AI in the Sciences and Engineering HS 2025: Lecture 12

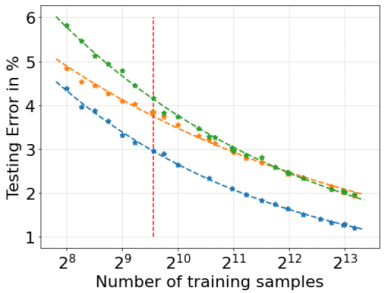
Siddhartha Mishra

Computational and Applied Mathematics Laboratory (CamLab)
Seminar for Applied Mathematics (SAM), D-MATH (and),
ETH AI Center (and) Swiss National AI Institute (SNAI) ,
ETH Zürich, Switzerland.

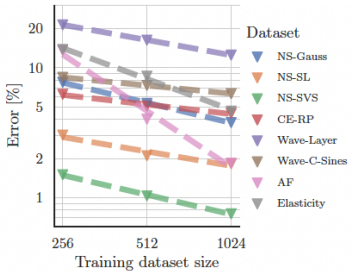
What we have learnt so far ?

- AIM: Learn PDEs using Deep Neural Networks
- Operator Learning: Learn the PDE Solution Operator from data.
- Examples: FNO, CNO, VIT, scOT etc.
- Very successful on PDEs on 2D Cartesian Domains !!
- Readily extended to time-dependent PDEs with all2all training.
- Extended to PDEs on Arbitrary domains with GNNs or GAOT.
- GenAI for PDEs with Chaotic multiscale solutions (GenCFD)

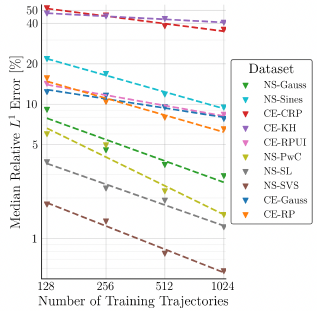
Where's the final CAVEAT ?



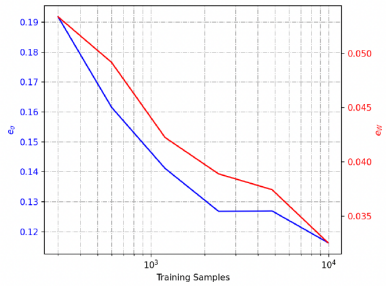
CNO/FNO



RIGNO



GAOT



GenCFD

- Models Scale with sample size: $\mathcal{E} \sim N^{-\alpha}$ but with α small
- Even more pessimistic rates with theory [1]
- ML models require Big Data: $\mathcal{O}(10^3) - \mathcal{O}(10^4)$ training samples per Task
- Very Difficult to obtain Data for PDEs.
- How to make models more Data Efficient ?

[1] Lanthaler, SM, Karniadakis, 2022, De Ryck, SM, 2023

Sample Complexity Solution I

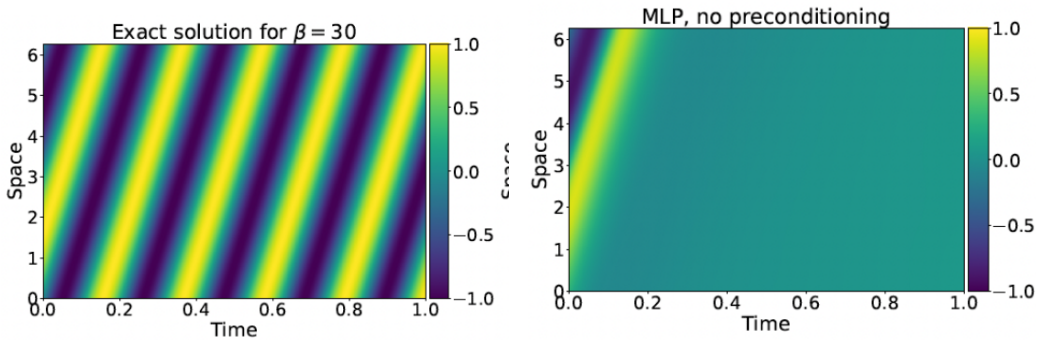
- Add PDE Residuals to loss functions.
- Basis of Physics Informed Neural Operators
- Given Abstract PDE $\mathcal{L}(u) = f$.
- Solution Operator: $\mathcal{S} : \mathcal{X} \mapsto \mathcal{Y}$, $\mathcal{S}(f) = u$
- Underlying Data Distribution: $\mu \in \mathrm{Prob}(\mathcal{X})$.
- Objective: Learn $\mathcal{S}_{\#\mu}$, given Data $(f_i, \mathcal{S}(f_i))_{i=1}^N$, $f_i \sim \mu$ + PDE
- Neural Operator (FNO, CNO etc): $\mathcal{S}^* : \mathcal{X} \mapsto \mathcal{X}$ with $\mathcal{S}^* = \mathcal{S}_{\theta^*}$

$$\theta^* = \mathrm{argmin}_{\theta \in \Theta \subset \mathbb{R}^M} \sum_{i=1}^{N} [\|\mathcal{S}(f_i) - \mathcal{S}_\theta(f_i)\|_{\mathcal{Y}} + \lambda \|\mathcal{L}(\mathcal{S}_\theta(f_i)) - f_i\|_Z]$$

Does it Work ?

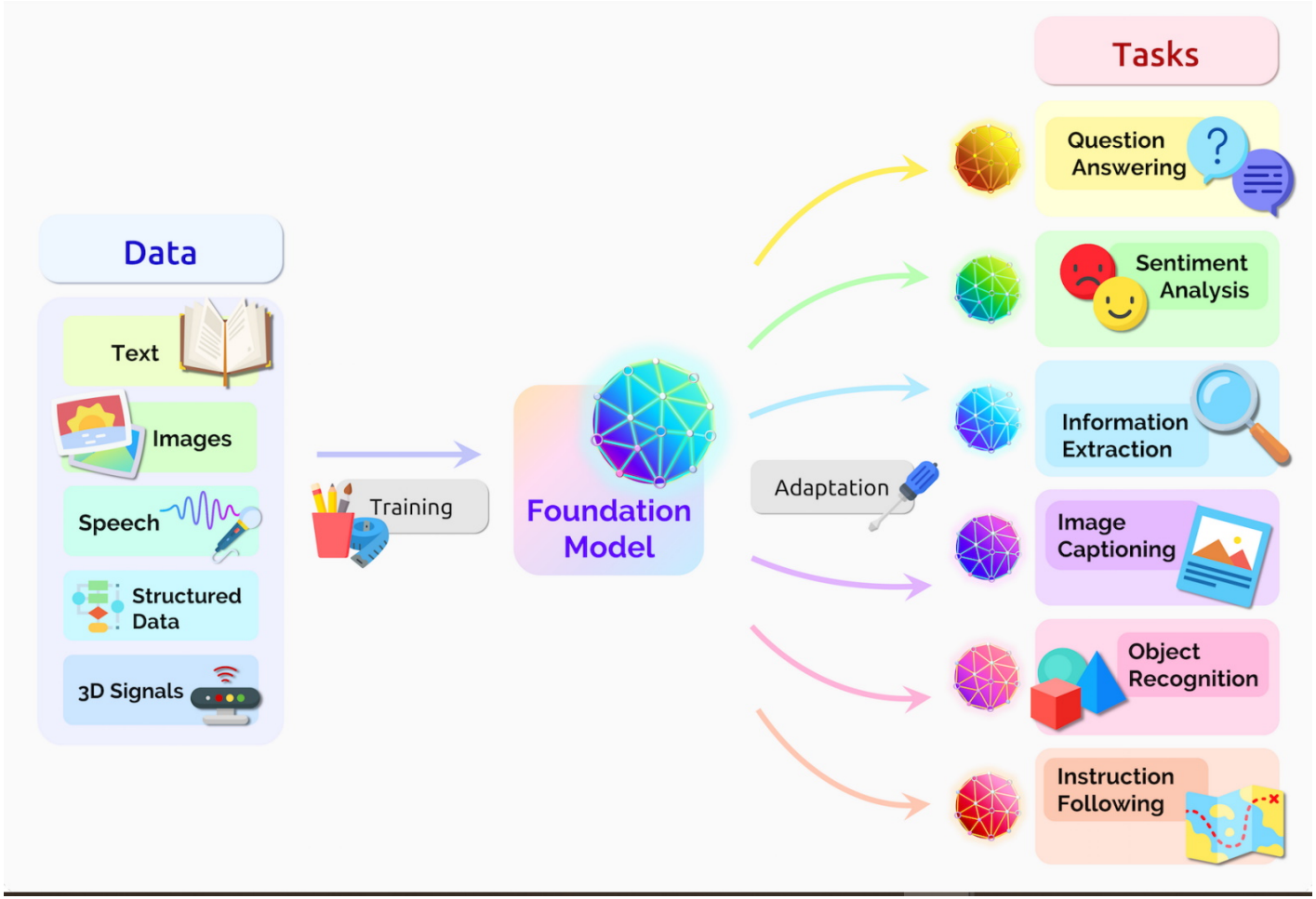
- Not even for Simple problems !!
- For Linear Advection:

$$u_t + \beta u_x = 0, \quad \beta = 30$$

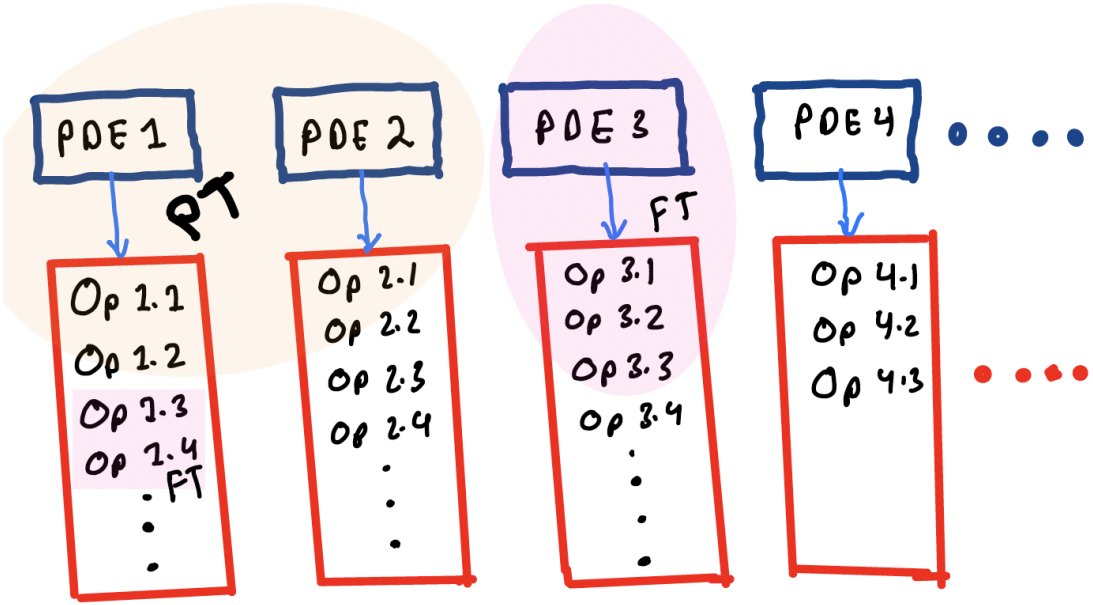


How do modern AI systems deal with it ?

Foundation Models are the Key !!

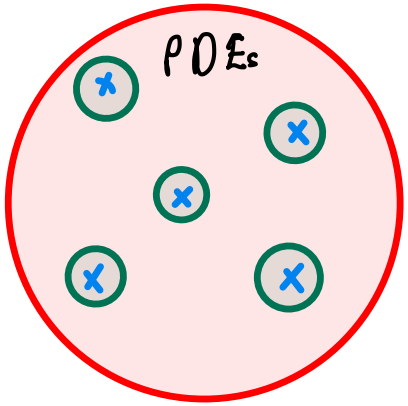


What would a Foundation Model for PDEs look like ?

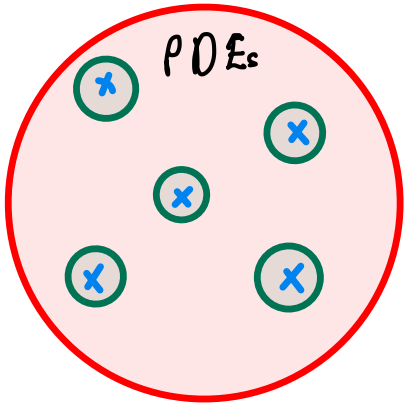


- Op: Operators need PDE + Data.
- PT: Pretraining.
- FT: Finetuning (Adaptation)

Can it Work ?

