A Variance Reduction Framework for Global Optimization

Stanley Osher[1]

Joint work with Samy Wu Fung[2] and Yat Tin Chow[3]

UCLA[1], Colorado School of Mines[2], UC Riverside[3]

Global Optimization

Goal: Solve the *global* optimization problem

$$\min_{x \in \mathbb{R}^n} f(x) \tag{1}$$

- f is highly non-convex and (potentially) non-smooth
- global optimization arises in many standard tasks, e.g., PDE parameter estimation, deep learning, phase retrieval, etc.
- convergence generally only guaranteed for *local* optimization algorithms, e.g., steepest descent, SGD, Newton, ADMM, etc.

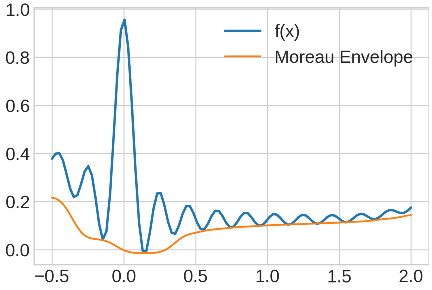
Prior Work: Finding Global Minima with Convergence Guarantees

Idea: minimize Moreau envelope $u(x, t)$ of f.

Theorem (Informal)

- *If f continuous and lower-bounded, and*
- *the set of global minimizers of f is compact*

then global minimizers of f are local minimizers of $u(x, T)$ for some T



[1] Global Solutions to Nonconvex Problems by Evolution of Hamilton-Jacobi PDEs. Comm App Math Comp Sci, Heaton, Wu Fung, Osher. 2022

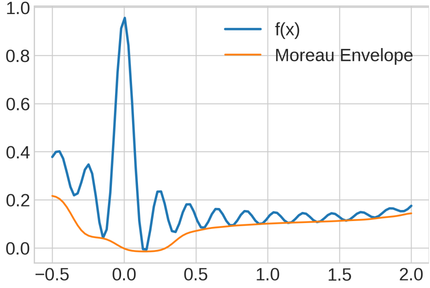
Prior Work: Finding Global Minima with Convergence Guarantees

Idea: minimize Moreau envelope $u(x,t)$ of f.

Theorem (Informal)

- *If f continuous and lower-bounded, and*
- *the set of global minimizers of f is compact*

then global minimizers of f are local minimizers of $u(x,T)$ for some T



Remark : Gradient descent on Moreau envelope u and converges to global minima of f, i.e., we want tractable way to compute ∇u[1]

[1] Global Solutions to Nonconvex Problems by Evolution of Hamilton-Jacobi PDEs. Comm App Math Comp Sci, Heaton, Wu Fung, Osher. 2022

Prior Work: Finding Global Minima with Convergence Guarantees

- Moreau envelope is solution to Hamilton-Jacobi Burgers' PDE $\implies$ difficult to compute

Prior Work: Finding Global Minima with Convergence Guarantees

- Moreau envelope is solution to Hamilton-Jacobi Burgers' PDE $\implies$ difficult to compute
- But one can leverage Hopf-Lax and Cole-Hopf transformations, we can approximate the gradient of Moreau envelope with the following formula

$$\nabla u(x,t) = \frac{1}{t} \cdot \frac{\mathbb{E}_{y \sim \mathcal{N}(x,\delta t)}\left[(x-y)\exp\left(-\delta^{-1} f(y)\right)\right]}{\mathbb{E}_{y \sim \mathcal{N}(x,\delta t)}\left[\exp\left(-\delta^{-1} f(y)\right)\right]}. \tag{2}$$

- Performing gradient descent on Moreau envelope is equivalent to proximal point algorithm on f

A highly non-convex 2D function (Griewank) example:

$$f(x) \triangleq 1 + \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right), \tag{3}$$

Relative Errors　　Objective Function Values　　Optimization Paths

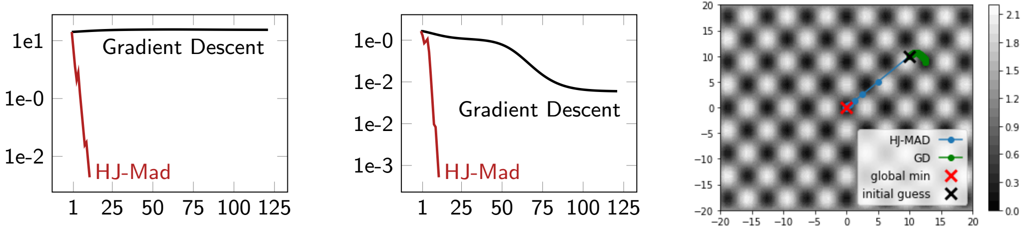


Figure 1: Gradient descent on Moreau Envelope converges to a tolerance of 5×10^{-2} of the global minimum, while traditional GD converges to local minima.

Nonconvex Benchmark Functions

	HJ-MAD	Pure Rand. Search	Diff. Evolution	Basin Hopping	Annealing
Griewank	167	460K	N	N	451.4K
Drop-Wave	9111	52.5K	1152	N	485.8K
Alpine N.1	635	755.6K	N	N	N
Ackley	498	243.2K	3003	476(116)	3.7M
Levy	5433	N	N	N	N
Rastrigin	500	660.2K	2223	48(12)	590.2K

Table 1: Comparison of global optimization algorithms. Rows represent benchmark functions and columns represent algorithms. The number in each box gives function (and gradient in parenthesis) evaluations used. An "N" indicates the method did not converge.

