BROWN

Temporal Learning Capacity of Transformers in Non-Markovian Dynamical Systems

Benjamin Shih
Zhongqiang Zhang, George Em Karniadakis, Khemraj Shukla

April 26, 2024

CRUNCH Group | Brown University

Table of contents

Introduction and Background

Tempered Fractional Derivative

Definition (σ-Tempered fractional derivative)

The σ-tempered fractional Caputo derivative of fractional order α is defined as

$$ {}^{\mathrm{C}}_{a}\mathcal{D}_{t}^{\alpha,\sigma}\, u(t) = \frac{e^{-\sigma t}}{\Gamma(n-\alpha)} \int_{a}^{t} (t-s)^{n-\alpha-1} \frac{d^n}{ds^n} (e^{\sigma s} u(s))\, ds $$

Clearly, if $\sigma = 0$, then this reduces to the standard Caputo fractional derivative with fractional order α.

Leaky Integrate-and-Fire Model

Integer order

$$\tau \frac{dV}{dt} = -(V - V_{\text{rest}}) + RI(t)$$

R cell constant

$I(t)$ spiking forcing term

τ membrane time constant

V_{rest} resting membrane potential

V membrane potential

σ-Tempered Fractional

$$\tau \, {}_{a}^{\text{C}}\mathcal{D}_{t}^{\alpha,\sigma}[V](t) = -(V - V_{\text{rest}}) + RI(t)$$

Major advantage:

- Non-Markovian (memory!): captures effect of residual current

Operator Learning

- Input: $f(t)$, output: $u(t)$. We denote the operator mapping f to u as

$$\mathcal{G}^* : \mathcal{F} \ni f \mapsto u \in \mathcal{U}$$

- On computers: use discrete samples of continuous function at "sensors."

$$f_O = \{f(t_1), \ldots, f(t_n)\}$$

- Operator learning: construction of a network

$$\mathcal{G}_\theta : \mathbb{R}^n \ni f_O \mapsto \mathcal{G}_\theta[f_O](\cdot) \in \mathcal{U}$$

  such that $\mathcal{G}_\theta$ approximates $\mathcal{G}^*$. We want

$$\arg\min_{\theta \in \Theta} \|\mathcal{G}_\theta[f](\cdot) - \mathcal{G}^*[f](\cdot)\|$$

Attention Mechanism

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d}}\right) V$$

$$\text{softmax}(\boldsymbol{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}}$$

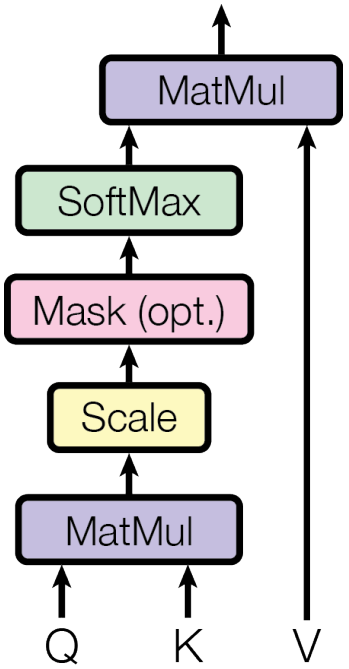

Figure 1: Scaled Dot-Product Attention[1]

[1] Ashish Vaswani et al. "Attention is All you Need". In: Advances in Neural Information Processing Systems. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017.

Methodology

Attention

Standard: $$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d}}\right) V$$

Fourier type: $$(z_i)_j = \frac{1}{n}\sum_{i=1}^{n}(q_i \cdot k_s)(v^j)_s \approx \int_\Omega \kappa(x_i, \xi)\, v_j(\xi)\, d\xi$$

Network Architecture & Loss

$$\mathcal{L}(\theta) = \frac{1}{m} \sum_{k=1}^{m} \frac{\|\mathcal{G}_\theta[f_k; \alpha, \sigma] - \mathcal{G}^*[f_k]\|_{\ell^2}}{\|\mathcal{G}^*[f_k]\|_{\ell^2}}$$

where

$$\|f\|_{\ell^2} = \left(\frac{1}{n} \sum_{i=1}^{n} |f(t_i)|^2 \right)^{\frac{1}{2}}$$