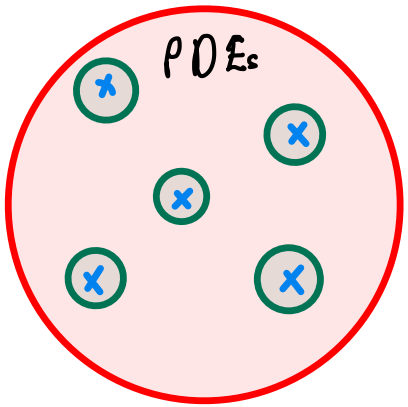


Can it Work ?

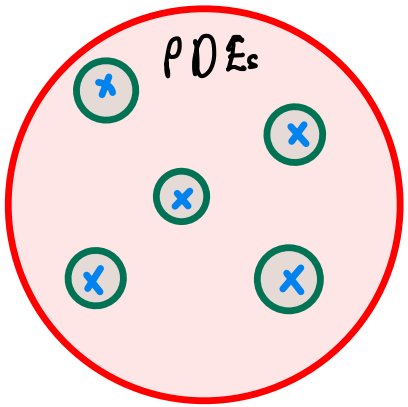


why pretraining a model on a (very) small set of PDEs and underlying data-distributions can allow it to learn effective representations and generalize to unseen and unrelated PDEs and data-distributions via finetuning?

Can it Work ?

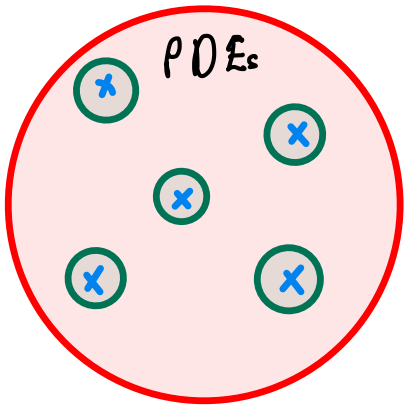


Can it Work ?



- Lets try it out !!

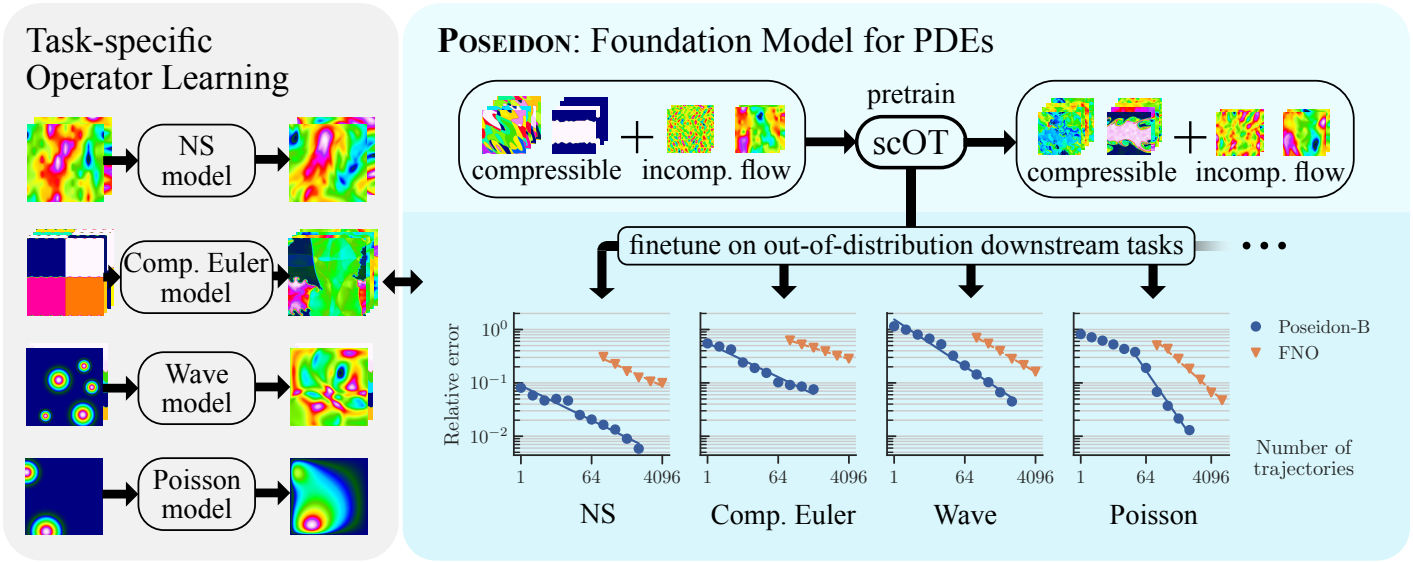
Can it Work ?



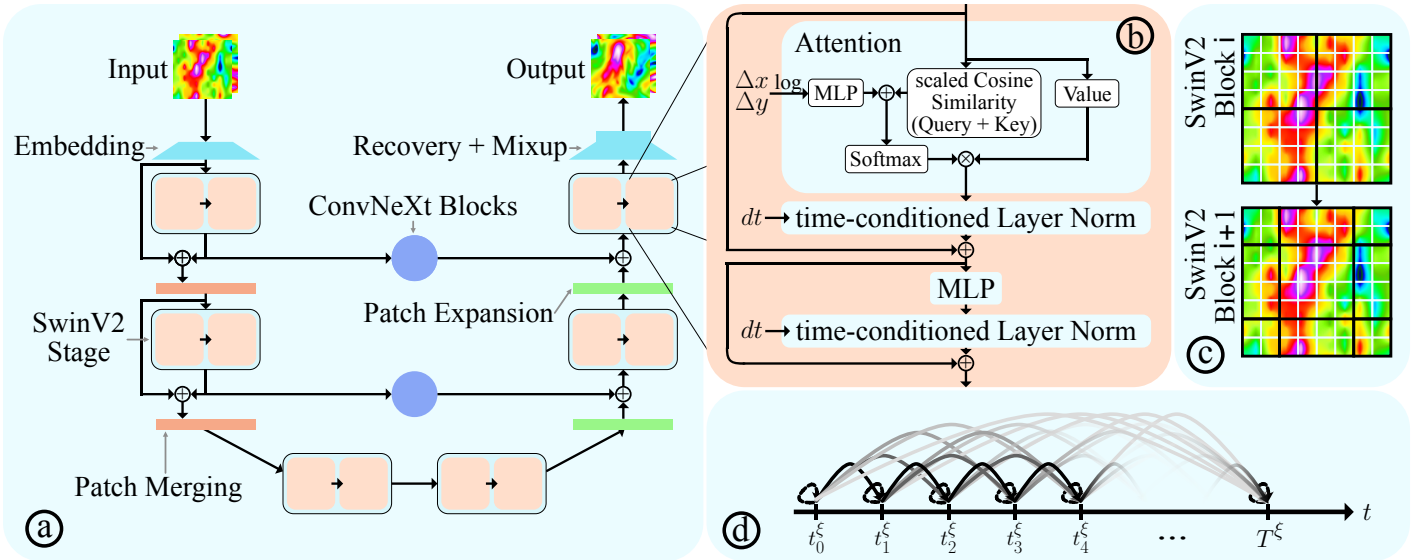
- Lets try it out !!

POSEIDON

- PDE foundation model of Herde, SM et. al, 2024
- Paper + Code: https://github.com/camlab-ethz/poseidon
- Model + Datasets: https://huggingface.co/camlab-ethz



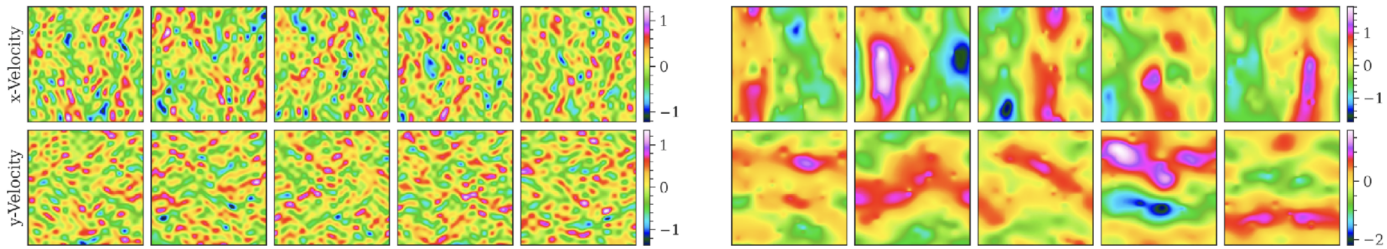
Model Backbone: scOT with all2all training



Pretraining Data I

Pretraining Data

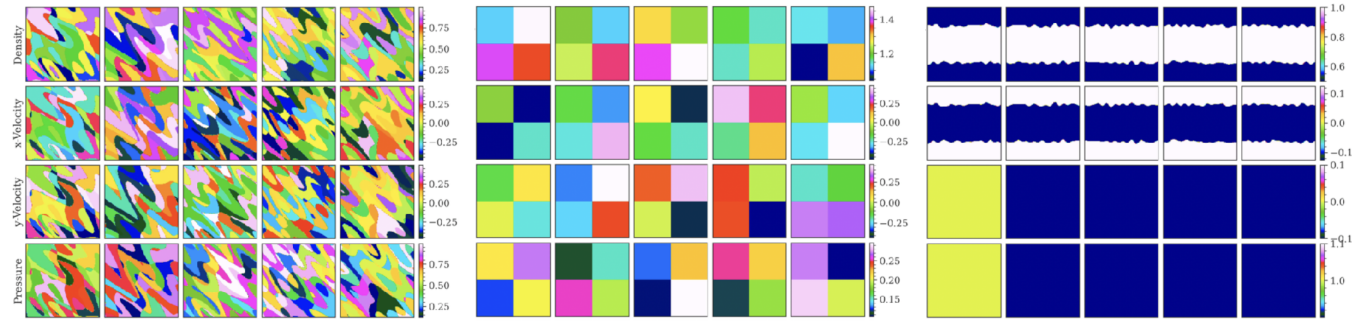
from incompressible Navier-Stokes eqns

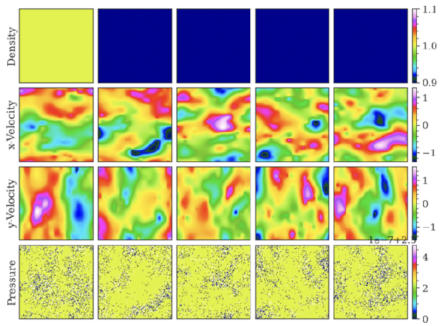


$$u_t + (u \cdot \nabla)u + \nabla p = \nu \Delta u, \quad \text{div } u = 0$$

- Each Dataset with 19 K trajectories.

Pretraining Data II





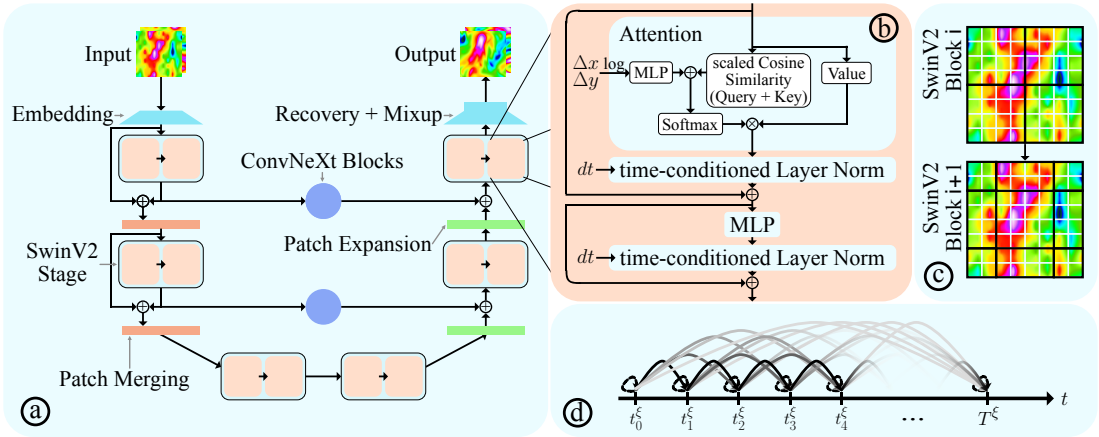
Pretraining Data

from compressible Euler eqns

$$u_t + \text{div}\, F(u) = 0,\ u = [\rho, \rho v, E]^\perp,\ F = [\rho v, \rho v \otimes v + p\mathbf{I}, (E+p)]v]^\perp$$