Variance Reduction for High-Dimensional Setting

- The gradient of envelope (or proximal of f) formula found success for moderately-dimensional problems (dim < 10) but struggles for higher dimensions due to sample requirements
- **This Project**: tackle the high-dimensional case.
- **Idea:** Use variance reduction schemes (e.g., SVRG) to estimate ∇u with much lower sample complexity.
- Variance reduction schemes can be applied to *any* empirical risk minimization problems, e.g., phase retrieval, deep learning, optimal control

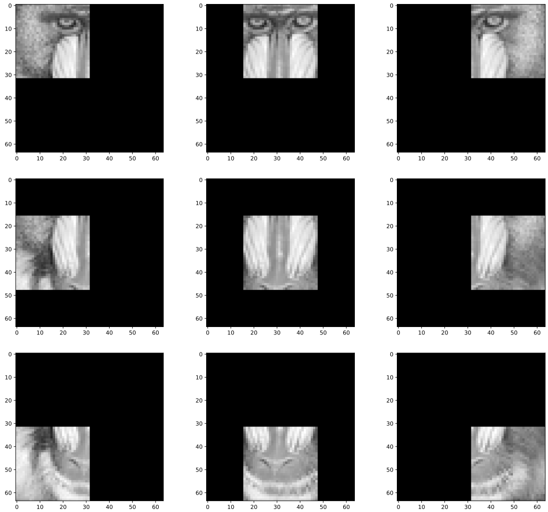
- Example: consider the ptychographic phase retrieval problem given by

$$\min_x \sum_{i=1}^{N} \| |\mathcal{F}(Q_i x)| - b_i \|, \qquad (4)$$

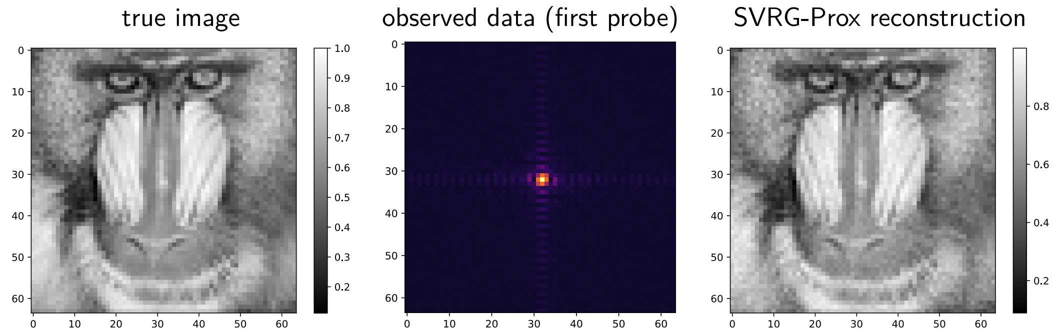
where $\mathcal{F}$ is the Fourier transform, Q_i are filters corresponding to different regions being scanned, b_i are observed measurements with 5% noise.

- Problem is highly-nonconvex[2] and is used in high-resolution electron microscopy
- We will consider problems of dimension $4096 \implies$ **expectation formula for Moreau envelope gradient (HJ-Prox) requires too many samples**

[2] "Phase Retrieval. What's New? ", D. R. Luke, SIAG/OPT Views and News, 25(1):1–5 (2017).


0
10
20
30
40
50
60
0 10 20 30 40 50 60

SVRG Proximals Preliminary Experiments



Reconstruction performed on measurements with 5% noise

Future Directions

- Theoretical framework for SVRG-Prox to converge to *global minimum*. Connections with Hamilton-Jacobi PDEs
- Application to other high-dimensional problems such as control, games, and deep learning
- Numerical considerations and fast implementation

References

- Heaton, H., Wu Fung, S., & Osher, S. (2023). Global solutions to nonconvex problems by evolution of hamilton-jacobi pdes. *Communications on Applied Mathematics and Computation*, 1-21.
- Osher, S., Heaton, H., & Wu Fung, S. (2023). A Hamilton-Jacobi-based Proximal Operator. *Proceedings of the National Academy of Sciences*, 120(14), e2220469120.

