Continual Learning of Physical Systems via Derivative Distillation

Claudia Gentili [1] **Alessandro Trenta** [1] **Andrea Cossu** [1] **Davide Bacciu** [1]

Abstract

Continual learning (CL) designs models that adapt to non-stationary data streams while preserving previously acquired knowledge. Moving away from current CL benchmarks and datasets, we turn our attention to scientific machine learning, and we study how CL can approach physical systems, which are inherently continuous in time and space and are naturally described by differential equations over spatio-temporal domains. We adopt Derivative Distillation, a distillation-based approach that leverages model derivatives as a compact representation of knowledge and integrates it within a physics-informed learning framework. Our results show that the Derivative Distillation enables stable adaptation with minimal forgetting. In some cases, the performance surpasses that of a model jointly trained on all data at once. These findings highlight physical systems as a promising benchmark for CL.

1. Introduction

Continual learning (CL) seeks to enable models to adapt to evolving data distributions while retaining previously acquired knowledge (Parisi et al., 2019). Despite a rich body of methods addressing catastrophic forgetting (Delange et al., 2021; Masana et al., 2023), empirical progress in CL has been largely driven by benchmarks made from dataset splits (van de Ven et al., 2022) or simple control environments (Khetarpal et al., 2022). More recently, the advent of foundation models enabled CL on large-scale datasets via parameter-efficient fine-tuning of large language models (Coleman et al., 2025).

We focus our attention on a different class of problems modeling real-world, physical systems, like diffusion processes. Crucially, many real-world systems are inherently continuous in both time and space, and are commonly described through differential equations defined over spatio-temporal domains. In these settings, learning unfolds over evolving regions of space, time, or system parameters, leading to non-stationary environments that differ fundamentally from available CL benchmarks. Time, in particular, plays a crucial role in physical systems and requires any learning model to incorporate and predict temporally correlated information. A skill perhaps undervalued in current CL models. Despite these characteristics, the intersection between CL and scientific machine learning is rarely studied (Trenta et al., 2025; Howard et al., 2024).

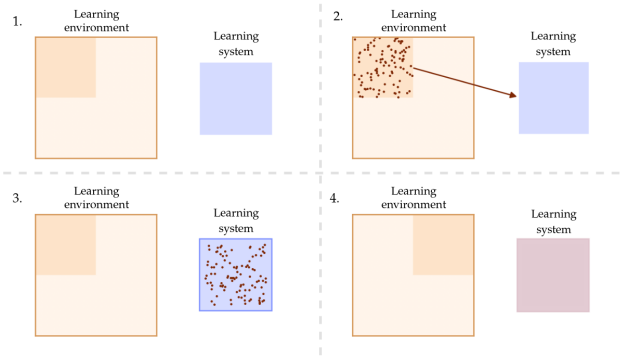

Figure 1. CL in a 2D environment. The shaded region of the learning environment represents the current task in a 2-task sequence.

In this work, we argue that physical systems offer a principled and realistic testbed for CL, and we investigate how different methods behave in this setting (Fig. 1). We formulate CL over initial-boundary value problems, where tasks correspond to incremental exposure along temporal or parametric dimensions. This formulation departs from the classification-based settings that dominate the continual learning literature and instead introduces regression-oriented CL benchmarks, which remain comparatively underexplored.

We adopt a recently-proposed approach, called Derivative Learning (DERL) (Trenta et al., 2025), and show that it is very effective in learning continuously a variety of physical systems. In particular, when integrated within a distillation-based view of CL, DERL leads to stable adaptation with minimal forgetting. Perhaps surprisingly, our empirical evaluation shows that distillation-driven CL not only outperforms standard CL baselines, but in some

[1]Computer Science Department, University of Pisa, Pisa, Italy. Correspondence to: Claudia Gentili <c.gentili3@studenti.unipi.it>.

Presented at the ICML 2026 Workshop "Continual Adaptation at Scale: Towards Sustainable AI".

cases surpasses the joint training performance, which has access to all data simultaneously.

Our analysis of CL in physical systems remains tightly connected to the CL literature, offering an easy way for researchers to test their approaches on another challenging application domain. In particular: i) our physical systems benchmark aligns with the popular domain-incremental and task-incremental CL benchmarks (see Section 3.1); ii) distillation-based approaches (Li & Hoiem, 2016) can be effectively repurposed for CL of physical systems, where distillation acts on physical variables.

We hope that introducing a new benchmark on physical systems where CL shows excellent performance will encourage more researchers to explore and design CL methods for scientific machine learning. An area that has the potential to play a key role in enabling adaptive, data-driven modeling of complex real-world processes.

