DeepOSets: Non-Autoregressive In-Context Learning of Operators

Shao-Ting Chiu [1*] Junyuan Hong [2] Ulisses Braga-Neto [1]

[1]Department of Electrical & Computer Engineering, Texas A&M University, College Station, TX
[2]Institute for Foundations of Machine Learning, University of Texas, Austin, TX

Highlights

- **DeepSets Operator Networks (DeepOSets)** is the first *non-autoregressive* neural network architecture for in-context operator learning.
- **DeepOSets combines the powerful *DeepSets* and *DeepONet* architectures** to perform operator learning across different numbers of in-context examples and noise levels with a permutation-invariance inductive bias.
- We show that **DeepOSets can learn classical least-squares linear regression** much more efficiently than a comparable transformer-based approach.
- **DeepOSets is parameter efficient, fast to train, and robust to noise.** It is also **more accurate** than the transformer approach under noisy conditions.

Introduction

- **In-context learning (ICL)** [1] refers to the ability of a pre-trained machine learning model to learn from examples in a user prompt without further training.
- In the context of **operator learning**, ICL is the ability of a pre-trained neural operator to **learn a new operator** from examples in the prompt.
- **Auto-regressive models**, typically based on the transformer architecture, abound in ICL due to its success in natural language processing (NLP).
- However, **auto-regressive transformers** are memory-intensive and computationally-intensive both at training and inference time.
- Furthermore, in operator learning, the examples in the prompt do not have an auto-regressive structure as in NLP, but instead are **permutation invariant**.
- Thus, we propose **DeepOSets**, an efficient non-autoregressive alternative for ICL of operators, which combines **DeepONets** [2] for operator learning with **DeepSets** [3] to allow prompts of different sizes with a permutation-invariance inductive bias.

Methods

Prompt for In-Context Learning

$$\overbrace{\underbrace{\underbrace{\mathbf{x}_1, f(\mathbf{x}_1)}_{\text{Example}_1}, \underbrace{\mathbf{x}_2, f(\mathbf{x}_2)}_{\text{Example}_2}, \dots, \underbrace{\mathbf{x}_n, f(\mathbf{x}_n)}_{\text{Example}_n}}_{\text{In-Context Examples } (D_n)}, \mathbf{x}_{query}}^{\text{Prompt}} \xrightarrow{\text{Prediction}} f(\mathbf{x}_{query}). \quad (1)$$

DeepOSets Architecture

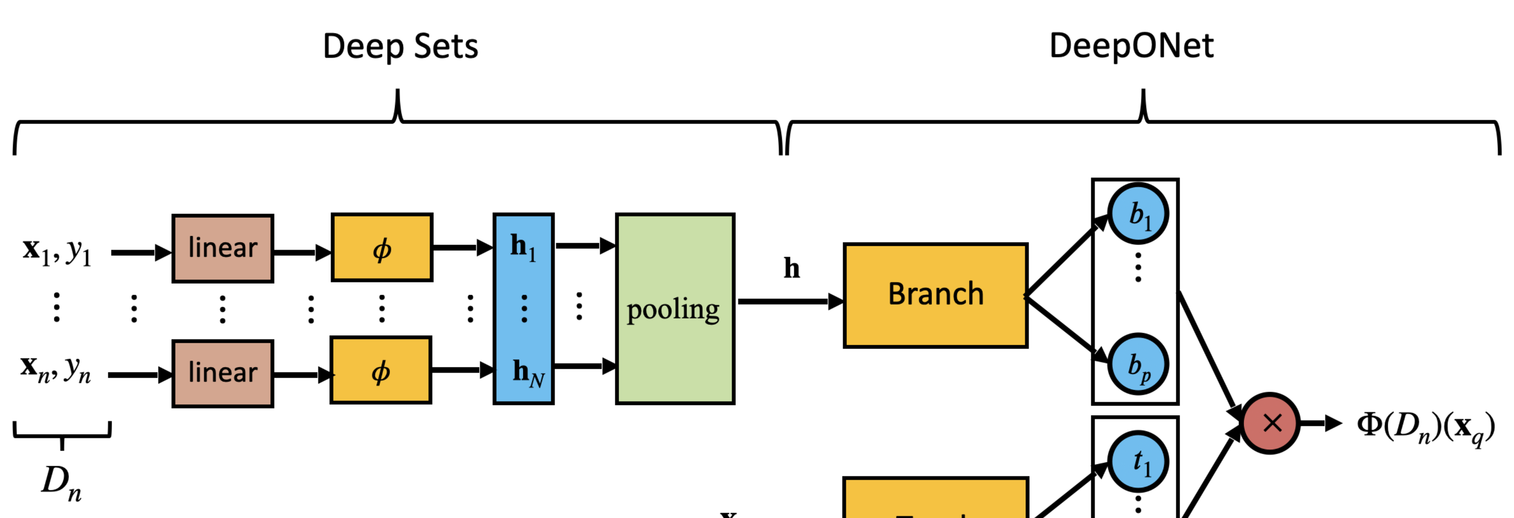


Figure 1. DeepOSets architecture for in-context learning of supervised learning operators.

