Active Learning with Selective Time-Step Acquisition for PDEs

M.S. Defense

Graduate School of Artificial Intelligence, KAIST

Yegon Kim (Advisor: Juho Lee)

June 6th 2025

Roadmap

1. Motivation
   - Why Active Learning (AL)?
   - Need for Time-Step Selective AL

2. Background
   - PDE
   - Surrogate Model
   - Active Learning

3. Method (STAP)
   - Framework
   - Acquisition Function
   - Batch Acquisition Algorithm

4. Experimental Setup
   - Baseline AL methods
   - Target PDEs
   - Surrogate Model

5. Results and Discussion
   - Performance of Active Learning Methods
   - Analysis of Timesteps Selected by STAP
   - Out-of-distribution Synthetic Inputs

Motivation

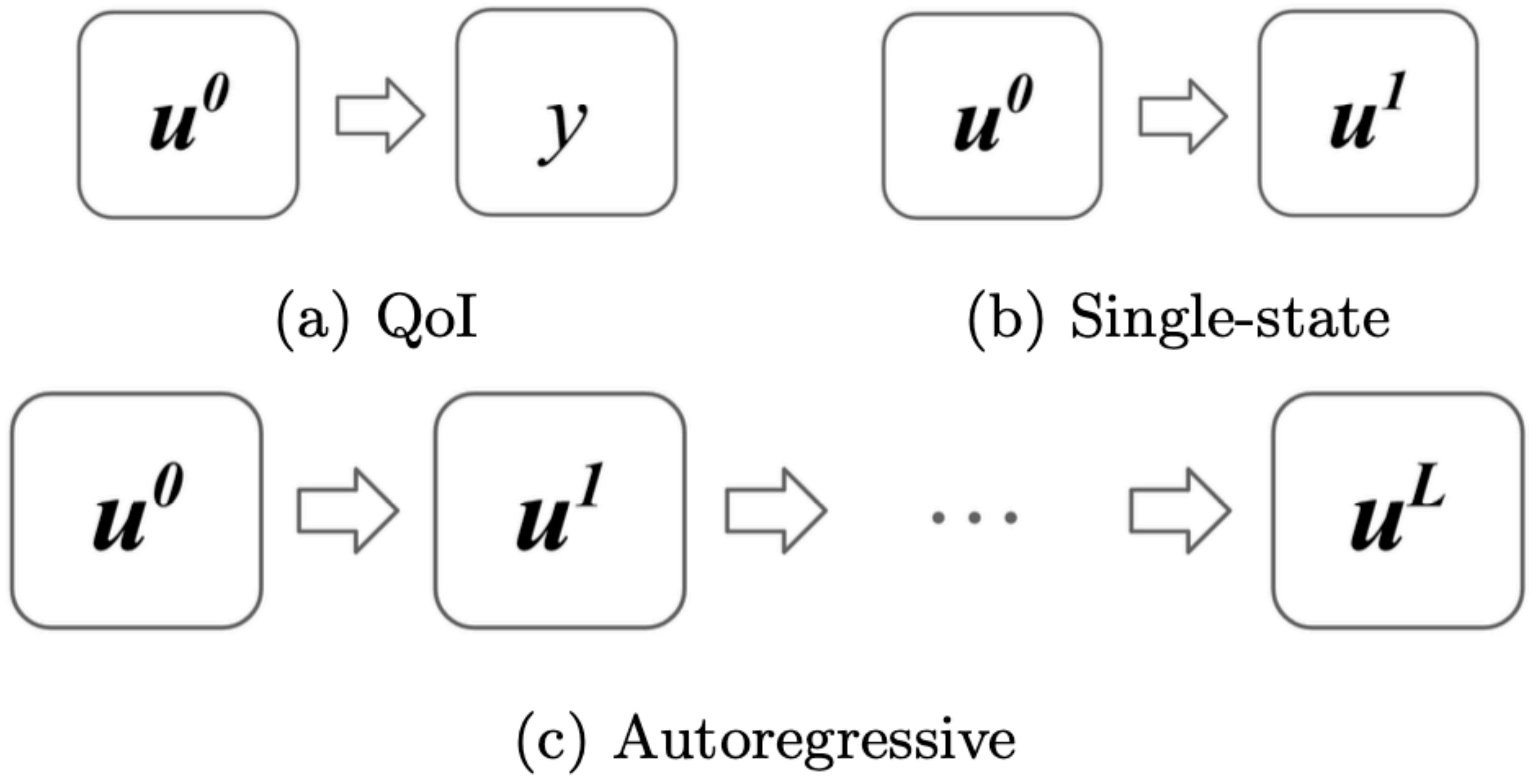
Why Active Learning (AL)?

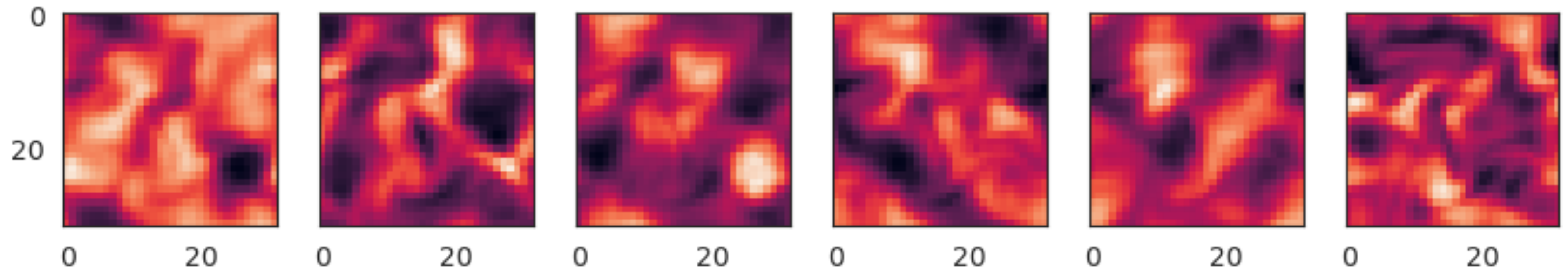
Partial differential equations (PDEs) model scientific phenomena.

We are interested in **autoregressive** prediction of future states.

Numerical solvers are expensive to run.

Surrogate models are cheap to run but require training data from numerical solvers.



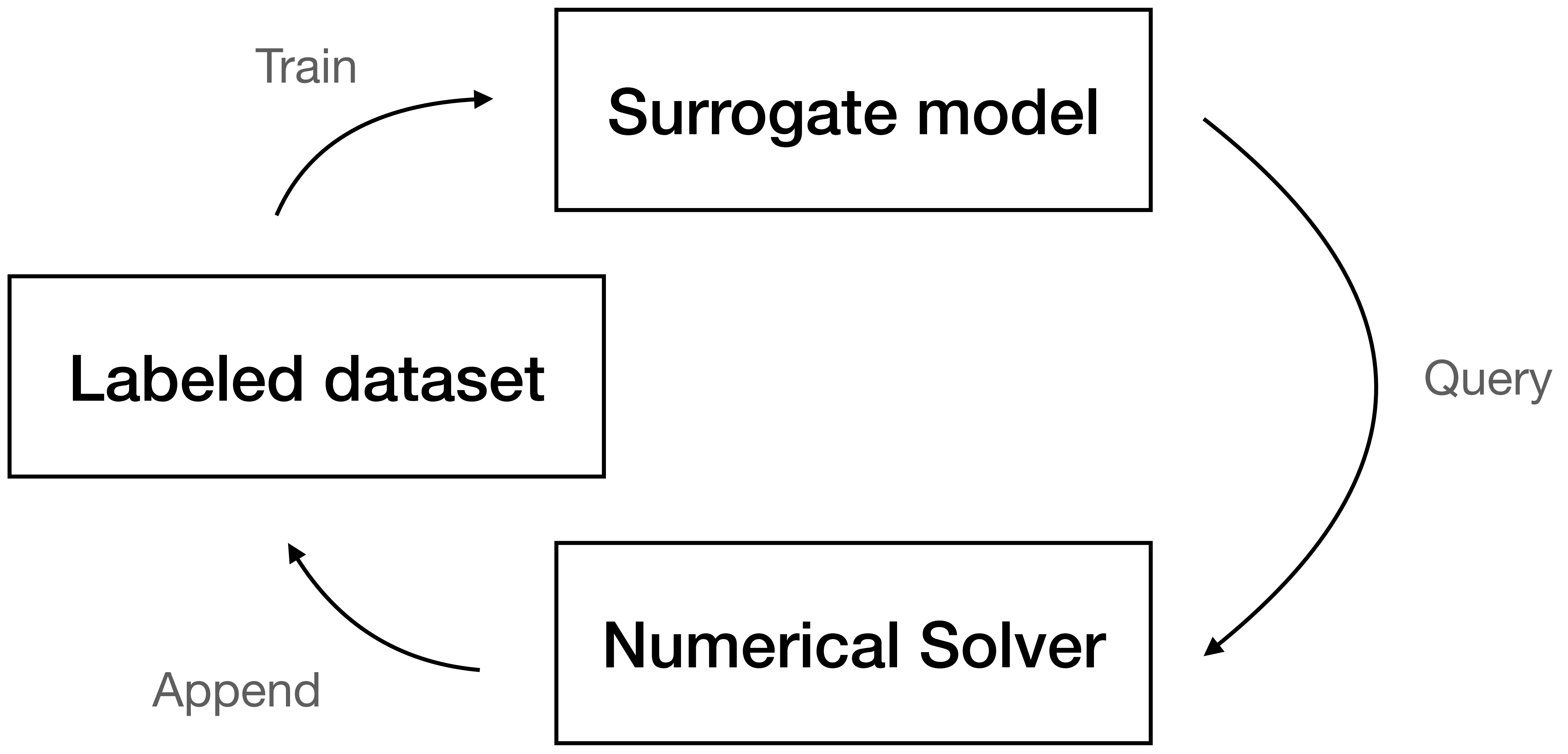


Sample Trajectory of the Compressible Navier-Stokes (CNS) equation

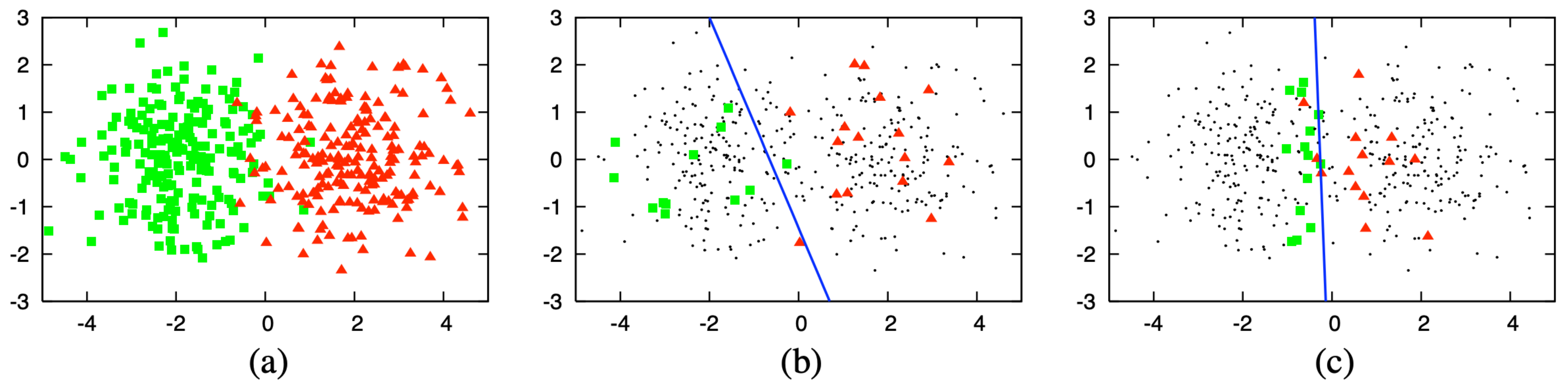
Motivation

Why Active Learning (AL)?