2. Related Work

A large body of work in scientific machine learning focuses on learning dynamical systems governed by differential equations. Existing approaches can be broadly categorized into closure learning (Chen et al., 2019; Melchers et al., 2023), Lagrangian/Hamiltonian methods (Lutter et al., 2019; Cranmer et al., 2019; Greydanus et al., 2019), physics-informed learning (Raissi, 2018; Raissi et al., 2019; Wang et al., 2023), and operator learning (Li et al., 2021; Lu et al., 2021). Closure models and neural ODEs learn system dynamics from trajectory data, often relying on numerical solvers for training and inference. Lagrangian and Hamiltonian neural networks incorporate physical priors by modeling energy-based formulations of the system. Physics-informed neural networks (PINNs) embed governing equations, boundary conditions, and initial conditions directly into the training objective, enabling learning with limited or no labeled data. Neural operators extend this paradigm by learning mappings between function spaces, approximating solution operators of partial differential equations. While powerful, these approaches are studied in static settings and assume access to the full training domain.

CL, instead, addresses the challenge of learning from sequential data streams while avoiding catastrophic forgetting. Popular approaches are replay-based methods (Hayes et al., 2021), which store and reuse subsets of past data, and regularization-based methods (Parisi et al., 2019), which constrain parameter updates (Kirkpatrick et al., 2017) or distill knowledge from previous models by matching their outputs or intermediate representations (Li & Hoiem, 2016). Among the most popular CL benchmarks (van de Ven et al., 2022) we find class-incremental streams on vision-based tasks, where a learning sequence is constructed by splitting a given dataset into multiple groups with different classes each. In other cases (e.g., reinforcement learning on Atari games), the CL model is trained on a task-incremental stream where a task identifier is explicitly provided. Finally, in a domain-incremental stream, the model encounters new instances of already seen classes, thus continuously refining the concepts associated with each class.

The intersection between continual learning and scientific machine learning remains relatively underexplored. Physical systems introduce unique challenges due to their continuous spatio-temporal structure, strong correlations, and multi-objective nature arising from governing equations and boundary conditions. Recent work (Trenta et al., 2025; 2026) has highlighted the potential of derivative-based supervision and distillation for encoding physical knowledge, showing that model derivatives can serve as compact representations of learned dynamics. However, a systematic study of continual learning in physics-informed settings is still missing.

Our work bridges these areas by formulating continual learning over physical systems modeled as initial-boundary value problems and by showing that distillation-based approaches, when integrated with physics-informed learning, provide an effective mechanism for continual adaptation. In contrast to prior work, we emphasize continuous task structures, solver-free training, and the surprising effectiveness of distillation in enabling positive transfer across sequential learning stages.

3. Methodology

We consider physical systems modeled by initial-boundary value problems on general domains $\Omega \subset \mathbb{R}^d$ with boundary $\partial\Omega$ through the conditions:

$$\begin{cases} \mathcal{F}[\boldsymbol{u};\xi](t,\boldsymbol{x}) = 0 & t \in [0,T], \boldsymbol{x} \in \Omega, \quad \text{(PDE)} \\ \boldsymbol{u}(0,\boldsymbol{x}) = g(\boldsymbol{x}) & \boldsymbol{x} \in \Omega, \quad \text{(IC)} \\ \boldsymbol{u}(t,\boldsymbol{x}) = b(t,\boldsymbol{x}) & \boldsymbol{x} \in \partial\Omega, \quad \text{(BC)} \end{cases} \tag{1}$$

where $\mathcal{F}$ is a differential operator $\xi \in \mathbb{R}^l$ is a set of parameters that regulates the dynamics, and $g(\boldsymbol{x})$ and $b(t,\boldsymbol{x})$ are functions that define the Initial condition (IC) and Boundary condition (BC), respectively. Problems of this kind are completely determined by the Partial Differential Equation (PDE), which describes the evolution and propagation of information from the IC and BC to the rest of the domain through time and space (Evans, 2022). Our base learning model is a PINN $\hat{\boldsymbol{u}}$ (Raissi et al., 2019), which is widely used to learn PDEs. Given points $\{(t_i,\boldsymbol{x}_i)\}_{i=1,\dots,N_d}$ in the domain, the PINN loss is composed of 3 terms associated with the PDE, ICs and BCs: $L_{\text{PINN}} = \lambda_u \|\mathcal{F}[\hat{u}]\|^2_{L^2(\Omega)} + \lambda_B \|\hat{\boldsymbol{u}}(t,\boldsymbol{x}) - b(t,\boldsymbol{x})\|^2_2 +$

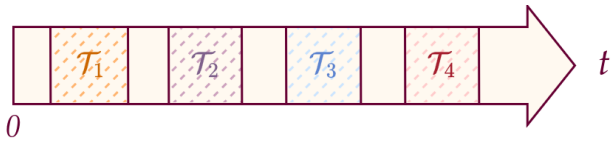

Figure 2. Time-incremental scenario

$\lambda_I \|\hat{\boldsymbol{u}}(0,\boldsymbol{x}) - g(\boldsymbol{x})\|_2^2$, with $\lambda_{u,B,I}$ hyper-parameters. To continuously train a PINN, we use Derivative Distillation (Trenta et al., 2025), which is implemented by adding an additional term to the loss function (like any regularization approach). Let the student S be the PINN trained on the current task and the teacher T the frozen previous model, the loss becomes: $L_{\text{S}} = L_{\text{PINN}} + \|\text{D}\hat{u}_T - \text{D}\hat{u}_S\|_2$.

