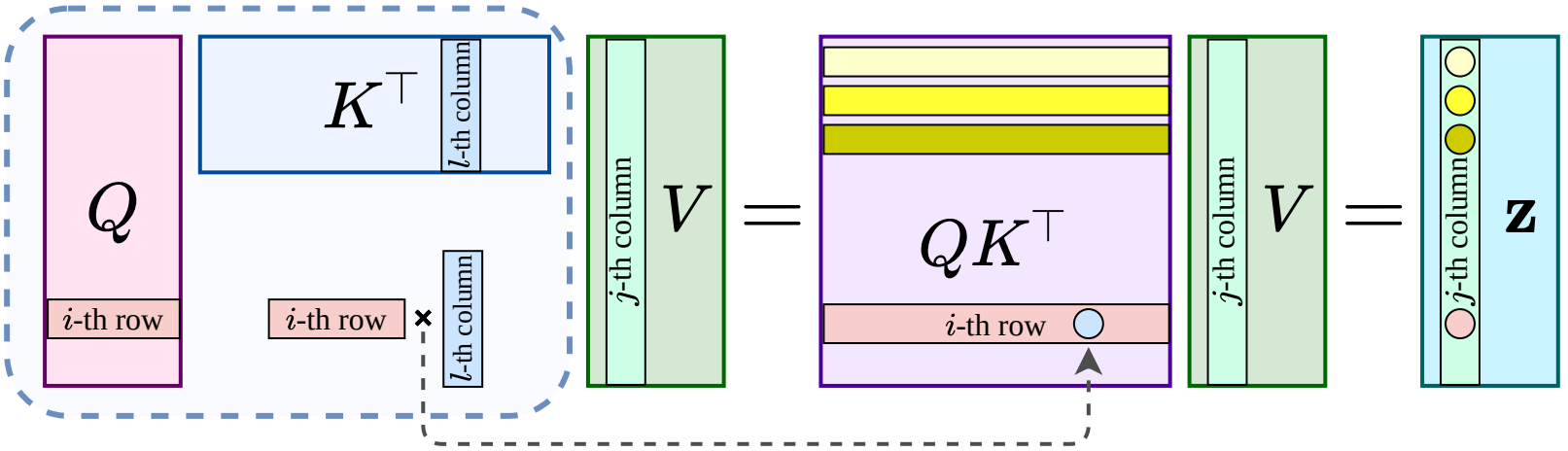
- Classic vanilla Transformer (encoder only) + original hyperbolic manifold'ish positional encodings.
- Fourier Neural Operator (FNO) + Euclidean coordinate positional encodings.
- Grid size: $n = 2048$. Inference either on $n = 2048$ or 8192.
- Training pairs: 1024; testing: 100. Loss: weighted $\|\cdot\|_{H^1}$.
- Trainer: 25600 iterations of ADAM (100 epochs), `1cycle` scheduler.

	$n = 2048$ (eval)	GFLOP/backprop	$n = 8192$ (zero-shot eval)	# params
FNO1d original[3]	1.46×10^{-2}	361.94 ± 1.65	1.26×10^{-2}	570k
FNO1d re-implement	4.37×10^{-3}	369.13 ± 1.81	4.18×10^{-3}	549k
Vanilla Transformer	1.41×10^{-1}	$\approx 12T$	1.47×10^{-1}	22m

Not good... End of the story (2020 Nov)?

[3] Z. Li et al. In *International Conference on Learning Representations*, 2021.

