ACCELERATING PINN TRAINING VIA RL-BASED ADAPTIVE LOSS CONTROL

Vladislav Kolzhetsov
Novosibirsk State University,
Novosibirsk, Russia
v.kolzhetsov@g.nsu.ru

Andrei Zakharov
AXXX, Moscow, Russia
Innopolis University,
Innopolis, Russia
andzkrv@gmail.com

Ilya Makarov
AXXX, Moscow, Russia
Innopolis University,
Innopolis, Russia
Trusted AI Center, RAS,
Moscow, Russia
iamakarov@hse.ru

ABSTRACT

Physics-Informed Neural Networks have demonstrated strong performance across scientific and engineering applications by integrating governing physical laws, expressed as partial differential equations, into neural training. However, their practical use is often limited by slow and unstable training. A key factor is the balancing multiple terms within the loss function, typically with static, user-defined weighting coefficients. In this work we show that PINN training can be sufficiently accelerated by employing a reinforcement learning agent to dynamically adapt these loss weights. We formulate loss balancing as a sequential decision-making problem and demonstrate that an RL-driven policy reduces the number of training iterations required to reach a target accuracy compared to static weighting schemes. Numerical experiments on canonical PDE benchmarks show consistent convergence acceleration without loss of solution fidelity, highlighting the potential of adaptive loss control for more efficient physics-informed learning.

1 INTRODUCTION

Physics-Informed Neural Networks (PINNs) Raissi et al. (2019) have emerged as a powerful and flexible tool for solving forward and inverse problems described by partial differential equations (PDEs). PINNs incorporate fundamental physical laws, expressed in the form of PDEs, directly into the loss function as a regularizer. This allows them to achieve high accuracy even with limited observational data, expanding their application to complex multi-scale systems.

However, PINNs still face significant limitations. One of the most fundamental is the problem of gradient imbalance, which arises due to the competitive nature of the multi-component loss function. Following Wang et al. (2020), the total loss is a weighted sum of multiple terms. In this work we consider the PDE, initial condition (IC), and boundary condition (BC) losses: $L_{total} = \lambda_{PDE} L_{PDE} + \lambda_{IC} L_{IC} + \lambda_{BC} L_{BC}$.

To address this issue, a number of adaptive weight balancing methods have been proposed. These include approaches based on tracking gradient norms Chen et al. (2018), Bischof & Kraus (2025) or curriculum regularization Krishnapriyan et al. (2021). These methods adapt weights based on the current or recent state of the system, without considering the long-term consequences of these adjustments for the optimization process. Such local adaptation can lead to suboptimal learning trajectories, oscillations, and a lack of convergence.

In this work, we propose a different perspective on the balancing problem by formulating it as a sequential decision-making problem. The PINN training process is considered as an interaction with an environment, where at each step an agent must choose optimal weights λ for the loss components to maximize the cumulative reward corresponding to the ultimate goal: fast and accurate convergence. To approach this problem, we propose applying reinforcement learning (RL) Sutton & Barto (2018) methods, enabling the agent to develop a proactive strategy that not only responds to the current imbalance but also anticipates the future evolution of the optimization process, selecting actions that are optimal in the long term (The general scheme is shown in the 1).

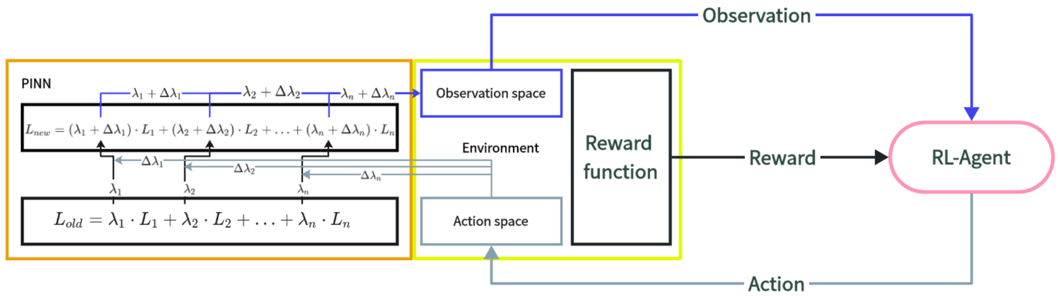

Figure 1: General outline of the proposed approach

The main contributions of this work are following: (i) the formulation of PINN loss weighting as an RL control problem, (ii) the design of a minimal and scalable RL environment, and (iii) empirical validation on benchmark PDEs.

2 Environment and Agent Development

We define the RL environment by constructing three core components: the state space, the action space, and the reward function. The underlying PINN architecture and PDE problem are treated as modular components and are not fixed by the environment design, enabling broad applicability. Thus, specific details of the PINN are not discussed.