3.1. Physical systems

We consider two physical systems, which can be associated with domain-incremental (time-incremental, in our case) and task-incremental. Task-incremental physical systems are parameterized physical systems, where each task requires the model to learn from K parameterizations of the system. Each parameterization is only present in a single task. In time-incremental learning instead (Fig. 2), each task considers a bounded time domain, which does not intersect between one task and the others. Below, we provide the general formulation of such systems. The setup used in the experiments is provided in Appendix B, which specifically describes the training stream built from the physical system.

Heat diffusion. Heat diffusion systems (Fig. 3) describe how thermal energy propagates in a material, from high temperature regions to low temperature regions. We consider the following form for heat diffusion:

$$\begin{cases} \dfrac{\partial u}{\partial t} = \overbrace{\xi^2 \Delta_x u}^{\text{diffusion}} & \text{in } \Omega \times (0,T] \quad (GE) \\ \langle \nabla_x u, \hat{n} \rangle = g & \text{in } \partial\Omega \times (0,T] \quad (BC) \\ u = h & \text{in } \bar{\Omega} \times \{0\} \quad (IC). \end{cases} \tag{2}$$

The system evolves to minimize the associated energy functional.

Allen-Cahn The Allen-Cahn equation is a reaction-diffusion equation that describes the process of phase separation. In this context, a phase is intended as any region of material whose all possible subregions share the same physical properties. Two different phases present different physical properties. The considered form for Allen-Cahn systems comprises a governing equation including a diffusion term and a reaction term, as follows.

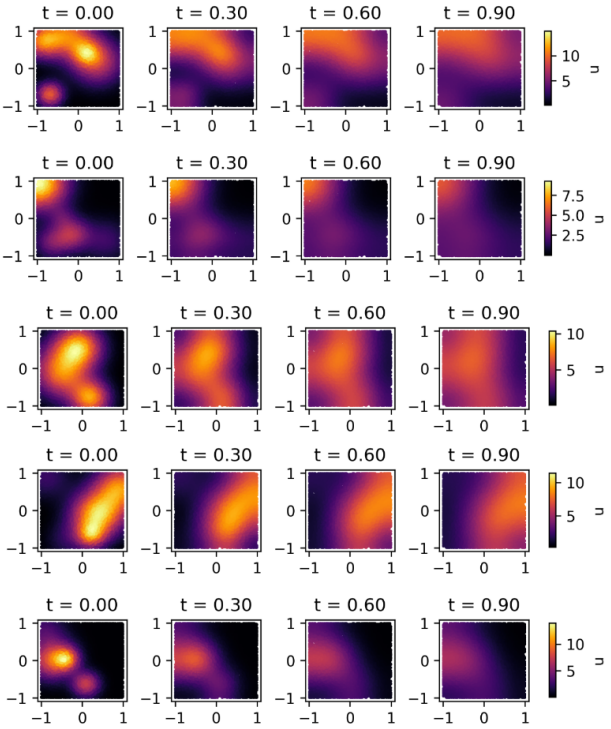

Figure 3. Heat diffusion dataset generated from Eq. 2.

$$\begin{cases} \dfrac{\partial u}{\partial t} = \overbrace{\xi^2 \Delta_x u}^{\text{diffusion}} - \overbrace{\rho u(u^2-1)}^{\text{reaction}} & \text{in } \Omega \times (0,T] \quad (GE) \\ \langle \nabla_x u, \hat{n} \rangle = g & \text{in } \partial\Omega \times (0,T] \quad (BC) \\ u = h & \text{in } \bar{\Omega} \times \{0\} \quad (IC). \end{cases} \tag{3}$$

The continual stream includes 4 different parameterizations per task. The Allen-Cahn dataset is shown in Appendix A.

Evaluation metrics. We report the prediction error on both the domain points and their derivatives. We call these metrics $u_{\text{err}} = \text{MSE}(\hat{\boldsymbol{u}}, \boldsymbol{u})$ and $D_{err} = \text{MSE}(D\hat{\boldsymbol{u}}, D\boldsymbol{u})$, respectively. While the former metric measures the prediction error on the physical system dynamics, the latter measures the discrepancy in predicting the derivative, and it is not necessarily linked with a good predictive model.

4. Results

We compare the performance of Derivative Distillation with EWC and Replay strategies. We also report the performance of Naive fine-tuning and joint training[1]. In all tables, OutD denotes the distillation of *output* $\hat{u}$ instead of the derivative

[1]The code to reproduce all the experiments, including dataset generation, can be found at https://github.com/claudia181/PinnsStudy.