Rethinking the scaled dot-product Softmax($QK^\top$)V



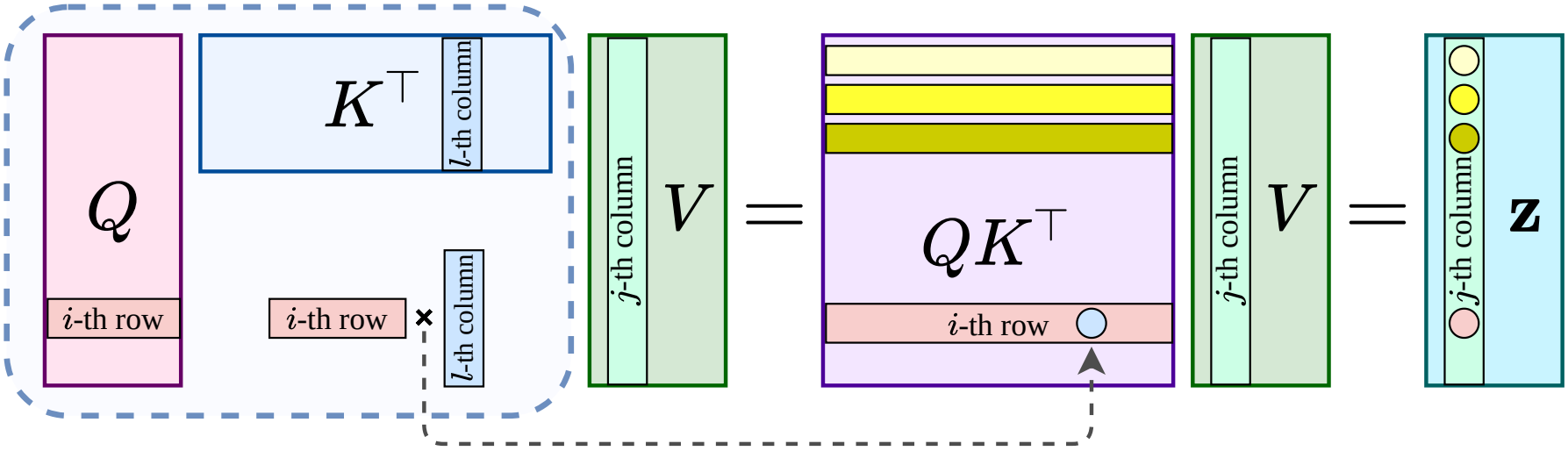
$$(\boldsymbol{z}_i)_j = h(QK^\top)_{i\bullet}\,\boldsymbol{v}^j = h\big(\boldsymbol{q}_i\cdot\boldsymbol{k}_1,\ldots,\boldsymbol{q}_i\cdot\boldsymbol{k}_l,\ldots,\boldsymbol{q}_i\cdot\boldsymbol{k}_n\big)^\top\cdot\boldsymbol{v}^j$$

$$= h\sum_{l=1}^{n}(\boldsymbol{q}_i\cdot\boldsymbol{k}_l)(\boldsymbol{v}^j)_l \approx \int_\Omega \big(\zeta_q(x_i)\cdot\phi_k(\xi)\big)v_j(\xi)\mathrm{d}\xi,$$

The i-th row in the output computes approx. an integral transform with a non-symmetric learnable low-rank kernel function $\kappa(x,\xi) := \zeta_q(x)\phi_k(\xi)$:

$$\boldsymbol{z}_i \approx \int_\Omega \big(\zeta_q(x_i)\cdot\phi_k(\xi)\big)\psi_v(\xi)\mathrm{d}\xi, \quad \text{where } \boldsymbol{q}_i = \zeta_q(x_i), \boldsymbol{k}_i = \phi_k(x_i), \boldsymbol{v}_i = \psi_v(x_i).$$

Standing on the shoulder of Giants



- The positive attention kernel Softmax(QK^T) is very familiar to the FNO's spectral convolution (treating FFT/iFFT as change of basis).
- But FNO does not have softmax!
- Softmax is extremely expensive (global dependence in backprop).
- Computational cost of QK^T scales quadratically w.r.t. n.

