SFU

BERKELEY LAB

UC Berkeley

INTERNATIONAL COMPUTER SCIENCE INSTITUTE

NEURAL INFORMATION PROCESSING SYSTEMS

Data-Efficient Operator Learning via Unsupervised Pretraining and In-Context Learning

Wuyang Chen*, Jialin Song*, Pu Ren,
Shashank Subramanian, Dmitriy Morozov, Michael W. Mahoney

Motivation

SciML is **short of "labeled" data**

1.7M snapshots in total across 4 benchmarks: *FNO, PDEBench, PDEArena, CFDBench*

Compared to *Computer Vision*

- **14M** images (256x256) from *ImageNet*
- **300M** images from *JFT-300M*

Generation/Collection of *SciML* data **has high cost**

Simulations

0.15s for one 512x512 N.S. snapshot (A100) ⇒ **~1.5 GPU days for 1M** snapshots with temporal dynamics

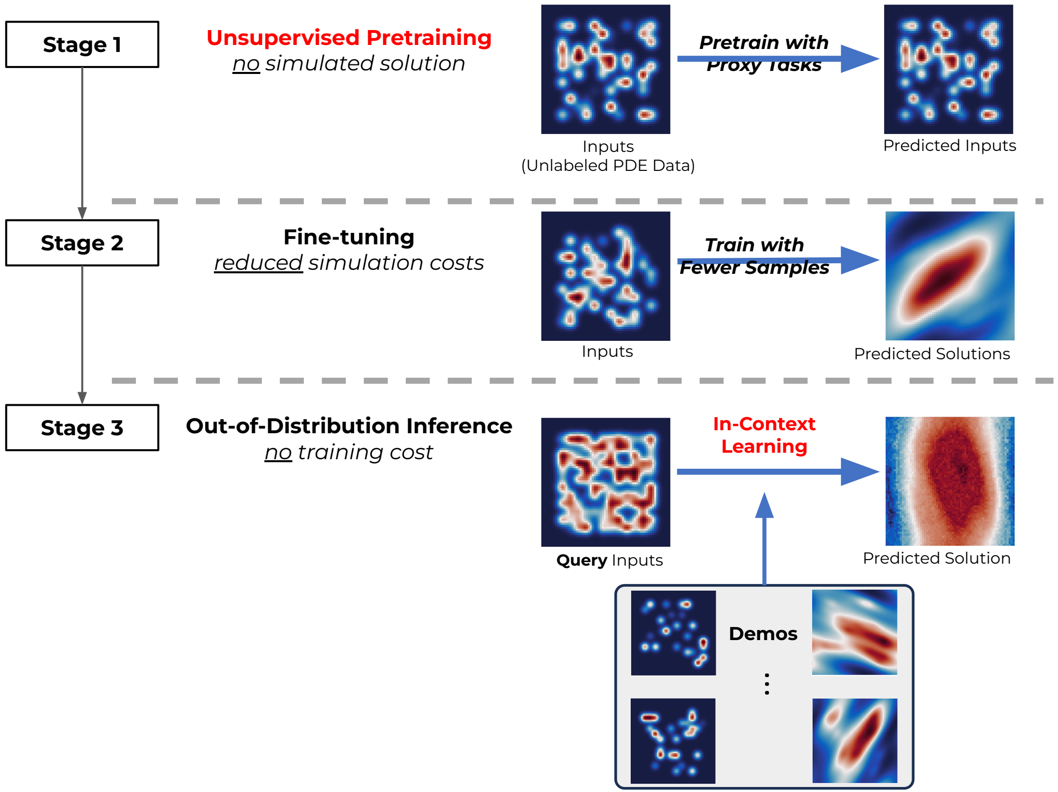
Observations

ERA5: 14600 snapshots for **10 years**

Success of Unsupervised pretraining in other domains

- NLP & CV: **Large-scale** unsupervised pretraining
  - With next/masked-token prediction; contrastive + masked-autoencoder
  - Fine-tune on downstream tasks
- SciML:
  - Multi-physics pretraining; Contrastive/Augmentations (e.g. Lie transform)
  - All happen in the **solution** space