Active learning (AL) reduces the cost of data acquisition by selecting the most important inputs to query.



Active learning loop. Warm-starts with an initial labeled dataset.



(a) Full dataset (b) Random sampling (c) Active learning

Motivation

Need for Time-Step Selective AL

A good AL algorithm should pick samples that are (1) **informative**, (2) **diverse**, and (3) **representative** [Wu 2018].

(1): *High mutual information* between samples and hypothesis

(2): *Small overlap* between samples

(3): Samples don't deviate too much from the *data distribution*

Motivation

Need for Time-Step Selective AL

AL4PDE [Musekamp et al. 2024] is the first work that applies AL to PDE trajectory learning.

Acquires **full trajectories** starting from initial conditions selected by AL.

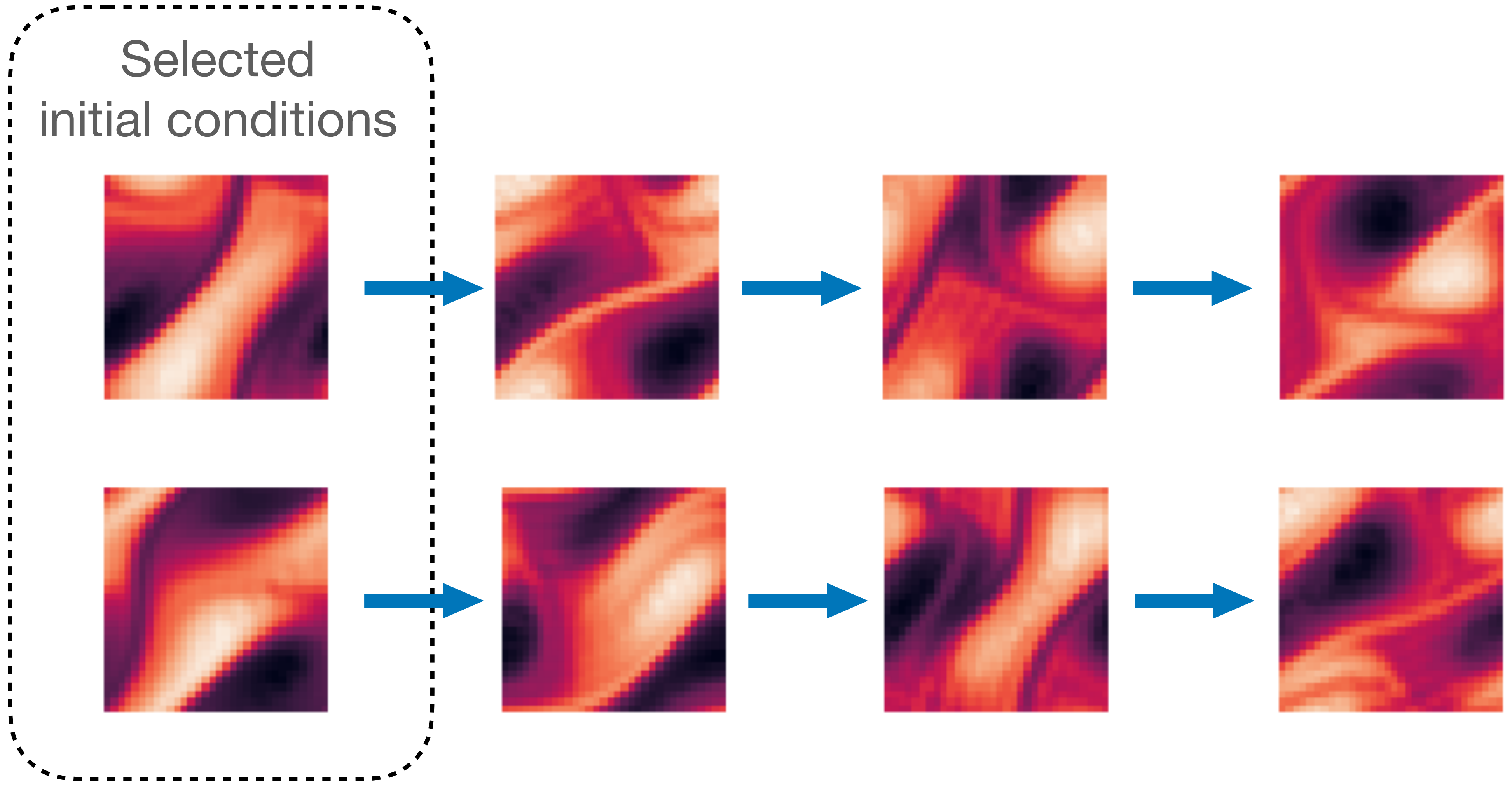


Illustration of full trajectory acquisition. The above example would incur a cost of 6 calls to the numerical solver.

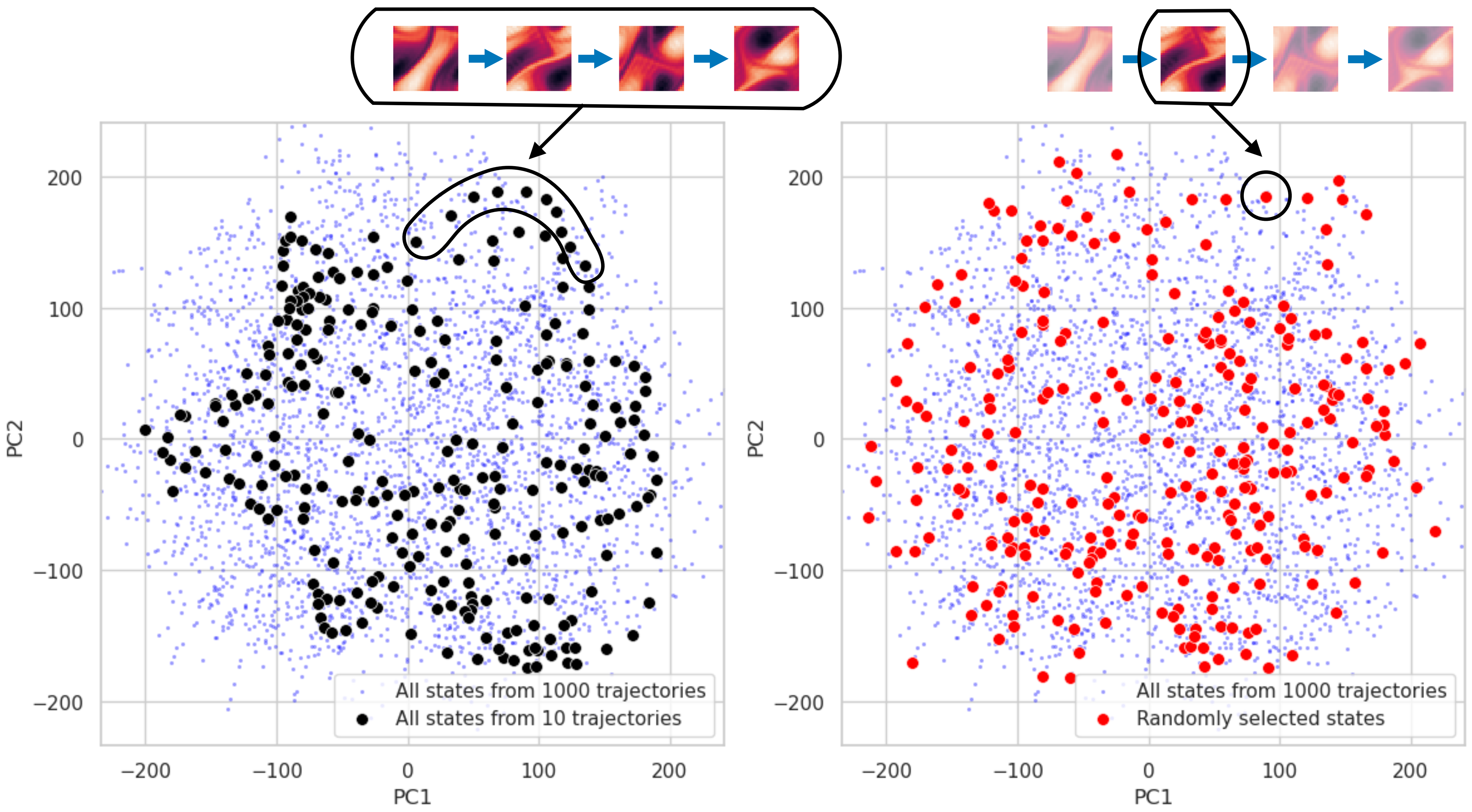
Motivation

Need for Time-Step Selective AL

Is full trajectory acquisition the only option? Because...

(1) States at certain time-steps are more **informative** than others

(2) States from different trajectories are more **diverse** than states from the same trajectory



Left: 130 PDE states from 10 trajectories of length 13
Right: 130 PDE states *randomly selected* from 1,000 trajectories

Motivation

Need for Time-Step Selective AL

We propose **S**elective **T**ime-Step **A**cquisition for **P**DEs (STAP).

It acquires **partial trajectories** by approximating intermediate steps with the surrogate model.

