Learning and Transferring Physical Models through Derivatives

Alessandro Trenta, Andrea Cossu, and Davide Bacciu
Department of Computer Science, University of Pisa, Italy
Correspondence to: alessandro.trenta@phd.unipi.it

TRANSACTIONS tmlr on ML RESEARCH

ITALIAN NATIONAL PHD IN ARTIFICIAL INTELLIGENCE

IN SUPREMÆ DIGNITATIS 1343 UNIVERSITÀ DI PISA

Motivation

- Machine Learning can **model real world phenomena** such as climate, chemistry, fluid dynamics.
- However, they **fail to maintain consistency** with physical laws or have **many optimization issues**.
- **Transfer and distillation** of Physical models are understudied.

Greydanus et al., Hamiltonian Neural Networks, NeurIPS 2019
Wang et al., When and why pinns fail to train: A neural tangent kernel perspective. J. of Comp. Physics, 2022

The Problem

We model physical systems described by Ordinary Differential Equations (ODEs)

$$\dot{\boldsymbol{u}} = \boldsymbol{f}(\boldsymbol{u}(t), t; \xi), \qquad \boldsymbol{u}(0) = \boldsymbol{u}_0$$

or Partial Differential Equations (PDEs)

$$\begin{cases} \mathcal{F}[\boldsymbol{u}; \xi](t, \boldsymbol{x}) = 0 & t \in [0, T], \boldsymbol{x} \in \Omega, & \text{(PDE)} \\ \boldsymbol{u}(0, \boldsymbol{x}) = g(\boldsymbol{x}) & \boldsymbol{x} \in \Omega, & \text{(IC)} \\ \boldsymbol{u}(t, \boldsymbol{x}) = b(t, \boldsymbol{x}) & \boldsymbol{x} \in \partial\Omega, & \text{(BC)} \end{cases}$$

Classical Methods

- Rely on discretizations in time and in space with a **pre-determined mesh**.
- Strong theoretical guarantees on convergence as the mesh becomes finer.
- **Slow at inference** as they need to recalculate the solution or interpolate.
- **Do not work out of distribution** without re-initialization.

Related Works

- Classical methods: convergence guarantees but slow at inference and do not work OOD.
- NeuralODEs: require external solvers and lose consistency over time.
- Hamiltonian NNs: can be applied only on certain problems.
- PINNs: impose the equations as targets but hard to optimize.

Greydanus et al., Hamiltonian Neural Networks, NeurIPS 2019
Wang et al., When and why pinns fail to train: A neural tangent kernel perspective. J. of Comp. Physics, 2022

- Given a set of coordinates $\{\boldsymbol{x}_i\}_{i=1,\ldots,N},\ \boldsymbol{x}_i \in \mathbb{R}^d$ and the solution at those points $\{u(\boldsymbol{x}_i)\}_{i=1,\ldots,N}$ there are (infinitely) many curves that interpolate them. Only one is true.
- In ODEs, the IC and the definition of the derivatives uniquely determine the trajectory.

Our solution: DERL

- In ODEs, the solution is uniquely determined by the derivatives and the IC.
- In a supervised setting, we propose the following **DERL** loss:

$$L(\hat{\boldsymbol{u}}, \boldsymbol{u}) = \overbrace{\lambda_u \left\|\mathrm{D}\hat{\boldsymbol{u}}(t, \boldsymbol{x}) - \mathrm{D}\boldsymbol{u}(t, \boldsymbol{x})\right\|_2^2}^{\textbf{Derivative learning}} + \overbrace{\lambda_B \left\|\hat{\boldsymbol{u}}(t, \boldsymbol{x}) - b(t, \boldsymbol{x})\right\|_2^2}^{\text{Boundary cond.}} + \overbrace{\lambda_I \left\|\hat{\boldsymbol{u}}(0, \boldsymbol{x}) - g(\boldsymbol{x})\right\|_2^2}^{\text{Initial cond.}}$$

- We will also see how to use DERL to transfer information and build physical models incrementally