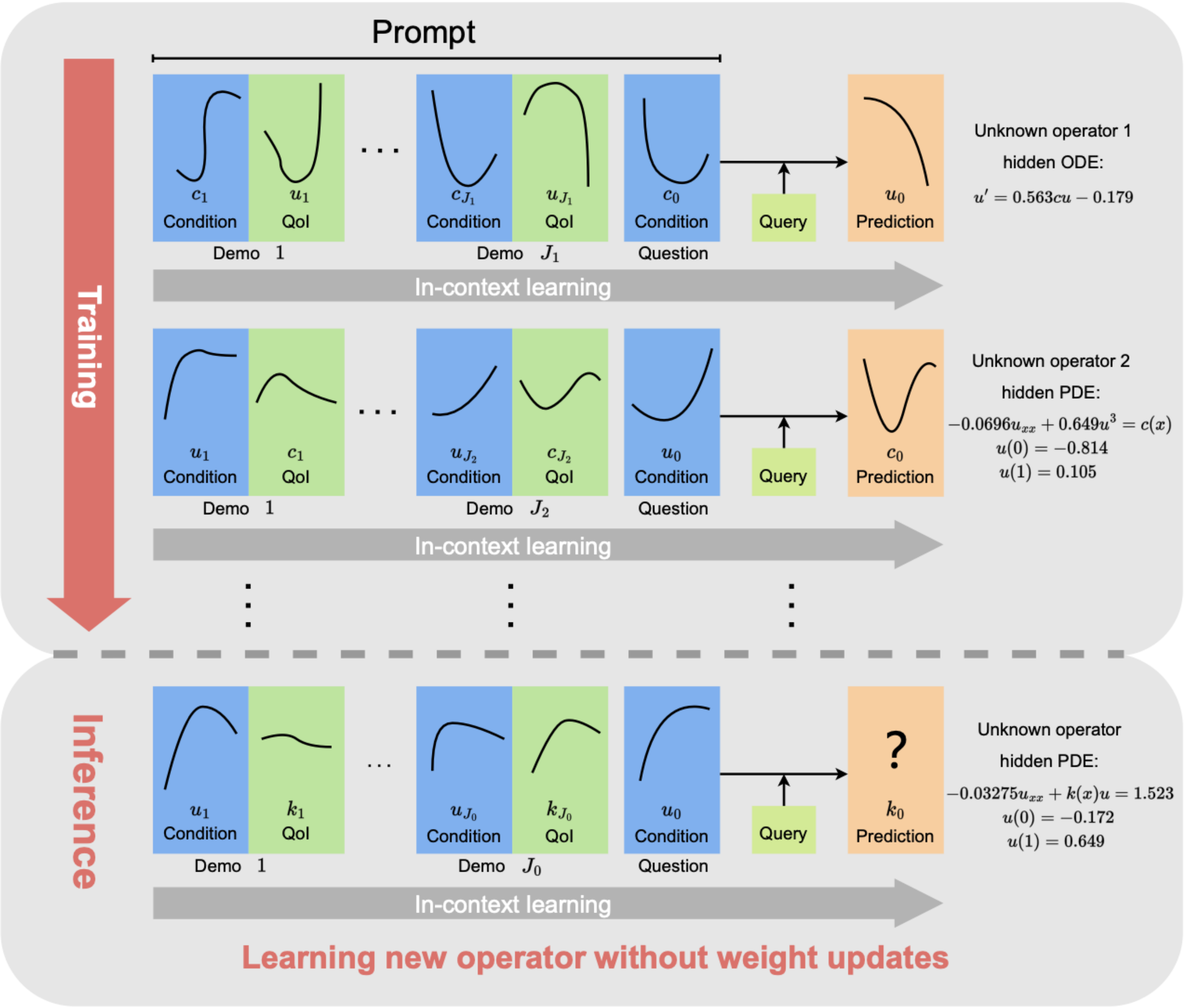
In-Context Operator Learning with Prompts for PDE and Mean Field Control Problems

Liu Yang, Siting Liu, Tingwei Meng, Stanley J. Osher

Preprint: https://arxiv.org/abs/2304.07993

Motivation

- Existing methods for solving differential equations via neural networks are limited by their equation specificity and need for frequent retraining when switching to new problems.
- We wish to solve multiple differential-equation-related tasks (including mean field control problems) with a single neural network, getting rid of retraining (even fine-tuning) for new tasks.
- In the journey toward Artificial General Intelligence (AGI), we also need networks that can adapt to new physical systems and tasks, just as a human would.
- Inspired by the "learning to learn" success in models like GPT-2 and GPT-3, we aim to adapt this concept for differential equation problems, leading to our proposal: In-Context Operator Networks (ICON).



Operator: mapping from condition to QoI, both are functions.

Prompt: the condition and QoI functions for demonstration, plus a question condition.

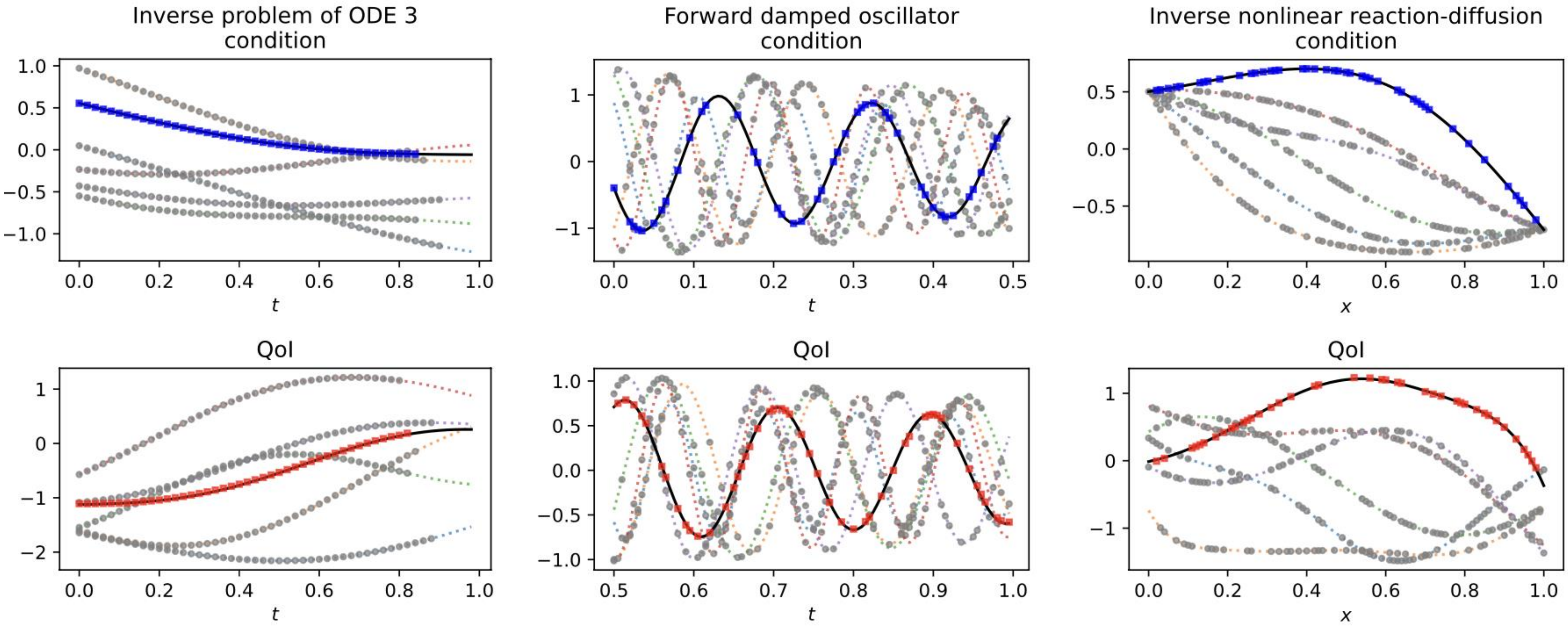
Training: ICON is trained to be an "operator learner", instead of an "operator approximator". It takes a prompt as input and predicts the question QoI. Here, query refers to where we want to evaluate the question QoI.