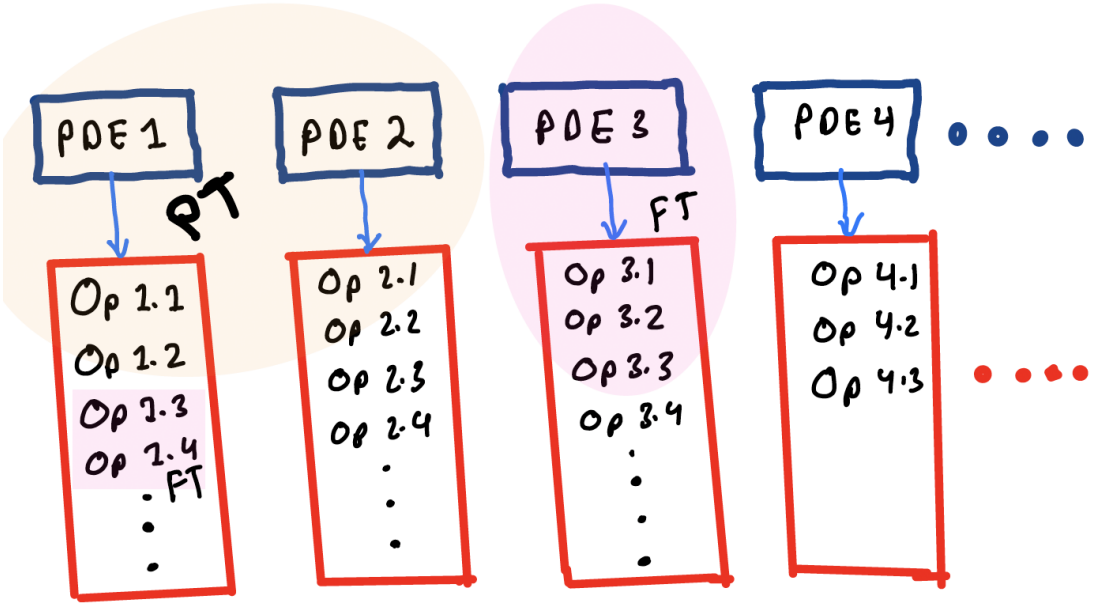
- Each Dataset with 9.5 K trajectories.

PreTraining Details



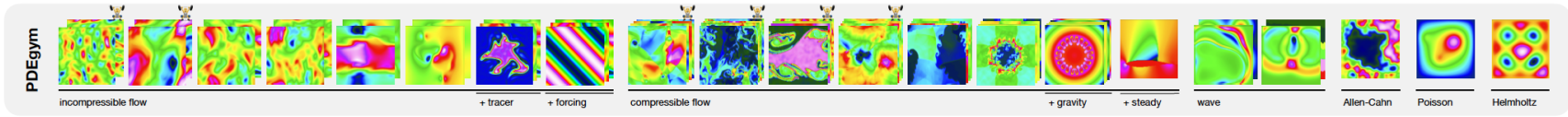
- Total of 5.11 M Images (due to all2all training)
  - Poseidon-T with 21 M parameters.
  - Poseidon-B with 158 M parameters.
  - Poseidon-L with 629 M parameters.
- Pretraining Compute: on 8 NVIDIA RTX-4090s
  - Poseidon-T: 22 hrs.
  - Poseidon-B: 118 hrs.
  - Poseidon-L: 165 hrs.
  - CNO-FM (109 M): 178 hrs.
- Pretraining Compute for Poseidon-XL (2.5 B parameters)
- on ALPS (NVIDIA-GH100): 194 hrs.

What would a Foundation Model for PDEs look like ?



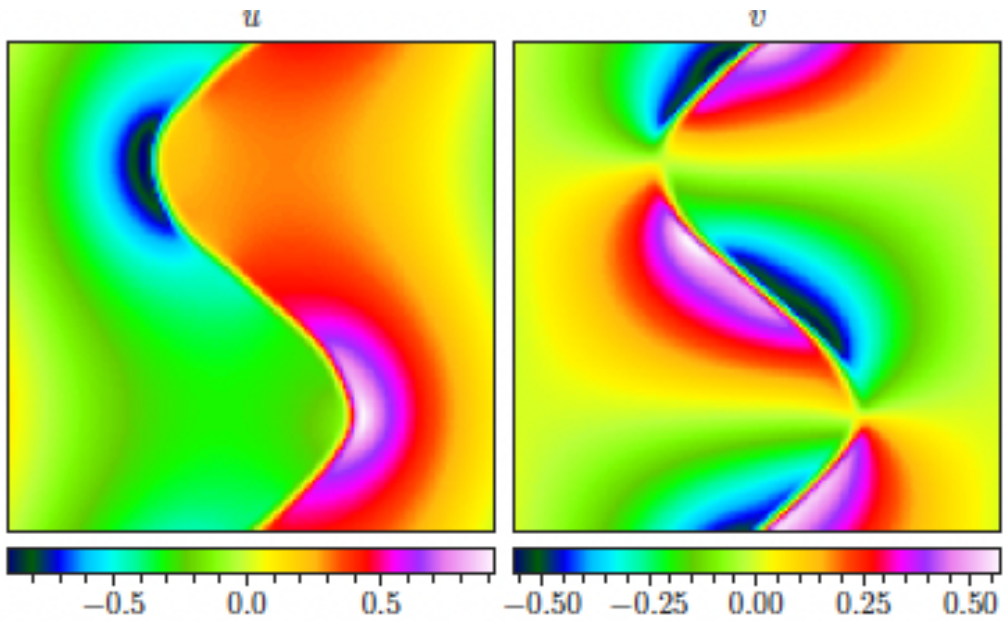
- Op: Operators need PDE + Data.
- PT: Pretraining.
- FT: Finetuning (Adaptation)

15 Downstream Tasks from PDEgym

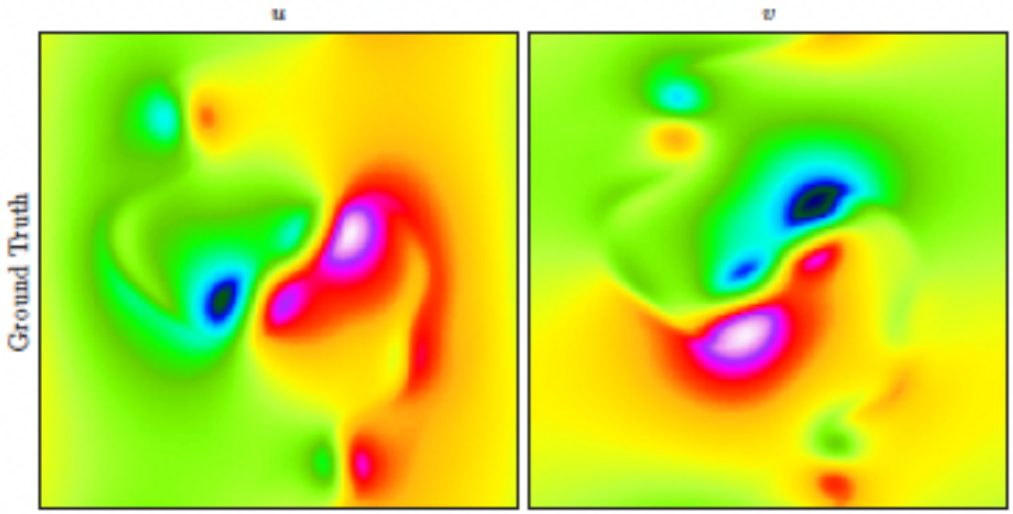


Abbreviation	PDE	Defining Feature	Visualization
NS-PwC	Navier-Stokes (B.1)	Piecewise constant vorticity IC	Fig. 56
NS-BB	Navier-Stokes (B.1)	Brownian Bridge IC	Fig. 57
NS-SL	Navier-Stokes (B.1)	Shear Layer IC	Fig. 58
NS-SVS	Navier-Stokes (B.1)	Sine Vortex sheet IC	Fig. 59
NS-Tracer-PwC	Navier-Stokes + Transport (B.21)	Scalar Advection	Fig. 60
FNS-KF	Forced Navier-Stokes (B.23)	Kolmogorov Flow	Fig. 61
CE-RPUI	Euler (B.7)	RP with uncertain interfaces	Fig. 62
CE-RM	Euler (B.7)	Richtmeyer-Meshkov	Fig. 63
GCE-RT	Euler+Gravity (B.27)	Rayleigh-Taylor	Fig. 64
Wave-Gauss	Wave Eqn. (B.34)	Waves in Gaussian medium	Fig. 65
Wave-Layer	Wave Eqn (B.34)	Waves in layered medium	Fig. 66
ACE	Allen-Cahn Eqn. (B.37)	Reaction-Diffusion	Fig. 67
SE-AF	steady state of Euler (B.7)	Flow past airfoil	Fig. 68
Poisson-Gauss	Poisson Eqn. (B.38)	Stationary diffusion	Fig. 69
Helmholtz	Helmholtz Eqn (B.39)	Waves in frequency domain	Fig. 70

Downstream Task: NS-SVS

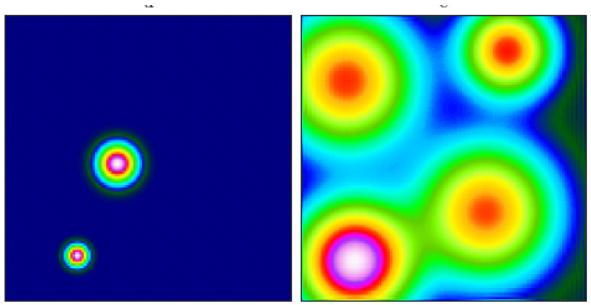


(a) Inputs: horizontal velocity u and vertical velocity v.

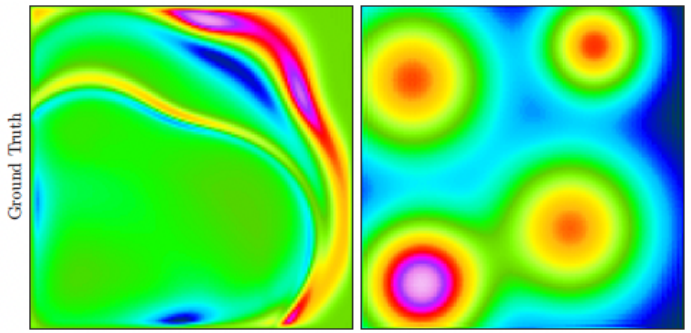


Downstream Task: Wave-Gauss

- Input ($T = 0$):

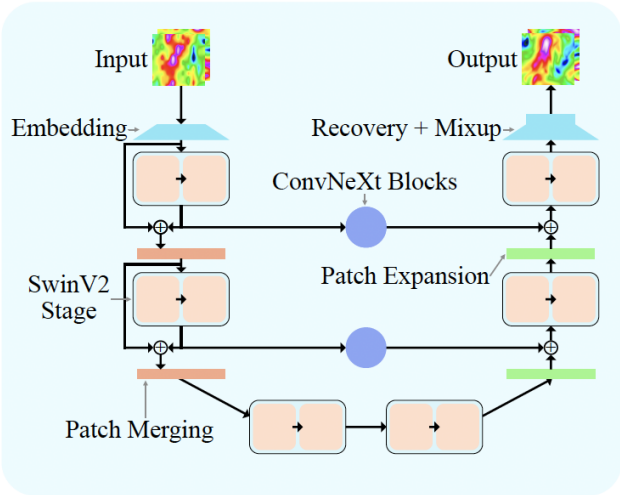


- Output t ($T = 1$):