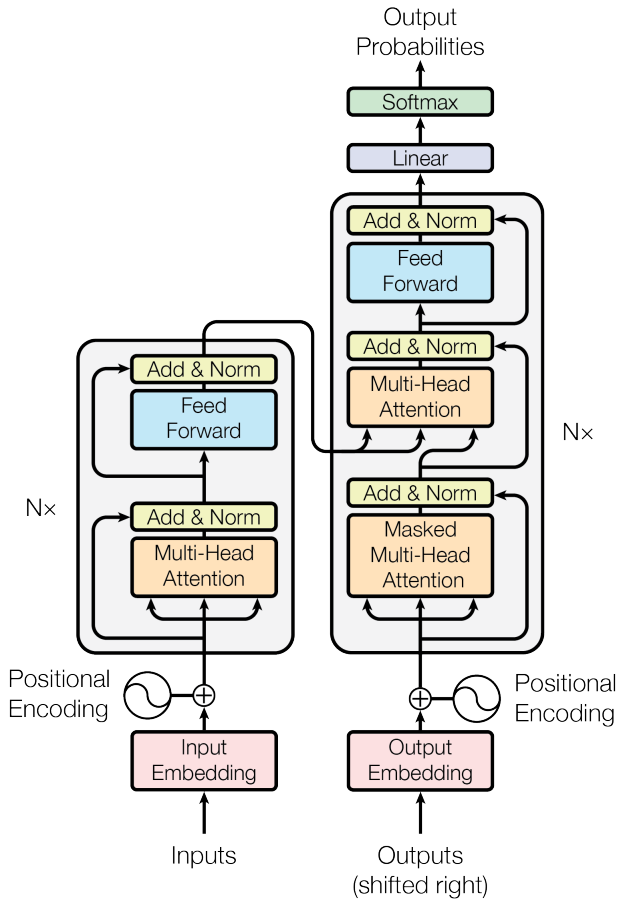

Figure 2: Transformer Architecture[2]

[2] Vaswani et al., "Attention is All you Need".

Experimental Results

Smooth TFIVP

$$\begin{cases} {}_{a}^{\mathrm{C}}\mathcal{D}_{t}^{\alpha,\sigma}[u](t) = \mu u(t) + f(t) \\ u(0) = u_0 \end{cases}$$

where $f \sim \mathcal{GP}$[3]. Operator:

$$\mathcal{S} : (f(t), \alpha, \sigma) \mapsto u(t; \alpha, \sigma)$$

- Case 1: $\mu = 0$, vary α, f, no tempering
- Case 2: $\mu = -1$, vary α, f, no tempering
- Case 3: $\mu = -1$, vary α, σ, f

[3] $\mathcal{GP}$ specifics: $\mathcal{N}(\mu, \sigma^2(-\Delta + \tau^2 I)^{\gamma})$ with $\mu = 1, \gamma = 2.5, \tau = 7, \sigma = 7$,

