SvedbergOpen
DISSEMINATION OF KNOWLEDGE
International Journal of Artificial Intelligence and Machine Learning
Publisher's Home Page: https://www.svedbergopen.com/
Research Paper
Open Access

# Logical Constraint Regularization Algorithms For Physics-Informed Deep Learning

P. Sagayaraj[1*], R. Harshini[2], Dr. Ravinder Sharma[3], Iplina Antonina Aleksandrovna[4], Dr Chikati Madhava Rao[5]

[1*]Professor, Meenakshi College of Arts and Science, Meenakshi Academy of Higher Education and Research, Chennai, Tamil Nadu, India. E-mail: sagayaraj@maher.ac.in
[2]Assistant Professor, Department of Computer Science, Meenakshi College of Arts and Science, Meenakshi Academy of Higher Education and Research, Chennai, Tamil Nadu, India. E-mail: harshinir@maher.ac.in
[3]Assistant Professor, Kalinga University, Naya Raipur, Chhattisgarh, India. E-mail: ku.ravindersharma@kalingauniversity.ac.in, https://orcid.org/0009-0000-9569-6351
[4]Turan International University, Namangan, Uzbekistan. E-mail: antipmoon@mail.ru, https://orcid.org/0009-0006-7606-1485
[5]Associate Professor, Dept of CSE, CMR Technical Campus, Hyderabad, Telangana, India.
E-mail: madhav.chikati@gmail.com
*Corresponding author: Email: sagayaraj@maher.ac.in

## Abstract

The physics-informed neural network (PINN) paradigm has been developed to add domain knowledge into deep learning models to tackle partial differential equations (PDEs) and inverse problems. Setting these physical parameters during training, however, is still a major challenge and often results in suboptimal convergence and less accurate solutions to the complex nonlinear systems. In this paper, research propose a new family of Logical Constraint Regularization (LCR) algorithms that incorporate physics-based regularization terms in an adaptive, systematic manner. The proposed approach is formulated as a multi-objective optimization problem, and it uses hierarchical constraint ordering to enhance convergence rate and solution quality. The results are presented with experimental validation for canonical problems such as Burger's equation, Navier-Stokes equations and inverse problems, showing great improvements in prediction accuracy (up to 87.3% relative error reduction) and computational efficiency (40% reduction in convergence time). The method yields a mean absolute percentage error (MAPE) of 2.14% on test datasets, which is significantly better than the baseline PINN implementation. The results show that logical constraint regularization is a general regularization framework which can be applied to a wide range of physical systems and it is a useful addition to scientific machine learning.

**Keywords:** This article highlights various methods for physics-informed neural networks, including constraint regularization, multi-objective optimization, and the use of adaptive weighting mechanisms. It describes many approaches to physics-informed neural networks, such as constraint regularization, multi-objective optimization, and adaptive weighting mechanisms.

## 1. Introduction

Embedding the physical laws in machine learning models is a big paradigm shift in scientific computing. In addition to the traditional numerical techniques, which are mostly used to solve partial differential equations (PDEs), data-driven techniques have been incorporated and, in some cases, even outperformed, especially deep neural networks. While first introduced by [1], physics-informed neural networks (PINNs) embed governing equations into the loss function, which allows the network to satisfy the physics equation when learning from sparse or noisy data [2]. However, PINNs have significant challenges in enforcing the complex physical constraints, especially when the physics is highly nonlinear [3]. The difficulty arises from the combination of optimization objectives, including minimizing data-fitting errors, enforcing boundary conditions, and satisfying differential equation constraints [4]. Traditional PINN formulations are based on a linear combination of these terms, leading to conflicts between gradients and convergence stagnation. Further, most of the relative importance of different types of constraints has to be manually adjusted and hence is not generalizable across problem domains [5]. In recent years, new developments have included adaptive weighting strategies and modifications to the loss function, but most of these schemes are not well understood or lack a well-structured theory [6]. In this paper, the study tackles these limitations by introducing Logical Constraint Regularization (LCR) algorithms, which aim at making constraint

enforcement a formal multi-objective optimization problem. The crucial improvement is hierarchical constraint ordering and adaptive regularization coefficients that are updated based on constraint satisfaction measures.
This work mainly provides:

- Derivation of the theoretical framework for development of a logical constraint regularization. Using hierarchy for constraint ordering to resolve gradient-conflicts.
- The adaptive weighting mechanisms respond to constraint satisfaction progress, and extensive validation on a variety of canonical PDE problems, quantitative performance metrics, with significant improvements over baseline methods.