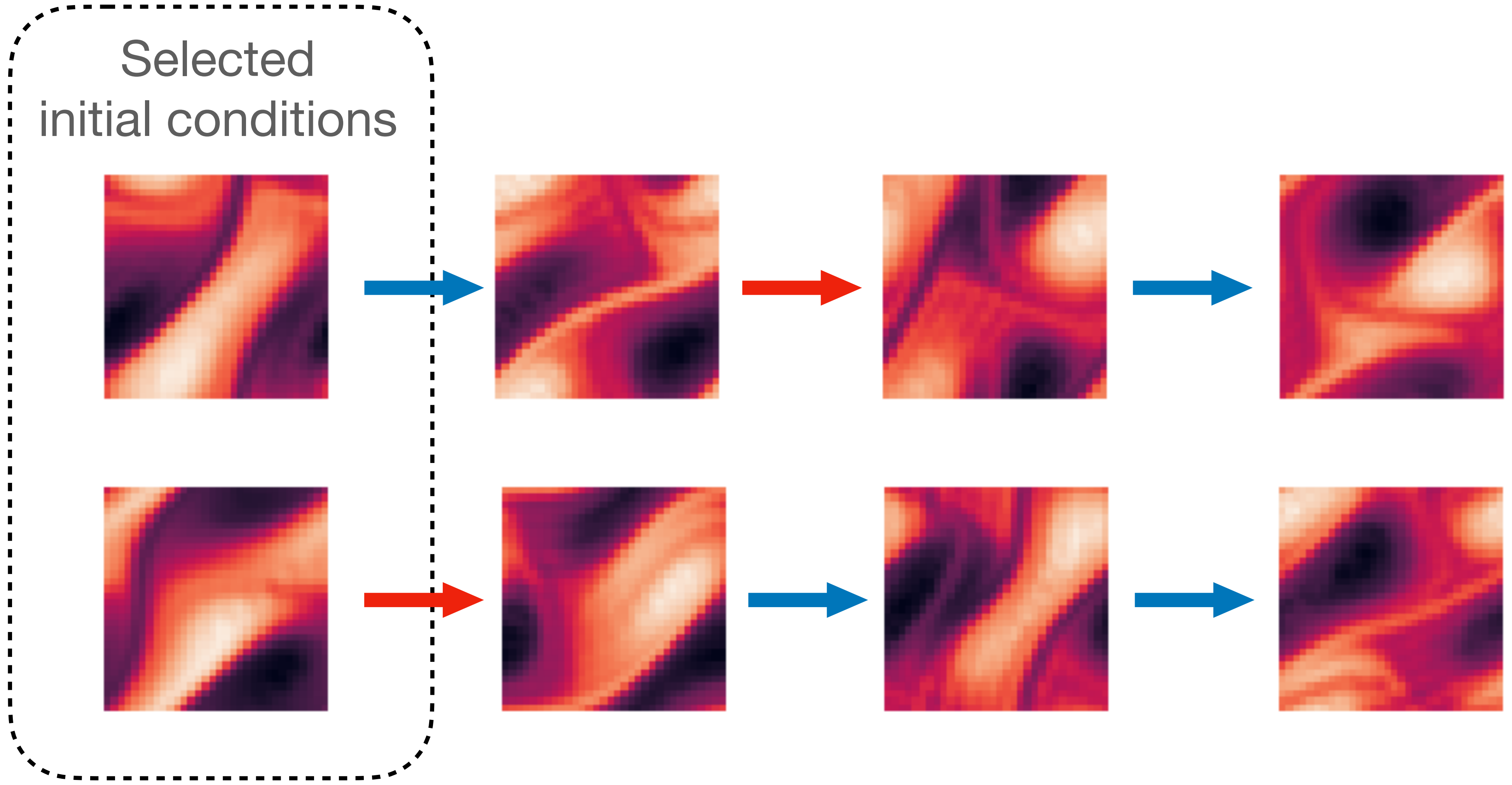


Illustration of STAP. Red arrows indicate the surrogate model, blue arrows indicate the numerical solver. The above example would incur a cost of 4 calls to the numerical solver.

Background

Background

PDE

Given a time domain $t \in [0,T]$ and spatial domain $\boldsymbol{x} \in \mathbb{X}$,

PDEs are written in the form

$$\partial_t \boldsymbol{u} = F(t, \boldsymbol{x}, \boldsymbol{u}, \partial_{\boldsymbol{x}} \boldsymbol{u}, \partial_{\boldsymbol{xx}} \boldsymbol{u}, \ldots)$$

where the solution is

$$\boldsymbol{u} : [0,T] \times \mathbb{X} \to \mathbb{R}^n.$$

Background

PDE

Given Δt, the numerical solver G maps the current state

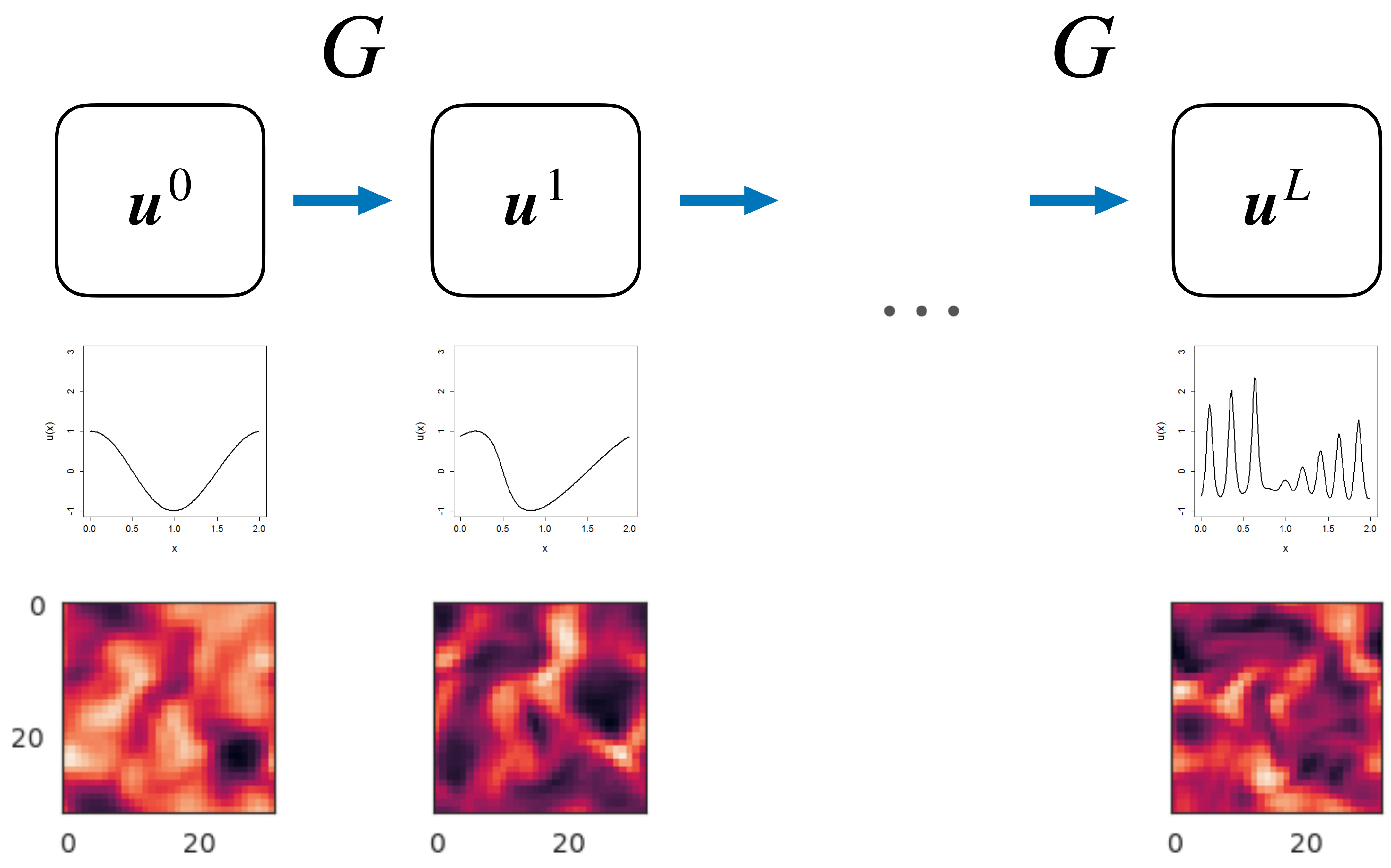
$$\boldsymbol{u}^0 := \boldsymbol{u}(t_0, \cdot\,)$$