Du. This is akin to traditional distillation in CL and acts as a further comparison. DerD^1 denotes Derivative Distillation. We also experiment with distillation of second-order derivatives D^2u (DerD^2). Each distillation has its own loss term during training. Finally, we also combine different distillations (output + first/second-order derivatives). *In all tables, highlighted values denote the best performance, while bold notation denotes any performance surpassing joint training.*

4.0.1. Heat Diffusion (Time-incremental)

Tables 1 and 2 report the results on the Heat diffusion experiments. It is easy to observe how Derivative Distillation consistently achieves the best results in all cases. Replay is able to accurately predict the derivatives as well, but not the result on the domain points. Moreover, it does so only when the examples require in-domain generalization (interpolation between seen training times). It instead fails when required to generalize out-of-distribution to unseen time periods. We highlight that in the former case, Derivative Distillation outperforms the joint training performance.

Table 1. Heat diffusion on unseen instances of training time intervals.

	u_{err}	D_{err}
Joint	2.64×10^{-3}	1.00×10^{0}
Naive	9.64×10^{-1}	4.85×10^{0}
Replay	7.30×10^{-3}	$\mathbf{9.34 \times 10^{-1}}$
OutD	1.80×10^{-2}	1.09×10^{0}
DerD	$\mathbf{1.13 \times 10^{-3}}$	$\mathbf{9.40 \times 10^{-1}}$
OutD+DerD	2.47×10^{0}	1.64×10^{1}
OutD+DerD^{1,2}	$\mathbf{1.87 \times 10^{-3}}$	$\mathbf{9.33 \times 10^{-1}}$

Table 2. Heat diffusion on time instants unseen during training.

	u_{err}	D_{err}
Joint	4.09×10^{-3}	1.89×10^{-2}
Naive	8.94×10^{-1}	7.29×10^{-1}
Replay	1.09×10^{-2}	7.36×10^{-2}
OutD	1.85×10^{-2}	6.25×10^{-2}
DerD	1.05×10^{-2}	4.70×10^{-2}
OutD+DerD	1.35×10^{-2}	5.70×10^{-2}
OutD+DerD^{1,2}	1.74×10^{-2}	8.65×10^{-2}

4.0.2. Allen-Cahn (Task-incremental)

Tables 3 and 4 report the performance on the Allen-Cahn experiments. These results confirm the findings of the Heat diffusion. Derivative distillation scores the best results both on in-domain generalization (unseen examples from training systems' parameters) and out-of-distribution generalization. In the latter case, this means that the model is able to predict examples from a physical system still belonging to the Allen-Cahn family, but configured with unseen parameters.

Table 3. MSE errors on unseen instances of the training system parameters.

	u_{err}	D_{err}
Joint	3.67×10^{-6}	2.07×10^{-4}
Naive	3.49×10^{-3}	4.49×10^{-2}
EWC	1.34×10^{-3}	1.80×10^{-2}
Replay	2.56×10^{-4}	6.46×10^{-3}
OutD	6.36×10^{-5}	3.70×10^{-3}
DerD	1.72×10^{-4}	1.35×10^{-3}
DerD^2	7.35×10^{-4}	3.52×10^{-3}
OutD+DerD	1.74×10^{-5}	1.00×10^{-3}
OutD+DerD^{1,2}	1.28×10^{-5}	5.19×10^{-4}

Table 4. MSE errors on unseen system parameters.

	u_{err}	D_{err}
Joint	8.22×10^{-4}	1.05×10^{-2}
Naive	1.17×10^{-2}	1.49×10^{-1}
EWC	7.65×10^{-3}	8.11×10^{-2}
Replay	2.38×10^{-3}	3.81×10^{-2}
OutD	1.25×10^{-3}	2.56×10^{-2}
DerD	1.67×10^{-3}	1.65×10^{-2}
DerD^2	6.76×10^{-3}	4.44×10^{-2}
OutD+DerD	8.96×10^{-4}	1.45×10^{-2}
OutD+DerD^{1,2}	1.37×10^{-3}	1.88×10^{-2}

5. Discussion and Future Work

We explored CL in the context of physical systems formulated as initial-boundary value problems, a setting that departs from traditional CL benchmarks by introducing continuous spatio-temporal structure and intrinsically evolving environments. Our results demonstrate that distillation-based approaches, and in particular Derivative Distillation, are highly effective in this domain, surpassing traditional CL baselines, including distillation of outputs $\hat{\boldsymbol{u}}$. When combined with physics-informed approaches, distilling the derivatives enables stable adaptation with minimal forgetting, even without explicit replay mechanisms. These findings highlight that physical systems offer a promising direction for developing and evaluating CL algorithms. Future work may extend this framework to more complex and higher-dimensional systems or investigate the robustness of the approaches in noisy environments.

Acknowledgements

This work has been partially funded by the EU-EIC project EMERGE (grant number 101070918).

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Chen, R. T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. Neural ordinary differential equations, 2019. URL https://arxiv.org/abs/1806.07366.

Coleman, E. N., Quarantiello, L., Liu, Z., Yang, Q., Mukherjee, S., Hurtado, J., and Lomonaco, V. Parameter-Efficient Continual Fine-Tuning: A Survey, August 2025.

Cranmer, M., Greydanus, S., Hoyer, S., Battaglia, P., Spergel, D., and Ho, S. Lagrangian neural networks. In *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations*, 2019. URL https://openreview.net/forum?id=iE8tFa4Nq.

