

Addressing the gravitational collapse of a massless scalar field with Physics-Informed Neural Networks

* Antonio Ferrer-Sánchez¹, Nino Villanueva-Espinosa^{1,3}, Carlos Hernani Morales^{1,2}, Roberto Ruiz de Austri-Bazan⁵, José A. Font^{3,4}, José David Martín-Guerrero^{1,6} and Matthew W. Choptuik⁷

¹Intelligent Data Analysis Laboratory (IDAL), Department of Electronic Engineering, ETSE-UV, University of Valencia, Spain.

²Quantum Spain, 46100 Burjassot, Valencia, Spain.

³Departamento de Astronomía y Astrofísica, Universitat de València, Dr. Moliner 50, Burjassot, 46100, Valencia, Spain.

⁴Observatori Astronòmic, Universitat de València, Catedrático José Beltrán 2, Paterna, 46980, Valencia, Spain.

⁵Instituto de Física Corpuscular CSIC-UV, c/Catedrático José Beltrán 2, Paterna, 46980, Valencia, Spain.

⁶Valencian Graduate School and Research Network of Artificial Intelligence (ValgrAI), Spain.

⁷Department of Physics and Astronomy, University of British Columbia, Vancouver BC, V6T 1Z1, Canada.

*e-mail: Antonio.Ferrer-Sanchez@uv.es

Abstract—The gravitational collapse of a massless scalar field remains a demanding benchmark for numerical methods in numerical relativity, as exhibits critical behavior at the boundary between dispersion and black hole formation. In this work we revisit this problem by relying on Physics-Informed Neural Networks (PINNs) as flexible solvers for partial differential equations, thereby providing a comparative assessment of several recent neural architectures. Building on the Einstein-massless-Klein-Gordon formulation in polar-areal coordinates, we consider four initial-value problems encompassing subcritical, critical, and supercritical regimes and use high-resolution finite-difference simulations as reference solutions. Our study is primarily comparative: we evaluate several state-of-the-art deep learning architectures, including vanilla and high-precision PINNs, sinusoidal-feature and quadratic-residual variants, and Kolmogorov-Arnold Networks, all trained under a common loss design that encodes the field equations, boundary conditions, and causal time-space enforcement, together with a novel adaptive spacetime sampling. Within this framework we also introduce ModPINN, a modest modification of standard PINNs that augments standard multilayer perceptrons with coordinate embeddings, quadratic layers, and other common ingredients in recent literature. This study shows that deep-learning-based methods can reproduce finite-difference solutions for the scalar field and the spacetime metric with competitive accuracy using significantly fewer collocation points than more traditional methodologies. While no single architecture dominates in all regimes, ModPINN achieves particularly stable and accurate solutions near criticality, indicating that suitably designed embeddings and adaptive sampling can enhance the robustness of PINNs for challenging gravitational-collapse scenarios.

Keywords—physics-informed neural networks, gravitational collapse, critical phenomena, massless scalar field, numerical relativity

1. Introduction

1.1. Problem setup and basic equations

The study of gravitational collapse and black hole formation stands as one of the most interesting and revealing problems in modern physics. Beyond its astrophysical relevance, gravitational collapse provides an exceptional framework for testing the nonlinear, strong-field dynamics of general relativity (GR) and fundamental conjectures such as cosmic censorship. A particularly insightful and motivating aspect of this field arises in the so-called *critical collapse* problem, first discovered by Choptuik in the early 1990s [1], where the threshold between dispersion and black hole formation in the collapse of a massless scalar field was found to display a universal and self-similar behavior.

Within the framework of GR, one can consider a spherically symmetric massless scalar field ϕ minimally coupled to the spacetime metric $g_{\mu\nu}$. By characterizing the initial data of this system in terms of a one-parameter family labeled by p , Choptuik showed the existence of a critical value p^* that sharply separates solutions which disperse to flat spacetime ($p < p^*$) from those that form a black hole ($p > p^*$).

Near this threshold, the dynamics exhibit critical phenomena: (i) universality, in the sense that different families of initial data approaching p^* lead to similar universal behavior, largely independent of their detailed profile; (ii) a power-law scaling relation between the mass of the resulting black hole and the distance to the critical parameter, $M_{\text{BH}} \sim |p - p^*|^\gamma$, where $\gamma \approx 0.37$ is a universal critical exponent; and (iii) discrete self-similarity (or “echoing”), whereby the

critical solution reproduces itself on progressively smaller spacetime scales with a characteristic logarithmic periodicity.

These findings reveal that the dynamics in the proximity of the black hole threshold can be characterized by attractor solutions of codimension one in the phase space of GR. In this regime, spacetime curvature can grow without bound outside the event horizon, providing a natural path towards understanding how GR itself approaches the limits of predictability and potentially signaling where quantum gravitational effects must emerge. Consequently, the study of critical collapse helps to deepen our understanding of the nonlinear structure of Einstein’s equations and builds direct relations to fundamental issues in high-energy physics, quantum gravity, and the nature of singularity formation, thereby paving the way for both analytical and numerical methods in modern gravitational theory.