to the evolved state

$$\boldsymbol{u}^1 := \boldsymbol{u}(t_0 + \Delta t, \cdot\,).$$

We obtain a length-L trajectory $(\boldsymbol{u}^i)_{i=1}^L$

$$\boldsymbol{u}^i := G^{(i)}[\boldsymbol{u}^0].$$

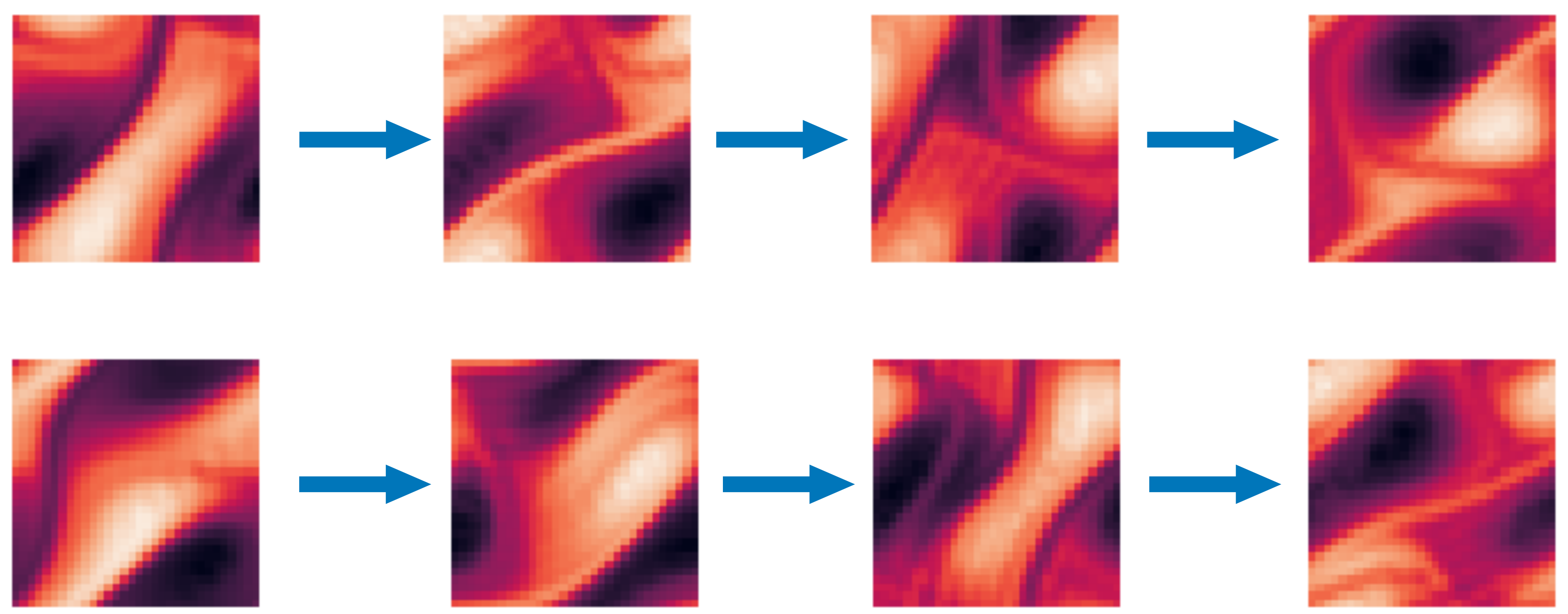


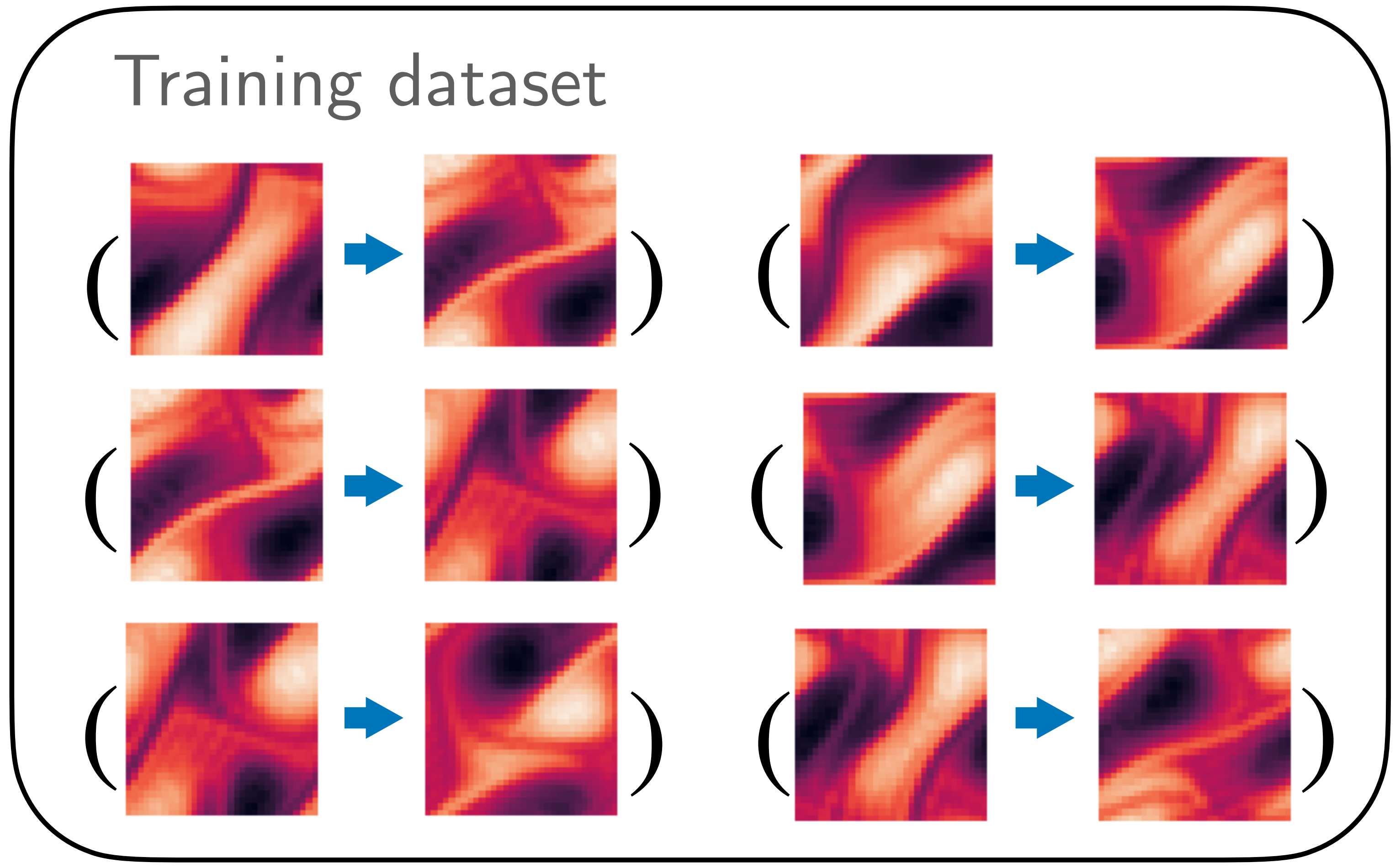
*$G^{(i)}$ is the i^{th} iterate of G

Background

Surrogate Model

We train a neural surrogate model $\hat{G}$ with input-output pairs $(\boldsymbol{u}, G[\boldsymbol{u}])$ from the numerical solver G.





Background

Surrogate Model

Our goal is to obtain a surrogate model $\hat{G}$ that approximates the numerical solver G with low error on length-L trajectories

$$\mathbb{E}_{\boldsymbol{u}^0 \sim p(\boldsymbol{u}^0)} \left[\mathrm{err} \left((G^{(i)}[\boldsymbol{u}^0])_{i=1}^{L}, (\hat{G}^{(i)}[\boldsymbol{u}^0])_{i=1}^{L} \right) \right].$$

For the purpose of AL, we sometimes train a **committee** of M surrogate models $\{\hat{G}_m\}_{m=1}^{M}$.

Background

Active Learning

We are given a *pool* of initial conditions $\boldsymbol{u}^0 \sim p(\boldsymbol{u}^0)$.

At each round:

1. We train a committee $\{\hat{G}_m\}_{m=1}^M$ with the current training dataset.
2. Using the committee, AL selects a batch of inputs $\boldsymbol{u}$ to the solver G.
   The batch size is limited to a certain budget B.
3. The input-output pairs $(\boldsymbol{u}, G[\boldsymbol{u}])$ are added to the training dataset.

Background

Active Learning

e.g. In AL4PDE [Musekamp et al. 2024], Query-by-Committee (QbC) [Seung et al. 1992] selects initial conditions $\boldsymbol{u}^0$ that *maximize*

$$a_{\mathrm{QbC}}(\boldsymbol{u}^0) = \sum_{i=1}^{L} \mathrm{Var}_m \left[\hat{G}_m^{(i)}[\boldsymbol{u}^0] \right]$$

and acquire full trajectories $\left(G^{(i)}[\boldsymbol{u}^0] \right)_{i=1}^{L}$.

a_{QbC} is called the *acquisition function* of QbC.

*Var is the total variance, i.e. trace of covariance matrix

Method - STAP

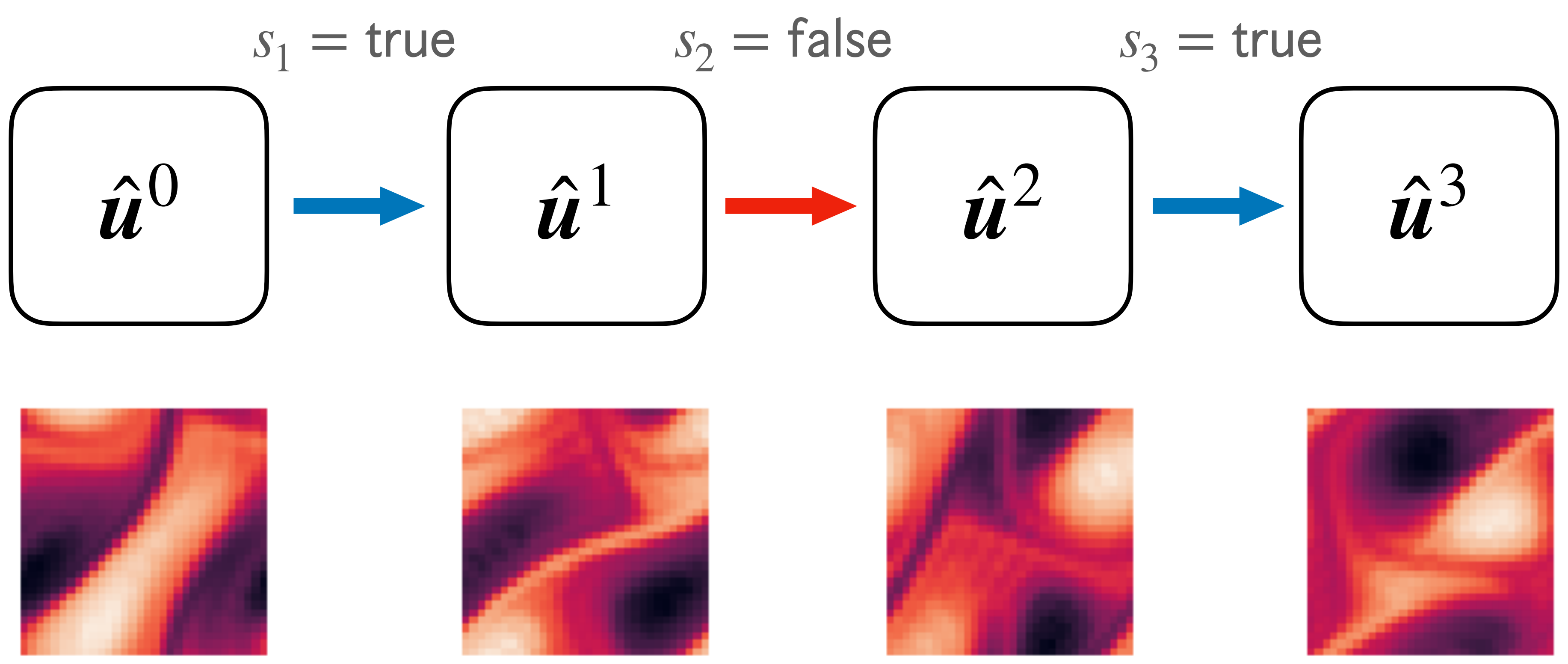
Method - STAP

Framework of Data Acquisition

For each initial condition $\boldsymbol{u}^0$, we choose a *sampling pattern* $S = (s_1, \dots, s_L)$.

Starting with $\hat{\boldsymbol{u}}^0 = \boldsymbol{u}^0$, we iterate over $1 \le i \le L$:

$$\hat{\boldsymbol{u}}^i = \begin{cases} G[\hat{\boldsymbol{u}}^{i-1}] & \text{if } s_i = \text{true} \\ \hat{G}[\hat{\boldsymbol{u}}^{i-1}] & \text{if } s_i = \text{false}\,. \end{cases}$$



Example with $S = (\text{true}, \text{false}, \text{true})$. Blue arrow indicates numerical solver G, red arrow indicates surrogate model $\hat{G}$.