Inference: learn and apply the new unknown operator, without weight updates.

#	Problem Description	Differential Equations	Parameters	Conditions	QoIs
1	Forward problem of ODE 1	$\frac{d}{dt}u(t) = a_1c(t) + a_2$ for $t \in [0,1]$	a_1, a_2	$u(0), c(t), t \in [0,1]$	$u(t), t \in [0,1]$
2	Inverse problem of ODE 1			$u(t), t \in [0,1]$	$c(t), t \in [0,1]$
3	Forward problem of ODE 2	$\frac{d}{dt}u(t) = a_1c(t)u(t) + a_2$ for $t \in [0,1]$	a_1, a_2	$u(0), c(t), t \in [0,1]$	$u(t), t \in [0,1]$
4	Inverse problem of ODE 2			$u(t), t \in [0,1]$	$c(t), t \in [0,1]$
5	Forward problem of ODE 3	$\frac{d}{dt}u(t) = a_1u(t) + a_2(t)c(t) + a_3$ for $t \in [0,1]$	a_1, a_2, a_3	$u(0), c(t), t \in [0,1]$	$u(t), t \in [0,1]$
6	Inverse problem of ODE 3			$u(t), t \in [0,1]$	$c(t), t \in [0,1]$
7	Forward damped oscillator	$u(t) = A\sin(\frac{2\pi}{T}t + \eta)e^{-kt}$ for $t \in [0,1]$	k	$u(t), t \in [0,0.5)$	$u(t), t \in [0.5,1]$
8	Inverse damped oscillator			$u(t), t \in [0.5,1]$	$u(t), t \in [0,0.5)$
9	Forward Poisson equation	$\frac{d^2}{dx^2}u(x) = c(x)$ for $x \in [0,1]$	$u(0), u(1)$	$c(x), x \in [0,1]$	$u(x), x \in [0,1]$
10	Inverse Poisson equation			$u(x), x \in [0,1]$	$c(x), x \in [0,1]$
11	Forward linear reaction-diffusion	$-\lambda a\frac{d^2}{dx^2}u(x) + k(x)u(x) = c$ for $x \in [0,1]$, $\lambda = 0.05$	$u(0), u(1), a, c$	$k(x), x \in [0,1]$	$u(x), x \in [0,1]$
12	Inverse linear reaction-diffusion			$u(x), x \in [0,1]$	$k(x), x \in [0,1]$
13	Forward nonlinear reaction-diffusion	$-\lambda a\frac{d^2}{dx^2}u(x) + ku^3 = c(x)$ for $x \in [0,1]$, $\lambda = 0.1$	$u(0), u(1), k, a$	$c(x), x \in [0,1]$	$u(x), x \in [0,1]$
14	Inverse nonlinear reaction-diffusion			$u(x), x \in [0,1]$	$c(x), x \in [0,1]$
15	MFC g-parameter 1D $\to$ 1D	$\inf_{\rho,m} \iint c\frac{m^2}{2\rho}dxdt + \int g(x)\rho(1,x)dx$ such that $\partial_t\rho(t,x) + \nabla_x m(t,x) = \mu\Delta_x\rho(t,x)$ for $t \in [0,1], x \in [0,1]$, $c = 20, \mu = 0.02$, periodic boundary condition in spatial domain	$g(x), x \in [0,1]$	$\rho(t=0,x), x \in [0,1]$	$\rho(t=1,x)$, $x \in [0,1]$
16	MFC g-parameter 1D $\to$ 2D			$\rho(t=0,x), x \in [0,1]$	$\rho(t,x)$, $x \in [0,1]$, $t \in [0.5,1]$
17	MFC g-parameter 2D $\to$ 2D			$\rho(t,x), t \in [0,0.5)$, $x \in [0,1]$	$\rho(t,x)$, $x \in [0,1]$, $t \in [0.5,1]$
18	MFC ρ_0-parameter 1D $\to$ 1D		$\rho(t=0,x)$, $x \in [0,1]$	$g(x), x \in [0,1]$	$\rho(t=1,x)$, $x \in [0,1]$
19	MFC ρ_0-parameter 1D $\to$ 2D				$\rho(t,x)$, $x \in [0,1]$, $t \in [0.5,1]$