The parameter p corresponds to a specific characteristic of the initial distribution of the scalar field, $\phi(t = 0, r) := \phi_0(r)$, where t and r denote the time and radial components, respectively, in polar-areal coordinates, $\{t, r, \varphi, \theta\}$. These coordinates help to express the time-dependent Schwarzschild-like metric

$$ds^2 = -\alpha^2(t, r) dt^2 + a^2(t, r) dr^2 + r^2 d\Omega^2, \quad (1)$$

where $d\Omega^2 = d\theta^2 + \sin^2\theta d\varphi^2$ is the metric on the unit 2-sphere, $\alpha(t, r)$ represents the lapse function encoding the proper time evolution and $a(t, r)$ stands for the spatial metric component in the radial direction.

The parameter p naturally divides the entire set of initial conditions into two distinct categories: those that lead to the formation of a black hole during evolution and those that do not. Therefore, we can introduce the concept of a “threshold” when discussing the formation of an event horizon. Spacetimes evolving from initial conditions near this threshold exhibit properties that, from a mathematical perspective, are analogous to a critical phase transition in statistical mechanics (interested readers are referred to Sections 2.2 and 3.3 in [2]). In this manner, the parameter p has a critical value p^* separating “subcritical” solutions for which the mass dissipates ($p < p^*$) from “supercritical” solutions for which an event horizon forms ($p > p^*$), leading to the emergence of an apparent horizon.

Near the critical region, $p \sim p^*$, any geometric quantity X characterizing the black hole at the time of its formation—particularly its mass—follows a power law [3],

$$\ln X = \gamma \ln |p - p^*| + f(\ln |p - p^*|), \quad (2)$$

where γ is a universal scaling exponent independent of both the type of initial data considered and the physical quantity X being measured. Moreover, f stands for a function that can be directly related to a nonuniversal constant, c_f , which depends on the shape of the initial field $\phi_0(r)$ for the particular case of the black hole mass M_{BH} . In this case, relation (2) reads

$$\ln M_{\text{BH}} = \gamma \ln |\Delta p^*| + \ln c_f, \quad (3)$$

highlighting the proportionality between the logarithm of M_{BH} (or,

equivalently, the event horizon radius via $r_{\text{BH}} = 2M_{\text{BH}}$) and the logarithm of the deviation of the relevant physical parameter from its critical value, i.e., $|\Delta p^*| := |p - p^*|$.

For a massless scalar field $\phi(t, r)$ minimally coupled to the metric (1), the Einstein tensor is given by

$$G_{ab} = 8\pi \left(\nabla_a \phi \nabla_b \phi - \frac{1}{2} g_{ab} \nabla_c \phi \nabla^c \phi \right). \quad (4)$$

Together with the mass conservation equation $\nabla_a \nabla^a \phi = 0$, the Einstein-massless-Klein-Gordon (EMKG) system in spherical symmetry, written in terms of the polar-radial coordinates (t, r) and using geometric units ($G = c = 1$), reads:

EMKG field equations

$$\frac{\partial \Phi}{\partial t} = \frac{\partial}{\partial r} \left(\frac{\alpha}{r} \Pi \right), \quad \frac{\partial \Pi}{\partial t} = \frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\alpha}{a} \Phi \right). \quad (5)$$

$$\frac{\partial \ln \alpha}{\partial r} - \frac{\partial \ln a}{\partial r} + \frac{1 - a^2}{r} = 0. \quad (6)$$

$$\frac{\partial \ln a}{\partial r} - \frac{1 - a^2}{2r} - \frac{a^2}{r} (X^2 + Y^2) = 0. \quad (7)$$

At this stage, one can identify the “primitive” set of variables, consisting of $(\phi, \alpha, a)(t, r)$, where the physically admissible ranges are $\alpha \in [0, 1]$ and $a \in (1, +\infty)$, while the values of ϕ can be considered numerically unbounded. In addition, some auxiliary variables have been introduced.

$$\Phi(t, r) = \frac{\partial \phi}{\partial r}(t, r), \quad \Pi(t, r) = \frac{a}{\alpha} \frac{\partial \phi}{\partial t}(t, r). \quad (8)$$

$$X(t, r) = \sqrt{2\pi} \frac{r}{a} \Phi(t, r), \quad Y(t, r) = \sqrt{2\pi} \frac{r}{a} \Pi(t, r). \quad (9)$$

The set $(\Phi, \Pi)(t, r)$ in (8) is directly associated with the spatial and temporal derivatives of the scalar field, respectively. In terms of these variables, the set $(X, Y)(t, r)$ in (9) can be defined, leading to some relevant mathematical relations involving the derivatives of the mass aspect function, $m(t, r)$,

$$\frac{\partial m}{\partial r} = X^2 + Y^2, \quad \frac{\partial m}{\partial t} = \left(\frac{a}{\sqrt{1 - a^2}} \right)^3 \alpha X Y. \quad (10)$$