Results

DeepOSets meta-learns noisy linear regression after training with noiseless data.

Noiseless training data from function set $\{f | f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}, \mathbf{w} \in \mathbb{R}^d\}$ where $\mathbf{w} \sim \mathcal{N}(0, \mathbf{I}_d)$ [4]. Linear regression on noisy prompt $\tilde{f}(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + \epsilon$, $\epsilon \sim \mathcal{N}(0, \sigma^2)$.

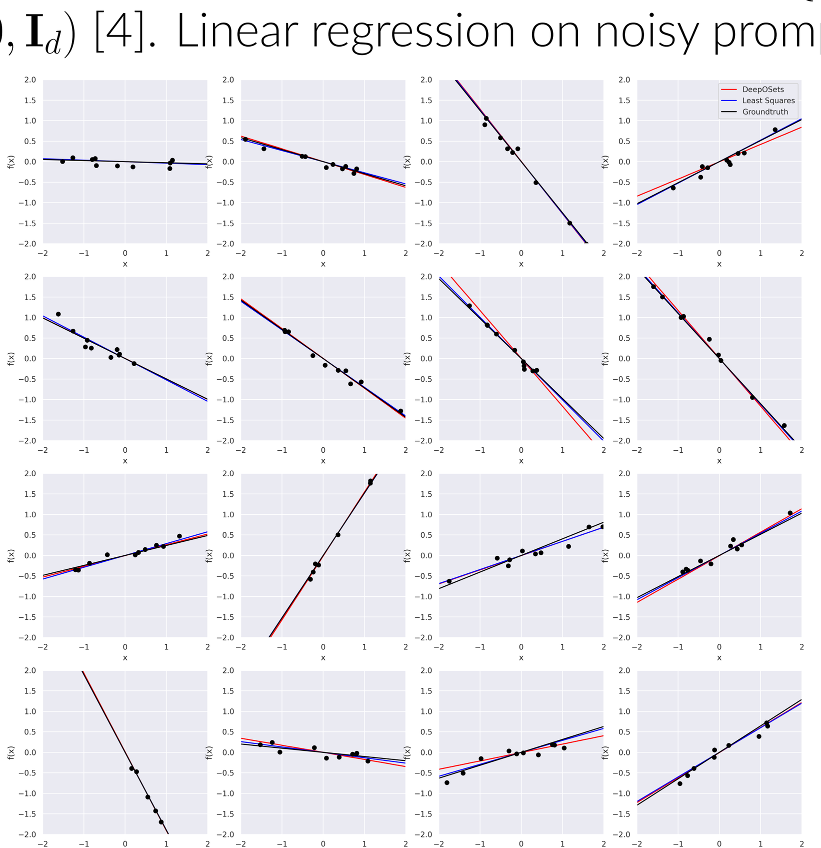


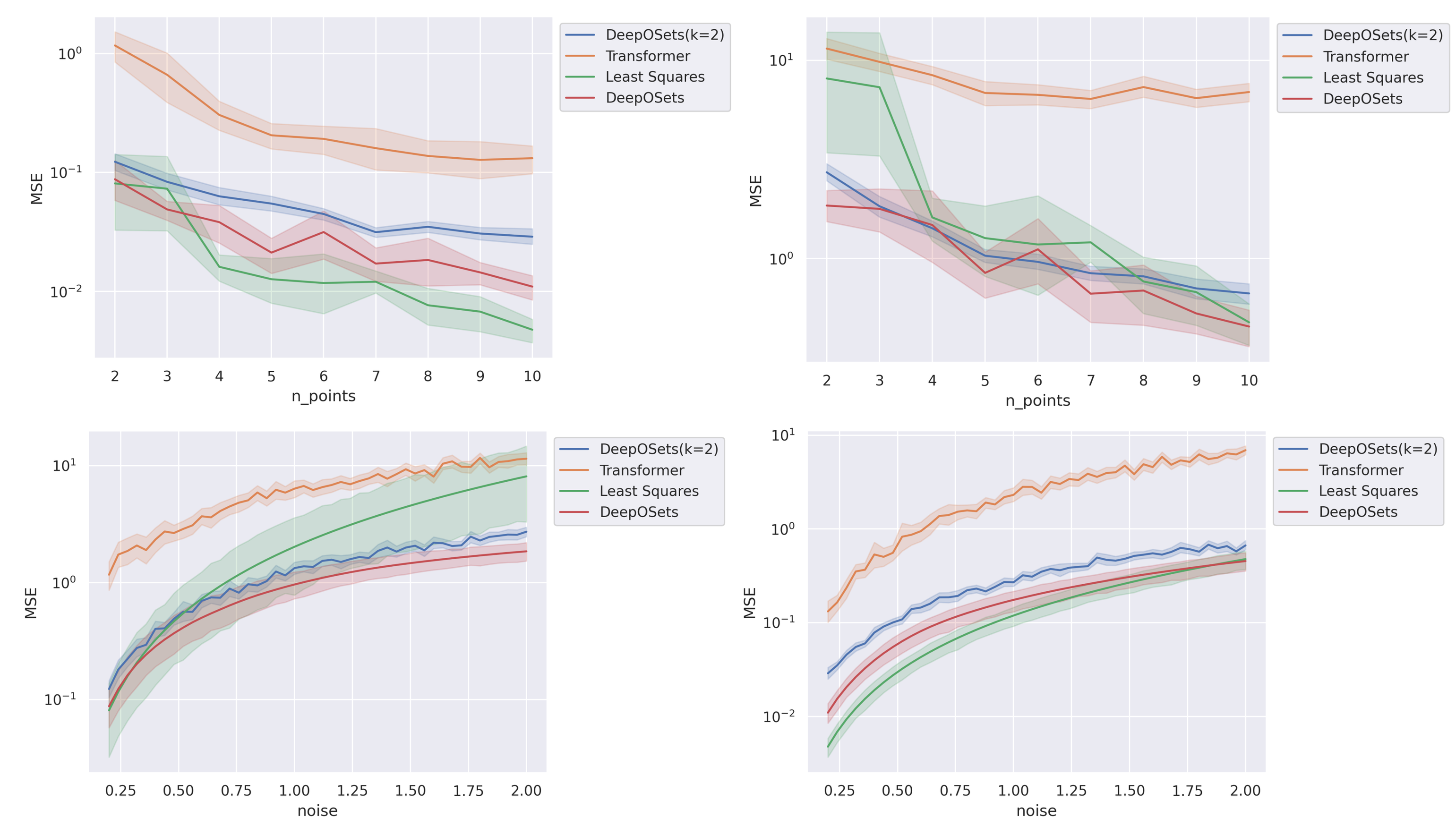
Figure 2. In-Context learn 1D linear regression from noisy examples ($\sigma^2 = 0.1$).

- The black dots (•) represent 10 noisy in-context examples.
- Blue line (—) represents ordinary least squares regression
- Red line (—) denotes DeepOSets prediction based on noisy examples.

DeepOSets is more accurate than a Tranformer [4] in low dimension or high noise.

Janossy pooling is used to consider permutations of k-tuples over the N examples. The case $k = 1$ is the baseline DeepOSets.