List of the problems solved with a single neural network

A Glance of ICON for ODE and PDE Problems



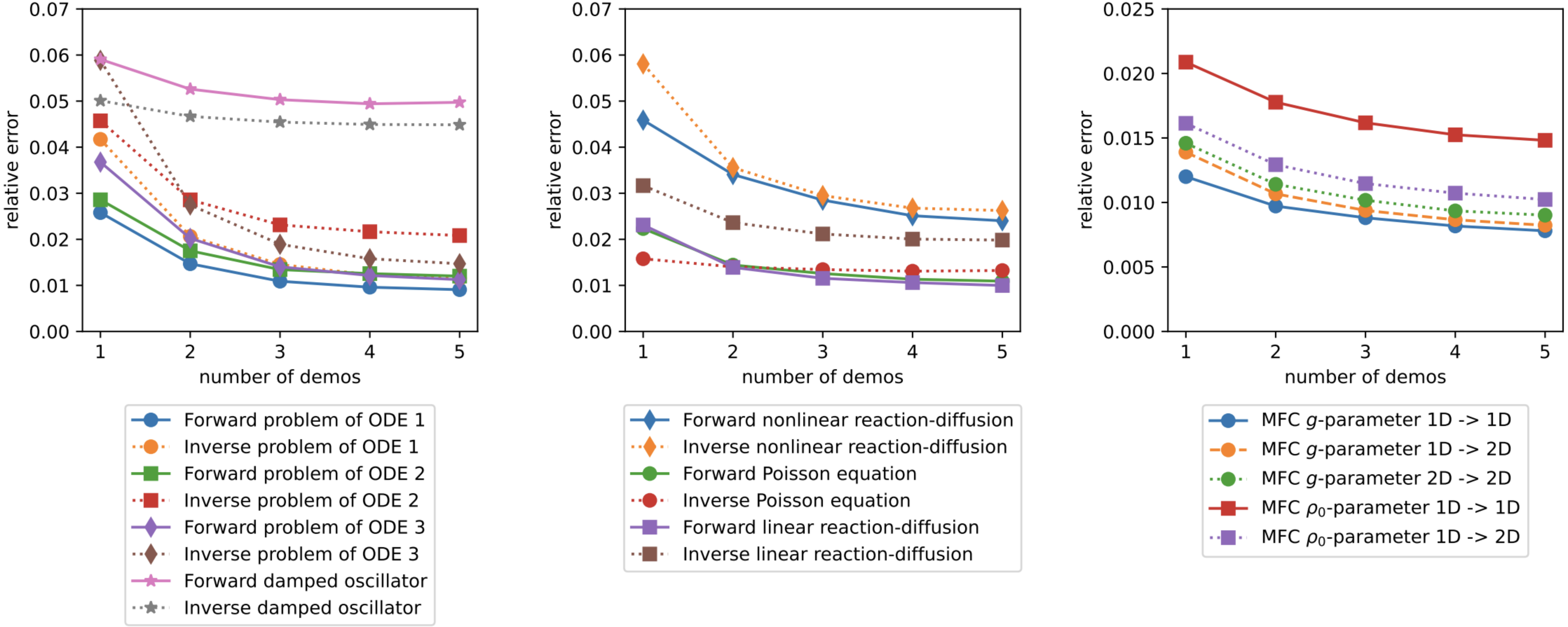
The colored dotted lines represent the of condition and QoI functions in demos.

The grey dots represent the data of the demo conditions and QoIs used in the prompts.

The blue dots represent the data in the question conditions.

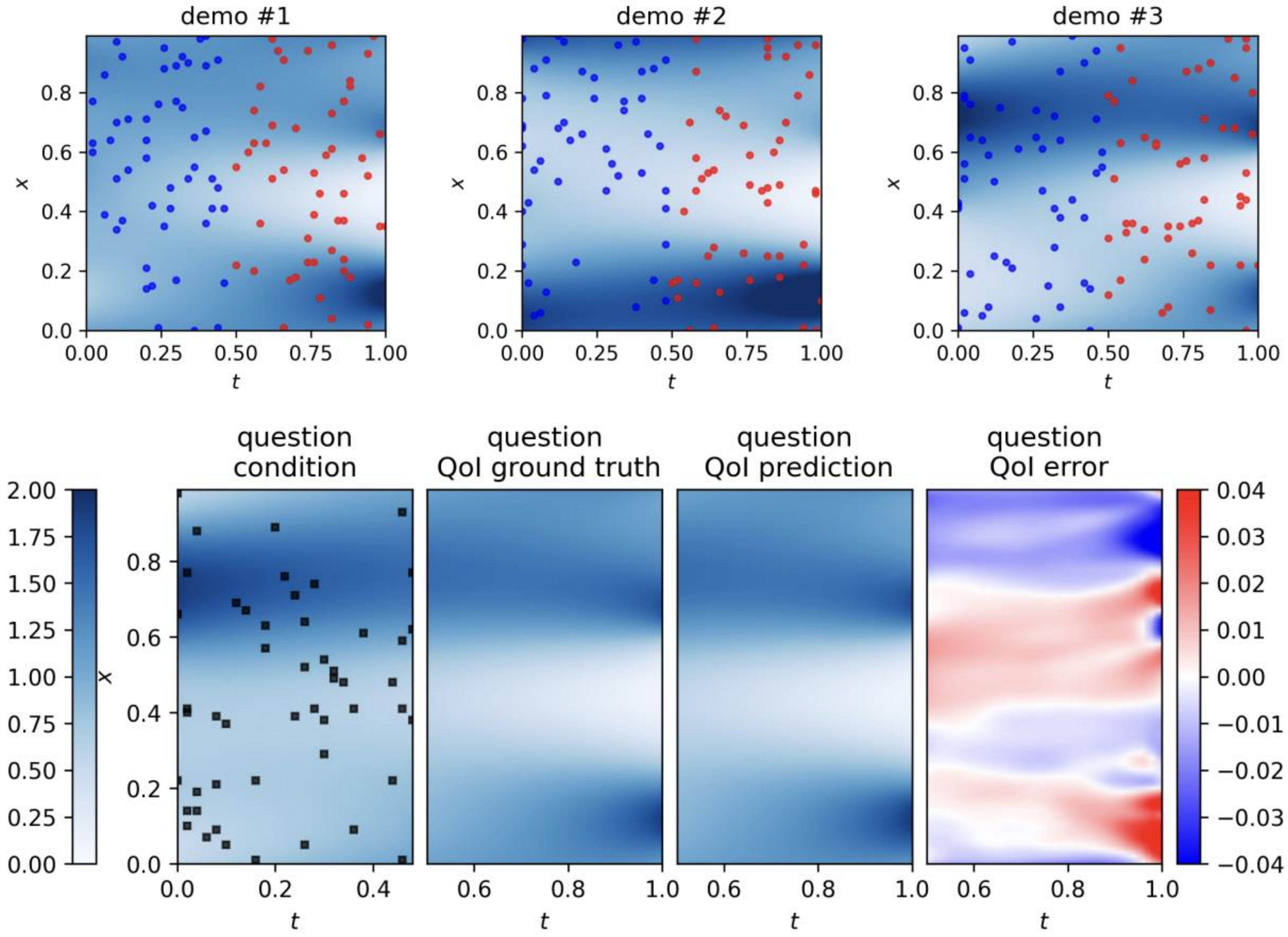
The red dots represent the prediction of the question QoI. One can see the consistency between the prediction and the ground truth **(solid black lines)**.

