Analyzing the Impact of Adaptive Weighting in Self-Adaptive Physics-Informed Neural Networks for Solving PDEs

TU Delft

1. PINN Introduction

Physics-Informed Neural Networks (PINNs) are a significant advancement in solving partial differential equations (PDEs) by incorporating physical laws directly into the neural network architecture [1]. Proposed by [2] they aim to minimize the error of the PDE, initial, and boundary conditions simultaneously, embedding domain knowledge into the learning process [3].

Here is how a standard PINN loss function looks like:

$$L(w) = L_r(w) + L_b(w) + L_0(w)$$

However, they are known to struggle when modeling PDEs that are "stiff", that means, with solutions characterized by sharp spatial transitions or fast time evolution [4].

2. SA-PINN Main Idea

Self-Adaptive Physics-Informed Neural Networks (SA-PINNs) introduced by [3] aim to solve the problem with convergence when predicting stiff nonlinear equations.

SA-PINNs applies fully trainable adaptation weight mask to each training point, which allows the network to autonomously focus on challenging regions of the solution. This mask is a core innovation of SA-PINNs and it works by introducing adaptation weights that are updated alongside the neural network parameters during training.

Here is how the loss function looks like for SA-PINNs:

$$L(w, \lambda_r, \lambda_b, \lambda_0) = L_r(w, \lambda_r) + L_b(w, \lambda_b) + L_0(w, \lambda_0)$$

Where:

- $L_r(w, \lambda_r)$ is the error caused by not satisfying the PDE
- $L_b(w, \lambda_b)$ is the error caused by not satisfying the boundary conditions
- $L_0(w, \lambda_0)$ is the error caused by not satisfying the initial condition

The self-adaptation weights are essentially vectors with dimensions equal to the number of points in the training set.

The adaptive weights are updated using gradient ascent and the neural network weights are updated using standard gradient descent.

3. Research Question

Which loss components contribute most significantly to the solution accuracy and how do the adaptive weights affect this contribution?

4. Methodology

Five different scenarios were trained, for 3 different epoch categories (100, 1000, and 10000 epochs).

Network configuration stays the same between the scenarios to ensure robustness of the experiments.

	All Masks Off	All Masks On	Residual Mask On	Boundary Condition Mask On	Initial Condition Mask On
Residual Mask	Off	On	On	Off	Off
Boundary Mask	Off	On	Off	On	Off
Initial Mask	Off	On	Off	Off	On

Each model undergoes two stages of optimization, first using Adam optimizer, and then L-BFGS.

Adaptive weights are only updated when training with Adam, and they are held constant when training with L-BFGS. This is the training setup used by [3].

The PDE the model is trained to predict is the Viscous Burgers equation defined as follows:

$$u_t + uu_x - (0.01/\pi)u_{xx} = 0, x \in [-1, 1], t \in [0, 1],$$

$$u(0, x) = -\sin(\pi x),$$

$$u(t, -1) = u(t, 1) = 0.$$

For each scenario, five independent models are trained to ensure robustness and reduce variability.

5. Results

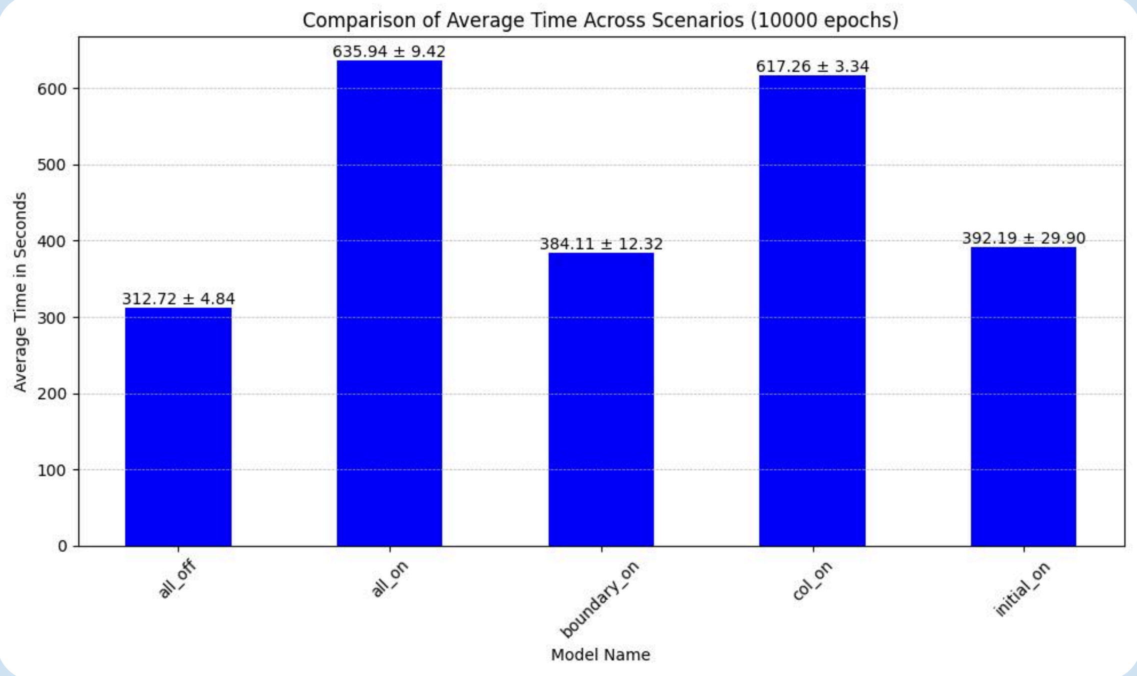
The results were evaluated based on two key metrics:

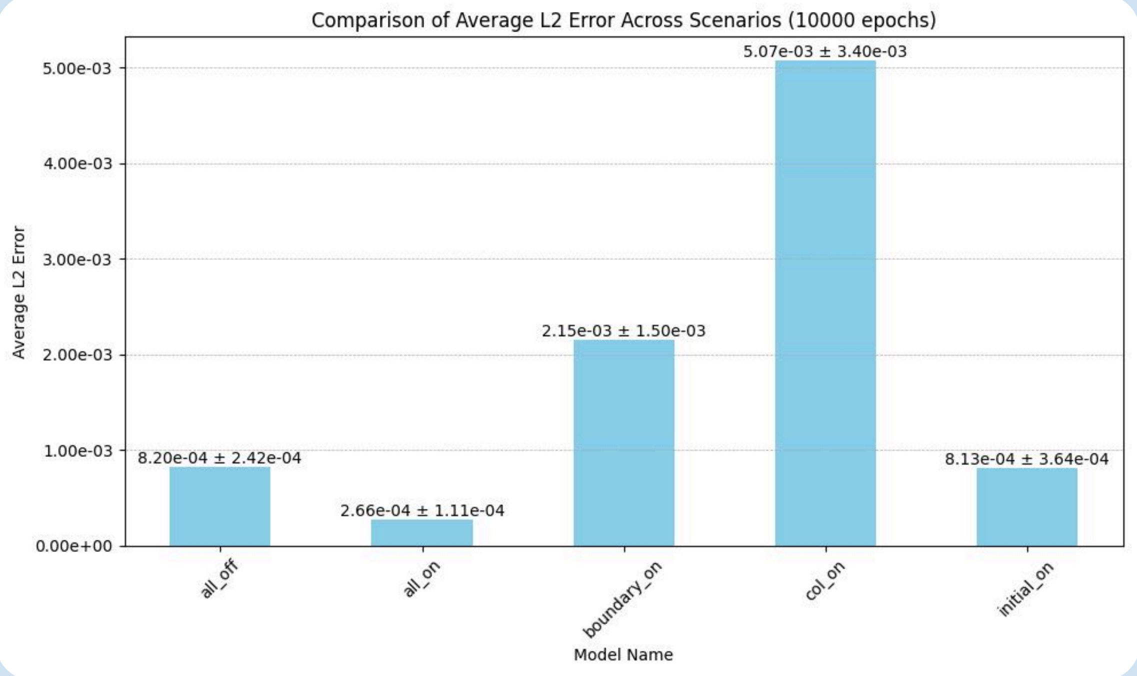
- Time to train in seconds
- L2-error given by:

$$\mathrm{L}_2 \text{ error} = \frac{\sqrt{\sum_{i=1}^{N_U} |u(x_i, t_i) - U(x_i, t_i)|^2}}{\sqrt{\sum_{i=1}^{N_U} |U(x_i, t_i)|^2}}$$

The graphs represent the average across the five runs, as well as the standard deviation. The L2-error directly translates to the accuracy of the model, while the time it took to train shows which configurations are the fastest.

These two benchmarks give a good overview of the model performance.





6. Conclusions and future work

The results align with the findings of [3], confirming that SA-PINNs benefit from the self-adaptive weights and demonstrate increased performance when tested on challenging problems like Viscous Burgers Equation. However, while the accuracy increases the training times do as well. Full SA-PINN implementation took over two times longer to train than the baseline PINN. Also, the result show that some masks might be worth using without the full SA-PINN implementation, like the initial conditions mask that demonstrated improved performance across scenarios when training for 1000 and 10000 epochs while not increasing the training time as much as the full SA-PINN implementation.

Future research could explore whether these results are replicable for other PDEs to assess the generalizability of the findings.

References

[1] - Sifan Wang, Xinling Yu, and Paris Perdikaris. When and why pinns fail to train: A neural tangent kernel perspective. Journal of Computational Physics, 449:110768, 2022
[2] - M. Raissi, P. Perdikaris, and G.E. Karniadakis. Physicsinformed 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.
[3] - Levi D. McClenny and Ulisses M. Braga-Neto. Selfadaptive physics-informed neural networks. Journal of Computational Physics, 474:111722, February 2023
[4] - Sifan Wang, Yujun Teng, and Paris Perdikaris. Understanding and mitigating gradient pathologies in physicsinformed neural networks, 2020.

Author: Jakub Mankowski
Supervisors: Dr. Jing Sun, Dr. Alexander Heinlein, Dr. Tiexing Wang