2.1 Observation Space Construction

The objective of an agent is to adapt the coefficients of the PINN loss function constituent terms to accelerate convergence. Therefore, a minimal representation of the state space consists of loss weights $s_t = (\lambda_{PDE}, \lambda_{IC}, \lambda_{BC}) \in \mathbb{R}^3$. Reducing the dimensionality of the state space accelerates learning and avoids unnecessary complexity of the policy search space Lu et al. (2024). Therefore, we include only the loss function coefficients as state variables.

2.2 Action Space Construction

Following the same principle, we constrain the action space to the smallest dimensionality sufficient to enable meaningful control. Since the agent must independently adjust each coefficient, the minimal action dimension is three, corresponding to the three coefficients for optimization. Each action is defined as an additive perturbation applied to the current state: $s_{t+1} = s_t + a_t$, where $s_t \in \mathbb{R}^3$ is the state vector of coefficients at timestep t, and $a_t \in \mathbb{R}^3$ is the action vector.

2.3 Reward Function Construction

The primary objective is to accelerate PINN training through adaptive loss function weighting. Thus, the reward function must encode a meaningful signal of relative improvement over a baseline training. To implement this, we define acceleration as follows:

1. **Baseline training:** Train the PINN using a fixed loss function with all coefficients set to one and save the loss trajectory $L_{baseline}(t)$.
2. **RL-Guided training:** Train the PINN under the guidance of the RL agent, with dynamically adjusted coefficients. Save the corresponding loss trajectory $L_{current}(t)$.
3. **Acceleration criterion:** At each evaluation timestep t, training is considered accelerated if $L_{current}(t) < L_{baseline}(t)$; otherwise, no training acceleration is observed.

The reward is based on two principles: **Reward shaping and dense rewards.** The idea is to divide the PINN's learning process into a number of checkpoints, and give the agent a reward at a checkpoint equal to the estimated number of epochs by which it is ahead of the baseline. We implement this temporal shaping as follows:

$$S(t) = -\frac{1}{\tau}(L_{baseline}^{-1}(L(t)) - L_{current}^{-1}(L(t-\tau))) + \frac{1}{\tau}(L_{baseline}^{-1}(L(t+\tau)) - L_{current}^{-1}(L(t))) \quad (1)$$

Scale invariance.

$$N(t) = \frac{1}{\tau}\sum_{i=1}^{\tau} \frac{L_{baseline}^{(i)}(t) - L_{current}^{(i)}(t)}{L_{baseline}^{(i)}(t)}, \quad (2)$$

where τ is the number of epochs between two checkpoints.

The total reward at timestep t is the sum of the normalized gain and the temporal shaping component $R(t) = N(t) + S(t)$.

2.4 Agent development

Given that both the state and action spaces in the proposed environment are continuous, several reinforcement learning (RL) agent architectures may be employed. Three distinct approaches were considered, differing in their treatment of state-action discretization, agent architecture, and multi-agent design:

(i) Discretization-based approach: The continuous state and action spaces may be discretized into finite bins, enabling the use of Deep Q-Networks (DQN) as the underlying RL algorithm Mnih et al. (2013).

(ii) Continuous-state, single-agent approach: To avoid the pitfalls of discretization, we employed a continuous control framework using Deep Deterministic Policy Gradient (DDPG) Lillicrap et al. (2019). In this setting, the state and the action spaces remain continuous. To enable the agent to model temporal dynamics, specifically, to distinguish between early and late phases of training, we augmented the state vector with a scalar representing elapsed training time. Rewards are evaluated periodically every 500 training epochs (i.e., 15 times per episode, where one episode corresponds to 8000 epochs of PINN training). Even this approach is sensitive to hyperparameter tuning, it achieves convergence in about 80 episodes, significantly improving sample efficiency.

(iii) Multi-agent, decentralized approach: We further propose a multi-agent architecture in which each agent controls the loss function coefficients over a fixed interval between consecutive reward checkpoints (i.e., one agent per 500-epoch segment) Lee & Jeong (2021). These agents operate independently, where each receives rewards only within its designated temporal window. Because each agent operates within a fixed, short horizon, the explicit inclusion of training time as a state variable becomes unnecessary, the temporal context is implicitly encoded by the agent's assigned interval. All agents are trained using DDPG, leveraging the continuous control capabilities of the algorithm. This approach decouples learning across time segments, potentially improving stability and reducing interference between distant phases of training.