Delange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G., and Tuytelaars, T. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021. ISSN 1939-3539. doi: 10.1109/TPAMI.2021.3057446.

Evans, L. C. *Partial differential equations*. American Mathematical Society, Providence, RI, March 2022.

Greydanus, S., Dzamba, M., and Yosinski, J. Hamiltonian neural networks. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/26cd8ecadce0d4efd6cc8a8725cbd1f8-Paper.pdf.

Hayes, T. L., Krishnan, G. P., Bazhenov, M., Siegelmann, H. T., Sejnowski, T. J., and Kanan, C. Replay in Deep Learning: Current Approaches and Missing Biological Elements. *Neural computation*, 33(11):2908–2950, October 2021. ISSN 0899-7667. doi: 10.1162/neco_a_01433.

Howard, A. A., Fu, Y., and Stinis, P. A multifidelity approach to continual learning for physical systems. *Mach. Learn. Sci. Technol.*, 5(2):25042, 2024. doi: 10.1088/2632-2153/AD45B2. URL https://doi.org/10.1088/2632-2153/ad45b2.

Khetarpal, K., Riemer, M., Rish, I., and Precup, D. Towards Continual Reinforcement Learning: A Review and Perspectives. *Journal of Artificial Intelligence Research*, 75:1401–1476, December 2022. ISSN 1076-9757. doi: 10.1613/jair.1.13673.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D., and Hadsell, R. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, March 2017. doi: 10.1073/pnas.1611835114.

Li, Z. and Hoiem, D. Learning Without Forgetting. In Leibe, B., Matas, J., Sebe, N., and Welling, M. (eds.), *Computer Vision – ECCV 2016*, Lecture Notes in Computer Science, pp. 614–629, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46493-0. doi: 10.1007/978-3-319-46493-0_37.

Li, Z., Kovachki, N. B., Azizzadenesheli, K., liu, B., Bhattacharya, K., Stuart, A., and Anandkumar, A. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=c8P9NQVtmnO.

Lu, L., Jin, P., Pang, G., Zhang, Z., and Karniadakis, G. E. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, March 2021. ISSN 2522-5839. doi: 10.1038/s42256-021-00302-5. URL http://dx.doi.org/10.1038/s42256-021-00302-5.

Lutter, M., Ritter, C., and Peters, J. Deep lagrangian networks: Using physics as model prior for deep learning. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=BklHpjCqKm.

Masana, M., Liu, X., Twardowski, B., Menta, M., Bagdanov, A. D., and van de Weijer, J. Class-Incremental Learning: Survey and Performance Evaluation on Image Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):5513–5533, May 2023. ISSN 1939-3539. doi: 10.1109/TPAMI.2022.3213473.

Melchers, H., Crommelin, D., Koren, B., Menkovski, V., and Sanderse, B. Comparison of neural closure models for discretised pdes. *Computers amp; Mathematics with Applications*, 143:94–107, August 2023. ISSN 0898-1221. doi: 10.1016/j.camwa.2023.04.030.

URL http://dx.doi.org/10.1016/j.camwa.2023.04.030.

Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., and Wermter, S. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, May 2019. ISSN 0893-6080. doi: 10.1016/j.neunet.2019.01.012.

Raissi, M. Deep hidden physics models: Deep learning of nonlinear partial differential equations. *Journal of Machine Learning Research*, 19(25):1–24, 2018. URL http://jmlr.org/papers/v19/18-046.html.

Raissi, M., Perdikaris, P., and Karniadakis, G. E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

Trenta, A., Cossu, A., and Bacciu, D. Learning and Transferring Physical Models through Derivatives. *Transactions on Machine Learning Research*, October 2025. ISSN 2835-8856.

Trenta, A., Cossu, A., and Bacciu, D. Comphy: Composing physical models with end-to-end alignment. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=ER7zDJXtRI.

van de Ven, G. M., Tuytelaars, T., and Tolias, A. S. Three types of incremental learning. *Nature Machine Intelligence*, 4(12):1185–1197, December 2022. ISSN 2522-5839. doi: 10.1038/s42256-022-00568-3.

Wang, S., Sankaran, S., Wang, H., and Perdikaris, P. An expert's guide to training physics-informed neural networks, 2023. URL https://arxiv.org/abs/2308.08468.

A. Allen-Cahn dataset

Similarly to what was reported for the Heat diffusion, Fig. 4 reports the dataset used for the Allen-Cahn experiments.