Positional token embedding to Hilbertian discretization

- Re-interpreting the latent representation in $\mathbb{R}^{n\times d}$ from:

Row = A word in a sentence to

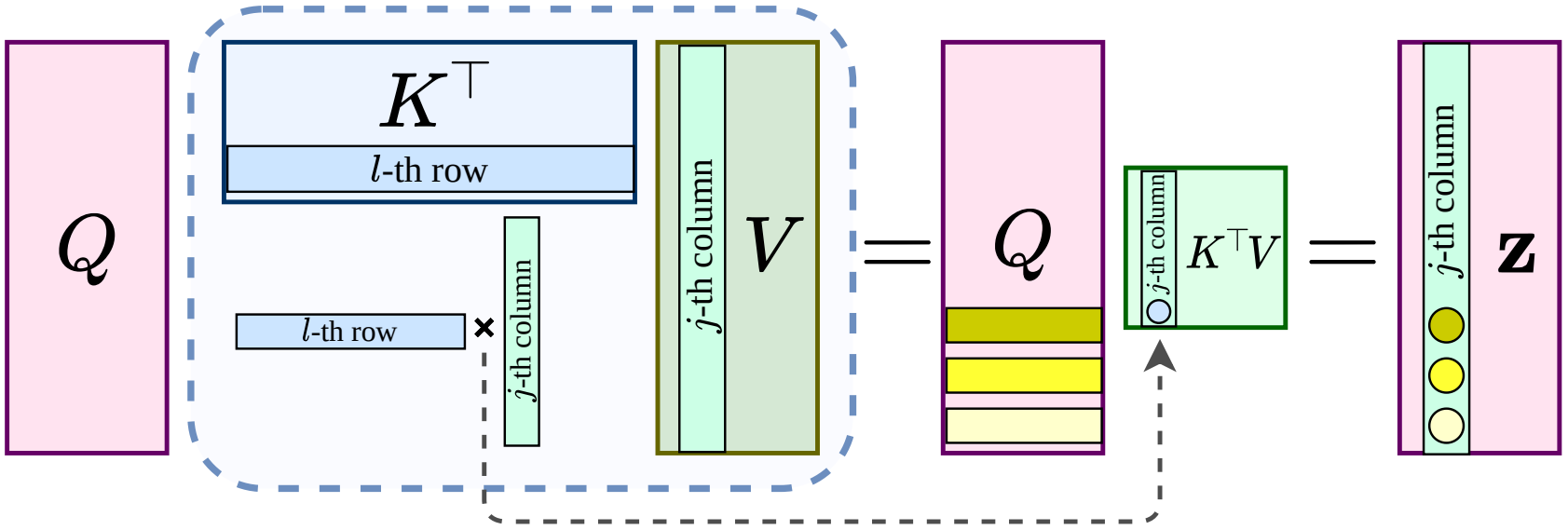
Column = A basis function in a Hilbert subspace.

$$\begin{matrix}(1,0)\\(0,1)\end{matrix} = \begin{matrix}\text{Machine}\\\text{Learning}\end{matrix} \subset \text{Col}\left\{\binom{1}{0}, \binom{0}{1}\right\}$$

- The columns of Query/Key/Value contain the learned basis functions spanning certain subspaces of different Hilbert spaces.

Linear attention: a learnable Petrov-Galerkin projection

- In linear attention: Q: values, K: query, V: keys.
- While it makes sense to ask the kernel to be positive (similarity between rows), it does not to ask the interaction between bases (columns) to be positive.



$$(\boldsymbol{z}^j)_i = z_j(x_i) = h\sum_{l=1}^{d}(\boldsymbol{k}^l \cdot \boldsymbol{v}^j)(\boldsymbol{q}^l)_i \approx \sum_{l=1}^{d}\left(\int_\Omega k_l(\xi)v_j(\xi)\mathrm{d}\xi\right) q_l(x_i).$$

First result on the approximation capacity of $Q(K^\top V)$

Theorem (Approximation capacity of a single layer of Galerkin attention)

Suppose there exists a continuous key-to-value map that is bounded below on the discrete approximation space, i.e., the functional norm of $v \mapsto b(q, v)$ is bounded below (discrete Ladyzhenskaya–Babuška–Brezzi inf-sup condition) for any q

$$\min_{\theta} \|f - g_\theta(\mathrm{y})\| \leq c^{-1} \underbrace{\min_{q \in \mathbb{Q}_h} \max_{v \in \mathbb{V}_h} \frac{|b(\Pi f - q, v)|}{\|v\|}}_{\text{(Error of the Petrov-Galerkin projection)}} + \underbrace{\|f - \Pi f\|}_{\text{(How good the current approximation space is)}}.$$

- Intepretation: for an incoming "query" (a function in some Hilbert space), to deliver the best approximator in "value" (trial function space), the "key" space (test function space) has to be big enough so that for every value there is a key to unlock it.