Pipeline


Stage 1
Unsupervised Pretraining
no simulated solution
Pretrain with Proxy Tasks
Inputs (Unlabeled PDE Data)
Predicted Inputs
Stage 2
Fine-tuning
reduced simulation costs
Train with Fewer Samples
Inputs
Predicted Solutions
Stage 3
Out-of-Distribution Inference
no training cost
In-Context Learning
Query Inputs
Predicted Solution
Demos

Stage 1 & 2: Unsupervised Pretraining

Unlabeled PDE Data

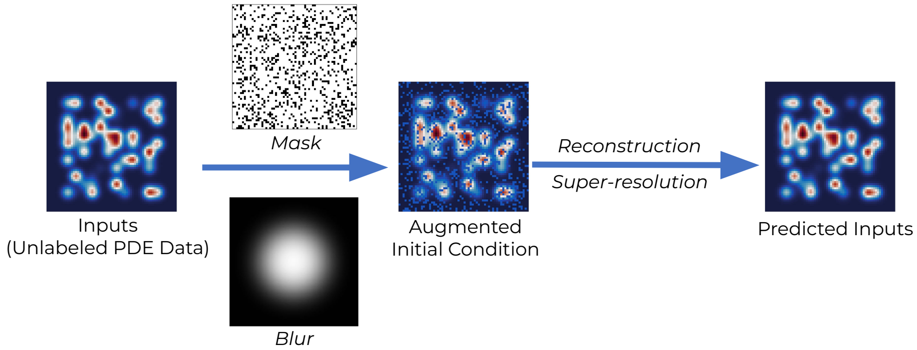
$$\sum_{i,j=1}^{n} a_{ij}(x) u_{x_i x_j} + \sum_{i=1}^{n} b_i(x) u_{x_i} + c(x) u = f(x)$$

physical space: $x \in \mathbb{R}^n$, *target solution:* u

- Time-independent equations
  - *PDE coefficients:* a_{ij}, b_i, c
  - *Forcing functions:* f
  - Coordinates
- Time-dependent equations
  - *Initial snapshot:* $u_0(x)$

Cheap generation, low simulation costs

Proxy Tasks



- Masked Autoencoder
  - Apply randomly sampled mask with various ratios
  - Model should predict the removed content
- Super-resolution
  - Apply Gaussian filter to the input
  - Model should recover the fine-scale data

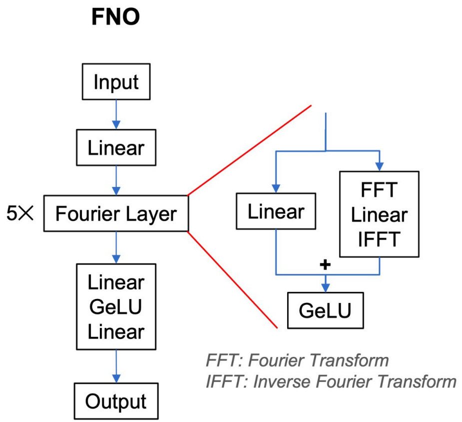
Sensor/Resolution-invariant, improve robustness

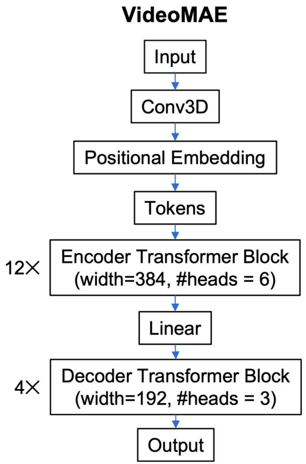
Stage 1 & 2: Unsupervised Pretraining

Settings

- **FNO** for **time-independent** PDEs
  - Pretrain:
    Encoder + Decoder
  - Fine-tune:
    Encoder
  - Input:
    coefficients, source functions, coordinates

- **FNO 3D/VideoMAE** for **time-dependent** PDEs
  - Pretrain on VideoMAE:
    Encoder + Decoder
  - Fine-tune on VideoMAE:
    Encoder + Decoder
  - Input:
    snapshots with no temporal dynamics