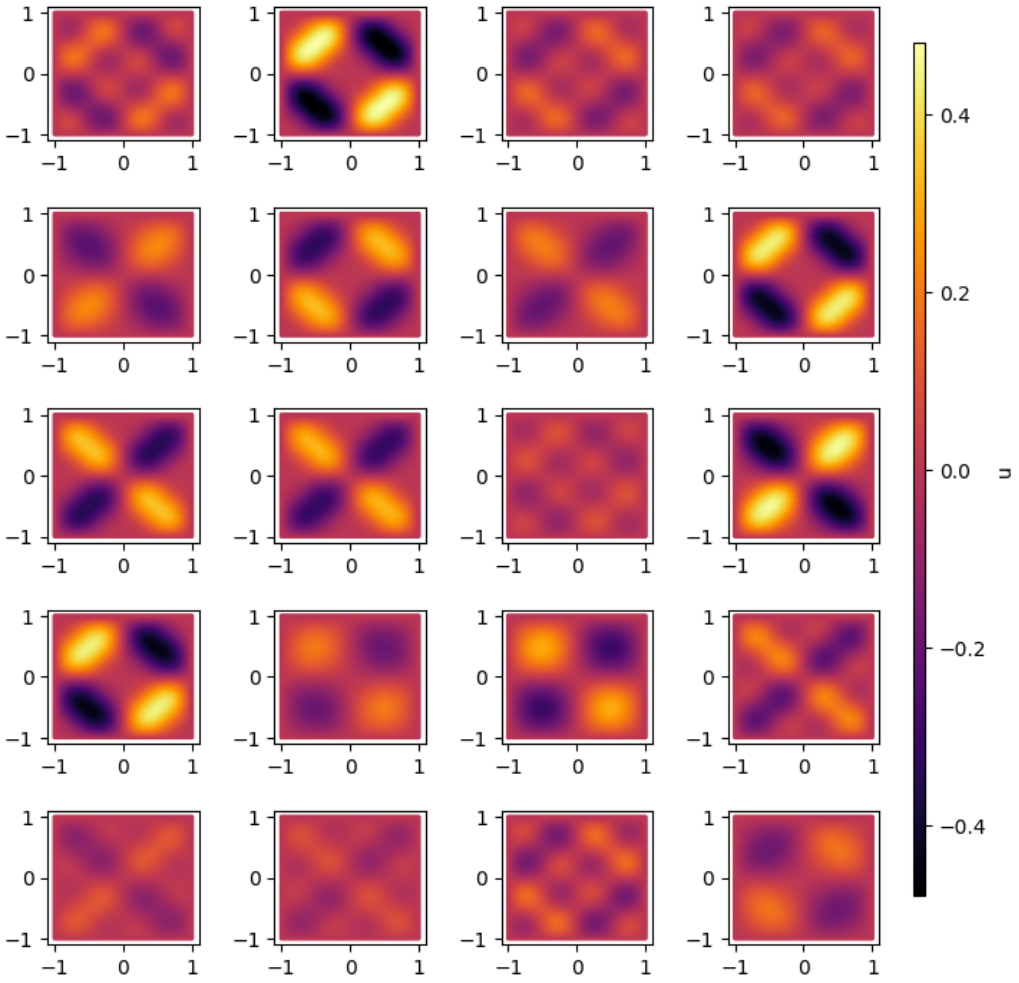

Figure 4. Allen-Cahn dataset.

B. Dataset setup

Table 5 describes the specific configuration used for Allen-Cahn and Heat diffusion systems.

B.1. Heat diffusion experiments

Heat diffusion experiments consider a bidimensional square spatial domain, a diffusion coefficient ξ^2 of 0.1 and 5 heat sources corresponding to 5 Gaussian bumps whose centres (μ_i) are sampled uniformly at random on the 0-valued surface. The standard deviations σ_i of the sources are sampled uniformly at random between 0.25 and 0.5, while the amplitudes A_i are sampled uniformly at random between 0 and 10. Any of the systems is defined using Neumann boundary conditions.

A total of 5 instances have been generated, and the average performances are analysed.

The trajectories are generated by running a fipy solver on the sampled parameters, with a spatio-temporal step of 0.02.

The training trajectories are long 10 time instants. Each training snapshot of the evolving system is made by 2048 points randomly sampled from the spatial surface.

In order to evaluate the models on unseen future temporal instants, for any diffusion system, the 10 snapshots successive to

Table 5. Systems on which the outlined continual physics-informed learning methods have been tested.

Physical system	**Instances**
Heat diffusion $\begin{cases} \frac{\partial u}{\partial t} = \xi^2 \Delta_x u & \text{in } \Omega \times (0,T] \\ \langle \nabla_x u, \hat{n} \rangle = g & \text{in } \partial\Omega \times (0,T] \\ u = h & \text{in } \bar{\Omega} \times \{0\} \end{cases}$	$\Omega = (-1,1)^2$, $\xi^2 = \frac{1}{10}$, $g = x \mapsto 0$, $h = x \mapsto \sum_{i=1}^{5} A_i e^{\frac{-\lVert x-\mu_i \rVert^2}{2\sigma_i^2}}$, where $\mu_i \sim \mathcal{U}(\Omega)$, $\sigma_i \sim \mathcal{U}([\frac{1}{4}, \frac{1}{2}])$, $A_i \sim \mathcal{U}([0,10])$
Allen-Cahn $\begin{cases} 0 = \xi^2 \Delta_{x,y} u_\lambda + u_\lambda(u_\lambda^2 - 1) - f_\lambda & \text{in } \Omega \\ u = g & \text{in } \partial\Omega \end{cases}$ $u_\lambda(x,y) = \sum_{i=1}^{n} \lambda_i \frac{\sin(i\pi x)\sin(i\pi y)}{i^2}$.	$\Omega = (-1,1)^2$, $\xi^2 = \frac{1}{100}$, $n = 2$, $\lambda \sim \mathcal{U}([-1,1]^n)$, $f_\lambda = \xi^2 \Delta_{x,y} u_\lambda + u_\lambda(u_\lambda^2 - 1)$, $g = u_\lambda$

the ones of training are used.