Theoretical Results

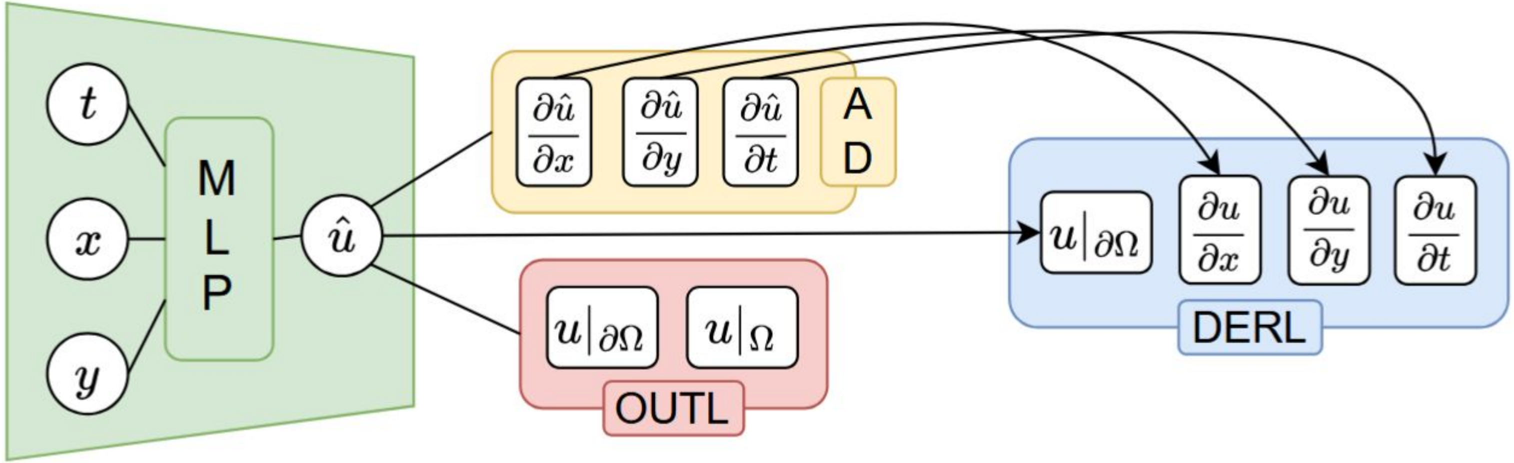
- If the loss converges to zero $L(u, \hat{u}) \to 0$, the model converges to the solution $\hat{u} \to u$
- Explicit bound on convergence + the final model also fulfills all the conditions of the PDE.
- DERL works with empirical derivatives.

Experiments: Learning Physical Systems

Learn the solution from a set of points and test the models on a finer grid to show generalization.

- We compare our methodology to:
  - OUTL, which learns directly u.
  - OUTL+PINN, which learns u and the equation.
  - SOB, which learns a combination of u and its derivative.
- We measure:
  - The distance to the true solution.
  - The PDE residual (as in PINNs).

Czarnecki et al., Sobolev training for neural networks, NeurIPS 2017



t
x
y
MLP
$\hat{u}$
$\frac{\partial \hat{u}}{\partial x}$
$\frac{\partial \hat{u}}{\partial y}$
$\frac{\partial \hat{u}}{\partial t}$
AD
$u|_{\partial\Omega}$
$u|_{\Omega}$
OUTL
$u|_{\partial\Omega}$
$\frac{\partial u}{\partial x}$
$\frac{\partial u}{\partial y}$
$\frac{\partial u}{\partial t}$
DERL

Learning Physics: In-domain generalization

Table 2: Results for the Allen-Cahn and continuity equation.

Model	Task	$\|u-\hat{u}\|_2$ $\times 10^{-4}$	$\|\mathcal{F}[\hat{u}]\|_2$ $\times 10^{-3}$	Task	$\|u-\hat{u}\|_2$ $\times 10^{-2}$	$\|\mathcal{F}[\hat{u}]\|_2$ $\times 10^{-1}$
DERL (ours)	Allen-Cahn	**6.836**	**1.930**	Continuity	9.059	**2.247**
OUTL		56.95	11.41		**8.749**	3.762
OUTL+PINN		39.13	7.105		9.378	2.8975
SOB		9.448	2.469		16.52	2.881

(AC) $\lambda(u_{xx}+u_{yy})+u(u^2-1)=f(\xi)$ (CO) $\frac{\partial u}{\partial t}+\nabla\cdot(vu)=0$

Università di Pisa

Table 3: Results for the Navier-Stokes Kovasznay flow experiment.