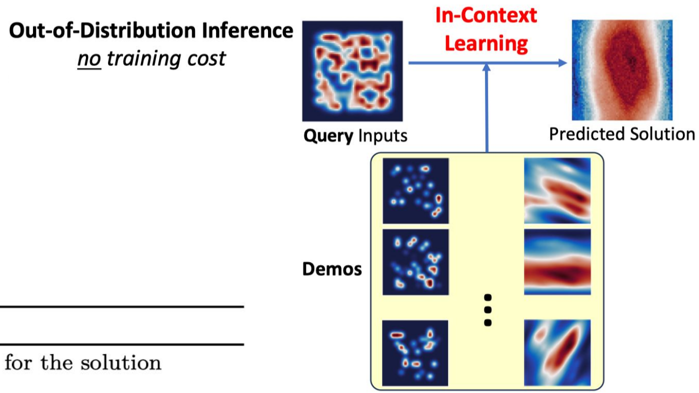




Stage 3: In-Context Learning

For Out-of-Distribution Generalization

- **No extra** training costs
- **Scalable** to any number of demos



Algorithm 1: Pseudocode of Test-time In-context Learning in a PyTorch-like style.

Data resolution: x-axis (W), y-axis (H), output temporal steps (T), output channel dimensions for the solution (C_{out}).

Input: Query input (x). Paired unlabeled PDE data (X) and solutions ($Y \in \mathbb{R}^{J\times H\times W\times T\times C_{out}}$) as J demos. Trained Neural Operator Model $\mathcal{M}$. TopK (k) demo solutions to aggregate.

$\hat{y} = \mathcal{M}(x)$ ▷ Shape: $H \times W \times T \times C_{out}$

$\hat{Y} = \mathcal{M}(X)$ ▷ Shape: $J \times H \times W \times T \times C_{out}$

$\hat{\delta} = \hat{y}.\text{reshape}(-1, 1, C_{out}) - \hat{Y}.\text{reshape}(1, -1, C_{out})$ ▷ Shape: $H \times W \times T \times (J \cdot H \cdot W \cdot T) \times C_{out}$

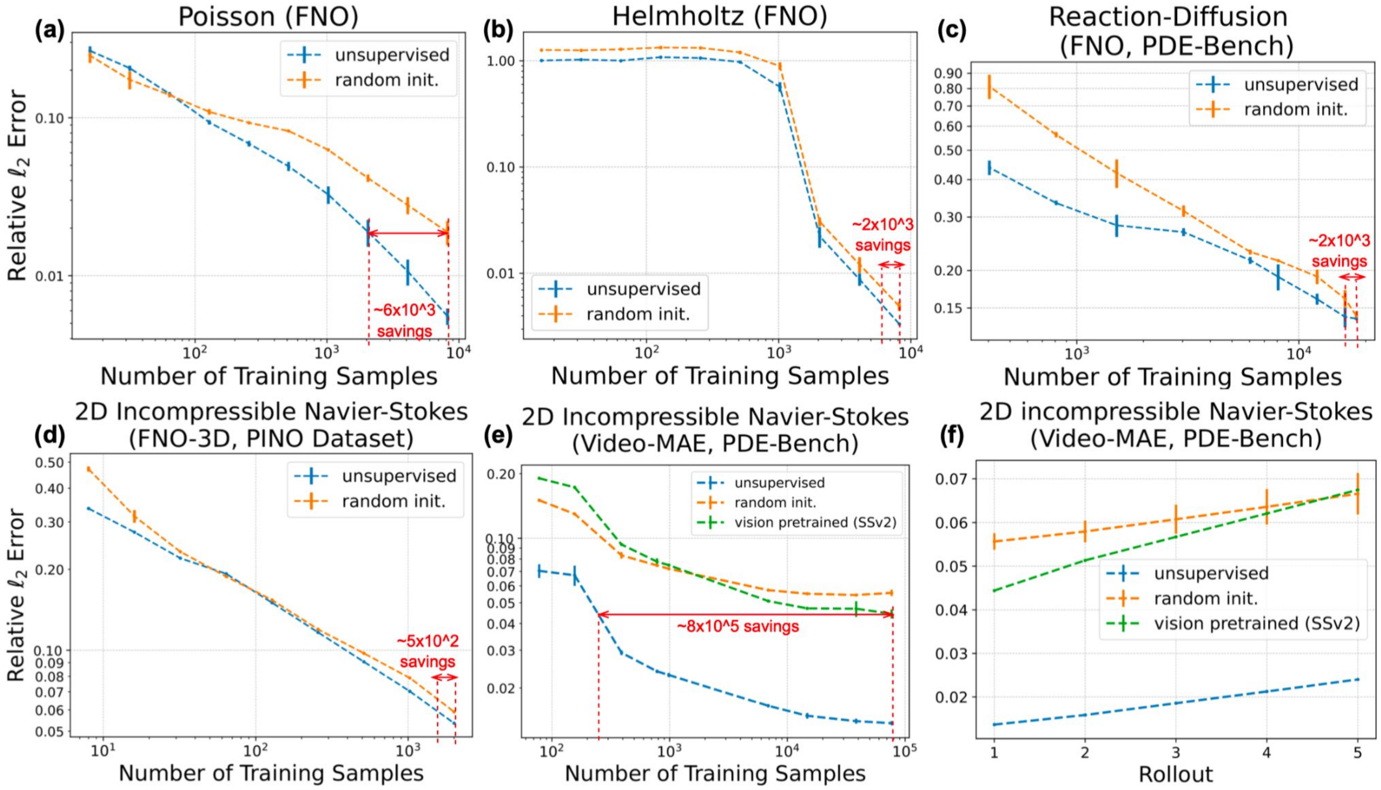
$\hat{\delta} = \text{absolute}(\hat{\delta}).\text{sum}(-1)$