Smooth TFIVP Case 1: Visually

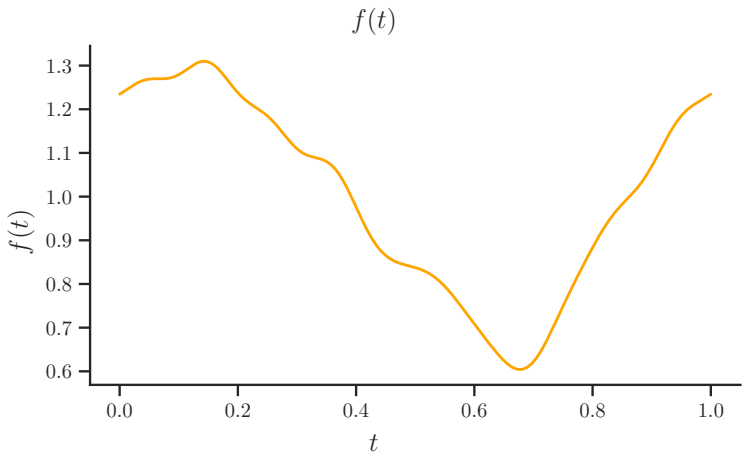

Figure 3: Input: Smooth forcing term

$\mathcal{S}$

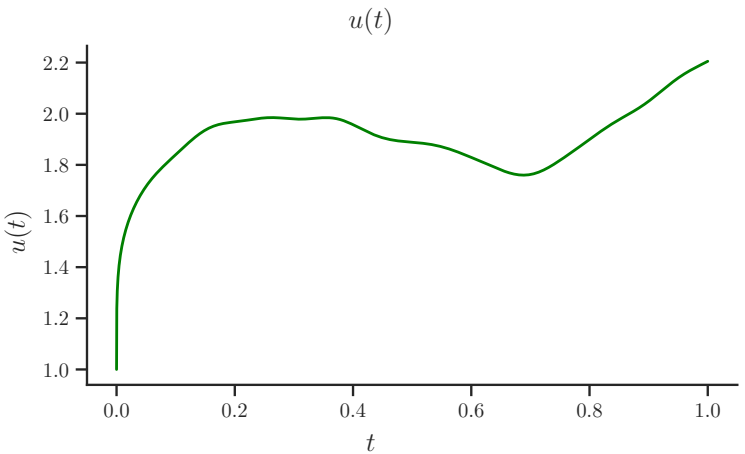

Figure 4: Output: Solution

TFIVP Results

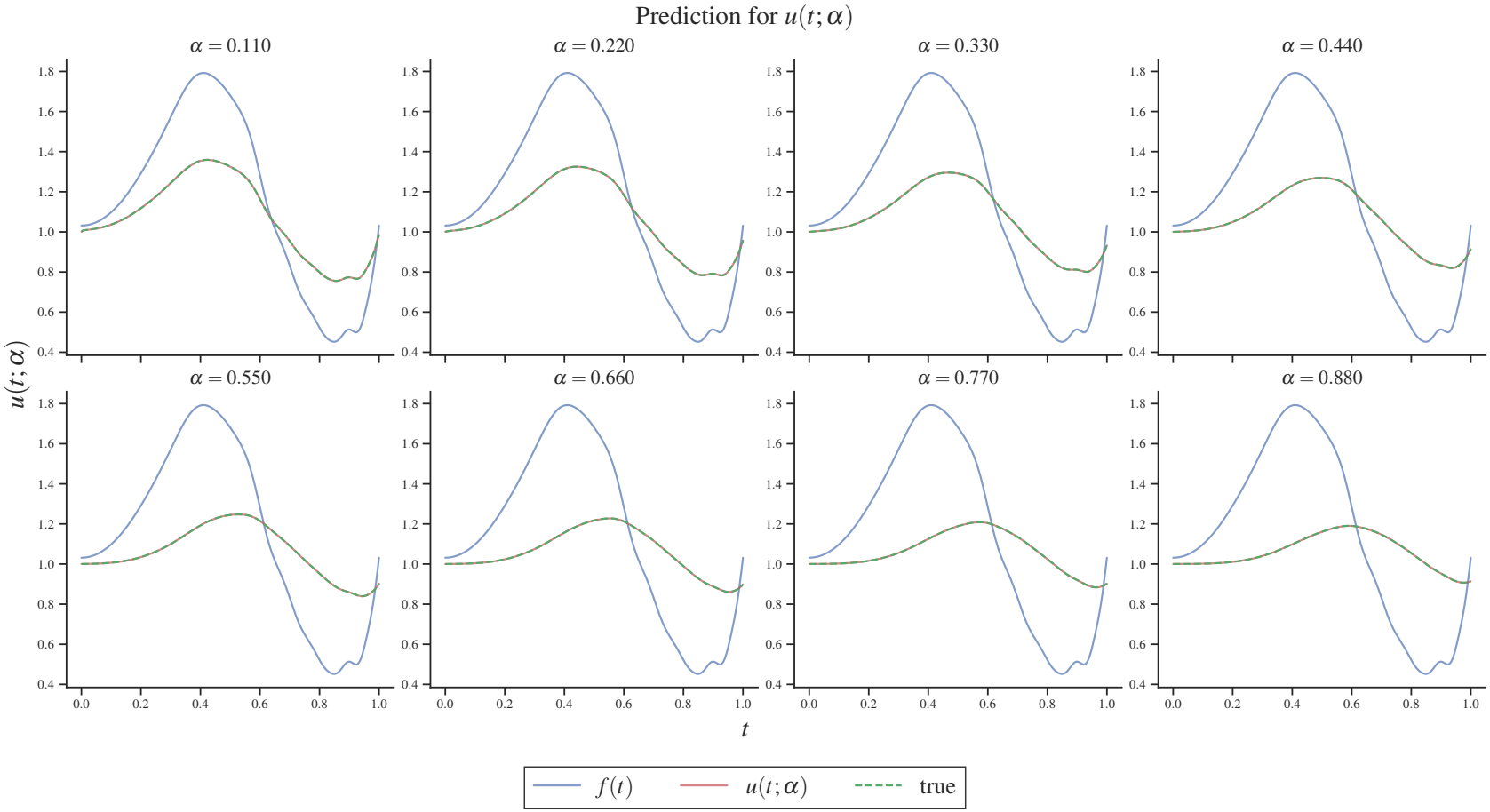

Figure 5: Results for TFIVP Case 1 (varying α and f)