Each in- and out-of-training test snapshot is composed of 512 points, uniformly sampled from the system spatial surface.

For the time-incremental experiments, any task involves 2 consecutive snapshots along the training trajectory.

A model selection phase has been performed only for the joint training and for the first task of each continual learning stream.

The joint model has been trained for 400 epochs with a batch size of 1024.

The total number of continual learning training steps matches the number of training steps of the joint training.

The various objectives (terms) taking part in the training loss function are combined in a weighted linear combination, where the balancing weights are initially set to 1 and then dynamically adapted during training using the loss gradient norm of each term ((Wang et al., 2023)).

B.2. Allen-Cahn experiments

The Allen-Cahn equation that is considered describes a stationary forced version of the corresponding advection-reaction system 3, where f_λ is an external forcing term, defined in such a way as to zero the left-hand side. In these systems, the boundary behaviour is modelled using Dirichlet boundary conditions.

The dataset points are sampled uniformly at random in a bidimensional square spatial domain. The diffusion coefficient ξ^2 is kept fixed to 0.01, and the components of the bidimensional parameter vector λ are sampled uniformly at random from the interval [-1, 1].

A total of 20 λ parameter configurations are generated, 16 used for training and 4 taking part in an external test set (out of the training distribution). For each configuration, 2048 spatial points are used as development set and 512 as test set.

The models taking part in the space-incremental experiments are input with the spatial information and the parameter components. Each spatial task is a quadrant of the square domain and involves all 16 parameter configurations.

The models taking part in the parameter-incremental experiments are input with the spatial information and the parameter components. Each task is a set of 4 parameter vectors, for a total of 4 tasks.

All the models approximating the considered Allen-Cahn systems make use of an encoding designed to tackle high-frequency functions, namely a random Fourier features encoding ((Wang et al., 2023)).

A model selection phase has been performed only for the joint training and for the first task of each continual learning stream.

The joint model has been trained for 200 epochs with a batch size of 1024.

The total number of continual learning training steps matches the number of training steps of the joint training.

The various objectives (terms) taking part in the training loss function are combined in a weighted linear combination, where the balancing weights are initially set to 1 and then dynamically adapted during training using the loss gradient norm of each term ((Wang et al., 2023)).

B.3. Non-stationary Allen-Cahn

Table 6. Non-stationary Allen-Cahn on unseen instances of training time intervals.

	$\hat{\mathcal{L}}[u^{\theta^*}, u]$	$\hat{\mathcal{L}}[\nabla_{x,t} u^{\theta^*}, \nabla_{x,t} u]$	$\hat{\mathcal{L}}[\mathcal{G}[u^{\theta^*}], \mathcal{G}[u]]$
Joint	1.21×10^{-5}	2.80×10^{-3}	5.72×10^{-5}
Forget	3.93×10^{-2}	1.64×10^{-1}	1.16×10^{-1}
Replay	9.36×10^{-4}	2.20×10^{-2}	1.83×10^{-4}
OutD	1.57×10^{-3}	2.53×10^{-2}	4.52×10^{-3}
DerD	2.83×10^{-5}	3.06×10^{-3}	2.51×10^{-4}
OutD+DerD	4.13×10^{-5}	3.37×10^{-3}	3.27×10^{-4}
OutD+DerD^{1,2}	8.48×10^{-5}	5.73×10^{-3}	1.02×10^{-3}

Table 7. Non-stationary Allen-Cahn on time instants unseen during training.

	$\hat{\mathcal{L}}[u^{\theta^*}, u]$	$\hat{\mathcal{L}}[\nabla_{x,t} u^{\theta^*}, \nabla_{x,t} u]$	$\hat{\mathcal{L}}[\mathcal{G}[u^{\theta^*}], \mathcal{G}[u]]$
Joint	2.28×10^{-4}	3.12×10^{-3}	2.05×10^{-3}
Forget	9.75×10^{-2}	2.74×10^{-1}	5.96×10^{-3}
Replay	2.23×10^{-3}	2.48×10^{-2}	9.94×10^{-3}
OutD	3.88×10^{-3}	3.56×10^{-2}	1.13×10^{-2}
DerD	1.08×10^{-3}	1.24×10^{-2}	8.08×10^{-3}
OutD+DerD	1.31×10^{-3}	1.47×10^{-2}	1.03×10^{-2}
OutD+DerD^{1,2}	1.13×10^{-3}	1.47×10^{-2}	8.27×10^{-3}

Table 8. Systems on which the outlined continual physics-informed learning methods have been tested.