Model	$\|(\boldsymbol{u}, p) - (\hat{\boldsymbol{u}}, \hat{p})\|_2$ $\times 10^{-5}$	$\|\omega - \hat{\omega}\|_2$ $\times 10^{-4}$	**(NS.M)** $\|\mathcal{F}[\hat{\boldsymbol{u}}, \hat{p}]\|_2$ $\times 10^{-4}$	**(NS.I)** $\|\mathcal{F}[\hat{\boldsymbol{u}}]\|_2$ $\times 10^{-4}$
DERL (ours)	**0.6719**	**0.6611**	0.8810	0.5945
OUTL	9.201	106.6	10.58	10.30
OUTL+PINN	3.331	5.042	**0.8465**	**0.5729**
SOB	1.125	1.077	1.660	1.177

$$\text{(NS.M)} \quad [\mathrm{D}\boldsymbol{u}] \cdot \boldsymbol{u} - \Delta \boldsymbol{u} + \nabla p = 0$$

$$\text{(NS.I)} \nabla \cdot \boldsymbol{u} = 0$$

Learning Physics: Generalization to New Parameters

We consider the damped pendulum equation.

- The models predict the entire trajectory given the initial conditions.
- We train on 30 trajectories, validate on other 10 and test on 10 unseen ICs.

$$(\mathrm{DP}) \quad \ddot{u} + \frac{g}{l}\sin(u) + \frac{b}{m}\dot{u} = 0$$

Table 4: Numerical results for the damped pendulum.

Model	$\|u - \hat{u}\|_2$ $\times 10^{-2}$	$\|\dot{u} - \dot{\hat{u}}\|_2$ $\times 10^{-2}$	$\|\mathcal{F}[\hat{u}]\|_2$ $\times 10^{-2}$	$\|\mathcal{G}[\hat{u}]\|_2$ $\times 10^{-2}$	$\|\mathcal{F}[\hat{u}]_{t=0}^{\text{full}}\|_2$ $\times 10^{-1}$
DERL (ours)	1.069	**1.296**	**1.277**	**9.244**	**4.445**
OUTL	**0.9607**	5.766	5.452	20.55	8.233
OUTL+PINN	1.118	7.614	7.070	30.07	10.65
SOB	1.133	2.824	2.803	15.17	8.054

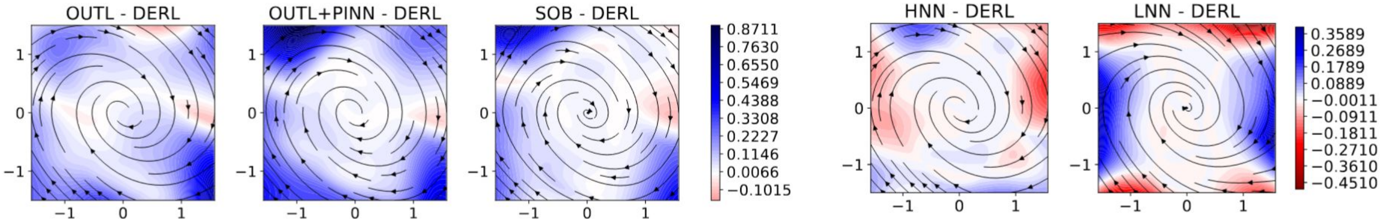


Table 5: Results for the parametric Allen-Cahn experiments, averaged over test parameters.

Model	Single-Parameter $\|u-\hat{u}\|_2$ $\times 10^{-3}$	Single-Parameter $\|\mathcal{F}[\hat{u}]\|_2$ $\times 10^{-2}$	Double-parameter $\|u-\hat{u}\|_2$ $\times 10^{-3}$	Double-parameter $\|\mathcal{F}[\hat{u}]\|_2$ $\times 10^{-2}$
DERL (ours)	**5.441**	**1.730**	**9.628**	**1.467**
OUTL	8.486	11.67	19.13	8.050
OUTL+PINN	6.413	2.466	27.70	8.446
SOB	9.736	2.276	21.26	1.477

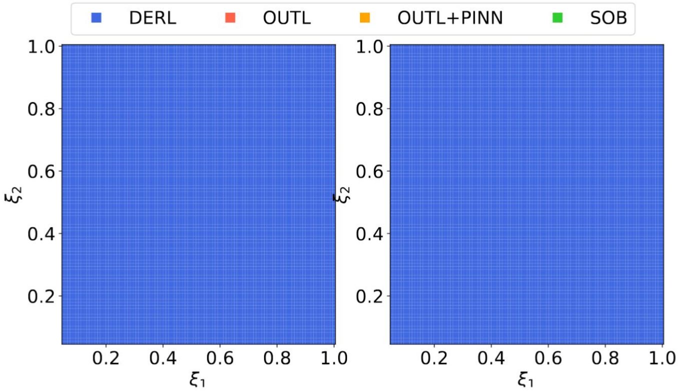


Figure 16: Double-parameter Allen-Cahn equation. The plot is obtained by taking a grid of points ξ_1, ξ_2 in the parameter space and coloring it with the color of the model with the lowest u error or PDE residual for such parameters.