TF-LIF

$$\tau \, {}_{a}^{\mathrm{C}}\mathcal{D}_{t}^{\alpha,\sigma}[V](t) = -(V - V_{\mathrm{rest}}) + RI(t)$$

Operator:

$$\mathcal{S} : (I(t), \alpha, \sigma) \mapsto V(t; \alpha, \sigma')$$

- Case 1: vary α, no tempering
- Case 2: vary α, $I(t)$ (both location and intensity)
- Case 3: vary α, σ

TF-LIF Case 1: Visually

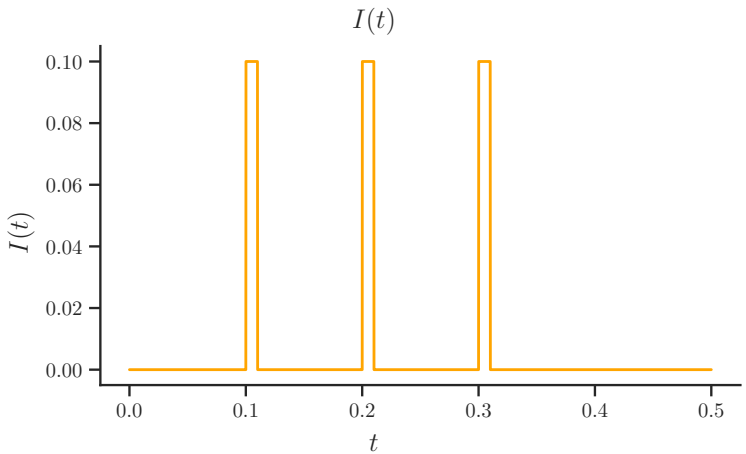

Figure 6: Input: Spiking current

$\mathcal{S}$

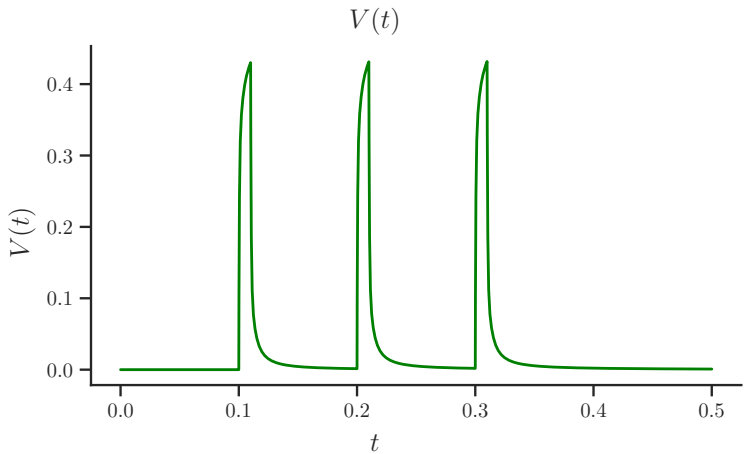

Figure 7: Output: Membrane Potential

TF-LIF Results

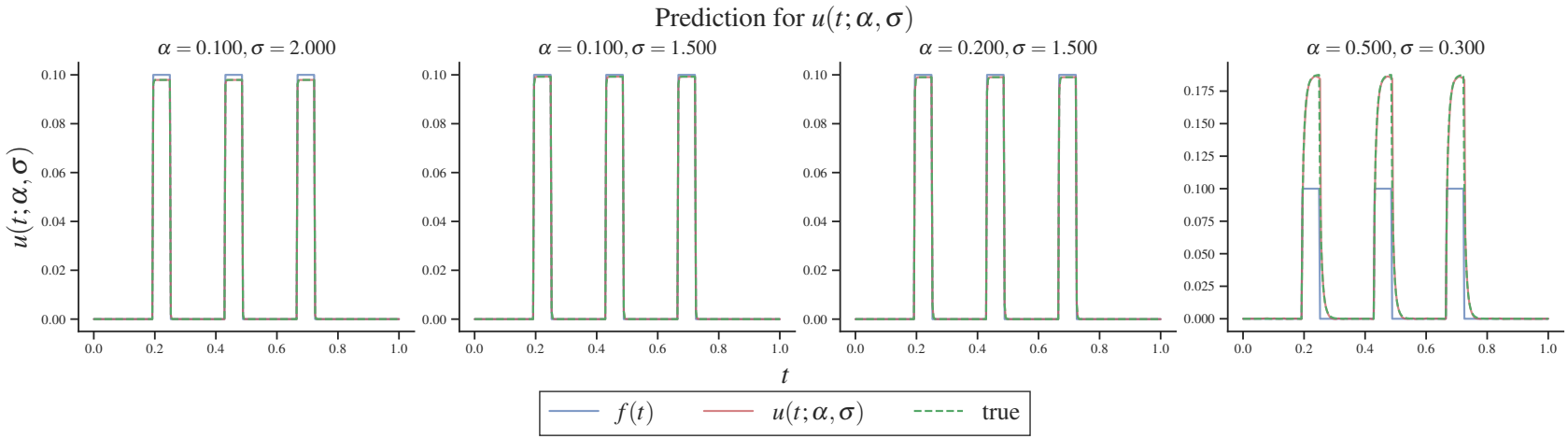

Figure 8: Results for LIF Case 3 (varying α and σ)

TF Inhomogenous Diffusion

$$\begin{cases} {}_{a}^{\mathrm{C}}\mathcal{D}_{t}^{\alpha,\sigma}[u](x,t) = \mu \frac{\partial^2 u}{\partial x^2}(x,t) + f(x,t) \\ u(x,0) = u_0 \\ u(0,t) = u(1,t) = 0 \end{cases}$$

Operator:

$$\mathcal{S} : (f(x,t), \alpha, \sigma) \mapsto u(x,t;\alpha,\sigma)$$

- vary α, f

TF Inhomogenous Diffusion: Visually

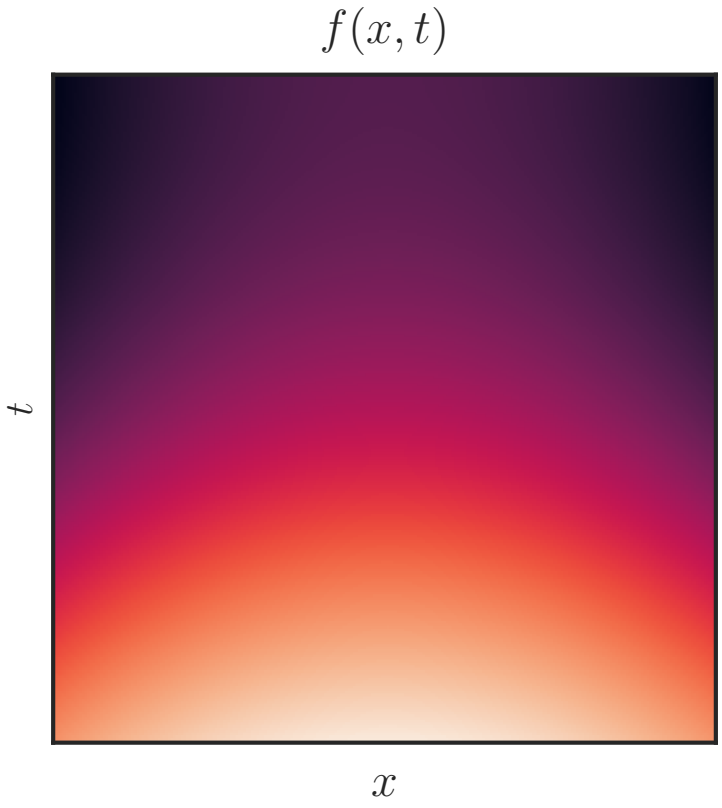

Figure 9: Input: Source term

$\xrightarrow{S}$