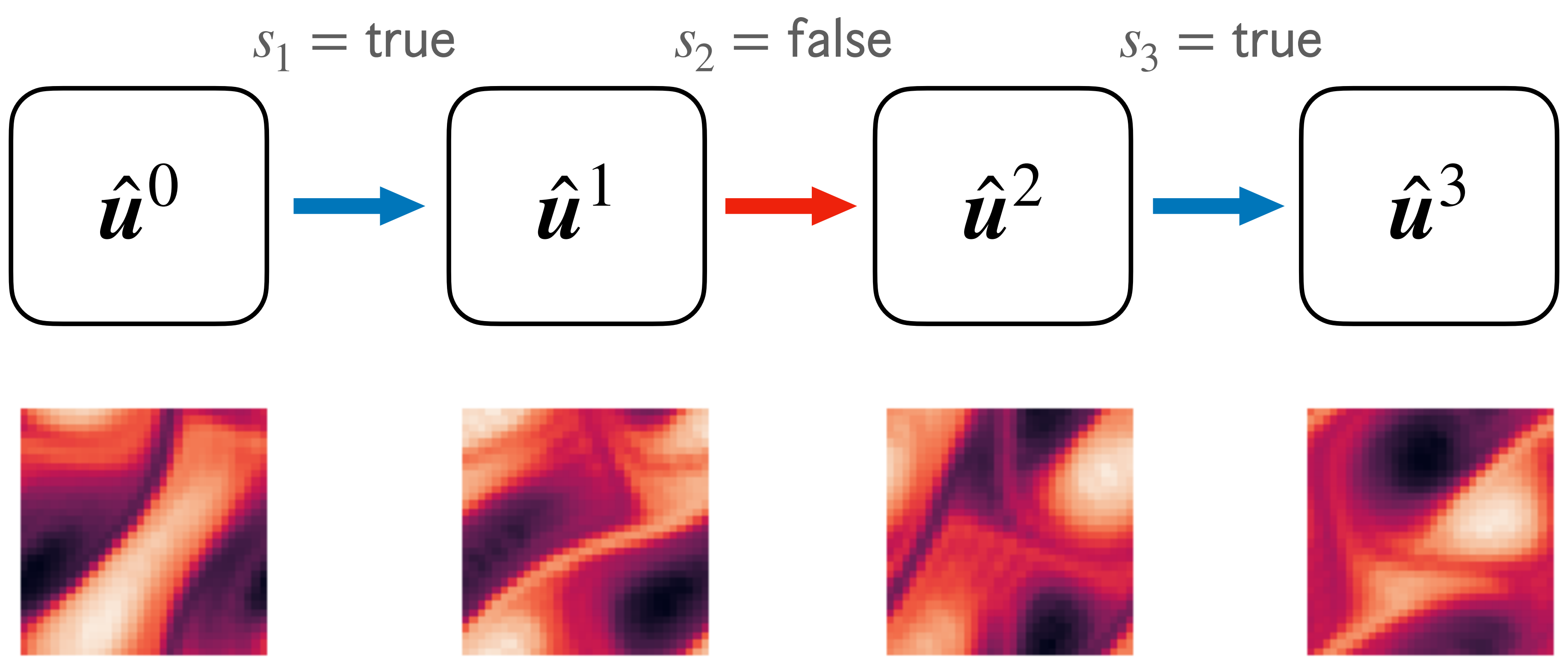
Method - STAP

Framework of Data Acquisition

We only add the numerical solver pairs ($\hat{\boldsymbol{u}}^{i-1} \rightarrow \hat{\boldsymbol{u}}^{i}$) to the training dataset.

Full trajectory acquisition, e.g. AL4PDE, is a special case with $S = (\text{true}, \ldots, \text{true})$.

The cost is reduced from L to $\|S\|$ (number of true entries).



Example with $S = (\text{true}, \text{false}, \text{true})$. Blue arrow indicates numerical solver G, red arrow indicates surrogate model $\hat{G}$.

Method - STAP

Acquisition Function

Consider a pair $(\hat{G}_a, \hat{G}_b)$ in the committee of M surrogate models.

Let $\hat{\boldsymbol{u}}_a$ and $\hat{\boldsymbol{u}}_b$ be the trajectories rolled-out with $\hat{G}_a$ and $\hat{G}_b$, starting from $\boldsymbol{u}^0$.

Let $\hat{\boldsymbol{u}}_{b,S,a}$ be the trajectory rolled-out with $\hat{G}_a$ when $s_i =$ true, and $\hat{G}_b$ when $s_i =$ false (as if $\hat{G}_a$ is the solver and $\hat{G}_b$ the surrogate).

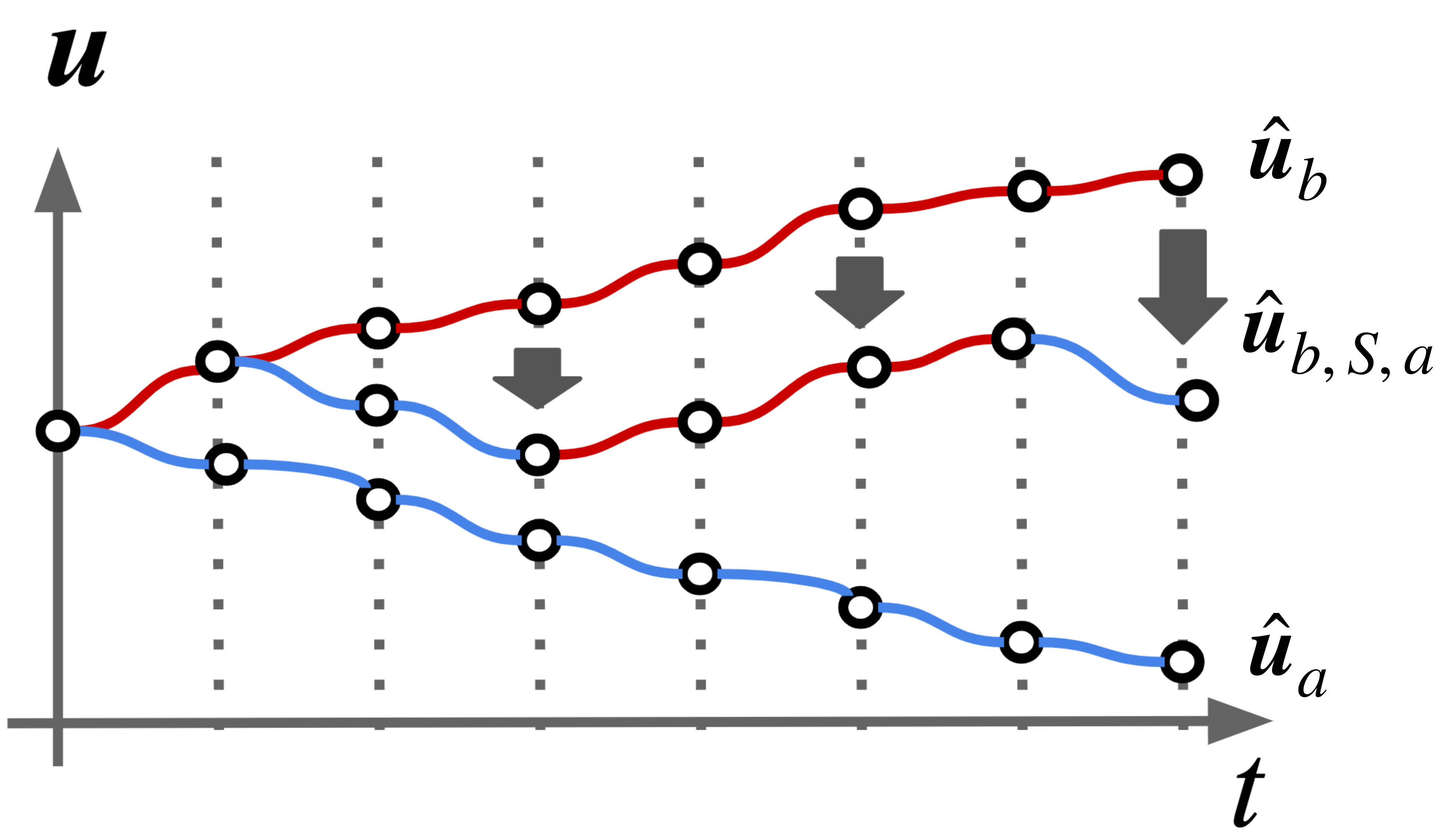


Illustration of $\hat{\boldsymbol{u}}_a$, $\hat{\boldsymbol{u}}_b$, and $\hat{\boldsymbol{u}}_{b,S,a}$. Blue lines represent $\hat{G}_a$, red lines represent $\hat{G}_b$.

Method - STAP

Acquisition Function

The variance reduction is defined as

$$R(a,b,S) := \sum_{i=1}^{L} \left(\|\hat{\boldsymbol{u}}_a^i - \hat{\boldsymbol{u}}_b^i\|^2 - \|\hat{\boldsymbol{u}}_a^i - \hat{\boldsymbol{u}}_{b,S,a}^i\|^2 \right)$$