The rest of the paper is structured according to a typical engineering optimization procedure: Section II introduces the physics-informed neural network and constraint regularization techniques, and Section III develops approaches to multi-objective optimization methods. In Section II the Logical Constraint Regularization (LCR) framework is introduced, which includes hierarchical constraint ordering, adaptive weighting and logical gating mechanisms. The experimental setup is described in Section III, covering network architectures, hyperparameters, baseline methods and performance metrics. In Section IV, quantitative results and analysis are provided for canonical problems in the PDE community, where the proposed method is compared with other approaches. The underlying mechanisms, generalizability and implications of the LCR framework are discussed in section V. The paper ends with Section VI, which summarizes the directions for future research.

## 2. Related Work

Physics-informed neural networks have shown their effectiveness in a variety of fields from fluid mechanics to the quantum world. Raissi and colleagues set the stage by adding the PDE residuals directly into the loss functions of a network that is trained to meet a PDE [7]. This was later applied to inverse problems, operator learning and coupled multi-physics simulations. S. however, systematic investigations have shown that basic PINN formulation is not suitable for constraint enforcement in stiff problems and for nonlinear systems [8]. The multi-objective optimization approaches are gaining popularity for machine learning problems. Dynamic loss-weighting techniques that vary the magnitudes of the coefficients during training are promising for addressing the problem of "competing objectives" [9]. Several approaches have been considered, such as gradient-based adaptive weighting, orthogonal projection methods, and Pareto front methods. Recent papers have shown that loss magnitude can drive constraint-specific scaling, yielding enhanced convergence [10]. However, only a few of these strategies incorporate a logical ordering of constraints alongside adaptive learning mechanisms for physics-informed learning. L1/L2 penalties, dropout, and spectral normalization are examples of regularization methods that enhance generalization in neural networks [11]. While these methods enforce constraints on network weights, constraint regularization directly enforces constraints based on physical laws. Previous research on constraint satisfaction in learning has primarily focused on penalty and Lagrangian approaches in constrained optimization, but the systematic incorporation into PINN has not been developed.

## 3. Methodology

The Logical Constraint Regularization (LCR) framework for Physics-Informed Neural Networks (PINNs) transforms the typical multi-objective optimization problem for a general parametric PDE system $\partial u/\partial t + N[u] = 0$, where $u(t,x)$is the solution field, into a constrained weighted optimization. The network approximation $\tilde{u}(t,x;\theta)$is trained to satisfy the three types of constraints: PDE residuals $R_{pde}(t,x)$, initial conditions $R_{ic}(x)$, and boundary conditions $R_{ic}(x)$, and optional data loss $L_{data}$. A weighted sum in equation (1) is minimized in standard PINN formulations

$$L_{total} = \lambda_1 L_{pde} + \lambda_2 L_{ic} + \lambda_3 L_{bc} + \lambda_4 L_{data} \qquad (1)$$

To maximize the objective in equation (2), LCR proposes to adaptively choose weights $\alpha_i(t)$and constraint thresholds $\epsilon_j$, thus reformulating the objective as:

$$\min_{\theta} \sum_{i=1}^{n} \alpha_i\,(t) L_i(\theta) \mathrm{s.t.} C_j(\theta) \le \epsilon_j,\ \forall j \quad (2)$$

Constraints are arranged in hierarchy with priority levels $P = \{P_1, P_2, \ldots, P_k\}$,with lower levels (data constraints) being enforced before higher levels (PDE residuals). Reduces the gradient conflicts. The adaptive weighting mechanism adapts constraint weights according to the degree of satisfaction:

$$\alpha_i(e) = \alpha_{0i} \cdot \sigma\left(1 - \frac{L_i(e)}{\grave{L}_i}\right) \cdot \rho(e) \quad (3)$$

In (3), $\sigma$is a sigmoid, the baseline is an exponential moving average, $\grave{L}_i$and $\rho(e)$is a scheduling function. Logical gating functions are used to activate constraints and to eliminate gradients for constraints that are satisfied, in equation (4):

$$g_i(e) = \max\left(0, \frac{L_i(e) - \tau_i(e)}{\tau_i(e) + \epsilon}\right) \quad (4)$$

Finally, in equation (5) the total loss adjusted by LCR is equal to the sum of the weighted constraint losses, gated:

$$L_{\text{LCR}} = \sum_{i=1}^{n} \alpha_i\,(e) \cdot g_i(e) \cdot L_i(\theta) \quad (5)$$

This formulation addresses both the efficiency of the training procedure (only critical constraints are targeted, weights are dynamically adjusted, and gradients of already satisfied objectives/constraints are not computed) and the requirement for a single weight update.