Transferring Physical Systems

- A PINN is trained on the whole domain (teacher).
- Student models learn the IC, BC, and the teacher's derivatives (or outputs).
- Korteweg-de Vries equation: $(\mathrm{KdV}) \quad u + uu_t + \nu u_{xxx} = 0$
- We analyze the effect of higher-order derivatives.

Table 13: Results for the KdV equation, PINN distillation. Results are empirical L^2 norms over the time-space domain. Derivative and Hessian losses are computed with respect to the PINN. Best model(s) in bold, second best underlined.

Model	$\|\hat{u}_S - u\|_2$	$\|D\hat{u}_S - D\hat{u}_T\|_2$	$\|D^2\hat{u}_T - D^2\hat{u}_S\|_2$	$\|\mathcal{F}[\hat{u}_S]\|_2$	**BC loss**
PINN (teacher)	0.037171	/	/	0.16638	0.33532
DERL (ours)	0.038331	0.098188	3.9872	0.32480	0.014197
HESL (ours)	**0.037380**	0.065454	1.1662	**0.19153**	0.014220
DER+HESL (ours)	0.038524	**0.041988**	**0.85280**	0.19317	0.031850
OUTL	0.038589	0.22580	31.582	17.366	**0.012830**
SOB	0.037447	0.10097	4.0967	0.38523	0.013644
SOB+HES	0.041353	0.13119	3.2684	0.23184	0.016222

Building Physical Models Incrementally

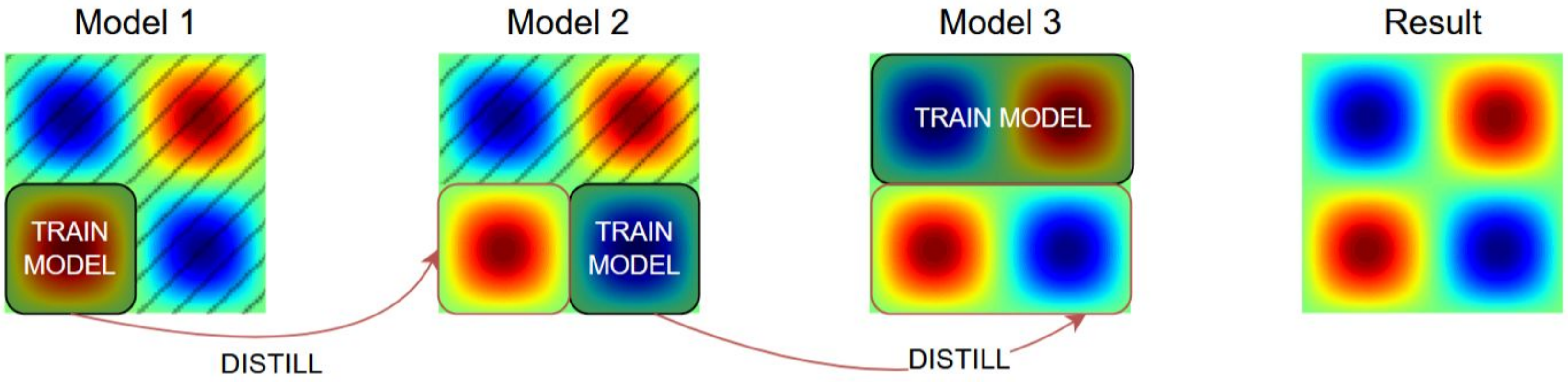
We employ DERL to develop a new methodology to train physical models incrementally.

1. Train a PINN on a sub-domain.
2. Collect samples of its outputs and derivatives.
3. Train a new model with a loss that combines a new PINN and distillation.

$$L_{\text{student}} = L_{\text{system}} + \|\mathrm{D}\hat{u}_T - \mathrm{D}\hat{u}_S\|_2$$

More results on knowledge distillation and higher-order derivatives in the paper.

Example - Incremental Space Domain Pipeline



Building Physical Models Incrementally

Three experiments:

- Larger time horizon.
- Larger domain space.
- New parameters.