index $= \text{argsort}(\hat{\delta}, -1)[:, :, :, :k]$ ▷ Spatial and temporal selection of demos similar to the query.

$\hat{y}_{icl} = \text{take\_along\_dim}(Y.\text{reshape}(-1, C_{out}), \text{index})$ ▷ Shape: $H \times W \times T \times C_{out} \times k$. Spatial and temporal aggregation of solutions from similar demos.

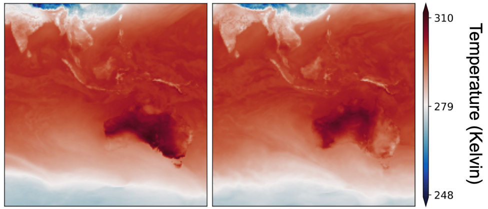
Return: $\hat{y}_{icl}.\text{mean}(-1)$

Results: Unsupervised Pretraining

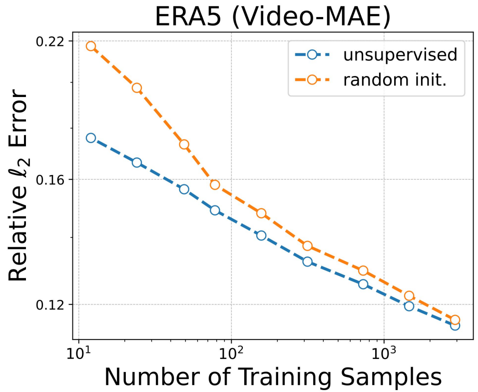


Results: Real-World Scientific Data

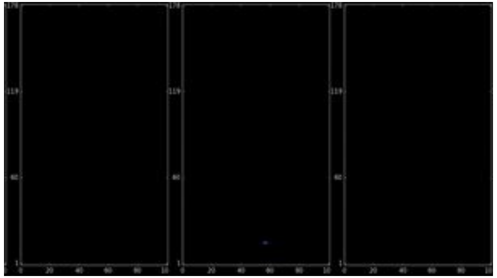
ERA5