For completeness, the spacetime curvature scalar R can also be written in terms of this pair of variables as

$$R = \frac{4}{r^2} (X^2 - Y^2). \quad (11)$$

The mass function represents the amount of mass enclosed within a radius r at a given time t during the evolution. It is defined in analogy to the mass in the static Schwarzschild metric in terms of $a(t, r)$,

$$m(t, r) = \frac{r}{2} \left(1 - \frac{1}{a^2} \right). \quad (12)$$

In order to determine whether a given initial data setup has resulted in gravitational collapse—and consequently, the formation of an apparent horizon—it is necessary to evaluate the ratio $2m/r$, commonly referred to as the compactness parameter (or, alternatively, the dimensionless mass aspect). We will denote this parameter by C , which can be easily defined in terms of the radial metric coefficient,

$$C(t, r) := \frac{2m(t, r)}{r} = 1 - \frac{1}{a^2(t, r)}. \quad (13)$$

Within this formulation, this parameter satisfies $C \in [0, 1]$, where the limit $C \rightarrow 1$ characterizes the onset of gravitational collapse. From a computational perspective, the formation of an apparent horizon in

numerical simulations would be identified by imposing a threshold ϵ such that $1 - C \leq \epsilon$. In this manner, classical numerical methods based on finite-difference methods (FDM) can effectively solve the field equations in both dispersive scenarios of $\phi(t, r)$ and cases involving gravitational collapse (e.g. see [4]). Despite the coupling between the different variables, the dynamical evolution is governed by (5), while (6) and (7) serve as metric constraints.

1.2. Deep-learning-based methodologies

Numerical solutions of the field equations (5-7) have been reported since the seminal work of [1]. Several studies (see [3, 5, 6]) focused on investigating critical behavior within generalizations of this framework to spacetimes of higher dimensions. Other extensions took into account massive fields [7], configurations involving multiple scalar fields [8], or scenarios incorporating more sophisticated or different metric structures, yielding interesting results, such as those reported in [9] or [10].

Despite the many numerical studies, it is worth noting that the most common methodology to solve the equations has been based on FDM. Only recent works have considered alternative methods to FDM to investigate gravitational collapse. In particular, the work of [11] introduced a neural network-based solver that mitigates artificial neural network (ANN) ill-conditioning while exhibiting linear scalability with respect to the number of parameters. Other studies, such as [12], developed a model also based on ANNs to investigate critical phenomena in Einstein-axion-dilaton models, finding no evidence of self-similar behavior. On the other hand, [13] and [14] employ Bayesian deep learning models to study the axion-dilaton system, utilizing Sequential Monte Carlo (SMC) sampling and related algorithms to obtain direct estimates of the critical exponent γ .

All of these recent studies establish a foundation for the incorporation of deep learning (DL) techniques in the investigation of gravitational collapse scenarios. The present work proposes the adoption of novel methodologies based on ANNs tailored to solve systems of partial differential equations (PDEs) and extract their underlying physics. Since their initial appearance in the scientific literature [15], Physics-Informed Neural Networks (PINNs) have emerged as a robust alternative for solving numerical problems governed by PDEs. These methods leverage the universal approximation theorem of ANNs [16] to minimize the governing physical equations, incorporating them into the loss function used to train the network parameters. This function is typically augmented with physical constraints—commonly referred to as “inductive biases”—such as initial and boundary conditions, additional physical information to be minimized, or even agreement with experimental data. Therefore, our primary objective will be to demonstrate that the PINN methodology is fully capable of solving the EMKG equations (5-7), effectively addressing both the field dispersion and the gravitational collapse scenarios. This approach is proposed as a viable alternative to traditional numerical methods.

The paper is structured as follows. Section 2 presents an overview of recent literature on PINNs, to motivate our study. Next, Section 3 introduces our PINN-based methodology, which consists of a modified version of the standard (vanilla) PINN; this section also details the specific formulation used to minimize the loss function. Section 4 presents the results obtained for various initial values of the scalar field, including the critical case, with particular emphasis on evaluating the predictive performance of the model in that regime. Finally, Section 5 presents our conclusions and outlines potential directions for future research.

2. Review of recent literature

In general terms, the choice of neural architecture is a critical factor, as the number of trainable parameters must be appropriate for the problem at hand, whatever the specific task may be—whether solving PDEs or addressing another application. Among the aspects

that define a neural architecture, several may be highlighted: the activation functions employed; the number of layers (depth) and their size (width); whether residual or skip connections are incorporated between layers; and whether any form of embedding is used, among other considerations. Several studies [17] conclude, in a range of independent physical problems, that relatively shallow but wide networks often outperform deeper but narrower ones. This reveals a concern that usually appears throughout the literature on multilayer perceptrons (MLPs) applied to physics, motivating more recent developments such as Physics-Informed Residual Adaptive Networks (PirateNets) [18], which are introduced as residual architectures that allow the model to start from shallow networks, progressively “revealing” additional layers only when they are advantageous in reducing the objective cost function. These networks also employ embeddings of the physical coordinates, mapping them into a high-dimensional feature space using random Fourier features [19]. However, although these embeddings can mitigate the frequency bias typically observed in PINNs [20] and facilitate the solution of physical problems involving high-frequency behavior, the selection of the maximum frequency scale and the standard deviation of the Gaussian distribution used to sample the frequencies are critical; if chosen poorly, these become problem-dependent hyperparameters that can hinder performance. Other architectures, such as the sf-PINNs proposed in [21], apply a mapping of the input coordinates to sinusoidal features, encoding them after the first layer, thus increasing the variability of the input gradient and also helping mitigate the said frequency bias.