We compare to:

- PINN on the entire domain,
- PINN trained incrementally with no distillation
- PINN trained incrementally with experience replay.

Table 6: Results for the physical knowledge transfer experiments. L^2 error and PDE residual on the whole domain (or on test parameters) at the final stage of incremental training.

Model		Larger timespan 5.1		Larger domain 5.2		New parameters 5.3	
		$\|u-\hat{u}\|_2$ $\times 10^{-2}$	$\|\mathcal{F}[\hat{u}]\|_2$ $\times 10^{-1}$	$\|u-\hat{u}\|_2$ $\times 10^{-3}$	$\|\mathcal{F}[\hat{u}]\|_2$ $\times 10^{-3}$	$\|u-\hat{u}\|_2$ $\times 10^{-3}$	$\|\mathcal{F}[\hat{u}]\|_2$ $\times 10^{-3}$
DERL	from-scratch	**3.331**	**2.932**	**3.332**	**7.453**	**7.736**	**7.537**
	continual	**2.379**	1.731	**2.377**	**3.873**	11.76	**7.275**
OUTL	from-scratch	9.302	8.966	6.850	15.10	9.757	10.27
	continual	9.334	7.571	3.399	8.995	11.49	11.87
SOB	from-scratch	3.662	4.327	5.151	9.480	11.03	11.63
	continual	2.535	**1.536**	2.642	5.284	**8.840**	7.645
PINN	full	9.801	1.730	2.356	6.275	14.96	6.302
PINN	no distillation	15.19	18.29	247.3	245.6	31.87	16.14
PINN	replay	5.371	1.889	5.281	10.54	13.72	6.289

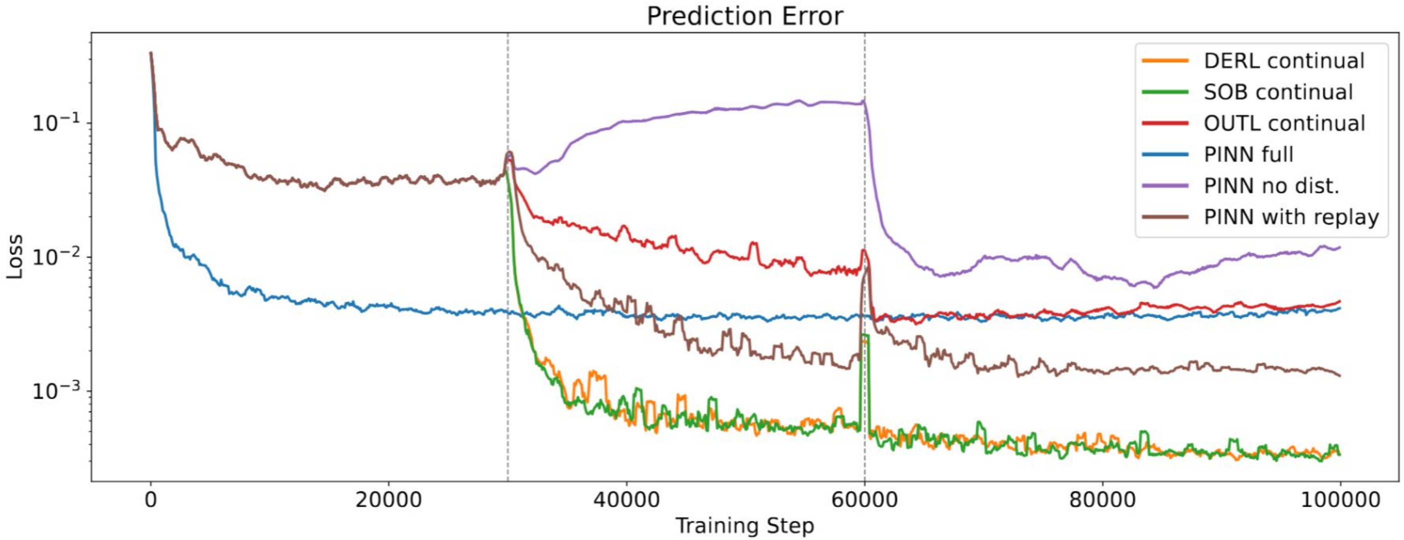


Figure 5: Prediction errors for the KdV transfer experiment for the continual models and the PINN variants. Vertical lines represent when a new step of incremental training is initiated.

Thank you for your Attention!

OPENREVIEW

CODE