Physical system	**Instances**
Allen-Cahn equations (reaction-diffusion) $\begin{cases} \frac{\partial u}{\partial t} = \xi^2 \Delta_x u - \rho u(u^2 - 1) & \text{in } \Omega \times (0, T] \\ \langle \nabla_x u, \hat{n} \rangle = g & \text{in } \partial\Omega \times (0, T] \\ u = h & \text{in } \bar{\Omega} \times \{0\} \end{cases}$	$\Omega = (-1, 1)^2$, $\xi^2 = \frac{1}{100}$, $\rho = 1$, $g = x \mapsto 0$, $h = x \mapsto \sum_{i=1}^{5} A_i e^{\frac{-\lVert x - \mu_i \rVert^2}{2\sigma_i^2}}$, where $\mu_i \sim \mathcal{U}(\Omega)$, $\sigma_i \sim \mathcal{U}([\frac{1}{4}, \frac{1}{2}])$, $A_i \sim \mathcal{U}([-1, 1])$

The experiments on non-stationary Allen-Cahn systems follow an experimental setup analogous to that of heat diffusion systems.

The additional element is the reaction parameter ρ, which is set to 1.

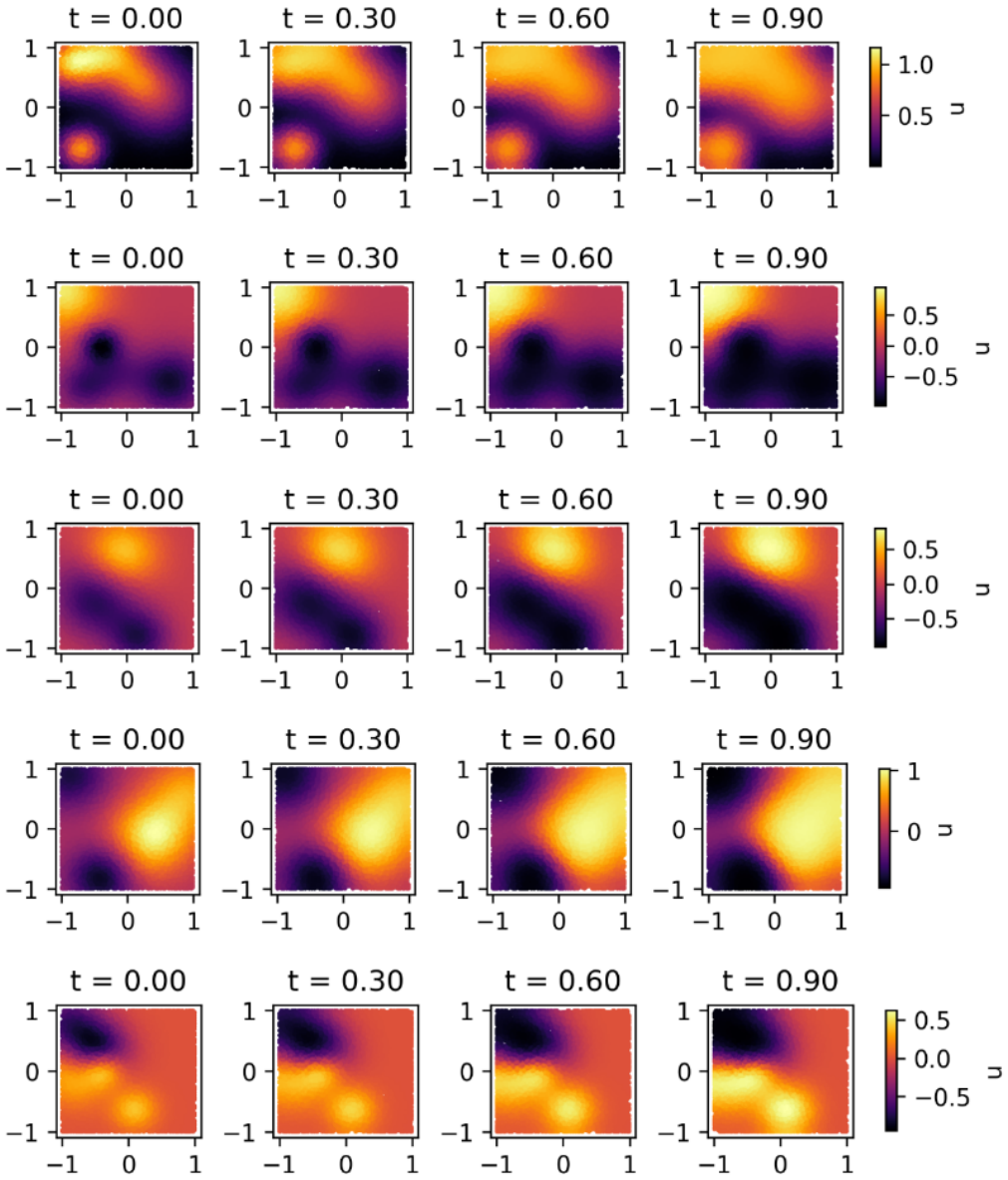

Figure 5. Non-stationary Allen-Cahn dataset.