Stepping somewhat away from traditional architectures, methodological variants such as the Quadratic Residual Networks (QRes) [22] aim to replace the standard linear layer with an alternative that introduces second-order interactions between the trainable parameters and the input, becoming more parameter-efficient and faster to train, in general, on scientific problems. Within the most recent DL literature, Kolmogorov-Arnold Networks (KANs) [23] have emerged as an alternative to conventional MLPs, which have also been applied to PINNs. These mainly consist of replacing the trainable weights of linear layers with learnable univariate functions located at the nodes, yielding high expressivity with substantially fewer parameters. Finally, architectures such as the PINNsFormer [24] implement a transformer-based methodology for time-dependent data, translating each point (t, r) into a short temporal sequence of k steps that is fed into a standard encoder-decoder architecture, thus learning and enforcing temporal dependencies. In contrast, although PINN-Mamba [25] shares certain similarities with the preceding approach, it replaces the MLP backbone with an encoder-only Mamba [26] (selective state-space) network that models short, overlapping temporal subsequences. This design yields a continuous solution over the time of the physical problem, with consistency across overlaps enforced via a contrastive loss between the subsequences. All of these architectural approaches present distinct strengths and limitations, and each has been evaluated on a broad range of physics problems in their respective references. In our case, in Section 4, we undertake a comparative assessment of their performance in solving a variety of sub- and supercritical gravitational collapse problems of the massless scalar field minimally coupled to the spacetime metric.

Regarding domain sampling, this step is also crucial to the methodology, since the collocation points—here the spacetime coordinates (t, r) —serve as training data for the network. Several strategies for generating these points may be considered, ranging from simple, uniformly spaced grids or randomly generated samples for a particular probability distribution to more sophisticated techniques such as Sobol sequences [27] or Latin hypercube sampling [28, 29]. Other approaches incorporating specialized deterministic distributions can also be considered. For example, methods that employ Chebyshev-distributed nodes in space combined with exponentially spaced temporal points enable more frequent and precise sampling near boundaries or localized features [30]. From a general perspective, any tech-

nique that involves non-uniform local refinement sampling can offer a strategy tailored to the specific problem, although such methods remain relatively uncommon in the PINN literature. Despite the fact that the performance of all these methods may vary depending on the specific PINN problem at hand, all of them share one common feature: they rely on fixed sampling strategies that remain static throughout the entire training process. In a comprehensive review, the authors of [31] concluded that there is no universal rule for defining a general and optimal sampling strategy for PINNs, saying that it is necessary to define the sampling in an intelligent manner and tailor the methodology to the particular scenario.

Within the range of adaptive sampling methods, those based on probability distributions deserve special mention, highlighting [32] where the authors introduced the Residual-based Adaptive Distribution (RAD) which focused on resampling points with a probability proportional to their PDE residual. In studies such as [33], the authors propose sampling techniques based on the significance assigned by the network to different regions of the training domain, finding that PINNs achieve faster convergence when the points are sampled according to a weighted probability density function (PDF) that depends on the loss. Some extensions of this idea are explored in works such as [34], where a PDF is defined to continuously concentrate training points in regions with high loss values while still maintaining coverage in areas with lower errors. This balance is crucial, as completely neglecting low-error regions could lead to systematic forgetting by the PINN. Other studies focus more on repositioning the collocation points. For example, the RANG method (Residual-based Adaptive Node Generation) proposed in [35] treats the sampling points as variable-density radial basis function (RBF) nodes, allowing adaptive redistribution based on the residual.

3. Methods

Incorporation of physical information into neural networks for use as universal approximators of differential equations has been discussed in the literature for several decades [36, 37, 38]. However, the emergence of computational techniques, such as automatic differentiation [39], marked the formal introduction of PINNs as a distinct concept, which were originally formulated as fully connected feed-forward neural networks composed of K hidden stacked layers [40]. These networks take a vector $\mathbf{x}$ as input, producing as output a tensor of target variables which we will denote by $\mathcal{F}(\mathbf{x}; \Theta) := \mathcal{F}_\Theta(\mathbf{x})$, where Θ represents the set of all trainable parameters of the architecture—typically, for a vanilla model, $\Theta = \{\mathbf{W}_k, \mathbf{b}_k\}_{1 \leq k \leq K}$, where $\mathbf{W}_k \in \mathbb{R}^{N_k \times N_{k-1}}$ and $\mathbf{b}_k \in \mathbb{R}^{N_k}$ are the network weights and biases of the k -th layer, respectively. Given the requirement of non-linearity, a set of activation functions denoted by $\{\sigma_k\}_{1 \leq k \leq K-1}$ is applied to the output of each hidden layer¹. These activation functions can be implemented either at the layer level or neuron-wise, which can also be trained [41, 42].