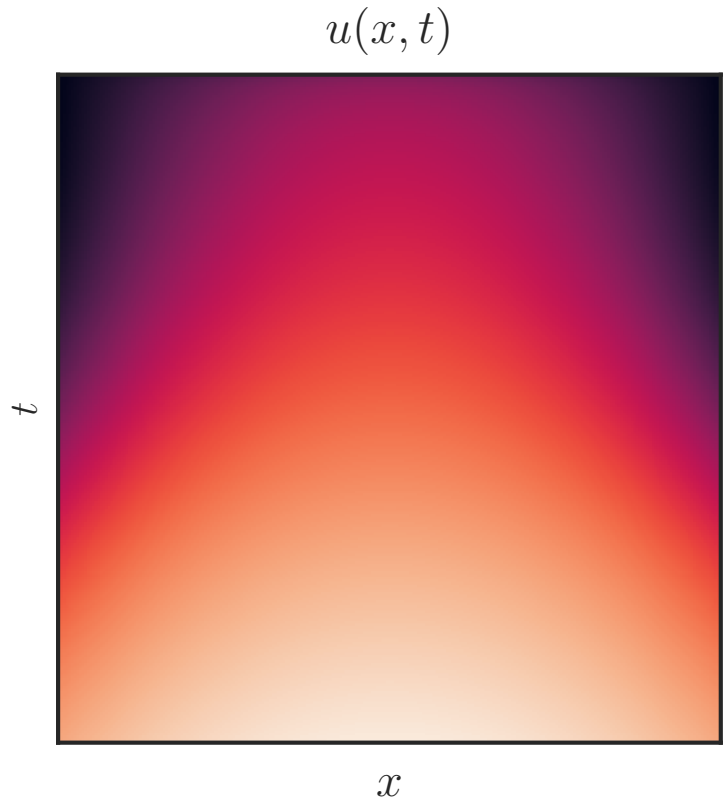

Figure 10: Output: Diffusive behavior

TF-Diffusion Results

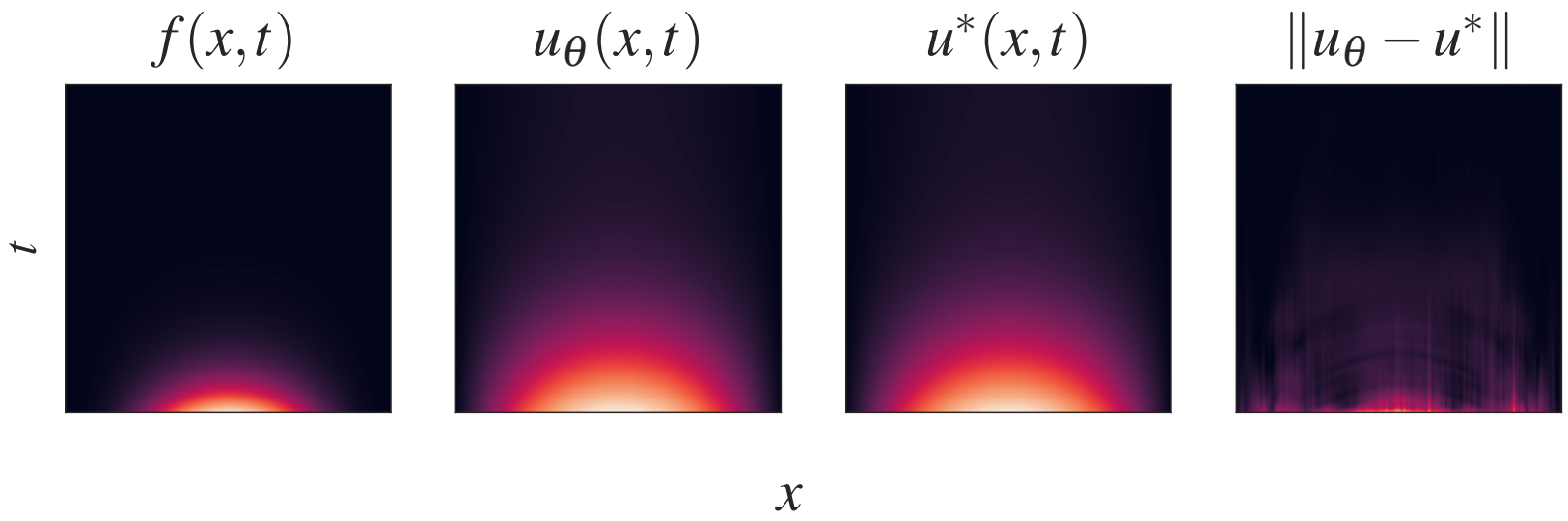

Figure 11: Results for Diffusion Case 1 (varying α, f)