Fine-Tuning

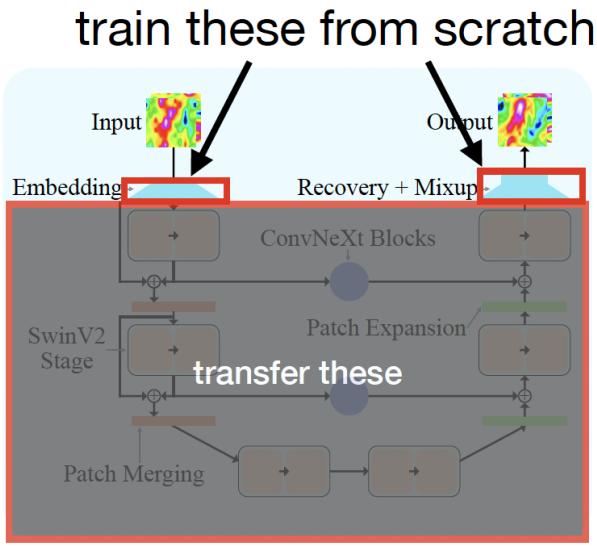
Native Usage



Transfer All

Small Learning rate

Non-Native Usage



Transfer Latent

Relearn Embeddings

Large Learning rate

Finetuning Poseidon

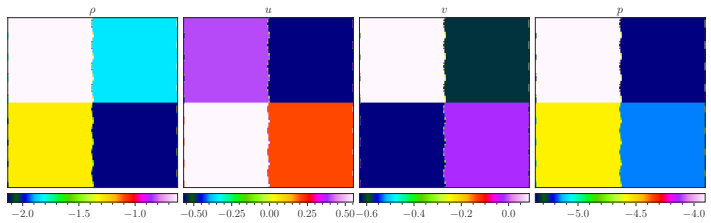
- Denote Pretrained Model as Π_{θ^*}
- Downstream Task: Learn Solution Operator $\mathcal{S}$ from Data.
- Denote trainable parameters by $\theta = [\hat{\theta}, \bar{\theta}]$
- With $\dim(\bar{\theta}) << \dim(\hat{\theta})$
- Gradient Descent step:

$$[\hat{\theta}_{r+1}, \bar{\theta}_{r+1}] = [\hat{\theta}_r, \bar{\theta}_r] - [\hat{\eta}_r, \bar{\eta}_r]\nabla_\theta \mathcal{L}(\theta_r), \quad r > 0,$$
$$\hat{\theta}_0 = \hat{\theta}^*, \quad \bar{\theta}_0 \sim \bar{P},$$

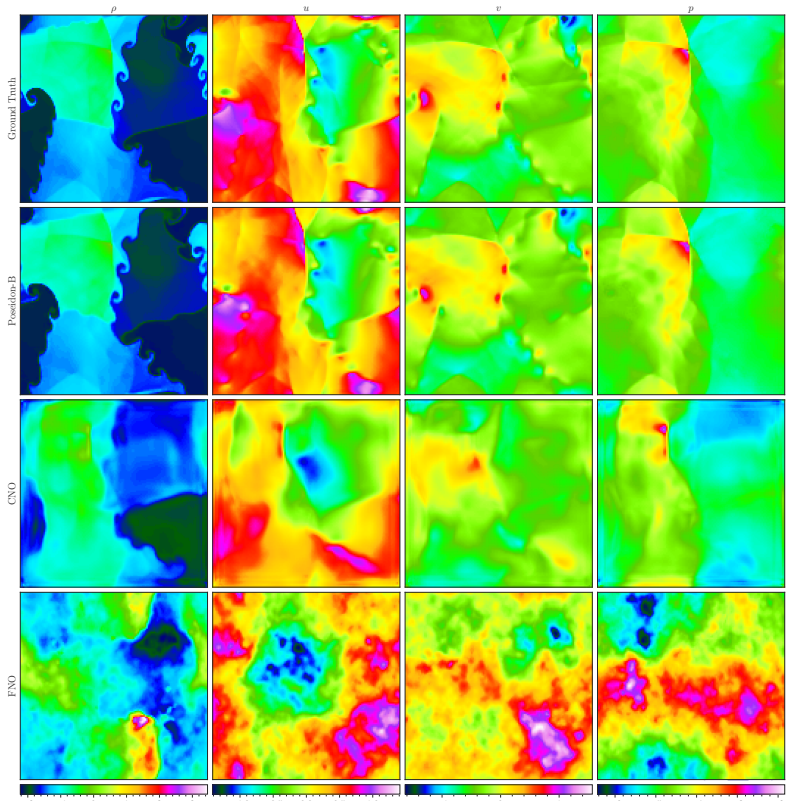
- Use maximum normalized lead times for time-independent PDEs.

Results: CE-RPUI

- Input (Initial Data):



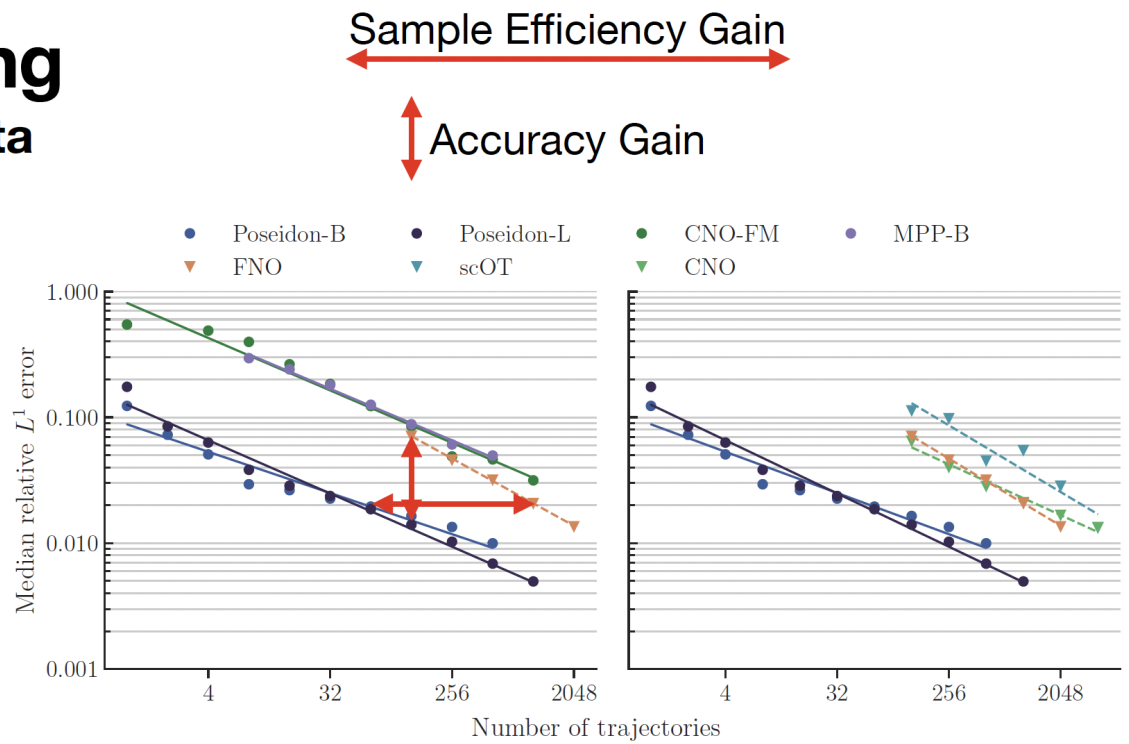
- Output (Solution at Final Time):



Performance Metrics

Scaling

w.r.t. data

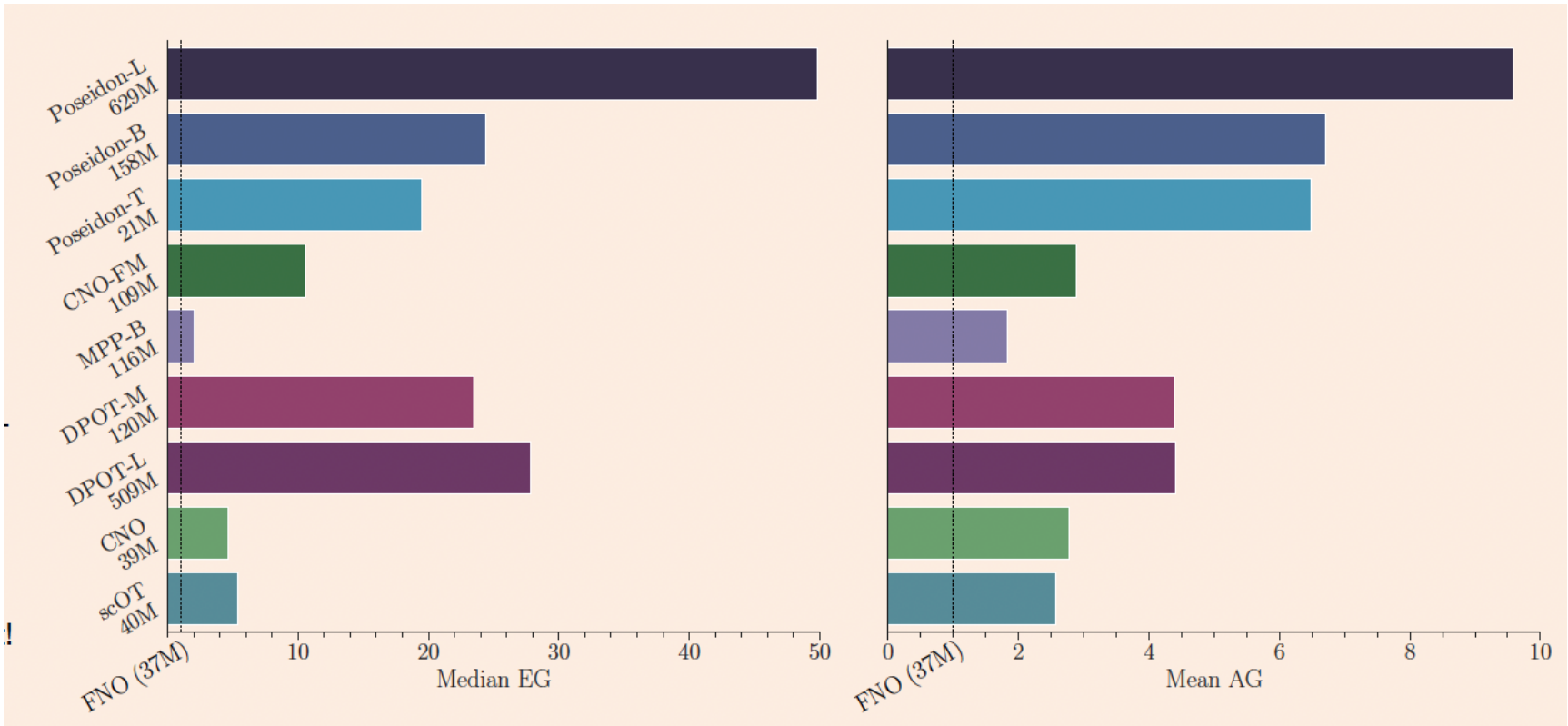


incompressible Navier-Stokes — Shear Layer

Sample Efficiency and Accuracy

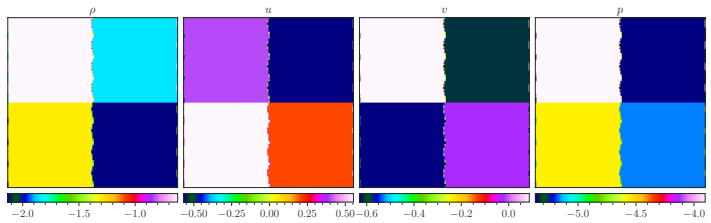
	Pretrained Models						Models trained from Scratch					
	Poseidon-L		CNO-FM		MPP-B		CNO		scOT		FNO	
	EG	*AG*	EG	*AG*	EG	*AG*	EG	*AG*	EG	*AG*	EG	*AG*
NS-PwC	**890.6**	***24.7***	16.6	*3.3*	7.4	*2.3*	3.7	*1.5*	5.4	*2.0*	1	*1*
NS-SVS	**502.9**	***7.3***	59.6	*3.1*	34.8	*2.2*	73.2	*3.4*	10.2	*1.2*	1	*1*
NS-BB	**552.5**	***29.3***	10.6	*3.9*	4.6	*2.6*	2.7	*1.7*	3.4	*2.1*	1	*1*
NS-SL	**21.9**	***5.5***	0.4	*0.8*	0.3	*0.8*	0.8	*1.2*	0.4	*0.5*	1	*1*
NS-Tracer-PwC	**49.8**	***8.7***	17.8	*3.6*	8.5	*2.7*	4.6	*1.9*	4.6	*1.9*	1	*1*
FNS-KF	**62.5**	***7.4***	13.2	*2.7*	2.0	*1.6*	3.1	*1.5*	3.3	*0.9*	1	*1*
CE-RPUI	**352.2**	***6.5***	33.2	*2.3*	0.0	*1.2*	12.5	*1.8*	15.6	*2.1*	1	*1*
CE-RM	**4.6**	***1.2***	0.6	*1.0*	0.0	*0.2*	1.7	*1.1*	0.4	*1.0*	1	*1*
SE-AF	3.4	*1.2*	4.8	*1.3*	2.2	*1.1*	**5.5**	***1.5***	1.2	*1.0*	1	*1*
GCE-RT	**5.3**	***2.0***	1.2	*1.0*	0.0	*0.3*	1.2	*1.4*	1.1	*1.1*	1	*1*
Wave-Layer	**46.5**	***6.1***	5.6	*2.2*	0.0	*0.9*	11.4	*3.0*	13.0	*2.9*	1	*1*
Wave-Gauss	**62.1**	***5.6***	6.0	*1.8*	0.0	*0.8*	14.0	*2.6*	9.2	*2.1*	1	*1*
ACE	**17.0**	***11.6***	1.7	*2.0*	0.0	*0.3*	4.5	*4.6*	6.5	*5.2*	1	*1*
Poisson-Gauss	**42.5**	***20.5***	25.0	*9.2*	17.0	*7.3*	21.1	*7.0*	9.8	*5.3*	1	*1*
Helmholtz	**78.3**	*6.1*	54.0	*5.1*	22.4	*3.0*	68.9	*7.3*	60.4	***9.0***	1	*1*