3 Experiments and results

We evaluate the proposed method using a PINN solving the one-dimensional homogeneous heat equation. Comparison of learning trajectories for the baseline and the RL-guided PINN is shown in Figure 2. The RL-guided approach achieves a consistent convergence acceleration of approximately 25%, measured as a reduction in the number of epochs required to reach a target loss. Importantly, the fidelity of the learned solution is preserved as the relative error between the RL-guided solution and the baseline solution remains below $10^{-8}\%$, with the overlapping confidence intervals in the solution profiles. This confirms that the speedup is achieved without compromising solution accuracy. Additional results for the nonlinear Schrödinger equation and the incompressible Navier–Stokes equations are provided in the Appendix (Figs. 3 and 4).

4 Directions for future research

To place these findings in broader context and to outline future research directions, we identify three promising avenues for extension:

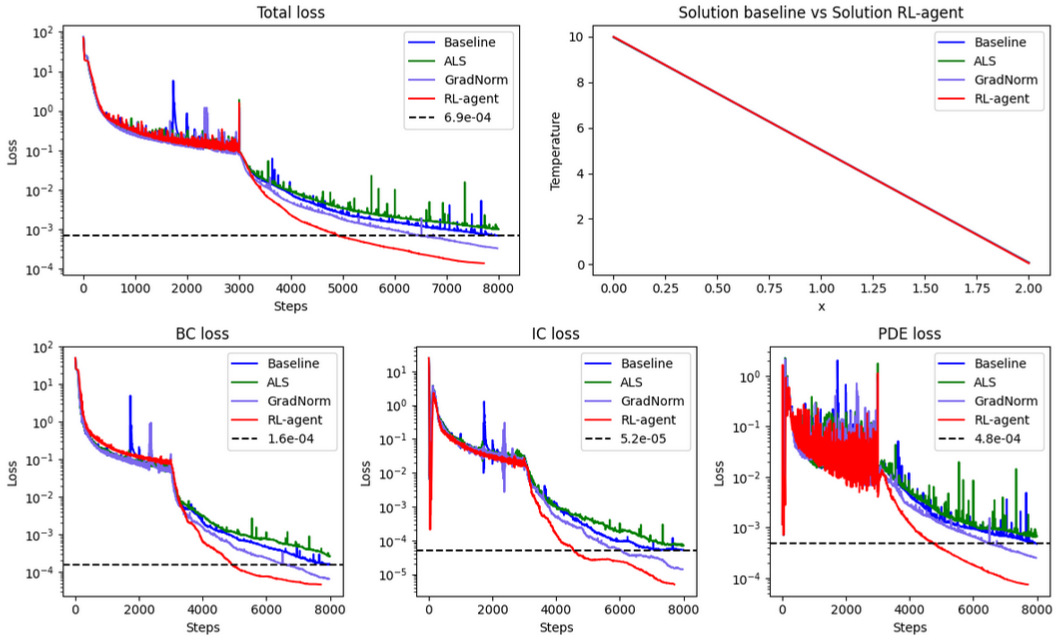

Figure 2: Comparative performance of loss weighting strategies for the 1D heat equation solved using PINNs. The blue, violet, green, and red curves correspond to the baseline, GradNorm, Adaptive Loss Scaling, and RL-based approaches, respectively. Top left: Evolution of the total composite loss during training. Top right: Predicted temperature profiles at the final training epoch for each method. Bottom row: Individual loss components. Boundary condition residual (left), initial condition residual (center), and PDE residual (right) highlight the balance and convergence behavior across methods. The black dashed line in each subplot denotes the minimum loss achieved by the baseline method, serving as a performance reference.

Generalization to Complex PDE Systems. Further extensions may include integro-differential equations, such as the Dyson-Schwinger, or operator-theoretic formulations. Such generalizations would test the robustness of adaptive loss weighting in regimes with complex solution manifolds and nonlocal dependencies.

Systematic Analysis of Hyperparameter Sensitivity. The performance of the RL agent is sensitive to the particular choice of hyperparameters (learning rates, reward scaling and network architecture). A systematic ablation study will be beneficial to quantify the impact of these parameters on convergence stability, solution accuracy, and generalization across problems. This would enable the development of guidelines for hyperparameter selection and potentially inform the design of meta-learning strategies for automatic configuration.

Architectural Robustness of PINNs under RL Guidance. It remains an open question how extensively the underlying PINN architecture can be modified without degrading the beneficial effects induced by the RL agent. It will be critical for designing hybrid architectures that jointly optimize learning dynamics and representational capacity.

In summary, this work demonstrates that RL can effectively accelerate PINN training while preserving solution fidelity. The proposed framework provides a scalable, paradigm for adaptive physics-informed learning, opening new pathways for the automated optimization of neural solvers in scientific computing.