Sample Attention Matrix Visualization: TF-LIF Case 1

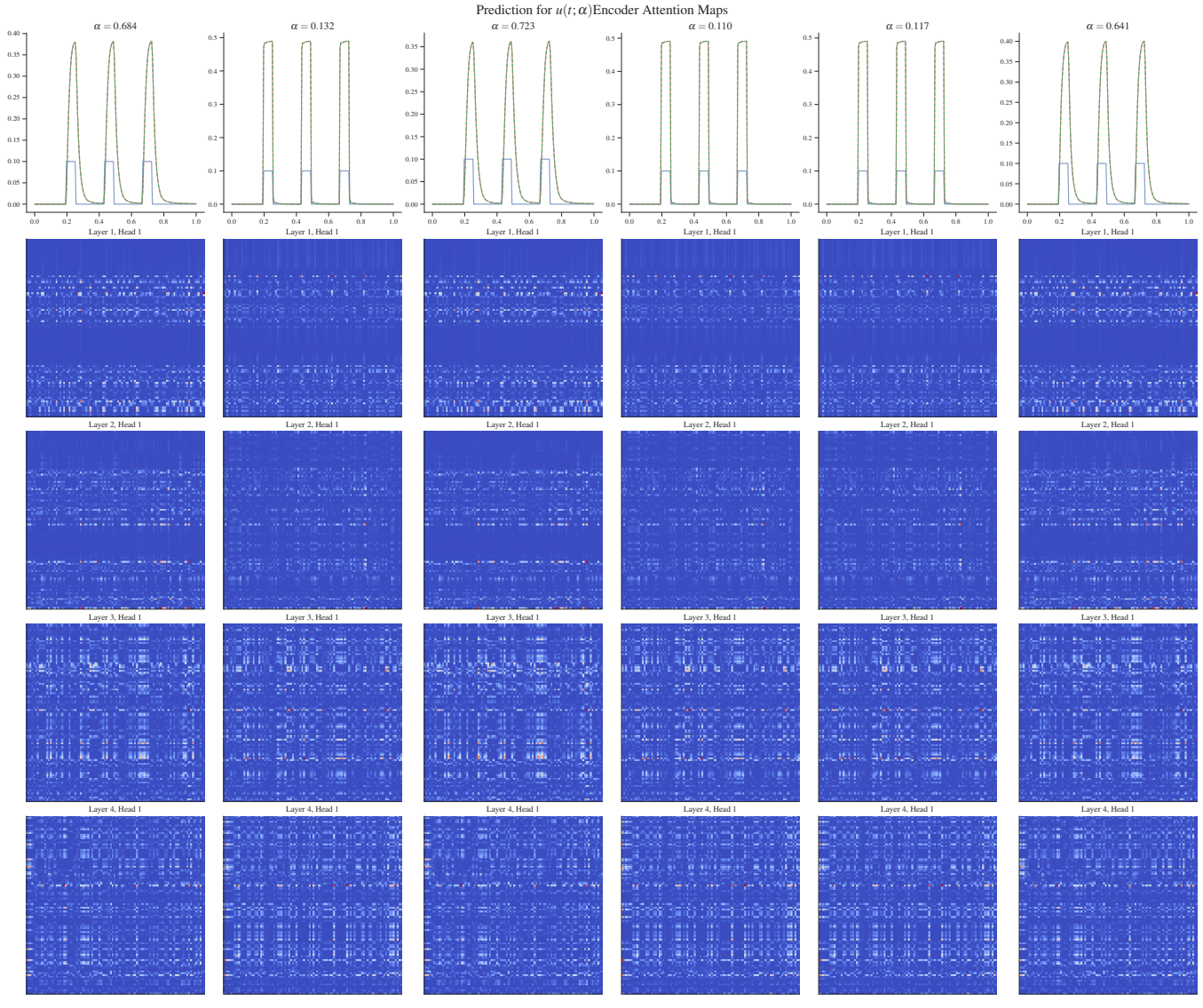

Figure 12: Transformer Encoder Attention: TF-LIF Case 1

Numerical Results

Table 1: Experiment details and relative L_2 error for each case of the tempered fractional smooth, LIF, and diffusion problems for the two different models tested.

Experiment settings									Results
Model	Experiment	Case	α	σ	n	Learning Rate	Batch Size	Iterations	$\mathcal{L}(\theta)$
Adam/1cycle	Smooth	1	$[0.11, 0.99]$	—	201	1.0×10^{-3}	32	1.0×10^{6}	4.7×10^{-4}
		2		—					9.2×10^{-4}
		3		$[0.11, 2]$					6.5×10^{-4}
	LIF	1	$[0.11, 0.99]$	—	204	1.0×10^{-4}	32	2.0×10^{6}	3.9×10^{-4}
		2		—	200				1.3×10^{-2}
		3		$[0.11, 0.99]$	314				1.3×10^{-3}
	Diffusion	1	$[0.11, 0.99]$	—	257^2	1.0×10^{-4}	32	5.0×10^{4}	6.1×10^{-3}
Lion/polynomial	Smooth	1	$[0.11, 0.99]$	—	201	1.0×10^{-4}	2000	1.0×10^{6}	5.5×10^{-5}
		2		—					7.9×10^{-5}
		3		$[0.11, 0.99]$					1.1×10^{-2}
	LIF	1	$[0.11, 0.99]$	—	204	1.0×10^{-4}	1000	2.0×10^{5}	3.9×10^{-5}
		2		—	200				3.0×10^{-1}
		3		$[0.11, 2]$	314				3.4×10^{-2}
	Diffusion	1	$[0.11, 0.99]$	—	257^2	1.0×10^{-4}	32	5.0×10^{4}	8.6×10^{-2}