Summary of Performance on all Downstream Tasks

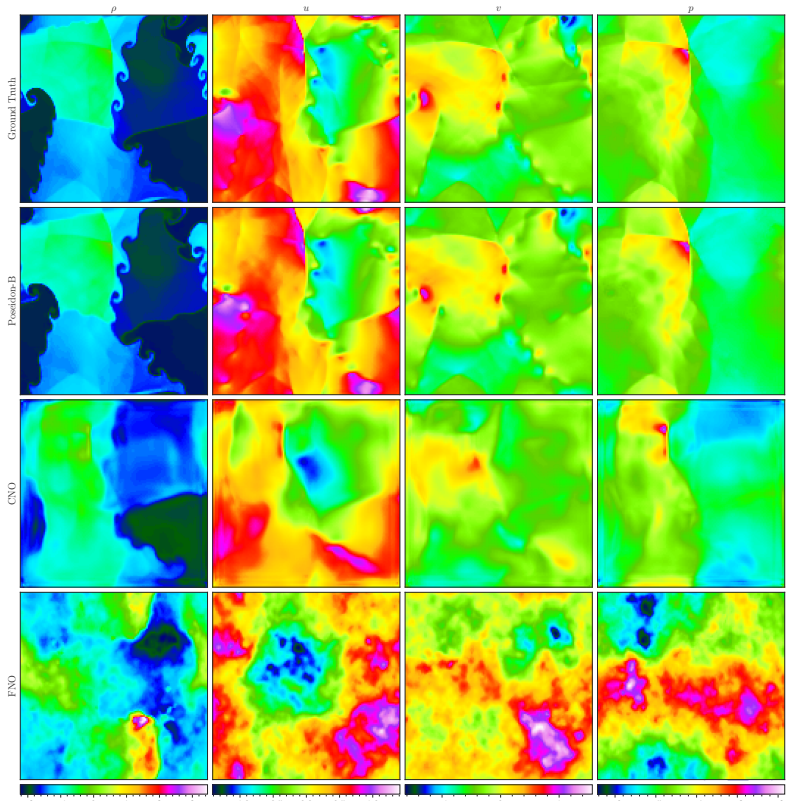


Deep Dive I: CE-RPUI

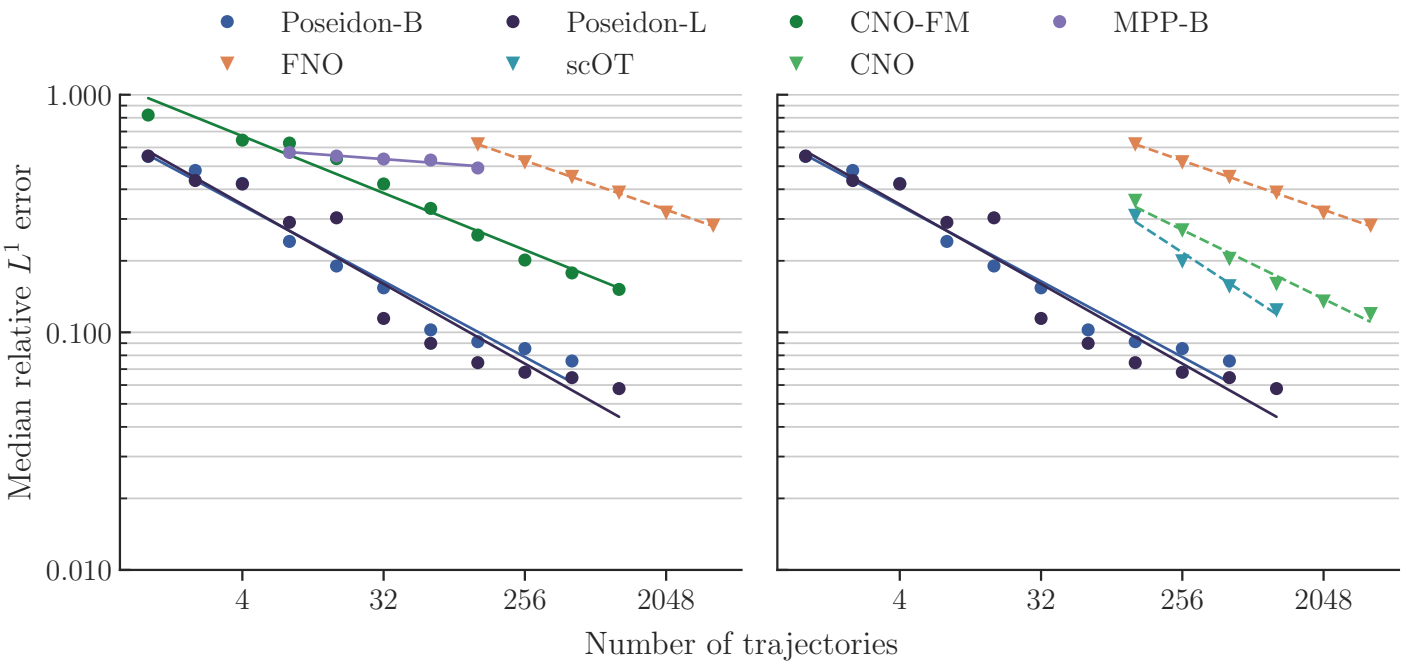
- Input (Initial Data):



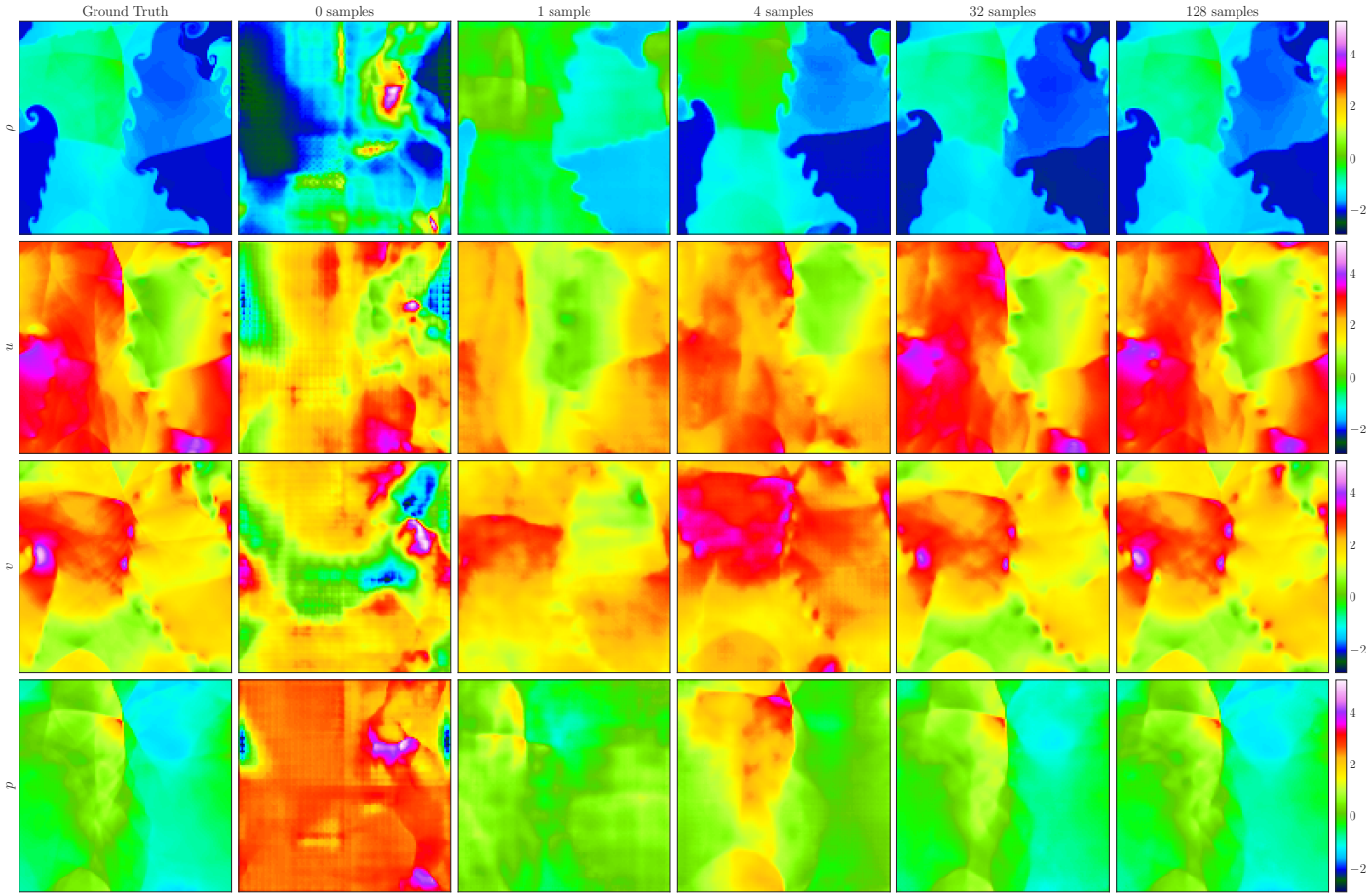
- Output (Solution at Final Time):



CE-RPUI: Scaling wrt Data

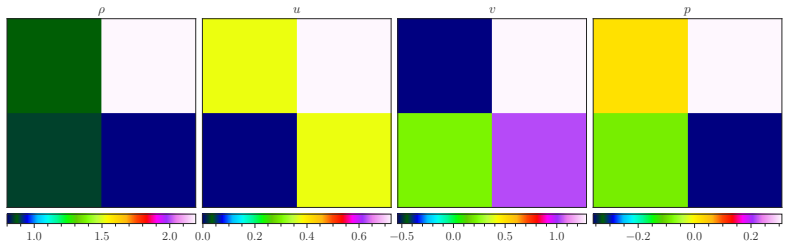


CE-RPUI: How does Poseidon Learn ?

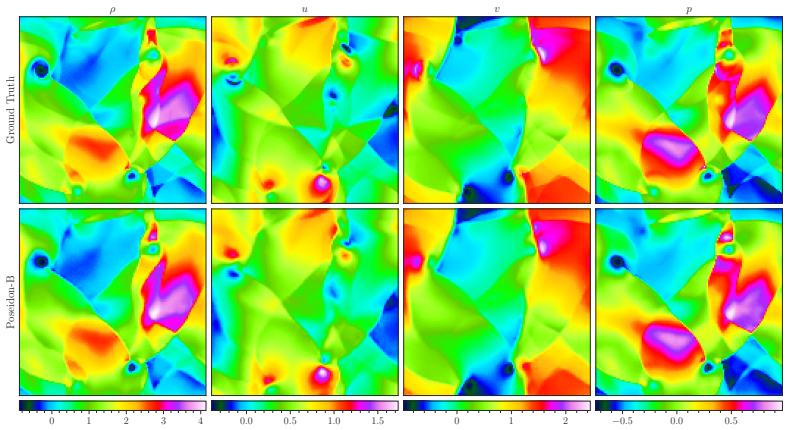


Learn Shocks from CE-RP

- Input ($T = 0$):

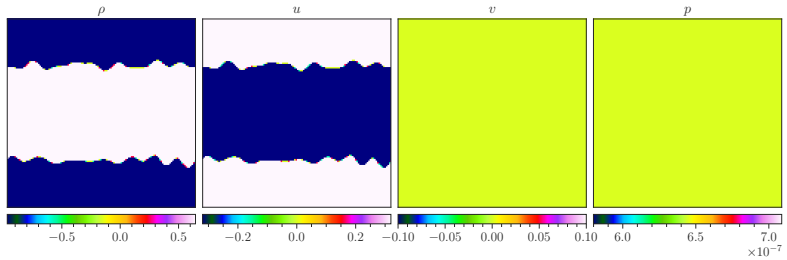


- Output ($T = 1$):