In the context of the present study, a spacetime domain defined as $(t, r) \in [0, T] \times \Omega$, where T denotes the maximum evolution time, is considered. A vanilla PINN is constructed by minimizing a given differential operator (14), typically subject to specific constraints that the predicted solution must satisfy, such as initial conditions in time (15) and boundary conditions in space (16).

$$\mathcal{P}\left(t, r; \frac{\partial \mathcal{F}}{\partial t}, \frac{\partial^2 \mathcal{F}}{\partial t^2}, \dots; \frac{\partial \mathcal{F}}{\partial r}, \frac{\partial^2 \mathcal{F}}{\partial r^2}, \dots; \frac{\partial^2 \mathcal{F}}{\partial r \partial t}, \dots\right) = 0. \quad (14)$$

$$\mathcal{IC}(t, r) = 0, \quad (t, r) \in \{0\} \times \Omega. \quad (15)$$

$$\mathcal{BC}(t, r) = 0, \quad (t, r) \in (0, T] \times \partial\Omega. \quad (16)$$

In general, for the case of vanilla PINNs, the loss function employed, $\mathcal{L}$, given by (17) below, involves both the minimization of

¹ Note that the set of activation functions spans only up to the penultimate layer. In general, the output layer corresponds to physical variables that typically require independent activation functions to restrict them within the acceptable physical range.

the governing PDEs, denoted by $\mathcal{L}_{\text{PDE}}$, and the incorporation of the additional constraints, $\mathcal{L}_i$, among which the initial and boundary conditions are considered. A weight vector $\vec{\lambda} = (\lambda_{\text{PDE}}, \lambda_{\mathcal{J}\mathcal{E}}, \lambda_{\mathcal{B}\mathcal{E}}, \dots)$ is defined to combine the different components of the loss into a single objective function.

$$\mathcal{L} = \lambda_{\text{PDE}} \mathcal{L}_{\text{PDE}} + \sum_i \lambda_i \mathcal{L}_i. \quad (17)$$

A quadrature point vector $\vec{N} = (N_{\text{PDE}}, N_{\mathcal{J}\mathcal{E}}, N_{\mathcal{B}\mathcal{E}}, \dots)$ is defined to specify the sampling locations used to evaluate each component. Although alternatives have been studied [43], these terms are typically defined individually as mean squared errors (MSE): in the case of $\mathcal{L}_{\mathcal{J}\mathcal{E}}$ and $\mathcal{L}_{\mathcal{B}\mathcal{E}}$, as the MSE between the predicted and true values; and in the case of $\mathcal{L}_{\text{PDE}}$, as the MSE of the residuals of the governing equations.

3.1. Construction of the loss function

Based on the definitions provided in the previous section and considering the compactness parameter C defined in (13), the evolution equation for the scalar field can be expressed as,

$$\text{eq}_{1a} := \frac{\partial \Phi}{\partial t} - \frac{\partial}{\partial r} (\alpha \Pi \sqrt{1-C}). \quad (18)$$

$$\text{eq}_{1b} := \frac{\partial \Pi}{\partial t} - \frac{1}{r^2} \frac{\partial}{\partial r} (r^2 \alpha \Phi \sqrt{1-C}). \quad (19)$$

On the other hand, the Hamiltonian constraint, which enforces the balance between the gravitational and scalar field energies at each radial coordinate r , along with the slicing condition, which stands for the momentum constraint under a zero-shift gauge and serves to fix the lapse function on each time slice, can be reformulated as

$$\text{eq}_2 := r(1-C) \frac{\partial \ln \alpha}{\partial r} - \frac{r}{2} \frac{\partial C}{\partial r} - C. \quad (20)$$

$$\text{eq}_3 := \frac{\partial m}{\partial r} - (X^2 + Y^2), \quad (21)$$

respectively. Considering that the evolution of $\phi(t, r)$ is second-order in time, it is necessary to specify its first temporal derivative as part of the initial data. Although there are several options that can be chosen, for the purpose of the present study we will use “time-symmetric initial data”, that is, $\frac{\partial \phi}{\partial t} \Big|_{t=0} = 0$, or equivalently, $\Pi(0, r) = 0$.

In our methodology, a two-dimensional spacetime domain is numerically discretized as $(t, r) \in [0, T] \times [0, R]$, where the parameters T and R denote the maximum values reached in the numerical simulation and must be chosen sufficiently large to encompass all relevant physical information. The resulting grid thus comprises N_t points in the temporal dimension and N_r in the radial direction. Therefore, the original loss function depends on both time and space, since a loss value is computed at each collocation point. In that sense, the first three terms in the loss equation

$$\mathcal{L}(t, r) = \begin{cases} \lambda_{\mathcal{J}\mathcal{E}} |\mathcal{F}_\Theta(0, r) - \mathcal{F}_{\text{GT}}(0, r)|^2, & (t, r) \in \{0\} \times [0, R], \\ \lambda_{\mathcal{B}\mathcal{E}, \text{left}} |\mathcal{F}_\Theta(t, 0) - \mathcal{F}_{\text{GT}}(t, 0)|^2, & (t, r) \in [0, T] \times \{0\}, \\ \lambda_{\mathcal{B}\mathcal{E}, \text{right}} (|\mathcal{R}_\phi|^2 + |\mathcal{R}_\alpha|^2 + |\mathcal{R}_C|^2), & (t, r) \in [0, T] \times \{R\}, \\ \sum_i \lambda_i |\text{eq}_i(t, r)|^2, & (t, r) \in (0, T] \times (0, R]. \end{cases} \quad (22)$$