ACKNOWLEDGMENTS

This work was supported by the Ministry of Economic Development of the Russian Federation (agreement No. 139-10-2025-034 dd. 19.06.2025, IGK 000000C313925P4D0002).

REFERENCES

Rafael Bischof and Michael A. Kraus. Multi-objective loss balancing for physics-informed deep learning. *Computer Methods in Applied Mechanics and Engineering*, 439:117914, May 2025. ISSN 0045-7825. doi: 10.1016/j.cma.2025.117914. URL http://dx.doi.org/10.1016/j.cma.2025.117914.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks, 2018. URL https://arxiv.org/abs/1711.02257.

Aditi S. Krishnapriyan, Amir Gholami, Shandian Zhe, Robert M. Kirby, and Michael W. Mahoney. Characterizing possible failure modes in physics-informed neural networks, 2021. URL https://arxiv.org/abs/2109.01050.

Heunchul Lee and Jaeseong Jeong. Multi-agent deep reinforcement learning (madrl) meets multi-user mimo systems, 2021. URL https://arxiv.org/abs/2109.04986.

Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning, 2019. URL https://arxiv.org/abs/1509.02971.

Chenbei Lu, Laixi Shi, Zaiwei Chen, Chenye Wu, and Adam Wierman. Overcoming the curse of dimensionality in reinforcement learning through approximate factorization, 2024. URL https://arxiv.org/abs/2411.07591.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning, 2013. URL https://arxiv.org/abs/1312.5602.

M. Raissi, P. Perdikaris, and G.E. Karniadakis. 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. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2018.10.045. URL https://www.sciencedirect.com/science/article/pii/S0021999118307125.

Pratik Rathore, Weimu Lei, Zachary Frangella, Lu Lu, and Madeleine Udell. Challenges in training pinns: A loss landscape perspective, 2024. URL https://arxiv.org/abs/2402.01868.

Richard S. Sutton and Andrew G. Barto. *Reinforcement learning: an introduction*. Adaptive computation and machine learning series. The MIT Press, Cambridge, Massachusetts, second edition edition, 2018. ISBN 978-0-262-03924-6.

Sifan Wang, Yujun Teng, and Paris Perdikaris. Understanding and mitigating gradient pathologies in physics-informed neural networks, 2020. URL https://arxiv.org/abs/2001.04536.

A APPENDIX

The GradNorm method is adopted from Chen et al. (2018). Method, called Adaptive Loss Scaling (ALS), is the author's adaptation presented in the Bischof & Kraus (2025). RL-based approach is represented by multiagent variation. During training, Adam and L-BFGS are used, which explains the break in the graphs at the point corresponding to 3000 epochs. Rathore et al. (2024).

We also provide results obtained by applying the proposed method to the one-dimensional nonlinear Schrodinger equation with cubic nonlinearity, which governs the dynamics of the wave function describing a Bose-Einstein condensate in the form of a soliton (Fig.3). Also, we show the performance of the method applying it to the solution of the incompressible Navier–Stokes equations for laminar flow past a circular cylinder confined within a two-dimensional flat channel (Fig.4).