Rethinking Burgers' benchmark

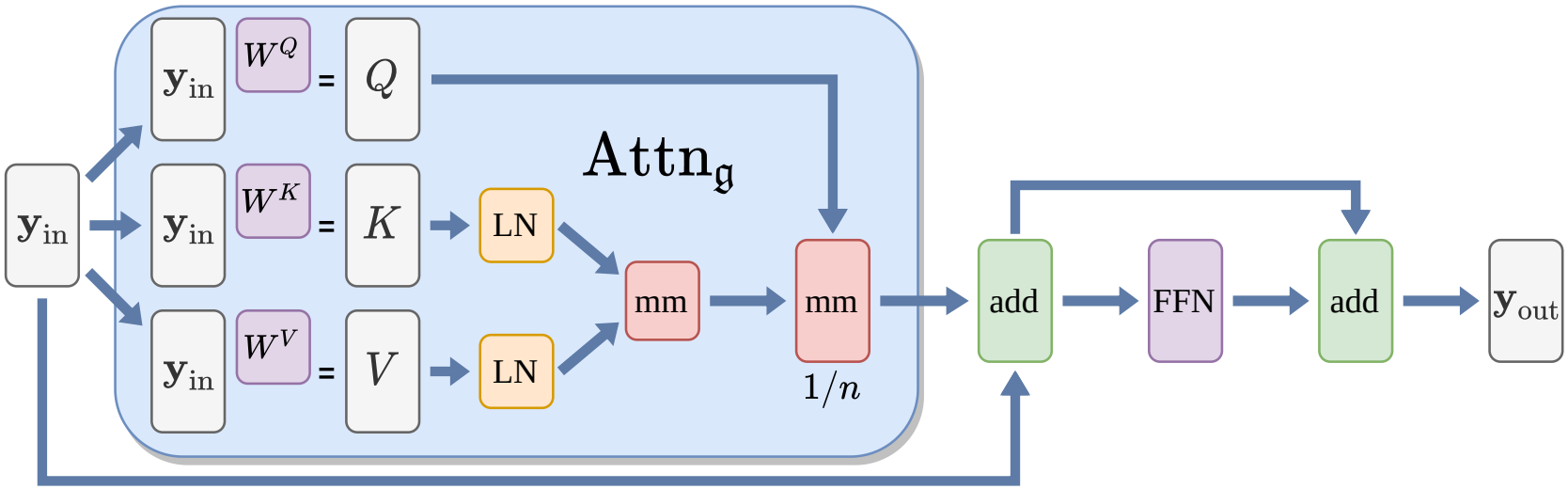
- Fourier Neural Operator (FNO) + Euclidean coordinate positional encodings.
- Efficient Transformer[4]: first Transformer encoder without softmax, decoder is 2 layers of spectral conv smoother.

	$n = 2048$ (eval)	GFLOP/backprop	$n = 8192$ (zero-shot eval)	# params
FNO1d original	1.46×10^{-2}	361.94 ± 1.65	1.26×10^{-2}	570k
FNO1d re-implement	4.37×10^{-3}	369.13 ± 1.81	4.18×10^{-3}	549k
Vanilla Transformer	1.41×10^{-1}	$\approx 12T$	1.47×10^{-1}	22m
Efficient Transformer	2.08×10^{-1}	405 ± 2	2.13×10^{-1}	523k

 It is even worse! 

[4] Z. Shen et al. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3531–3539, 2021.