(a) Low noise ($\sigma^2 = 0.2$) (b) High noise ($\sigma^2 = 2$)



(c) 2 in-context training examples. (d) 10 in-context training examples.

Figure 3. Linear regression results in $d = 1$ dimensions.

Regression Problems	Transformer[4]	DeepOSets	DeepOSets (k=2)
	1 dimension		
Linear Regression ($\sigma^2 = 0.0$)	6.744e-04	**1.548e-04**	2.710e-03
Linear Regression ($\sigma^2 = 0.04$)	6.454e-03	**1.450e-03**	4.759e-03
Linear Regression ($\sigma^2 = 0.2$)	1.917e-01	**3.158e-02**	4.479e-02
Linear Regression ($\sigma^2 = 2.0$)	6.692	1.113	**9.622e-01**
# of Parameters	22M	**71151**	84296
	5 dimensions		
Linear Regression ($\sigma^2 = 0.0$)	**7.509e-02**	1.642	5.221e-01
Linear Regression ($\sigma^2 = 0.04$)	**1.379e-01**	1.611	5.462e-01
Linear Regression ($\sigma^2 = 0.2$)	1.502	1.609	**6.716e-01**
Linear Regression ($\sigma^2 = 2.0$)	3.172e+01	**4.361**	6.837
# of Parameters	22M	**568561**	1942411