Our acquisition function is defined as

$$a(\boldsymbol{u}^0, S) = \frac{1}{M(M-1)} \sum_{a,b} R(a,b,S)$$

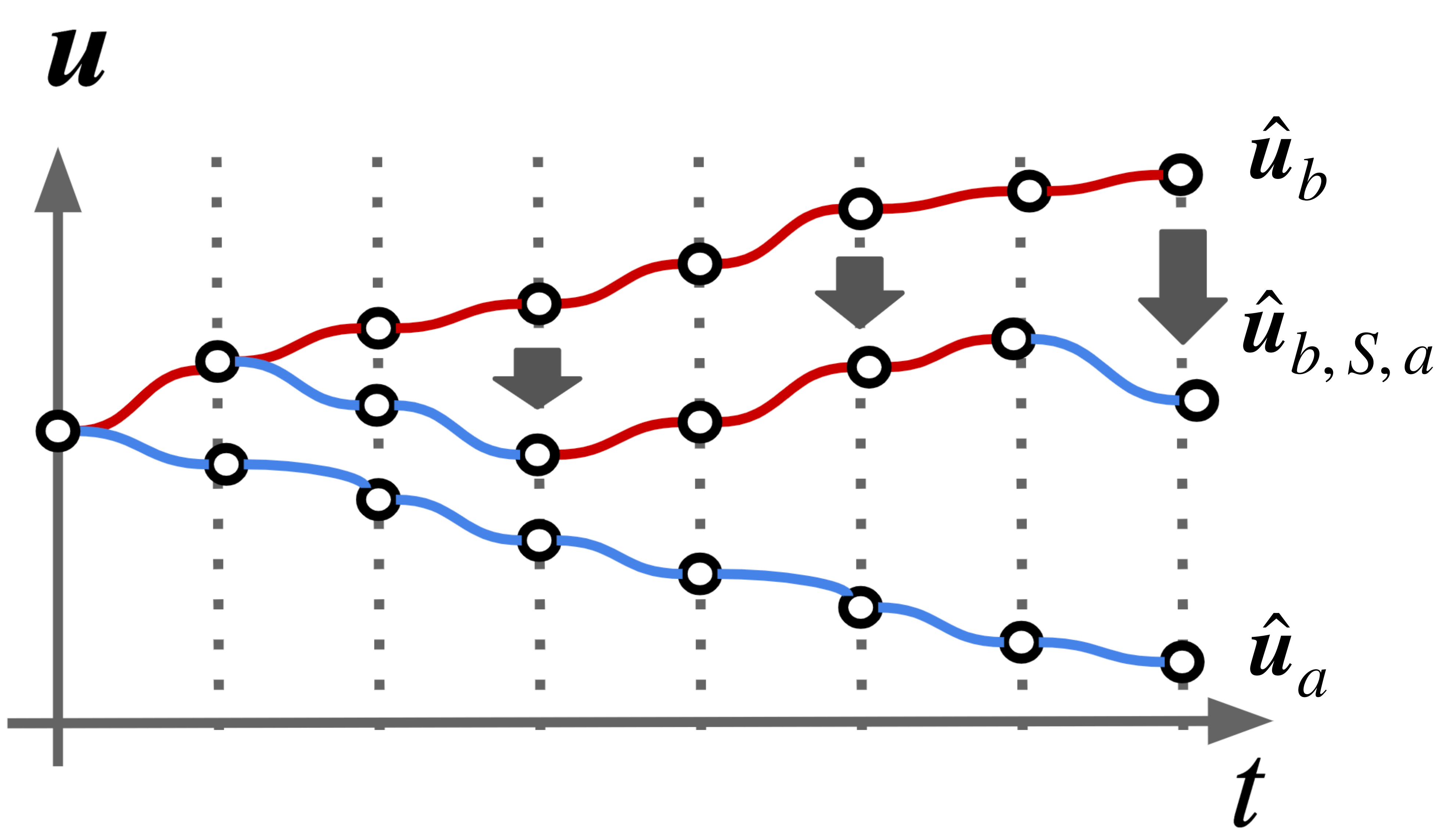


Illustration of $\hat{\boldsymbol{u}}_a$, $\hat{\boldsymbol{u}}_b$, and $\hat{\boldsymbol{u}}_{b,S,a}$. Blue lines represent $\hat{G}_a$, red lines represent $\hat{G}_b$.

Method - STAP

Batch Acquisition Algorithm

A batch is a set $\{(\boldsymbol{u}_j^0, S_j)\}_{j=1}^N$ of pairs of initial condition $\boldsymbol{u}_j^0$ and sampling pattern S_j.

The cost of a batch is $\sum_{j=1}^N \|S_j\|$, the total number of true entries.

One possible optimization target is the sum of acquisition values $\sum_{j=1}^N a(\boldsymbol{u}_j^0, S_j)$ under the constraint $\sum_j \|S_j\| \leq B$. Greedy maximization of $a^*(\boldsymbol{u}_j^0, S_j) = \frac{a(\boldsymbol{u}_j^0, S_j)}{\|S_j\|}$ is near-optimal.[1]

[1] Harvey M Salkin and Cornelis A De Kluyver. "The knapsack problem: a survey". In: *Naval Research Logistics Quarterly* 22.1 (1975), pp. 127–144.

Method - STAP

Batch Acquisition Algorithm

Sum of individual acquisition values deviates from the true utility of a batch, harming diversity and representativeness.[1]

Moreover, optimizing $a^*(\boldsymbol{u}^0, S)$ over $\boldsymbol{u}^0$ and S is computationally expensive.

Therefore, we first select initial conditions $\boldsymbol{u}^0$ with a full-trajectory AL method that respects diversity and representativeness.

We then optimize cost-weighted value $a^*(\boldsymbol{u}^0, S)$ over S with fixed $\boldsymbol{u}^0$, using greedy optimization.

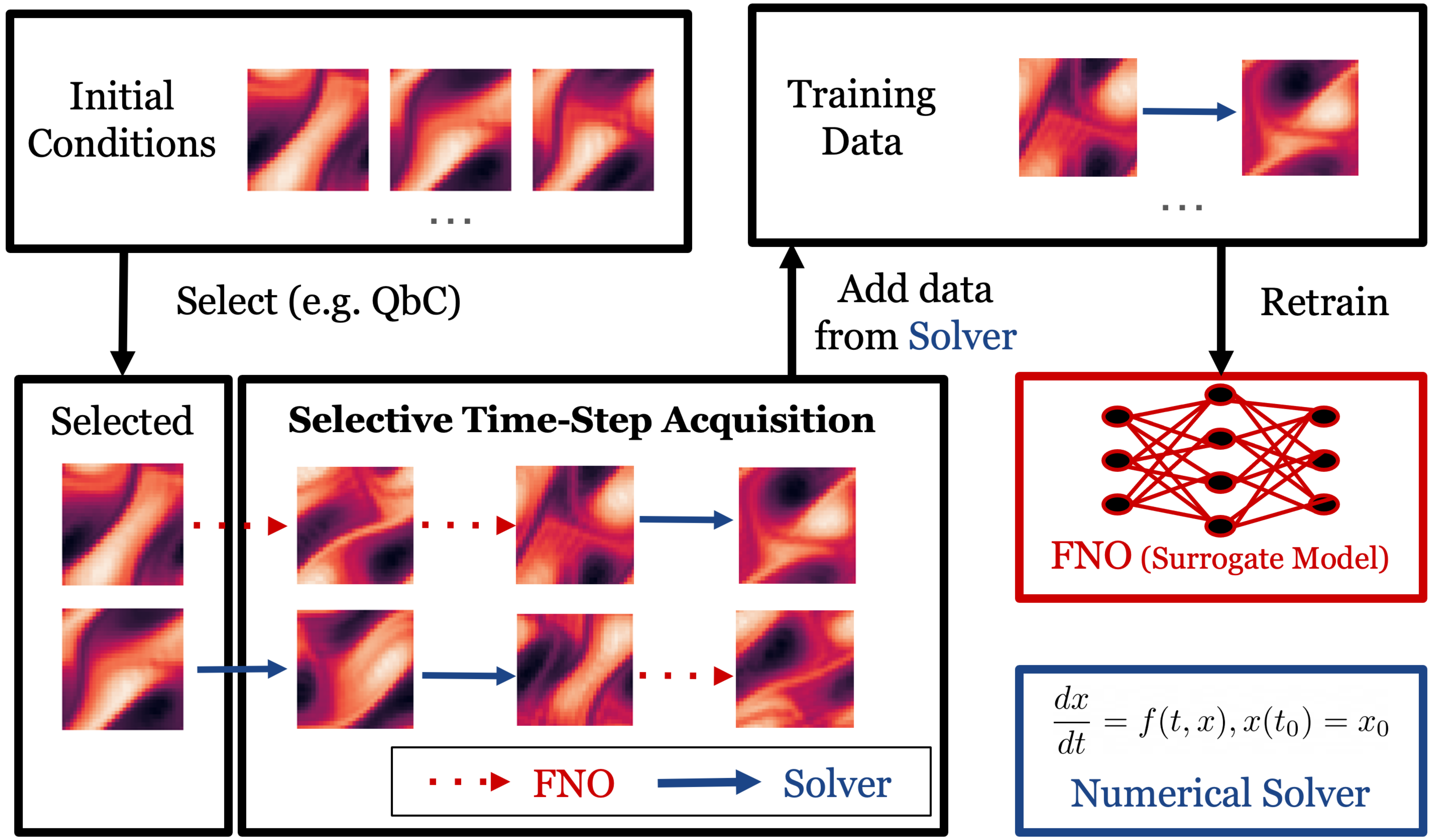
Initialize $S \leftarrow (\mathsf{true}, \ldots, \mathsf{true})$.
for $i = 1$ to T **do**
  $C = (C_1, \ldots, C_L)$ where $C_1, \ldots, C_L \overset{\text{i.i.d.}}{\sim} \mathrm{Ber}(\varepsilon)$.
  $S' = S \oplus C$
  if $a^*(\boldsymbol{u}^0, S') \geq a^*(\boldsymbol{u}^0, S)$ **then**
    $S \leftarrow S'$.
  end if
end for

Greedy optimization over S with bit-flip mutations.

[1] Andreas Kirsch, Joost Van Amersfoort, and Yarin Gal. "Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning". In: *Advances in neural information processing systems* 32 (2019).

Method - STAP

Illustration



Illustrated overview of STAP

(Algorithm 1 on page 7 provides an overview of STAP)

Experimental setup

Baseline AL Methods

We compare with the full trajectory AL methods in AL4PDE:

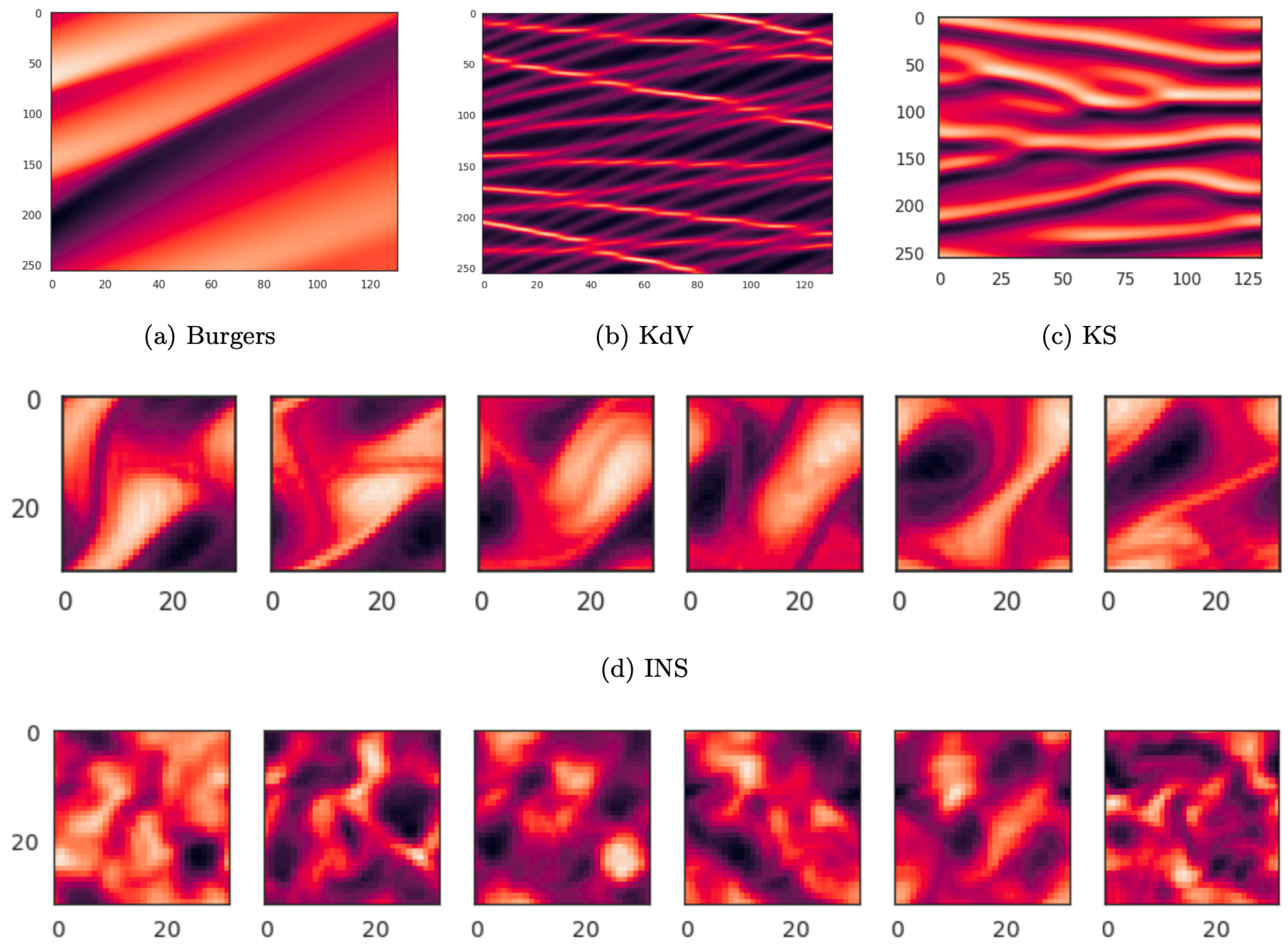
1. Query-by-Committee (QbC) [Seung et al. 1992]
2. Largest Cluster Maximum Distance (LCMD) [Holzmüller et al. 2023]
3. Stochastic Batch Active learning (SBAL) [Kirsch et al. 2023]