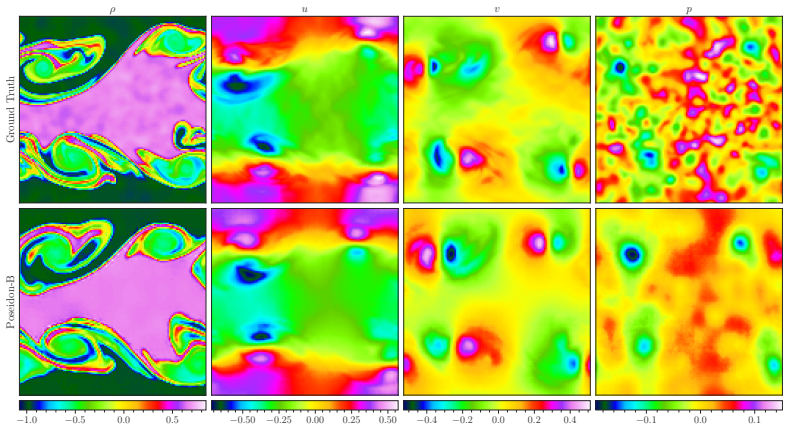


Learn Large Vortices from CE-KH

- Input ($T = 0$):

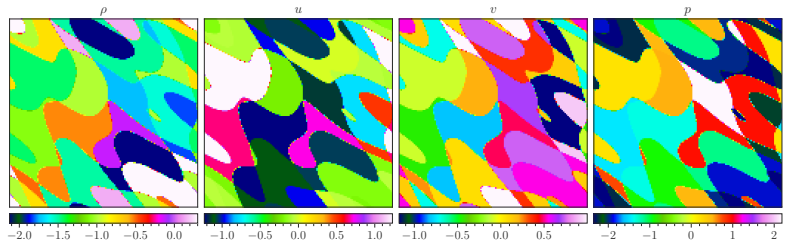


- Output ($T = 1$):

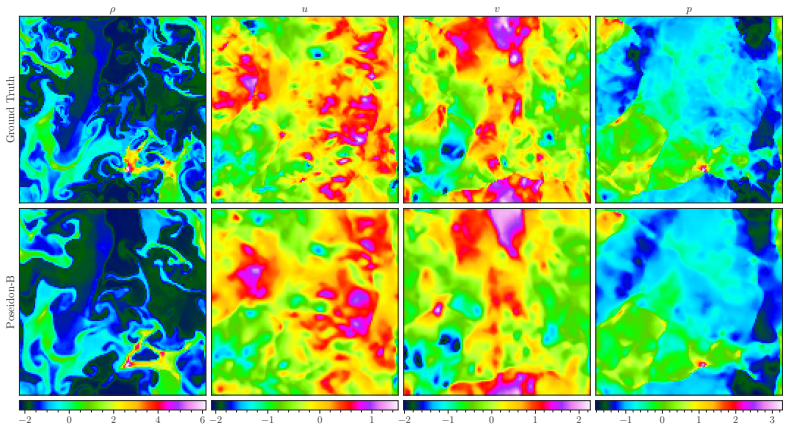


Learn small vortices from CE-RPC

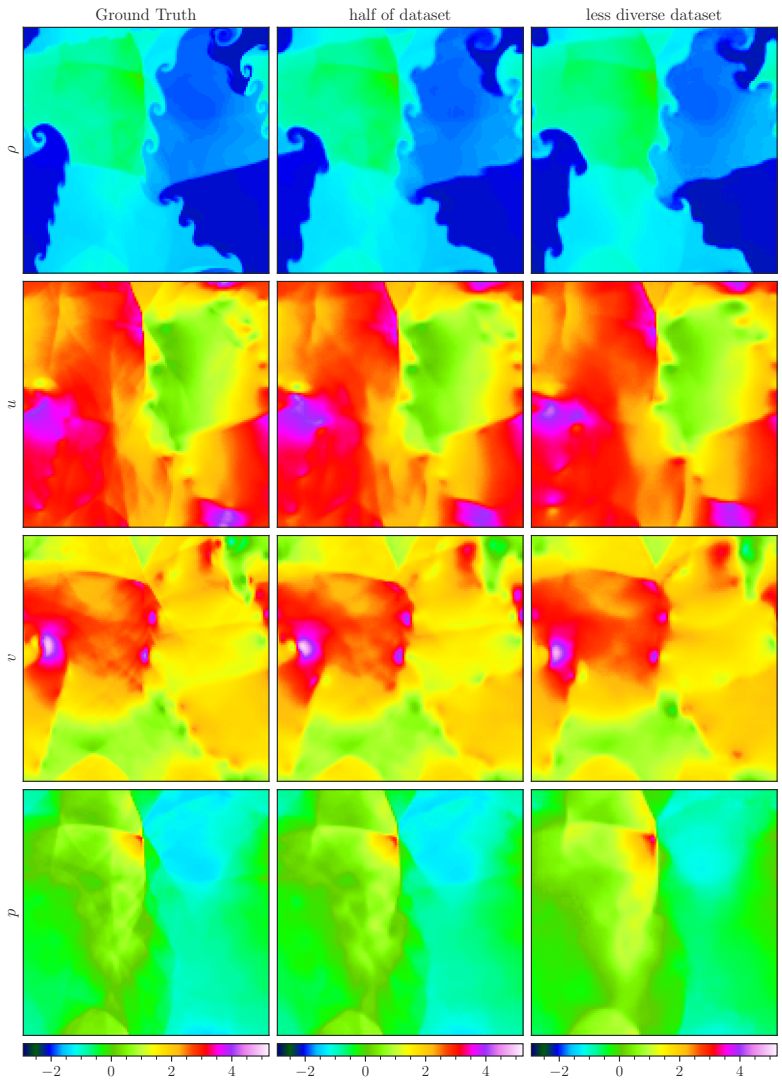
- Input ($T = 0$):



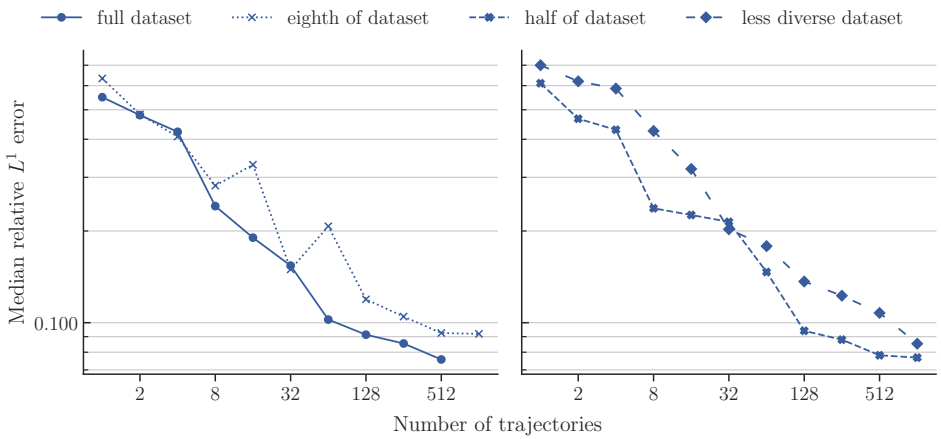
- Output ($T = 1$):



CE-RPUI: Less-Diverse Data



CE-RPUI: Less-Diverse Data Scaling

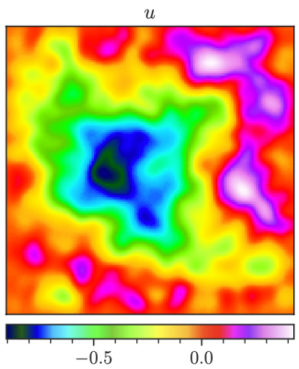


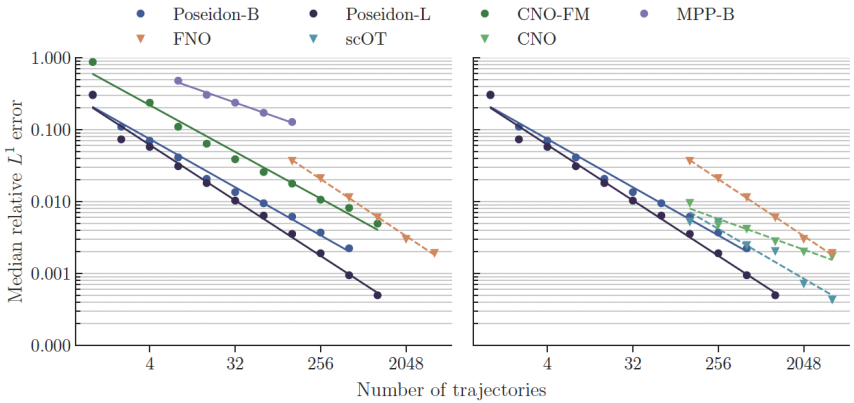
- Poseidon leverages diverse datasets.
- Link with Data Mixture Models for NLP/Vision.

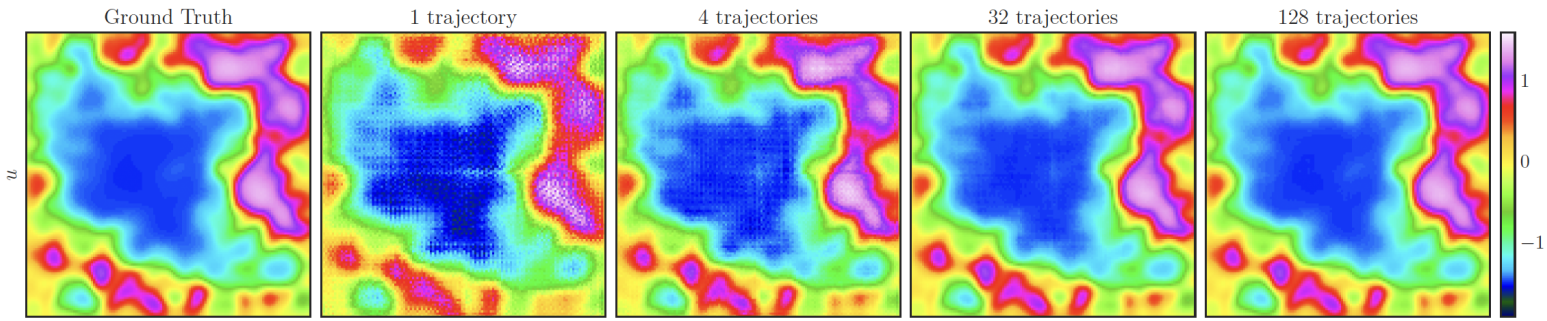
Deep Dive II: Allen-Cahn Eqn (Reaction-Diffusion).

In-depth

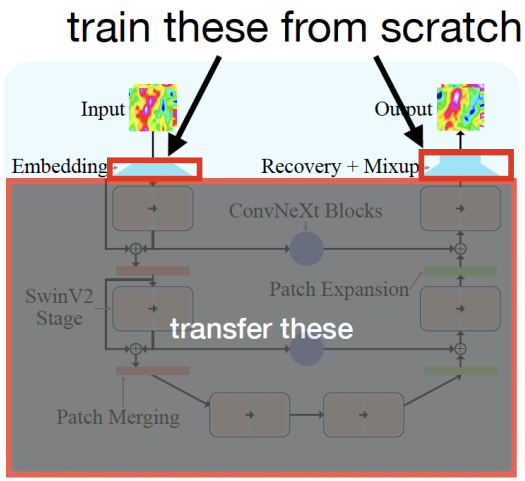
ACE



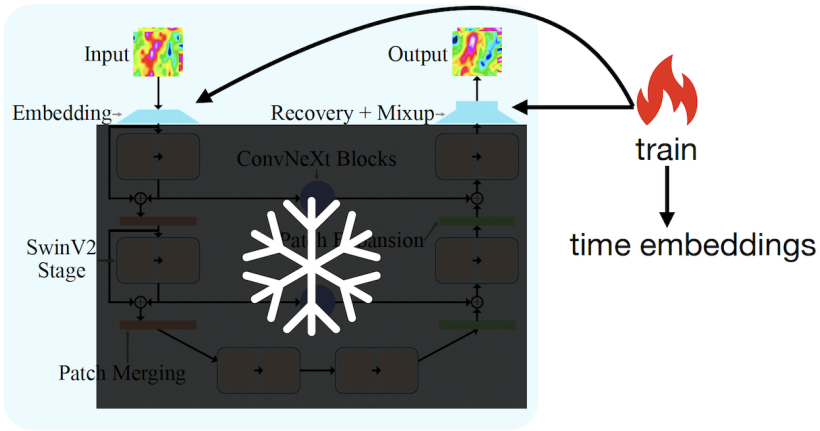




How much Physics has been learnt in PreTraining ?