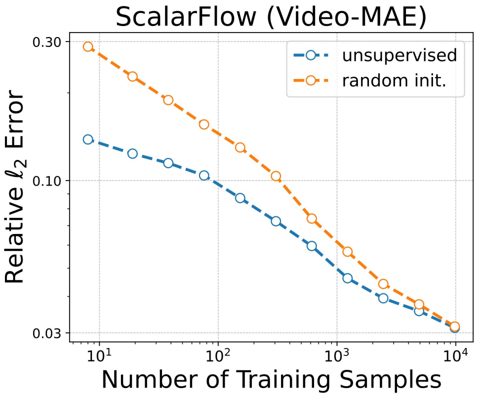


(a) t = 0 (b) t = 17

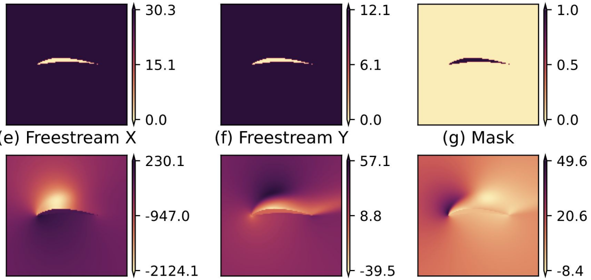


ScalarFlow



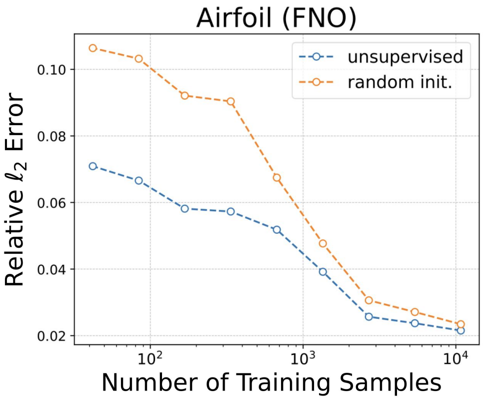


Airfoil

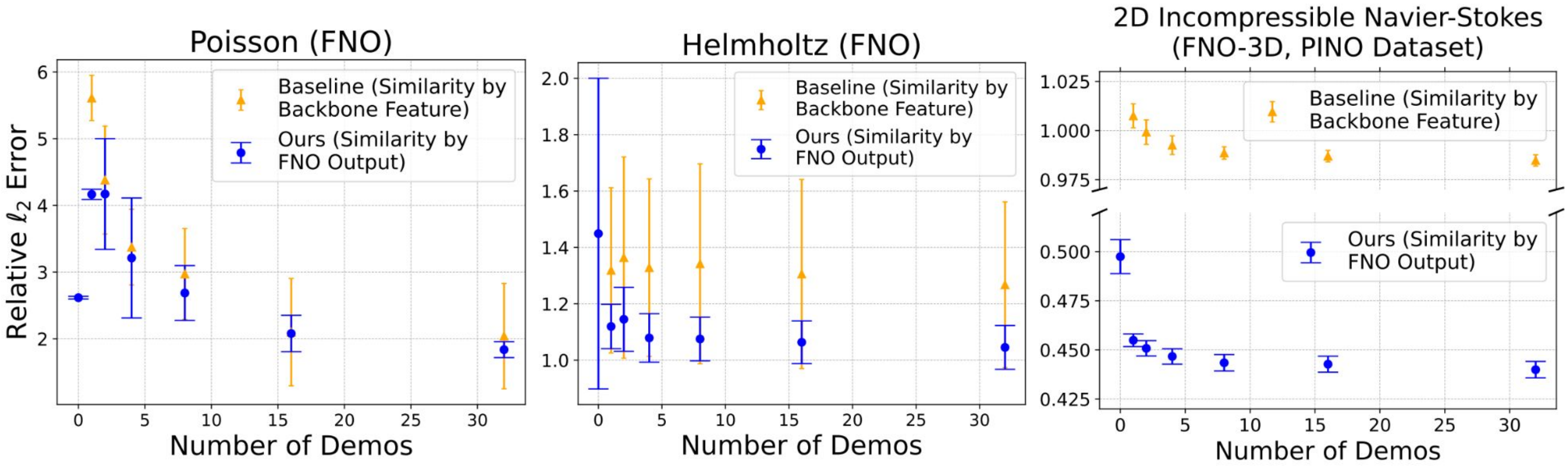


(e) Freestream X (f) Freestream Y (g) Mask

(h) Pressure (i) Velocity X (j) Velocity Y



Results: In-Context Learning



Physics Parameters	Poisson (diffusion)	Helmholtz (wave number)	Naiver Stokes (Reynolds number)
Pretraining	[1, 20]	[1, 20]	{100, 300, 500, 800, 1000}
Training (or Fine-tuning)	[5, 15]	[5, 15]	300
Out-of-Distribution Testing	[15, 50]	[15, 20]	10000

Thank you