We can combine any of these base AL methods with STAP, e.g. SBAL+STAP

Experimental setup

Target PDEs

1. Burgers
2. Korteweg–De Vries (KdV)
3. Kuramoto–Sivashinsky (KS)
4. Incompressible Navier–Stokes (INS)
5. Compressible Navier–Stokes (CNS)



(a) Burgers (b) KdV (c) KS

(d) INS

(e) CNS

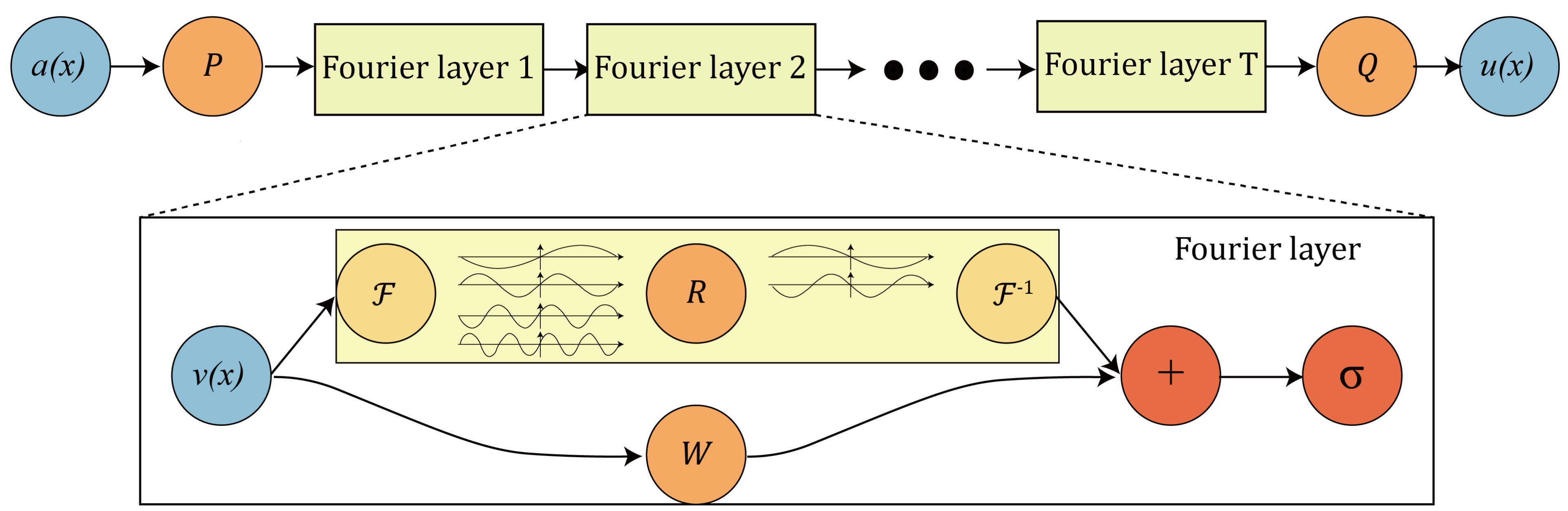
Experimental setup

Surrogate Models

Fourier Neural Operator (FNO) [Li et al. 2020] with 4 layers and 64 channels.

Trained from scratch at every round, with constant number of epochs. Trained with teacher forcing.

Committee size of $M = 2$, common in AL [Pickering et al. 2022; Musekamp et al. 2024].



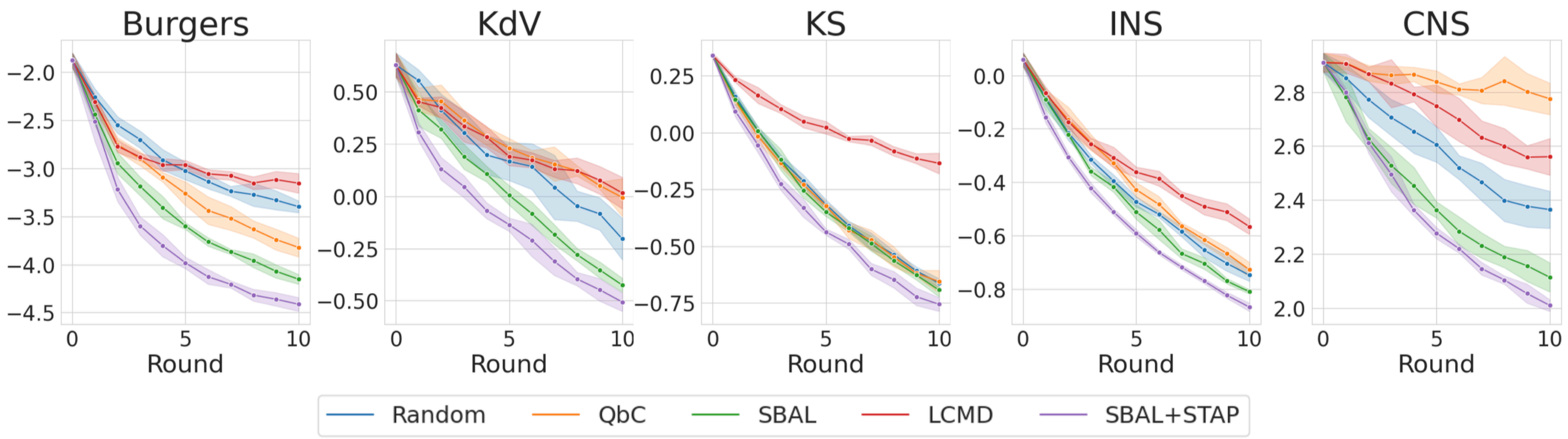
Architecture of Fourier Neural Operator (FNO)

Results and Discussion

Performance of Active Learning Methods

Pool of 10,000 initial conditions, initial training dataset of 32 full trajectories

10 rounds of acquisition, with constant budget $8 \times L$ at each round



Results and Discussion

Performance of Active Learning Methods

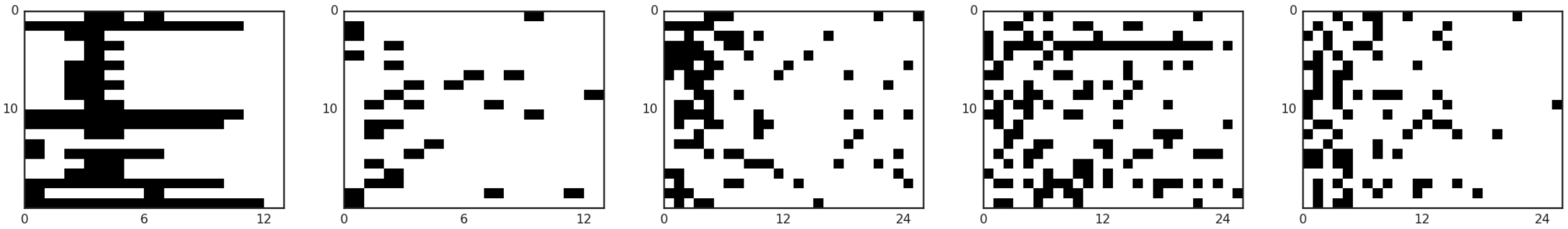
	Burgers	**KdV**	**KS**	**INS**	**CNS**
Random	-2.881±0.060	0.191±0.058	-0.258 ±0.003	-0.422±0.010	2.603±0.038
QbC	-3.121±0.065	0.266±0.027	-0.268 ±0.003	-0.385±0.014	2.844±0.021
LCMD	-2.847±0.027	0.256±0.030	0.046 ±0.013	-0.320±0.011	2.736±0.029
SBAL	-3.388±0.052	0.030±0.029	-0.275 ±0.014	-0.461±0.012	2.422±0.045
SBAL+STAP	**-3.674**±0.071	**-0.088**±0.040	**-0.349** ±0.003	**-0.525**±0.005	**2.363**±0.018

Table of log RMSE averaged across 10 rounds

Results and Discussion

Analysis of Timesteps Selected by STAP

STAP exhibits different patterns of timestep acquisition across tasks



(a) Burgers (b) KdV (c) KS (d) INS (e) CNS

Timesteps chosen by SBAL+STAP on the first AL round of each experiment.
Each row corresponds to an acquired trajectory, where the black cells indicate the selected time steps.

Results and Discussion

Analysis of Timesteps Selected by STAP

Does it matter which particular timesteps are chosen?

We test randomly selecting timesteps, with Ber(p).