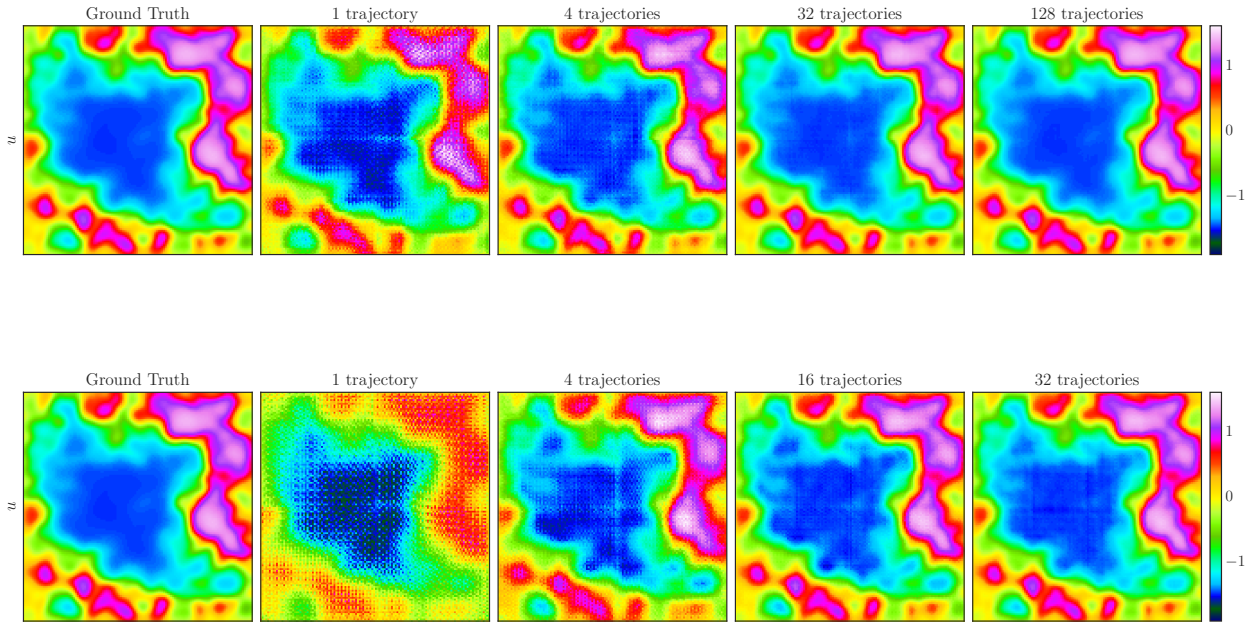
Transfer Latent



Freeze Latent

- Fine-Tune 158 M Parameters vs. Fine-Tune 293 K Parameters.

How much Physics has been learnt in PreTraining ?

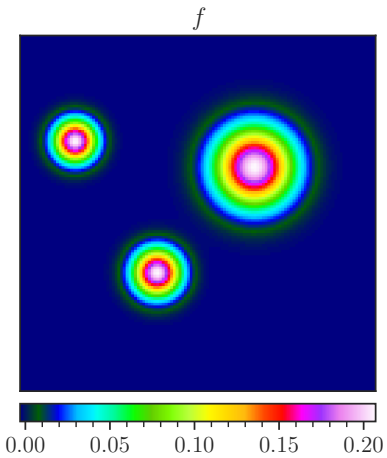


- Transfer Latents (Top) vs. Frozen Latents (Bottom)
- Error with 32 Trajectories:

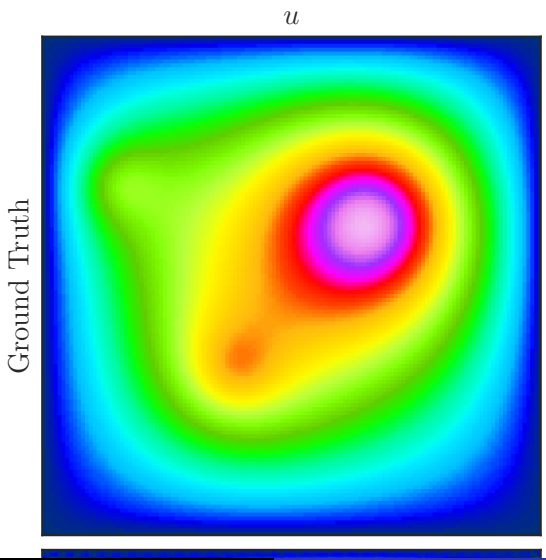
FNO	Frozen Latents	Transfer Latents
3.69%	3.1%	1.35%

Deep Dive: Poisson Eqn.

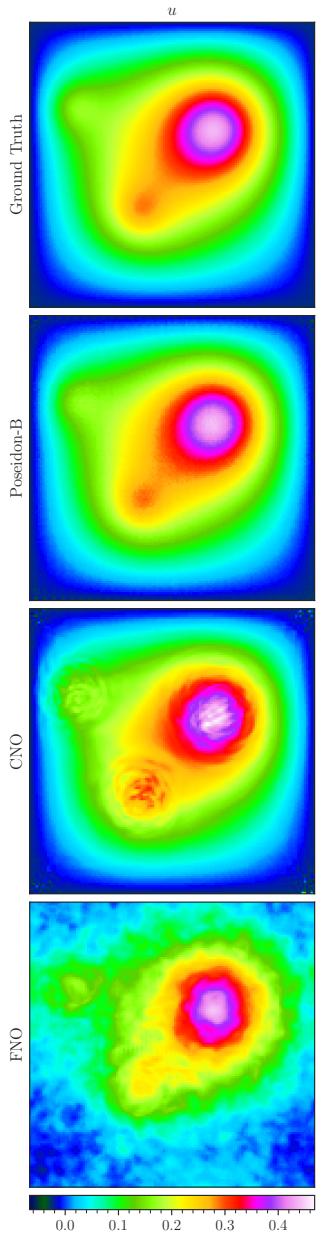
- Input (Source):



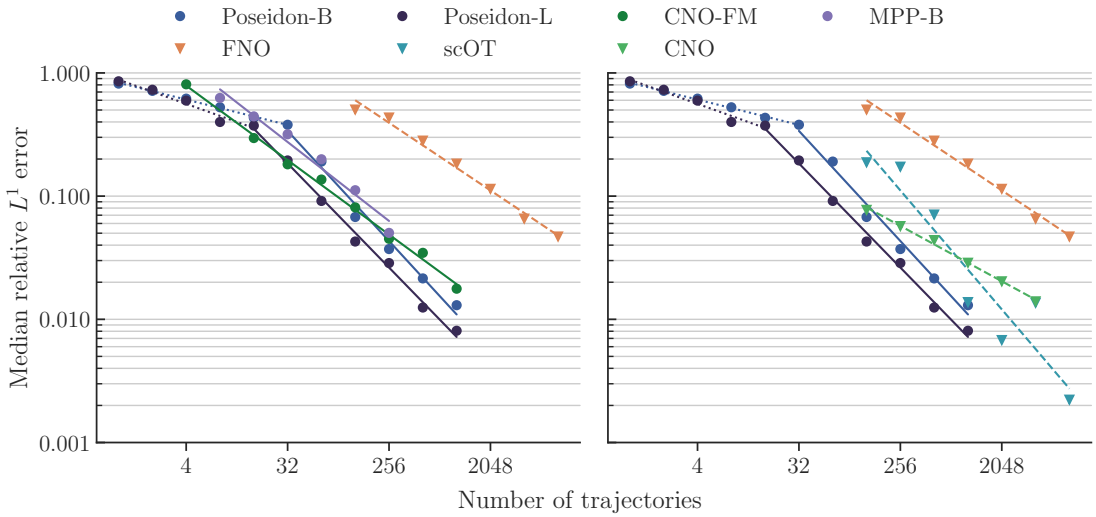
- Output (Solution):



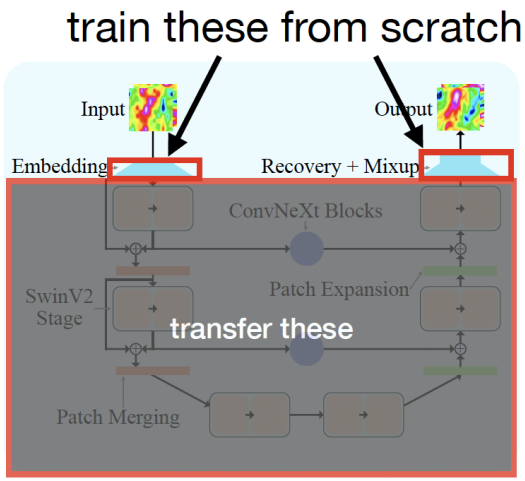
Poisson Eqn: Model Comparisons with 128 samples



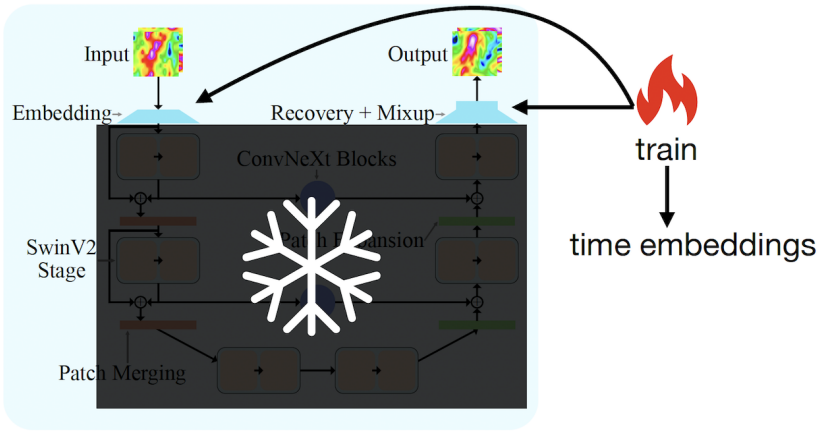
Poisson Eqn: Model Scalings



How much Physics has been learnt in PreTraining ?



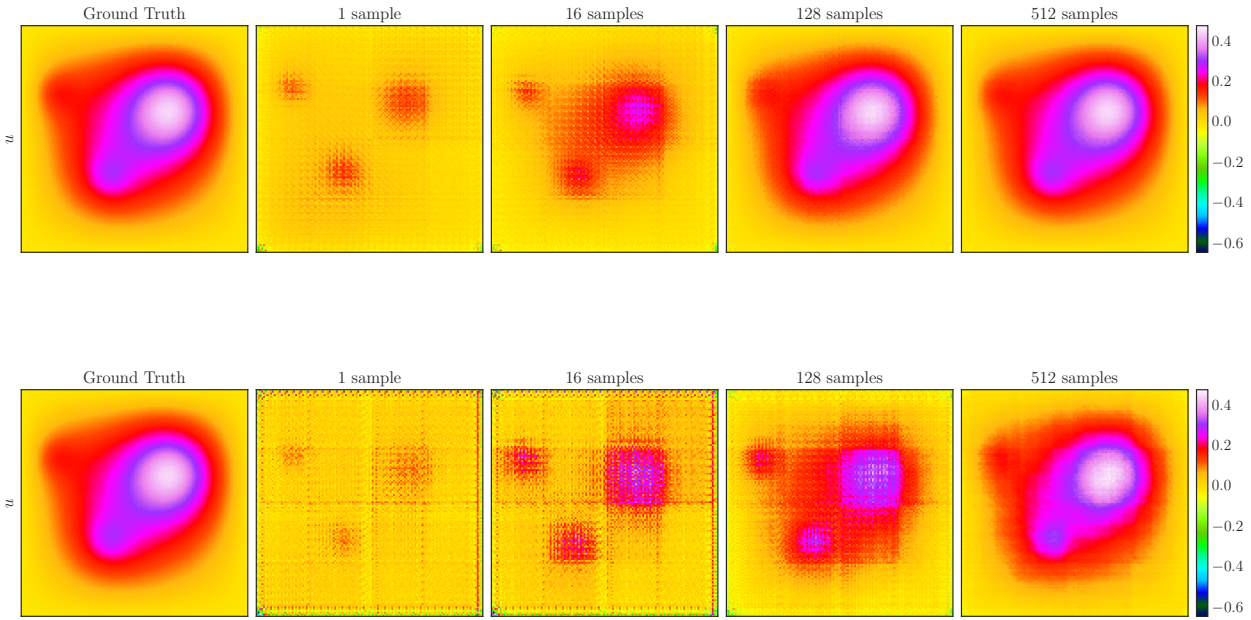
Transfer Latent



Freeze Latent

- Fine-Tune 158 M Parameters vs. Fine-Tune 293 K Parameters.

How much Physics has been learnt in PreTraining ?