highlight the conditions that need to be treated separately: the initial temporal slice ($t = 0$), which corresponds to the initial conditions, and the first and last spatial slices ($r = 0$ and $r = R$), corresponding to the boundary conditions. In this context, the subscript “GT” refers to “ground truth”, referring to some baseline numerical solution that is being used. The sets of conditions are computed a priori for the three variables of interest, $\mathcal{F}_{\text{GT}} = (\phi, \alpha, C)_{\text{GT}}$, imposing $t = 0$ and $r = 0$

in the field equations (18–21) and solving them directly using FDM implemented in modern numerical frameworks [44]. In addition, for points sufficiently far from the source—that is, the limit $r \gg 0$ (or $r \rightarrow R$)—we impose the standard $1/r$ falloffs of the metric and field variables to ensure asymptotic flatness,

$$\mathcal{R}_\phi = \frac{\partial}{\partial r} [r\phi], \quad \mathcal{R}_\alpha = \frac{\partial}{\partial r} [r(\alpha - 1)], \quad \mathcal{R}_C = \frac{\partial}{\partial r} [rC]. \quad (23)$$

The enforcement of restrictions $|\mathcal{R}_\alpha|^2, |\mathcal{R}_C|^2 \rightarrow 0$ yields the asymptotics $C \sim \frac{2M_{\text{ADM}}}{r}$ and $\alpha = \sqrt{1 - \frac{2M_{\text{ADM}}}{r}} = 1 - \frac{M_{\text{ADM}}}{r} + \mathcal{O}(r^{-2})$, the latter being the lapse for the exterior Schwarzschild metric and where M_{ADM} represents the time-conserved Arnowitt-Deser-Misner mass. Finally, demanding $\frac{\partial}{\partial r} (r\phi) = 0$ yields $\phi \sim \frac{\text{const.}}{r}$, the expected dispersal of a massless scalar. These conditions guarantee that the solution given by the network approaches the Schwarzschild asymptotics of an asymptotically flat spacetime, tying the falloff of α and C directly to the ADM mass.

Regarding the resolution of PDEs, the final term in (22) consists of several terms, each representing the squared modulus of a given equation. On the other hand, the set $\{\lambda_i\}$ represents the weights assigned to the different equations. In addition, $\lambda_{\mathcal{J}\mathcal{E}}$ denotes the coefficient associated with the initial conditions over time, while the pair $(\lambda_{\mathcal{B}\mathcal{E}, \text{left}}, \lambda_{\mathcal{B}\mathcal{E}, \text{right}})$ specifies the weights assigned to the boundary conditions at $r = 0$ and $r = R$, respectively.

This construction yields a quadrature tensor for the loss function, $\mathcal{L}(t_n, r_i)$, evaluated at each collocation point. To advance further, one may adopt the approach proposed by [45], who proved that PINNs tend to exhibit biased learning solutions at later times prematurely, often causing the model to converge to failure modes. Consequently, for a given time value t_n , a weighting factor $\omega_t(t_n) \in [0, 1]$ is introduced,

$$\mathcal{L} = \frac{1}{N_t N_r} \sum_{n=0}^{N_t} \omega_t(t_n) \sum_{i=0}^{N_r} \omega_r(t_n, r_i) \mathcal{L}(t_n, r_i). \quad (24)$$

$$\omega_t(t_n) = \exp \left(-\epsilon_t \sum_{m=0}^{n-1} \mathcal{L}(t_m) \right). \quad (25)$$

This weight is computed using an exponential function of the cumulative sum of all loss values evaluated at earlier time steps, $\mathcal{L}(t_{m < n})$, modulated by a hyperparameter ϵ_t , which governs the strength of this causality enforcement mechanism. Within the framework of the original method, the time-dependent loss function $\mathcal{L}(t_n)$ is obtained by averaging (or, alternatively, summing) over the spatial dimension. In our approach, we propose an analogous strategy for handling the radial dimension by introducing an additional weighting factor $\omega_r(t_n, r_i) \in [0, 1]$,

$$\omega_r(t_n, r_i) = \exp \left(-\epsilon_r \sum_{j=0}^{i-1} \mathcal{L}(t_n, r_j) \right). \quad (26)$$

This weight allows sequential learning along the spatial coordinate for a fixed temporal slice, progressively unlocking spatial points in an iterative manner as the previous losses $\mathcal{L}(t_n, r_{j < i})$ are minimized. Similarly, a new hyperparameter ϵ_r is introduced to regulate the strength of this *spatial enforcement* mechanism. In general, ϵ_r will be different from its temporal counterpart ϵ_t , allowing independent treatment of the learning dynamics in both dimensions.

The use of a loss function of the form (24)–(26) enables the network to perform sequential learning, mimicking the behavior of traditional FDM. In this way, learning unfolds progressively through the training process, gradually opening the learning space to the model and preventing the network from “getting ahead” by trying to learn regions that are posterior to those that have not yet been properly minimized. However, despite its clear advantages during training, enforcing sequentiality in learning can also present some drawbacks, including

Algorithm 1 Adaptive weighted-CDF remeshing of radial coordinate.**Require:** Spacetime tensor $(t, r) \in \mathbb{R}^{N_t \times N_r \times 2}$, residuals $\mathcal{L}(t, r) \in \mathbb{R}^{N_t \times N_r}$, adaptivity parameter $\Lambda > 0$, domain bounds $[r_{\min}, r_{\max}]$.**Ensure:** Updated (t, r^{new}) with remeshed radial coordinates.