In-Distribution Operators



Average relative testing errors for all 19 problems listed in the table.
The error decreases with an increasing number of demos in each prompt.
With only five demos, the error goes down to about 1%-2% for most cases.

Mean-Field Control Problem (Problem #17)



Plots: density field in the temporal-spatial domain. Three demos and one question share the same terminal cost as the unknown parameter in the operator.

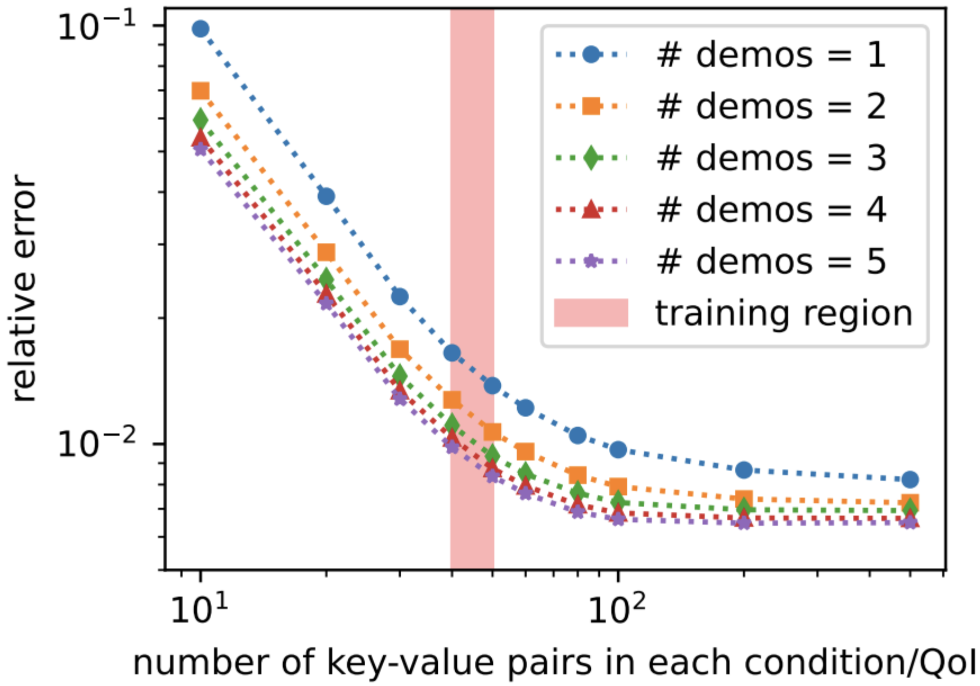
Blue dots: data for demo condition (density in the first half of the time).

Red dots: data for demo QoI (density in the second half of the time).

Black dots: data for question condition.

We make the prediction on $\rho(t,x), (t,x) \in [0.5,1] \times [0,1]$

More/Less Data Points (Super/Sub-Resolution)