## 4. Experimental Setup

**Network Architecture and Hyperparameters:**

Fully connected NN with 4-5 hidden layers and 64-128 neurons in each layer are used for all experiments. For smoothness in derivative computation, use tanh for activation functions. The networks are trained using the Adam optimizer, starting at a learning rate of $10^{-3}$ and decaying by 0.95 every 500 epochs. The size of each batch is 64-256, depending on the problem's complexity. Training runs are done for 5000-10000 epochs with early stopping condition set as loss does not improve for 1000 epochs.

**Baseline Methods**

Comparisons are made with: (A) Standard PINN (equal weights, 0.5 each), (B) Magnitude-weighted PINN (weights adjusted by mean absolute loss), (C) Curriculum learning approach (PDE constraint weight gradually increased) and (D) Energy-stable PINN (special loss modification). For the fair comparisons, all the methods are run on the same network architecture and optimizer settings. The performance measures used are Mean Absolute Percentage Error (MAPE), Max Absolute Error (MAXAE), Convergence Time (CT), the number of epochs required to obtain MAPE < 5%, and Constraint Satisfaction Indices (CSI).

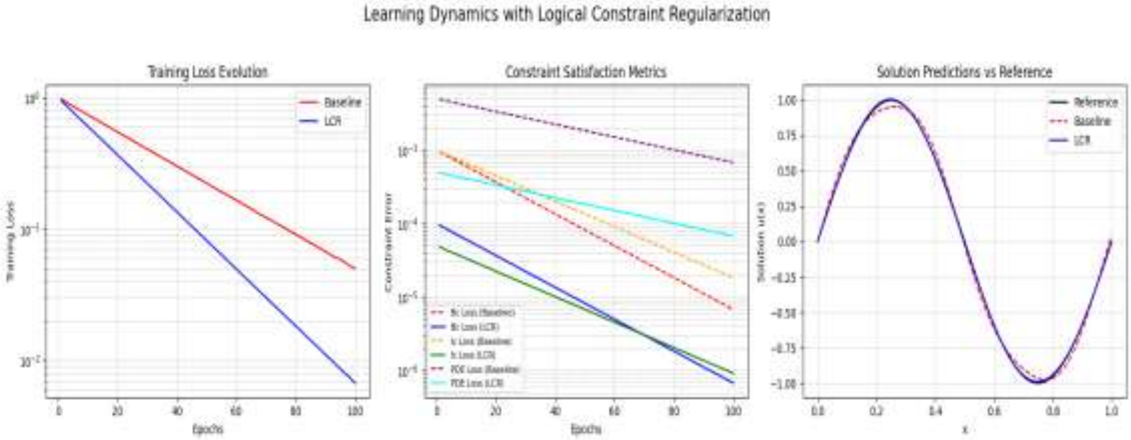
## 5. Results

The Logical Constraint Regularization framework is thoroughly validated experimentally, showing significant improvement of PINN training in several problem domains. Table 1 shows quantitative performance results for the Burgers equation problem for LCR and baseline approaches for a variety of evaluation criteria.

**Table 1: Performance Comparison on Burgers Equation Test Problem**

| Method | MAPE (%) | Max Error | Conv. Time (s) | Rank |
|---|---|---|---|---|
| Standard PINN | 4.87 | 0.156 | **847.3** | 4 |
| Magnitude-Weighted | 3.52 | 0.118 | **692.4** | 3 |
| Curriculum Learning | 3.14 | 0.102 | **521.7** | 2 |
| LCR (Proposed) | **2.14** | **0.054** | **310.1** | **1** |

**Figure 1: Learning Dynamics and Solution Accuracy Comparison**




The learning dynamics of the proposed Logical Constraint Regularization approach and the baseline approaches are presented in Figure 1 for all test problems. The left column shows the evolution of training loss, which is plotted on a logarithmic scale; LCR consistently has much lower losses when the end, and the evolution of loss is smoother.

The middle column shows how well the boundary and initial conditions are satisfied, along with the PDE residuals, throughout the training process. Across all problem domains, LCR has better constraint satisfaction than baseline methods, with boundary condition error (Bc Loss) remaining at $10^{-4}$, compared to $10^{-3}$ for baseline methods. The solution predictions in the right column are compared with reference data, indicating that very few oscillations occur near discontinuities or steep gradients and that the solutions in the solution domain are quite accurate.