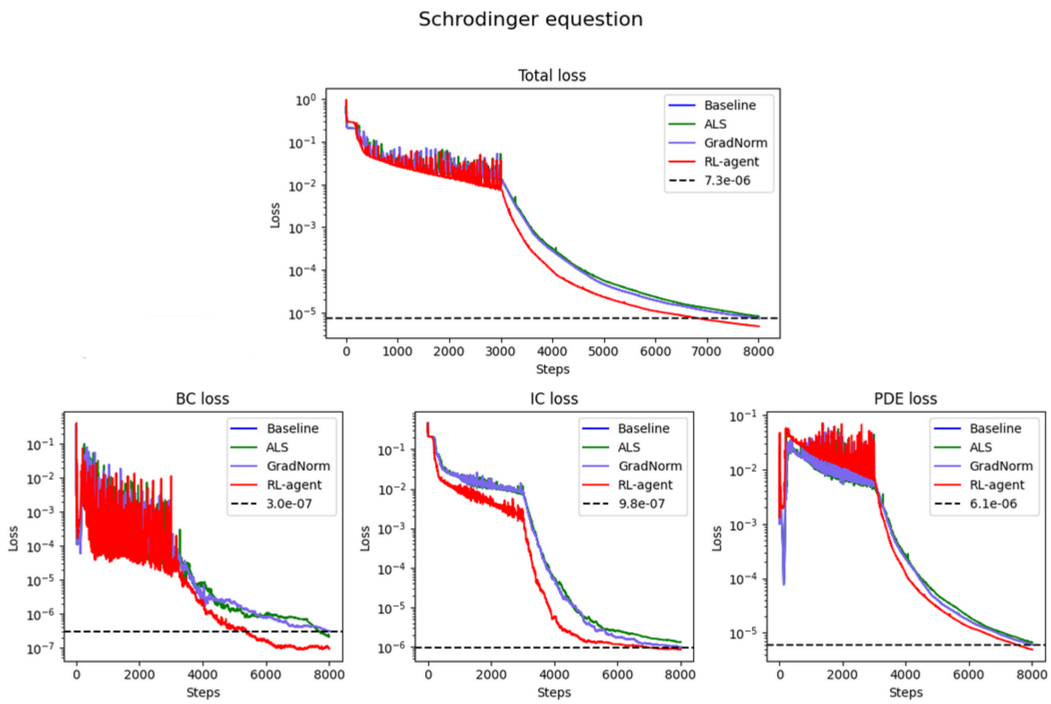

Figure 3: Comparative performance of loss weighting strategies for the Schrodinger equation solved via PINNs. The blue, violet, green, and red curves correspond to the baseline, GradNorm, Adaptive Loss Scaling, and RL-based approaches, respectively. Top: Evolution of the total composite loss during training. Bottom row: Individual loss components – boundary condition residual (left), initial condition residual (center), and PDE residual (right) – highlighting the balance and convergence behavior across methods. The black dashed line in each subplot denotes the minimum loss achieved by the baseline method, serving as a performance reference.

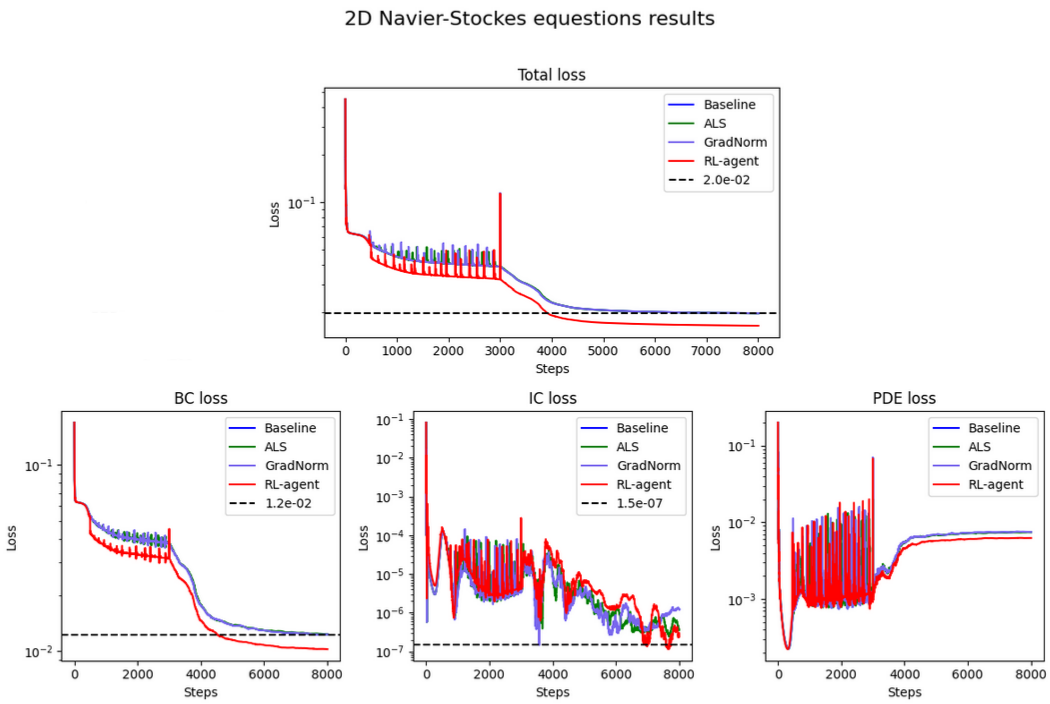

Figure 4: Comparative performance of loss weighting strategies for the Navier-Stokes equation solved via PINNs. The blue, violet, green, and red curves correspond to the baseline, GradNorm, Adaptive Loss Scaling, and RL-based approaches, respectively. Top: Evolution of the total composite loss during training. Bottom row: Individual loss components – boundary condition residual (left), initial condition residual (center), and PDE residual (right) – highlighting the balance and convergence behavior across methods. The black dashed line in each subplot denotes the minimum loss achieved by the baseline method, serving as a performance reference.