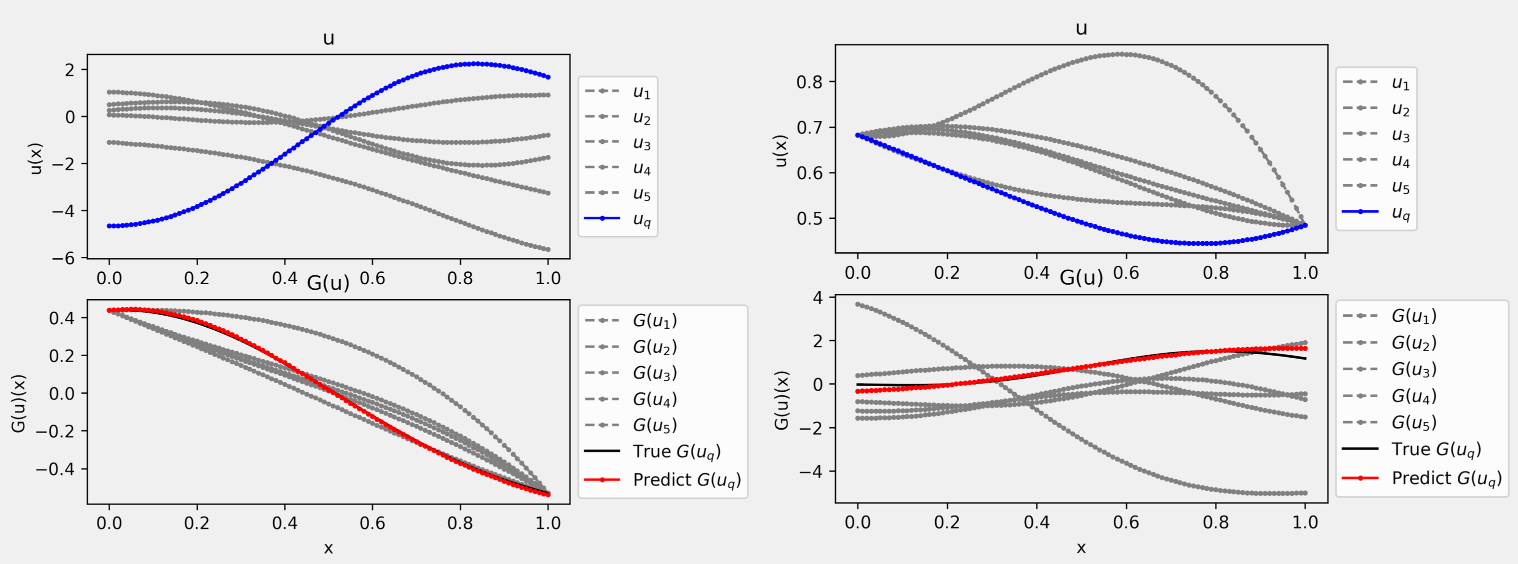
Table 1. Mean square error (MSE) and number of parameters benchmark with 30 test functions and 6 in-context examples.

DeepOSets is faster and lighter than a Transformer [4].

	Transformer[4]	DeepOSets	DeepOSets (k=2)
Parameters	22M	**72K**	84K
Training time	3 hours	9 min	**8min**
Complexity for first x_{query} on n examples	$O(n^2)$	$\boldsymbol{O(n)}$	$O(n^2)$
Complexity for second x_{query} on n examples	$O(n^2)$	$\boldsymbol{O(1)}$	$\boldsymbol{O(1)}$
Memory complexity for n examples	$O(n^2)$	$\boldsymbol{O(1)}$	$\boldsymbol{O(1)}$
Inference time per query ($n = 10$)	7.11 ms	**0.087 ms**	0.18ms
Test MSE	0.132	**1.12e-2**	2.89e-2

Table 2. Benchmark with $d = 1$, $n = 13$, $\sigma^2 = 0.2$ where n is the training sample size for DeepOSets.

DeepOSets for Multiple PDE Operator Learning



(a) Poisson Forward Problem $\frac{d^2 G(u)(x)}{dx^2} = u(x)$, given $u(0), u(1)$

(b) Poisson Inverse Problem $\frac{d^2 u(x)}{dx^2} = G(u)(x)$, given $u(0), u(1)$

Acknowledgements

Chiu and Braga-Neto were supported by NSF Award CCF-2225507.

References

[1] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, *et al.*, "Language models are few-shot learners," *arXiv preprint arXiv:2005.14165*, vol. 1, 2020.

[2] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis, "Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators," *Nature Machine Intelligence*, vol. 3, pp. 218–229, Mar. 2021.

[3] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov, and A. J. Smola, "Deep sets," *Advances in neural information processing systems*, vol. 30, 2017.

[4] S. Garg, D. Tsipras, P. Liang, and G. Valiant, "What Can Transformers Learn In-Context? A Case Study of Simple Function Classes," *Advances in Neural Information Processing Systems*, vol. 35, 2022.

*stchiu@tamu.edu