1. **Reshape input:** Reshape $r \rightarrow r_{\text{old}}[n, i]$, $\mathcal{L} \rightarrow \mathcal{L}[n, i]$ of shape $N_t \times N_r$.
2. **Sort each time slice:** Sort $r_{\text{old}}[n, \cdot]$ along r and permute $\mathcal{L}[n, \cdot]$ accordingly to obtain $(r_{n,j}, \mathcal{L}_{n,j})$.
3. **Normalize residuals:** Linearly map $\mathcal{L}_{n,j}$ to $[0, 1]$ as

$$\mathcal{L}_{n,j}^{\text{norm}} = \frac{\mathcal{L}_{n,j} - \min_j \mathcal{L}_{n,j}}{\max(\max_j \mathcal{L}_{n,j} - \min_j \mathcal{L}_{n,j}, \epsilon)}.$$

4. **Compute density:** Define $m_{n,j} = 1 + \Lambda \mathcal{L}_{n,j}^{\text{norm}}$.
5. **Compute weights:** For each interval,

$$\Delta r_{n,j} = r_{n,j+1} - r_{n,j}, \quad w_{n,j} = \frac{1}{2}(m_{n,j} + m_{n,j+1})\Delta r_{n,j}.$$

6. **Build cumulative distribution:** Let $W_{n,0} = 0$ and compute cumulative sums: $W_{n,k} = \sum_{j=0}^{k-1} w_{n,j}$.
7. **Define uniform target masses:** For each i ,

$$c_{n,i} = \frac{i}{N_r - 1} W_n^{\text{tot}}, \quad \text{where } W_n^{\text{tot}} = W_{n,N_r-1}.$$

8. **Invert weighted-CDF:** For each $c_{n,i}$, find j such that $W_{n,j} \leq c_{n,i} < W_{n,j+1}$ and interpolate:

$$\beta = \frac{c_{n,i} - W_{n,j}}{w_{n,j}}, \quad r_{n,i}^{\text{new}} = r_{n,j} + \beta \Delta r_{n,j}.$$

Clamp β to $[0, 1]$.

9. **Handle endpoints:** Set $r_{n,0}^{\text{new}} = r_{n,0}$ and $r_{n,N_r-1}^{\text{new}} = r_{n,N_r-1}$.

If $W_n^{\text{tot}} \leq 0$, mark slice as degenerate and set $r_{n,j}^{\text{new}} = r_{n,j}$ for all j .

10. **Restore original order:** Unsort $r_{n,i}^{\text{new}}$ to match the original index order.

11. **Clamp to domain:** Enforce $r_{n,i}^{\text{new}} \in [r_{\min}, r_{\max}]$.

12. **Preserve initial slice:** For $n = 0$, set $r_{0,i}^{\text{new}} := r_{\text{old}}[0, i]$.

13. **Reassemble tensor:** Overwrite the radial coordinates in the original (t, r) spacetime with the new set $\{r_{n,i}^{\text{new}}\}$ once flattened.

increased computational time and potential issues such as the cost gets stuck in a specific subset of points (t, r) if the hyperparameter pair (ϵ_t, ϵ_r) is set too high. Therefore, a thorough sensitivity analysis of the network is necessary with respect to these parameters.

3.2. Adaptive spacetime sampling

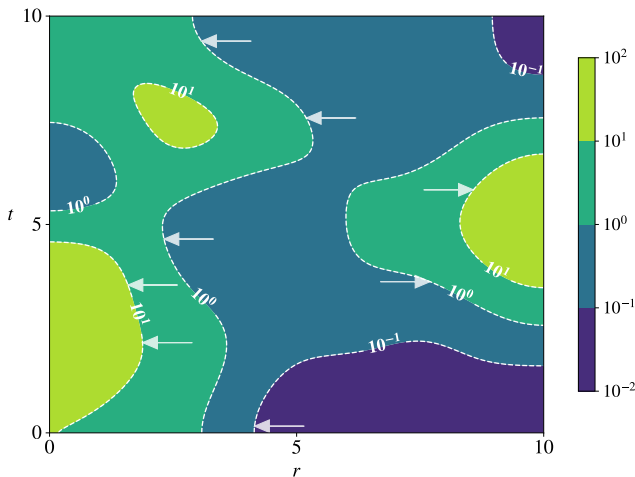


Figure 1. Diagrammatic representation of the adaptive model. The figure shows the landscape of a sample loss function with respect to the (t, r) sampling, whose values span several orders of magnitude. The arrows indicate the direction of movement of the collocation points, which intuitively shift towards regions having higher PDE residuals.