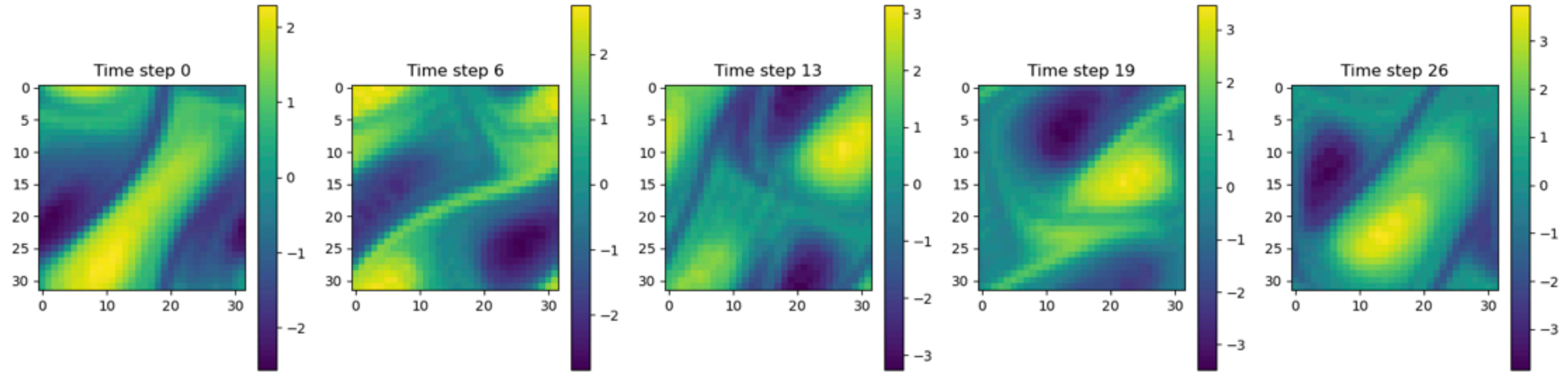
	Burgers	**KdV**	**KS**	**INS**	**CNS**
SBAL	-3.388$\pm$0.052	0.030$\pm$0.029	-0.275$\pm$0.014	-0.461$\pm$0.012	2.422$\pm$0.045
+STAP	**-3.674**$\pm$0.071	**-0.088**$\pm$0.040	-0.349$\pm$0.003	-0.525$\pm$0.005	**2.363**$\pm$0.018
+Ber(1/16)	-3.231$\pm$0.163	0.053$\pm$0.014	**-0.365**$\pm$0.008	**-0.529**$\pm$0.006	2.375$\pm$0.069
+Ber(1/8)	-3.152$\pm$0.267	0.049$\pm$0.014	-0.359$\pm$0.006	-0.525$\pm$0.006	2.370$\pm$0.018
+Ber(1/4)	-3.102$\pm$0.441	0.018$\pm$0.024	-0.346$\pm$0.008	-0.515$\pm$0.007	2.372$\pm$0.012
+Ber(1/2)	-3.458$\pm$0.067	-0.064$\pm$0.031	-0.324$\pm$0.007	-0.500$\pm$0.009	2.392$\pm$0.012

Log RMSE averaged across 10 rounds of AL.
STAP's frequencies were 0.35, 0.11, 0.19, 0.22, and 0.16, per task.

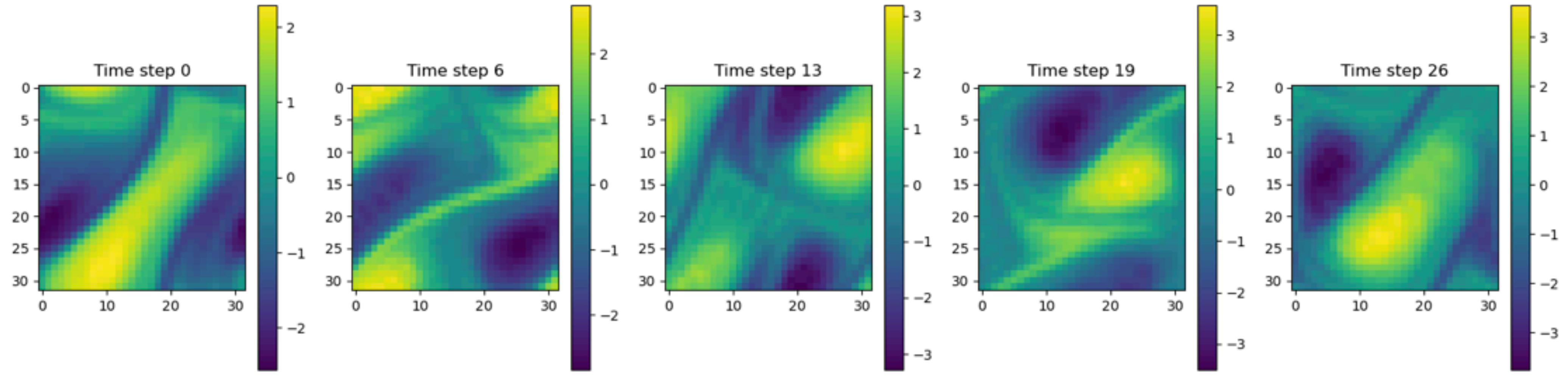
Results and Discussion

Out-of-distribution Synthetic Inputs

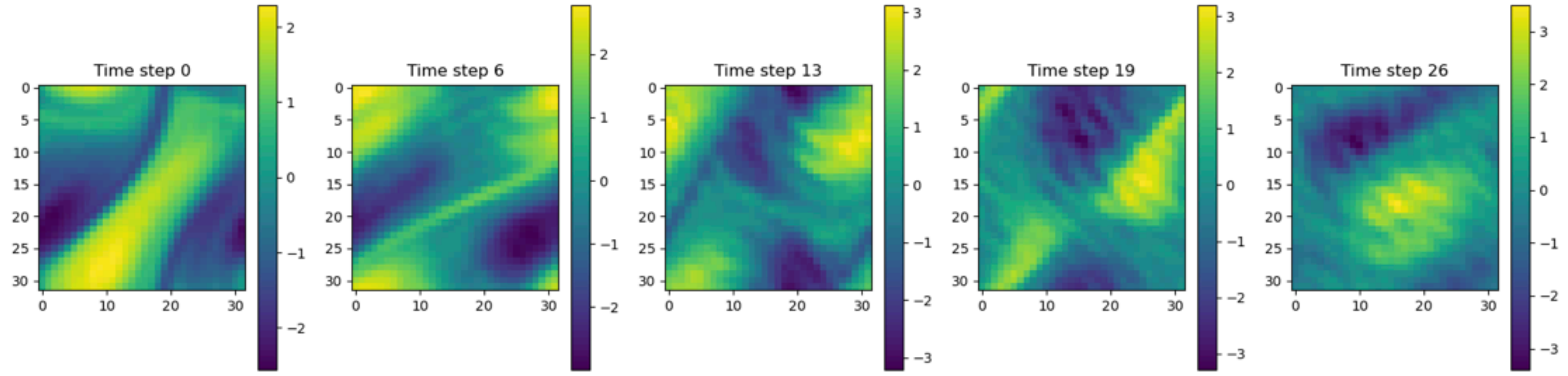
Inaccurate surrogate models might synthesize inputs that lie far from the ground truth data distribution, harming *representativeness*



Ground truth



Trained with 32 trajectories

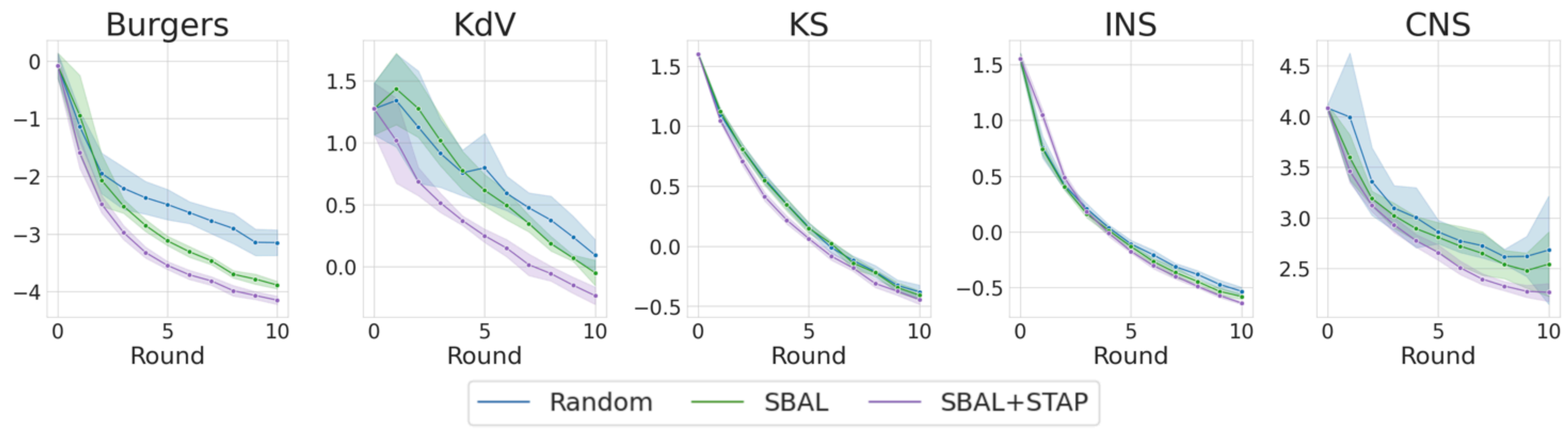


Trained with one trajectory

Results and Discussion

Out-of-distribution Synthetic Inputs

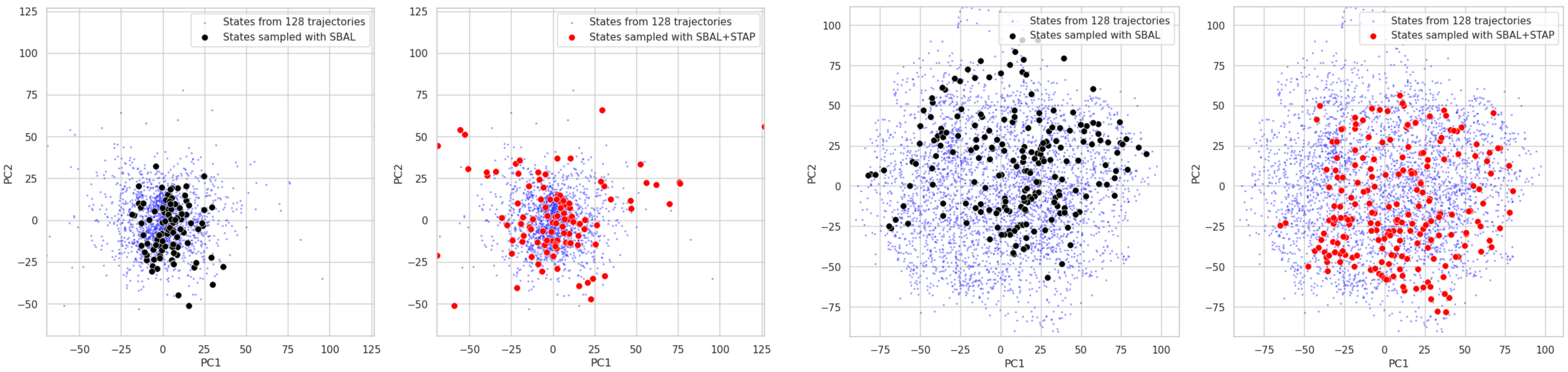
We stress test STAP by performing AL with initial dataset containing *one trajectory*, c.f. 32 trajectories in our main experiment



Results and Discussion

Out-of-distribution Synthetic Inputs

Explanation: Only a few synthetic inputs are severely out-of-distribution.



(a) KdV (b) KS

PCA of FNO activation for PDE states sampled by SBAL and SBAL+STAP during the first round of active learning.