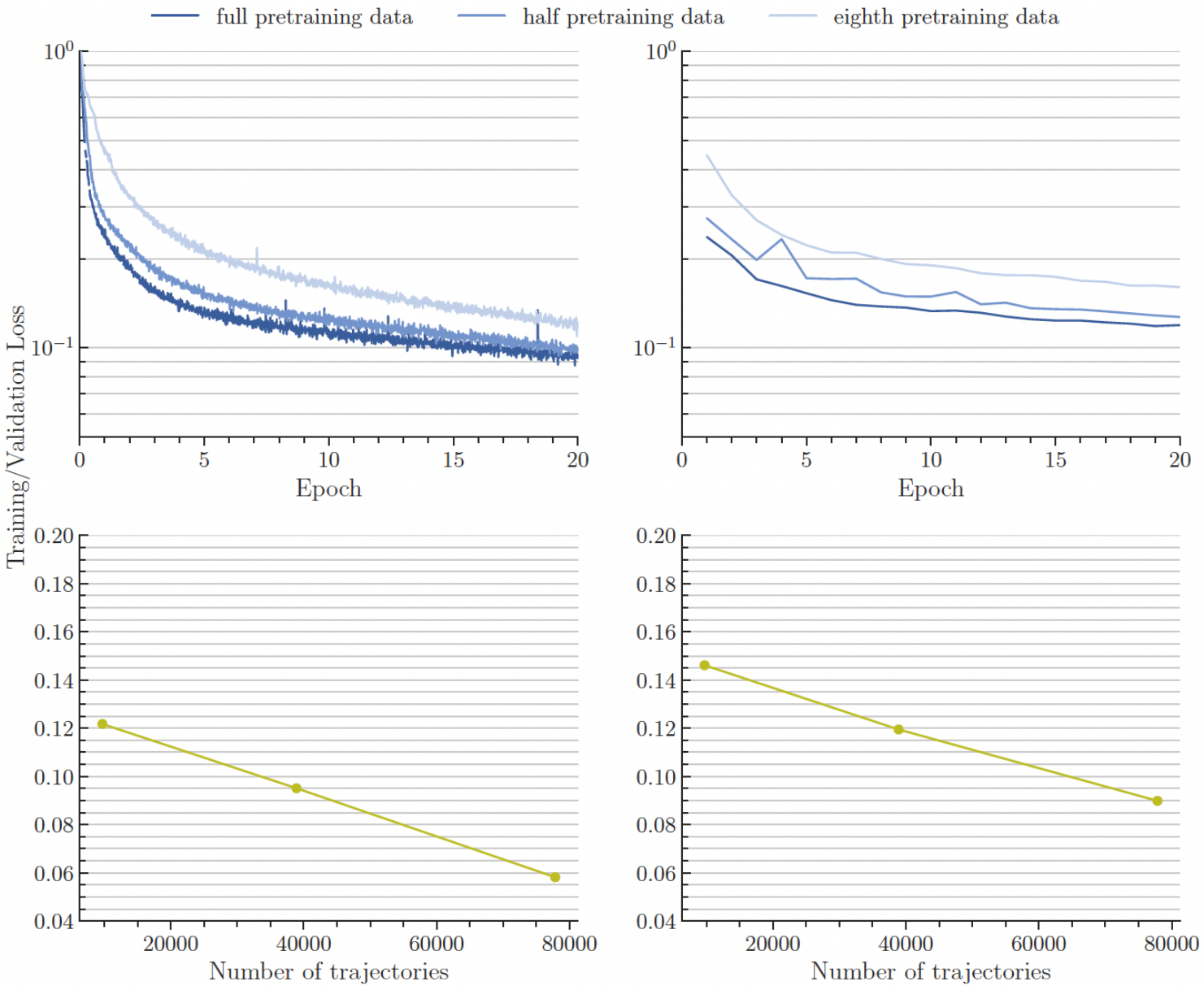
- Transfer Latents (Top) vs. Frozen Latents (Bottom)

Inference Costs on a NVIDIA RTX4090

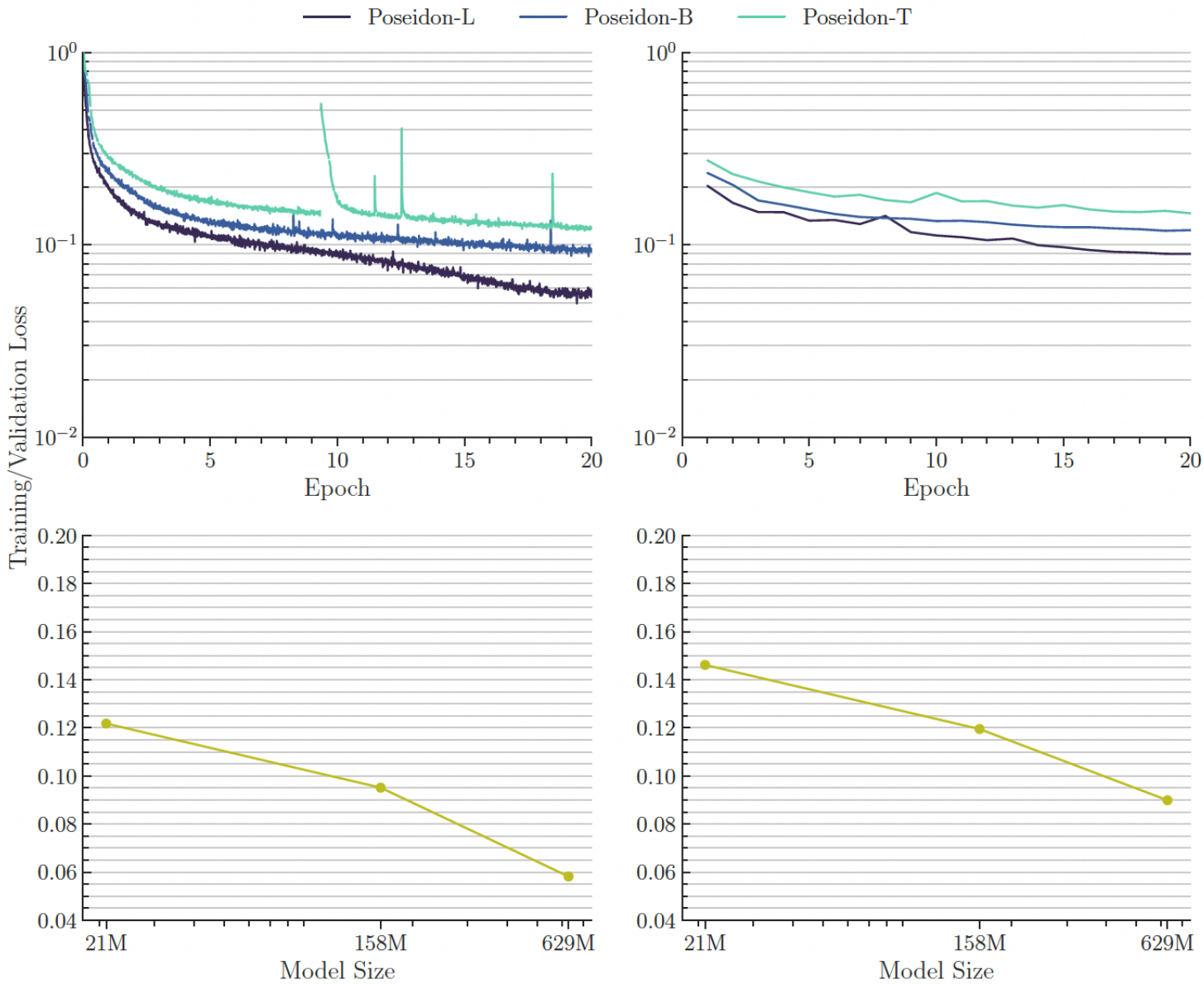
Model	Approximate inference time
POSEIDON-L	4 ms (16)
POSEIDON-B	2.9ms (40)
POSEIDON-T	1.6ms (40)
CNO-FM	1.8ms (32)
MPP-B	10ms (4)
CNO	0.9ms (32)
scOT	3ms (40)
FNO	2ms (40)

- Compared with $10^{-1} - 10^{2}$ secs for PDEgym data generation.

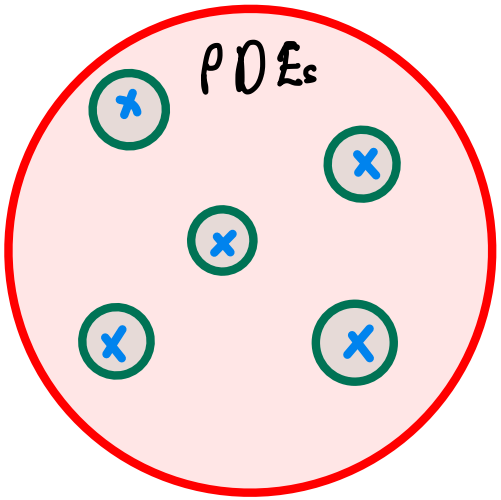
Poseidon scales with Data size

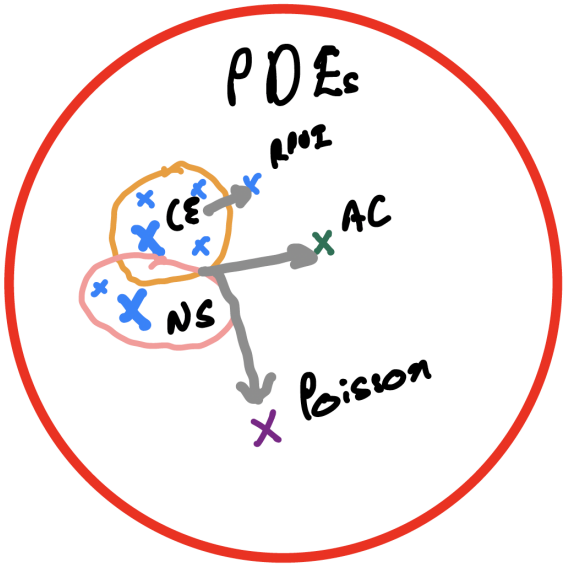


Poseidon scales with Model size

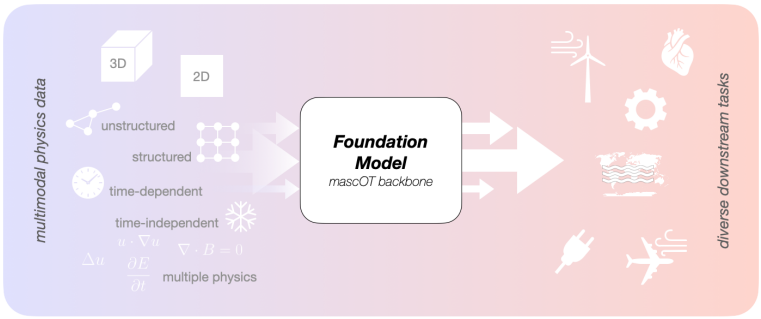


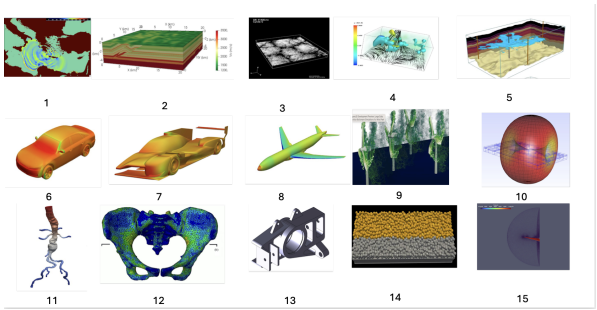
Poseidon shows feasibility of PDE Foundation Models



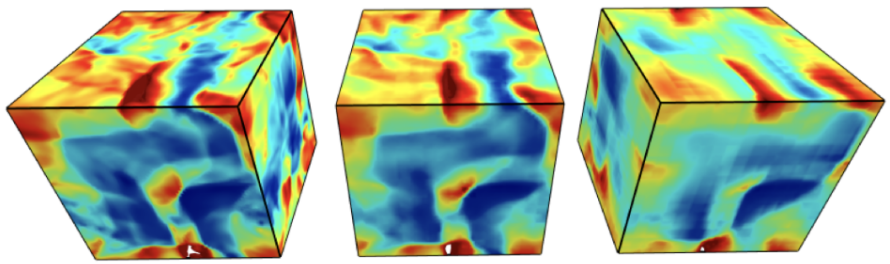


Ongoing@CAMLab: Multimodal FM for PDEs





- Augment Model Features:
  - 3D
  - Unstructured (point cloud) inputs
  - Genuine Multiphysics Training
  - Diffusion model for multiscale problems
  - PDE symbolic information



Course Summary + Perspectives