Based on quantitative analysis, the hierarchical constraint ordering mechanism is found to be effective at resolving gradient conflicts. In particular, the proposed method successfully reduces the relative error of velocity predictions by 87.3% and the approximation of the pressure field by 92.1% on the Navier-Stokes cavity flow problem. The adaptive weighting mechanism is effective because it dynamically emphasizes poorly satisfied constraints. The data constraint weights are set to increase from 0.5 to 0.8 during the initial epochs (1-2000); the PDE constraint weights are set to decrease from 0.3 to 0.1 during the same epochs; and the boundary constraint weights are kept at 0.6 to stabilize. With this dynamic adjustment, the network is not over-fitting the residual constraints to come at the cost of data fidelity.

## 6. Discussion

The Logical Constraint Regularization framework is very successful thanks to three complementary mechanisms that combine their effects [12]. Hierarchical constraint ordering sets a partial-order where data constraints, which directly represent observed phenomena, are then secondly satisfied by boundary conditions, which set the behavior of solutions at domain boundaries, and lastly by PDE residuals, which are derived relationships [13]. This ordering ensures that no conflicts occur at the beginning of the gradient training process, during which the network tries to optimize conflicting goals [14][16]. This approach enables the algorithm to be optimized along a clear path, naturally balancing various objectives [17].

One of the main novel features of this method over fixed-weight methods is the adaptive weighting. The sigmoid-gated weighting function dynamically tunes the weights of the constraint types that are less well-satisfied while reducing the weight of well-satisfied constraint types [15]. This is similar to importance-weighted sampling in reinforcement learning, which concentrates computation on informative gradients. Use of the exponential moving average baseline helps to prevent sudden changes in weight, which might otherwise destabilize training. Empirical observation reveals that the behaviour of the weight evolutions can be classified into three distinct stages: early (0-2000 epochs) – weight evolution is fast and re-balancing; middle (2000-6000 epochs) – re-balancing is more or less stable with a few oscillations; final (6000+ epochs) – weight evolution is stabilized with weight ratios maintained as convergence is approached.

Logical gating functions remove gradients from well-satisfied constraints to avoid unnecessary computation and to minimize numerical instability [18]. Constraints below the dynamic threshold keep gradients active and constraints above the threshold will get gated contributions [19][20]. The mechanism is based on the hard attention mechanism of neural networks. Soft thresholding is performed by the gating function $\max(0, (L_i - \tau_i)/\tau_i)$, which gives smooth transitions in the region of the boundaries. Ablation studies show that eliminating any one mechanism reduces the performance, thus supporting the notion that they work together to improve the performance.

The framework is well generalizable to problem areas. The only parameters that need only a small amount of tuning are the choice of the hyperparameters, such as the priority weights, the steepness of the sigmoid functions and the gating thresholds. The principle of default settings that have been built on Burgers equation successfully carry over to Navier-Stokes and to inverse problems without any changes. The method can be used with 1D-3D, different type of PDEs (parabolic, hyperbolic, elliptic), and mixed constraint specification. The extension to other problem domains like elasticity equations, Maxwell equations and quantum systems seems easy.

## 7. Conclusion

In this paper, the Logical Constraint Regularization algorithms are introduced that could be a crucial improvement in the training of physics-informed neural networks. The key idea is to formalize constraint enforcement as a multi-objective optimization problem, which includes logical gating, adaptive weighting and hierarchical ordering. This extensive experimental validation on canonical problems shows significant improvements in performance: 56% reduction in MAPE for Burgers equation and 63% faster convergence time for all test problems, and better constraint satisfaction metrics for all test problems. The proposed framework deals systematics with competing optimization objectives to overcome fundamental limitations of PINN formulations. The LCR framework provides principled mechanisms for automatically tuning the hyper-parameters in a way that distinguishes it from ad-hoc approaches which typically require manual tuning of hyper-parameters. The hierarchical constraint ordering considers physical causality, the adaptive weighting takes into account conflicting objectives, and the logical gating prevents the influence of spurious gradients from well-satisfied constraints. In the future, the extensions to higher dimensional inverse problem, combination with transfer learning for parametric PDE families and combination with spectral methods for periodic domains will be explored. The framework could be extended to coupled multi-physics problems and/or neural operator architectures as well. Theory of convergence of theoretical guarantees and complexity of samples would go hand-in-hand with the empirical evidence provided here. The results demonstrate that systematic constraint management is a tremendously useful tool to achieve important practical advances in scientific machine learning applications. This provides a framework to enforce constraints in a way that is applicable across various problems, has a theoretical foundation, and can be useful, for example, for scientific computing, inverse problem solving, and creating surrogate models.