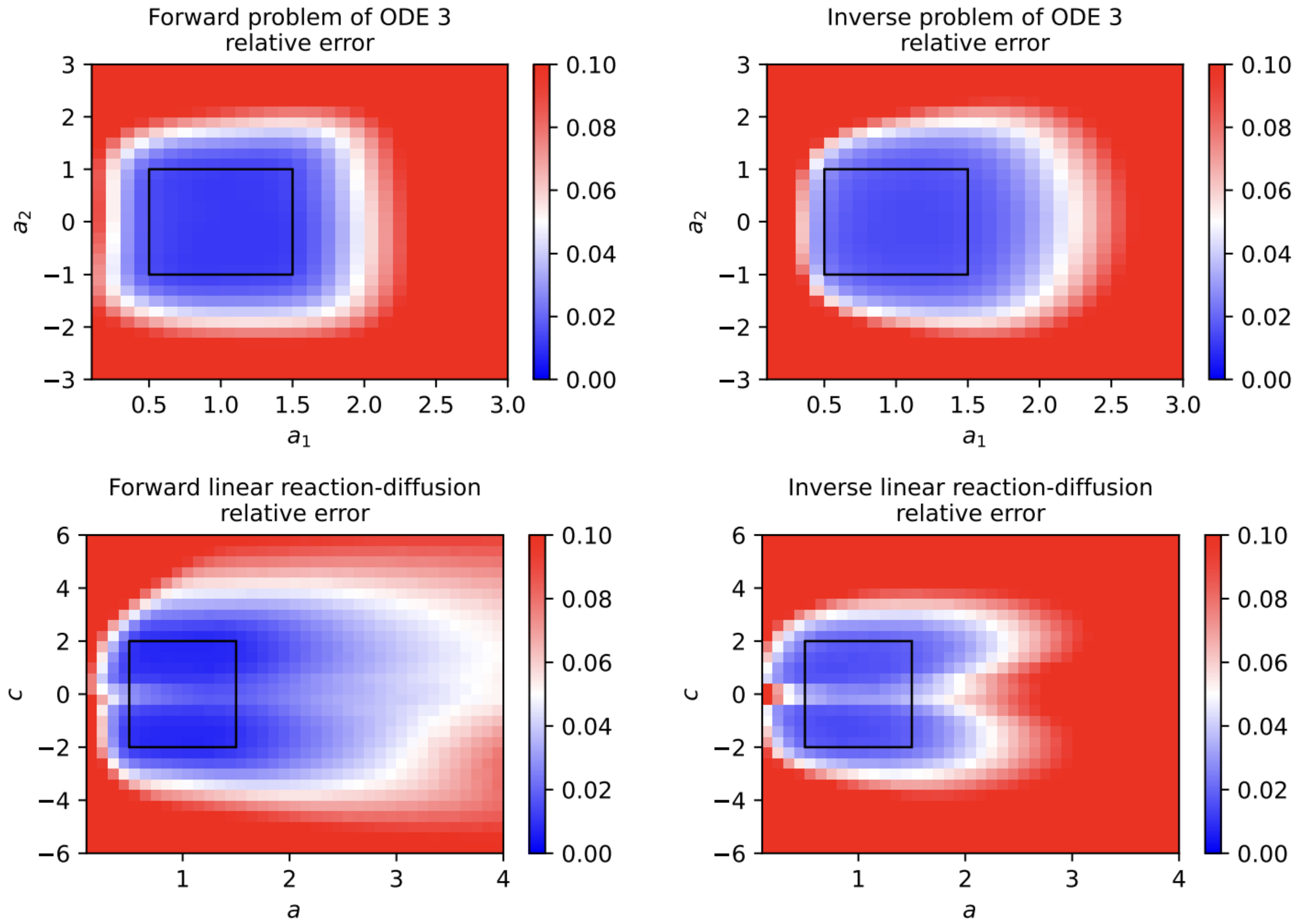


Still the same problem (mean-field control with terminal cost as the unknown parameters).

As we increase the number of data points in each condition/QoI function, the error decreases and finally converges below 1%.

ICON is trained using 41 to 50 data points, represented by the narrow **red region**.

Out-of-Distribution Operators



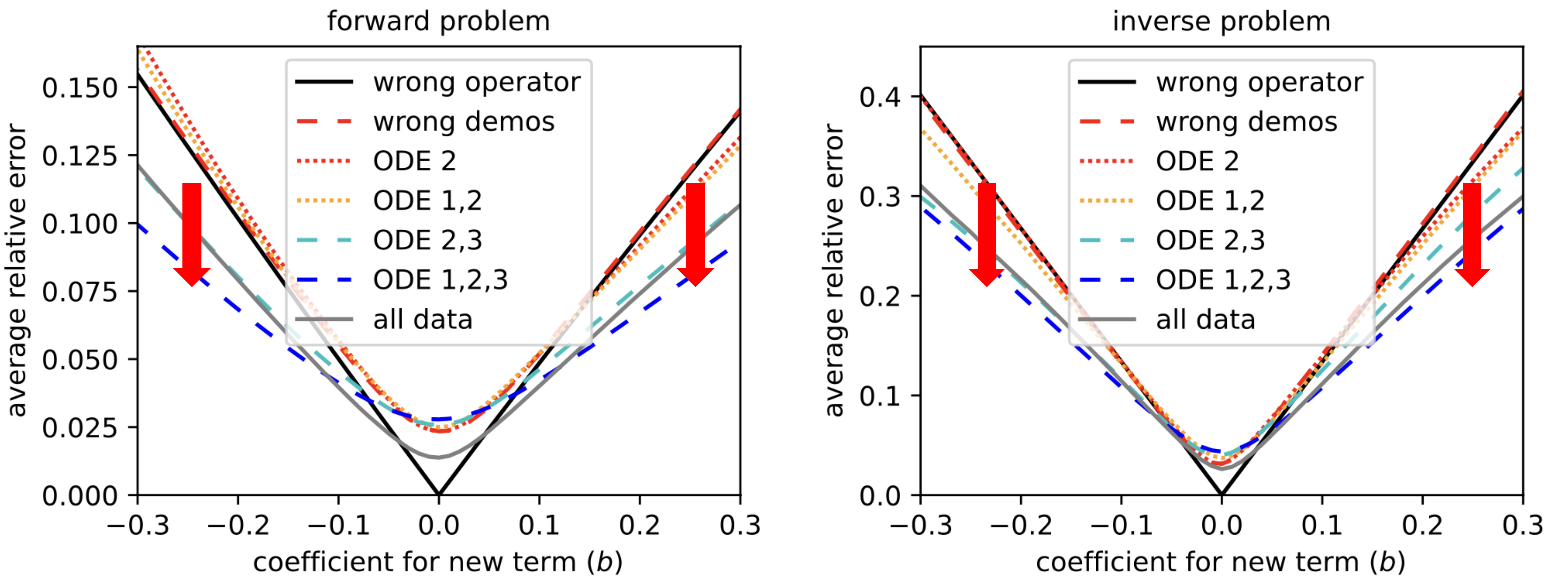
Taking forward and inverse problems of an ODE and a PDE as examples (problems 5, 6, 11, and 12 in the table).