Transformer encoder based on the Galerkin attention



- Linear complexity: a Galerkin-projection like layer normalization scheme with a mesh-weighted normalization instead of the softmax.
- Positional encoding concatenated in every encoder layer and every head, unlike only once in the classic Transformer. The importance of this trick is discovered concurrently in AlphaFold 2.[5]

[5] J. Jumper et al. *Nature*, 596(7873):583–589, 2021.

Re^{2}thinking Burgers' benchmark

- Fourier Neural Operator (FNO): 4 sc, ReLU changed to SiLU.
- Efficient Transformer: first Transformer with an encoder without softmax, decoder is 2 layers of spectral conv smoother.
- Galerkin Transformer (GT): 4 Galerkin attention encoder with the new layer normalization scheme + 2 sc decoder (same parameter quota with FNO).

	$n = 2048$ (eval)	GFLOP/backprop	$n = 8192$ (zero-shot eval)	# params
FNO1d reimplement	4.37×10^{-3}	369.13 ± 1.81	4.18×10^{-3}	549k
Vanilla Transformer	1.41×10^{-1}	$\approx 12T$	1.47×10^{-1}	22m
Efficient Transformer	2.08×10^{-1}	405 ± 2	2.13×10^{-1}	523k
Galerkin Transformer	1.012×10^{-2}	411.78 ± 1.83	1.09×10^{-2}	530k

Ten times better than Transformer without softmax applied directly.

Can we improve the accuracy even more?

- Rethinking the Galerkin-type attention operator

$$\boldsymbol{z} \leftarrow \boldsymbol{y} + \mathrm{Attn}(\boldsymbol{y}), \quad \boldsymbol{y} \leftarrow \boldsymbol{z} + g(\boldsymbol{z}).$$

  as a neural ODE like integration scheme due to the skip-connection:

$$\mathrm{y}_{k+1/2} \leftarrow \mathrm{y}_k + \Delta t\ \mathrm{Attn}\left(\mathrm{y}_k\right) \quad \text{and} \quad \mathrm{y}_{k+1} \leftarrow \mathrm{y}_{k+1/2} + \Delta t\, g\left(\mathrm{y}_{k+1/2}; \mathrm{x}\right)$$

- A new projection weights initialization scheme inspired by the interpretation above and the proof of the approximation theorem:

$$W^{\diamond}_{\mathrm{init}} \leftarrow \eta U + \delta I, \quad \text{for } \diamond \in \{Q, K, V\},$$

  where δ and η are small numbers. Similar tricks discovered concurrently in Csordás et al 2021 (accepted at EMNLP 2021)[6].

[6] R. Csordás, K. Irie, and J. Schmidhuber. In *Proc. Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, Punta Cana, Dominican Republic, 2021.