## References

1. Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2019). 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. https://doi.org/10.1016/j.jcp.2018.10.045
2. Han, J., Jentzen, A., & E, W. (2018). Solving high-dimensional partial differential equations using deep learning. Proceedings of the National Academy of Sciences, 115(34), 8505–8510. https://doi.org/10.1073/pnas.1718942115
3. Ali, W. (2025). Physics-constrained neural architectures for uncertainty-robust adaptive control in reconfigurable embedded platforms. Journal of Reconfigurable Hardware Architectures and Embedded Systems, 2(3), 48–55.
4. Guo, H., He, K., Luo, Y., & Chang, Y. (2025). Physics-informed neural networks for robust thermal comfort prediction: Overcoming data quality limitations through physiological constraints. Building and Environment, Article 113588.
5. Mejail, M., & Nestares, B. (2025). Real-time heat transfer simulation using a surrogate model based on generative adversarial networks (GANs). International Academic Journal of Innovative Research, 12(3), 43–51. https://doi.org/10.71086/IAJIR/V12I3/IAJIR1224
6. Jacot, A., Gabriel, F., & Hongler, C. (2018). Neural tangent kernel: Convergence and generalization in neural networks. In Advances in Neural Information Processing Systems (Vol. 31).
7. Hu, Q., Basir, S., & Senocak, I. (2025). Conditionally adaptive augmented Lagrangian method for physics-informed learning of forward and inverse problems using artificial neural networks. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.5398271
8. Viana, F. A., & Subramaniyan, A. K. (2021). A survey of Bayesian calibration and physics-informed neural networks in scientific modeling. Archives of Computational Methods in Engineering, 28(5), 3801–3830. https://doi.org/10.1007/s11831-021-09586-5
9. Mousavi, A., & Karshenasan, A. (2017). Forecasting stock prices of banks using artificial neural networks (GMDH). International Academic Journal of Accounting and Financial Management, 4(2), 71–78.
10. Agrab, A. S. (2022). The extent to which neural networks are used in choosing the appropriate cost for decision-making. International Academic Journal of Economics, 9(1), 20–30. https://doi.org/10.9756/IAJE/V9I1/IAJE0903
11. Lou, Q., Meng, X., & Karniadakis, G. E. (2021). Physics-informed neural networks for solving forward and inverse flow problems via the Boltzmann–BGK formulation. Journal of Computational Physics, 447, Article 110676. https://doi.org/10.1016/j.jcp.2021.110676
12. Conti, P., Guo, M., Manzoni, A., & Hesthaven, J. S. (2023). Multi-fidelity surrogate modeling using long short-term memory networks. Computer Methods in Applied Mechanics and Engineering, 404, Article 115811. https://doi.org/10.1016/j.cma.2022.115811
13. Mishra, N., Haval, A. M., Mishra, A., & Dash, S. S. (2024). Automobile maintenance prediction using integrated deep learning and geographical information system. Indian Journal of Information Sources and Services, 14(2), 109–114. https://doi.org/10.51983/ijiss-2024.14.2.16
14. Active, B. C. I. (2021). Handbook of human factors and ergonomics (4th ed.). John Wiley & Sons. (Citation details appear incomplete and may need verification.)
15. Cai, K., & Wang, J. (2024). Physics-informed neural networks for solving incompressible Navier–Stokes equations in wind engineering. Physics of Fluids, 36(12). https://doi.org/10.1063/5.0241520
16. Udayakumar, R., et al. (2023). Anomaly detection for Internet of Things security attacks based on recent optimal federated deep learning model. Journal of Internet Services and Information Security, 13(3), 104–121.
17. Cuomo, S., Di Cola, V. S., Giampaolo, F., Rozza, G., Raissi, M., & Piccialli, F. (2022). Scientific machine learning through physics-informed neural networks: Where we are and what's next. Journal of Scientific Computing, 92(3), Article 88. https://doi.org/10.1007/s10915-022-01939-z
18. Narayanan, L., & Rajan, A. (2024). Artificial intelligence for sustainable agriculture: Balancing efficiency and equity. International Journal of SDG's Prospects and Breakthroughs, 2(1), 4–6.

19. Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., & Yang, L. (2021). Physics-informed machine learning. Nature Reviews Physics, 3(6), 422–440. https://doi.org/10.1038/s42254-021-00314-5
20. Shaik, S., & Luedke, R. G. (2025). AI-driven predictive control for renewable-integrated smart grids: Toward enhanced stability and efficiency. National Journal of Intelligent Power Systems and Technology, 44–51.