Each pair of (a1, a2) or (a,c) defines an operator.

Black rectangle: training region.

ICON demonstrated accurate prediction capabilities even with operator parameters extending beyond the training region.

Generalization to Equations of New Forms



We designed a new ODE by adding a new term to ODE2. The new term is borrowed from ODE3.

The error shows a **decreasing trend** as the training dataset becomes larger and more diversified. This is preliminary evidence of learning operators for equations of new forms that were never seen in training data.

Discussion

Why a very few demos are sufficient to learn the operator?

We leveraged the commonalities shared in training and testing operators. ICON only need to identify the equation and hidden parameters.

Only need to learn the operator for a certain distribution of conditions.

For a larger family of operators, ICON requires more demos (especially for those complicated operators), as well as a larger neural network.

What's next?

Scale up. In the field of NLP, scaling up leads to emergent abilities beyond human expectations. We anticipate the possibility of witnessing artificial general intelligence for scientific computing with large ICON models.

Improvements in neural network architectures and training methods, as well as further theoretical and numerical studies of how ICON works.

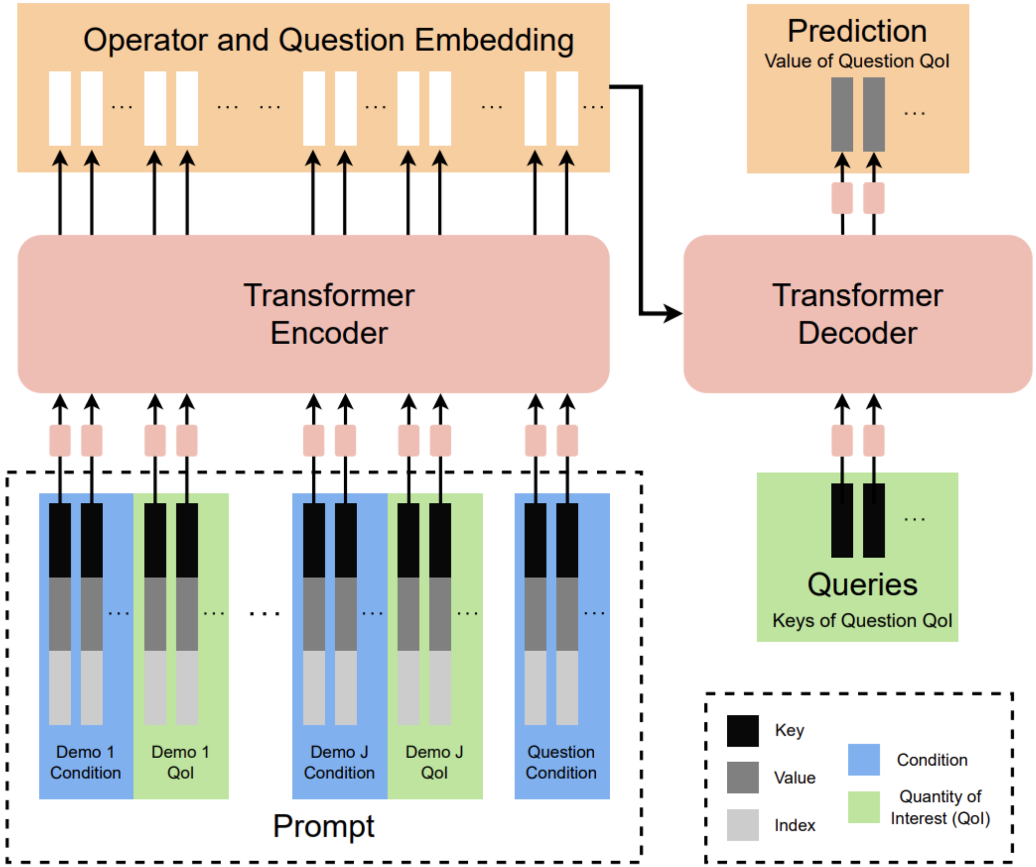


Figure 2: The neural network architecture for In-Context Operator Networks (ICON).

		condition					QoI			
key	term	0	0	$\dots$	0	1	0	0	$\dots$	0
	time	t_1	t_2	$\dots$	t_{n_j-1}	0	τ_1	τ_2	$\dots$	τ_{m_j}
	space	0	0	$\dots$	0	0	0	0	$\dots$	0
value		$c(t_1)$	$c(t_2)$	$\dots$	$c(t_{n_j-1})$	$u(0)$	$u(\tau_1)$	$u(\tau_2)$	$\dots$	$u(\tau_{m_j})$
index		$\mathbf{e}_j$	$\mathbf{e}_j$	$\dots$	$\mathbf{e}_j$	$\mathbf{e}_j$	$-\mathbf{e}_j$	$-\mathbf{e}_j$	$\dots$	$-\mathbf{e}_j$

Algorithm 2: The training and inference of In-Context Operator Networks (ICON).

// Training stage:
for $i = 1, 2, \ldots,$ *training steps* **do**
    for $b = 1, 2, \ldots,$ *batch size* **do**
        Randomly select a type of problem and a set of parameters from dataset;
        Randomly set the number of demos J, and the number of key-value pairs in each condition and QoI of the demos and question;
        From N pairs of conditions and QoIs, randomly select J pairs as demos and one pair as the question;
        Build a prompt matrix, query vectors, and the ground truth using the selected demos and question;
    end
    Use the batched prompts, queries and labels to calculate the MSE loss and update the neural network parameters with gradients;
end
// Inference stage:
Given a new system with an unknown operator, collect demos and a question condition, and design the queries;
Construct the prompt using the demos and question condition;
Get the prediction of the question QoI using a forward pass of the neural network;