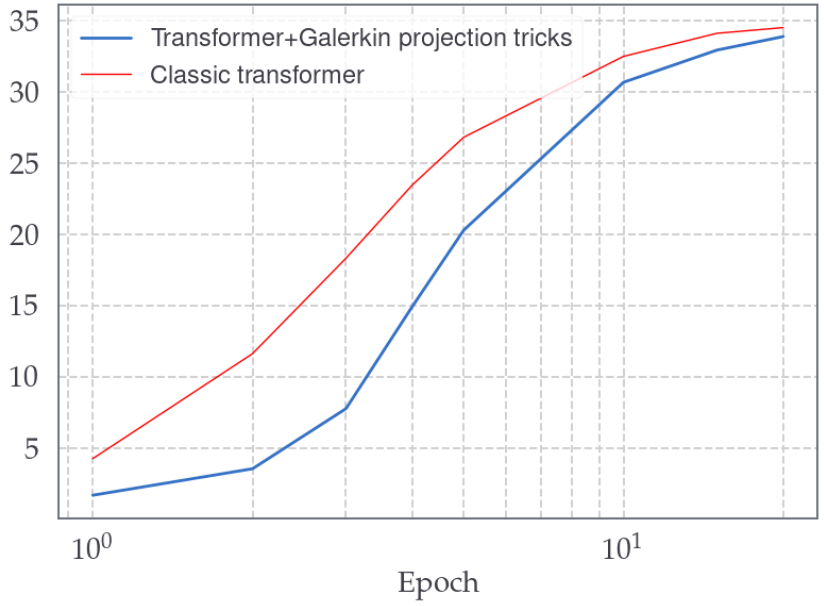
Re^{3}thinking Burgers' benchmark

- Fourier Neural Operator (FNO): 4 sc, ReLU changed to SiLU.
- Galerkin Transformer (GT): 4 Galerkin attention encoder with new tricks + 2 sc decoder.
- Softmax Transformer (ST): 4 softmax attention encoder with new tricks + 2 sc decoder.
- Random Feature Attention (RFA): H. Peng et al. In *International Conference on Learning Representations*, 2021 4 RFA + 2 sc.
- Performer (FAVOR$^+$): K. M. Choromanski et al. In *International Conference on Learning Representations (ICLR)*, 2021 4 FAVOR$^+$ + 2 sc.
- Multi-Wavelet Operator (MWO): G. Gupta, X. Xiao, and P. Bogdan. In *Thirty-Fifth Conference on Neural Information Processing Systems (NeurIPS 2021)*, 2021. eprint: 2109.13459 (cs.LG).
- CNN+Expressive Diagonalization (XD): N. C. Roberts et al. In *Thirty-Fifth Conference on Neural Information Processing Systems (NeurIPS 2021)*, 2021. eprint: 2103.15798 (cs.LG)

	$n = 2048$ (eval)	GFLOP/backprop	$n = 8192$ (zero-shot eval)	# params
FNO1d re-implement	4.37×10^{-3}	369.13 ± 1.81	4.18×10^{-3}	549k
ST with all tricks	2.31×10^{-3}	1876.36 ± 2.01	2.07×10^{-3}	523k
RFA	1.72×10^{-2}	480.11 ± 1.74	1.91×10^{-2}	523k
FAVOR$^+$	1.58×10^{-3}	510.90 ± 25.11	1.67×10^{-3}	523k
GT with some tricks	2.45×10^{-3}	411.78 ± 1.83	2.49×10^{-3}	530k
GT with all tricks	1.09×10^{-3}	411.78 ± 1.83	1.11×10^{-3}	530k
GT 500 epochs	7.79×10^{-4}	411.78 ± 1.83	7.90×10^{-4}	530k
FNO1d 500 epochs	2.47×10^{-3}	369.13 ± 1.81	2.40×10^{-3}	549k
MWO 500 epochs	1.86×10^{-3}	?	?	501k
XD 500 epochs	9.9×10^{-3}	?	?	?

How about traditional NLP tasks?

- Dataset: IWSLT14 De-En, translating German to English, train samples: $160,239$, valid samples $7,283$, test samples $6,750$.
- Training loss: standard word (token)-wise cross entropy.
- Evaluation metric: BLEU (bilingual evaluation understudy) score.
- Best BLEU on test set after 20 epochs, classic Transformer: 33.56; Galerkin projection inspired tweaks: 33.94.



- Encoder changed to Galerkin projection type layer normalization, softmax unchanged.
- New initialization tweaks.
- Decoder unchanged.

Acknowledgments

- We would like to thank the reviewers and the area chair.
- Dr. Long Chen (UC Irvine) for sharing a blog post which directly results the conceiving of this paper, sharing lecture notes about graph theory, open-sourcing the elegant vectorization of finite element methods in that inspires us to re-interpret the attention mechanism.
- Dr. Ari Stern (WUSTL) for the help during COVID-19.
- Dr. Ruchi Guo (UC Irvine) and Dr. Yuanzhe Xi (Emory) for the invaluable feedback on the numerical experiements.
- Mr. Zongyi Li (Caltech) for sharing an early dev code to verify our re-implementation in the new PyTorch `fft` interface and comments on Burgers' equation's viscosity.
- Joel Schlosser (facebook) for reviewing our code contributions to be incorporated into PyTorch's `Transformer` module.
- The PyTorch community on GitHub for selflessly code sharing.