Sample Loss Trajectory Comparison (Smooth Case 2)

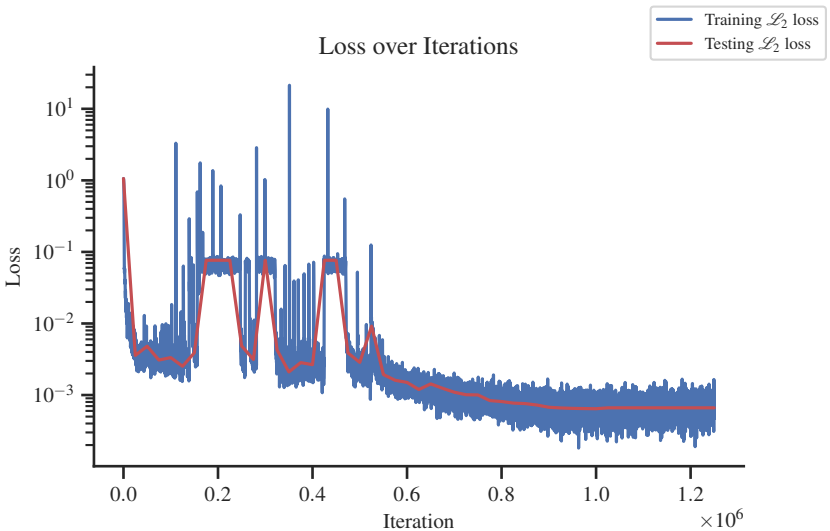

Figure 13: Adam/1cycle loss curve

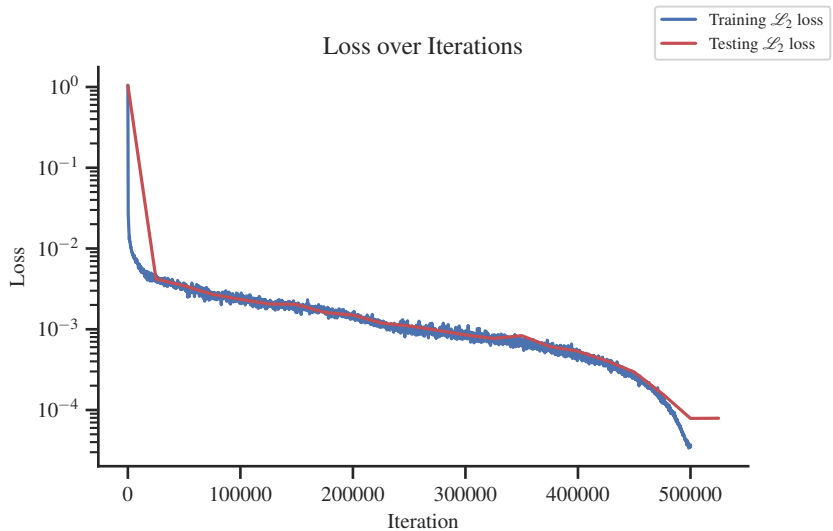

Figure 14: Lion/polynomial loss curve

Conclusion

Summary

- Novel application of the transformer architecture to the operator learning problem; specifically to tempered fractional operators
- Success on 3 different problems that vanilla DeepONet was unable to learn
- Visualization and interpretation of the attention in each of these models

Ongoing Work

- RiemannONet: success with massive jump discontinuities by decomposition in the trunk net:

$$\mathcal{L}^t(\Theta) = \left\| \sum_l T_l(\Theta)\mathcal{A}_l - U^* \right\|_{L_2}$$

$$\downarrow$$

$$Q^* R^* = \mathrm{qr}(T(\Theta))$$

- Comparison between different neural operators
- Ablation studies

Thank you!

References

Vaswani, Ashish et al. “Attention is All you Need”. In: Advances in Neural Information Processing Systems. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017.