Here we propose a method for redistributing the collocation points along the radial dimension r of the spacetime domain considered (see

visual aid in figure 1). This approach is guided by the performance of the model at each point, using a weighted cumulative distribution function (CDF). Starting from a two-dimensional sampling in time and space, (t, r) , we develop an algorithm that adaptively shifts the spatial points for each fixed temporal slice. Specifically, for a given time n , the algorithm starts by computing the normalized error (residual) as

$$\mathcal{L}_{\text{norm}}(t_n, r_i) = \frac{\mathcal{L}(t_n, r_i) - \min \mathcal{L}(t_n, r_i)}{\max \mathcal{L}(t_n, r_i) - \min \mathcal{L}(t_n, r_i)} \in [0, 1], \quad (27)$$

where the loss is calculated using (22).

Points within the domain can be interpreted as “particles” subjected to a virtual “force” that drives them from their original locations to new positions where the model can utilize them more effectively. To formalize this behavior, a target density function is defined,

$$m(t_n, r_i) = 1 + \Lambda \mathcal{L}_{\text{norm}}(t_n, r_i), \quad (28)$$

which depends on the normalized loss and a hyperparameter Λ that regulates how aggressive the displacement of the points is.

With this framework in place, the core idea is to use the numerically approximated cumulative integral of the target density to determine the new collocation positions. First, we compute the weights between adjacent spatial locations as the average of the density values at two consecutive spatial points for a given time step, as

$$w(t_n, r_j) = \frac{m(t_n, r_j) + m(t_n, r_{j+1})}{2} (r_{j+1} - r_j). \quad (29)$$

These weights are used to approximate the area under the density curve between each pair of points, allowing for the computation of

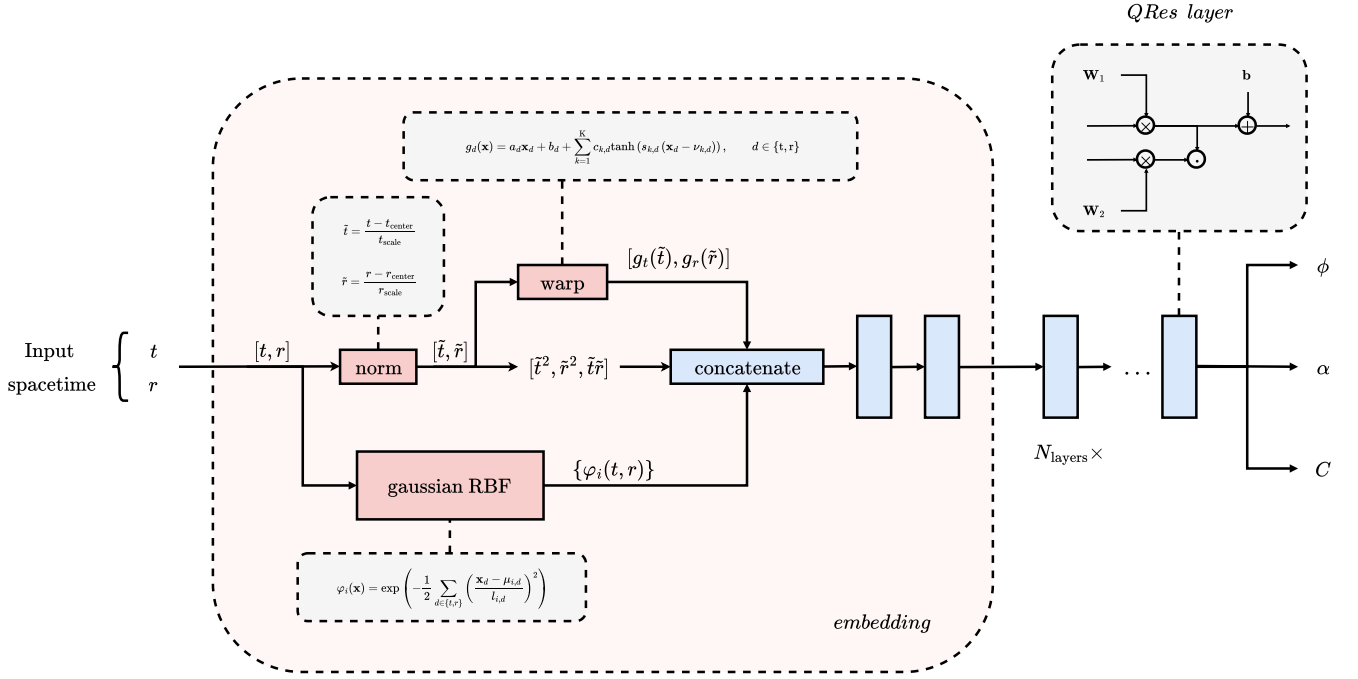


Figure 2. The proposed methodology for the ModPINN. The embedding block of the model modulates (t, r) via a hyperbolic-tangent “warp”, augments them with low-order polynomials of the normalized coordinates together with a dictionary of M Gaussian transforms, and projects the concatenated features through stacked quadratic (QRes) layers to output the physical set (ϕ, α, C) .

the cumulative “mass” via

$$W(t_n, r_k) = \sum_{j < k} w(t_n, r_j). \quad (30)$$

Target mass values are then selected to be evenly spaced between the original and the total accumulated mass W_n^{tot} . By inverting the CDF, we obtain a new set of locations that satisfy this distribution, except at $(t, r) \in [0, T] \times \{0\}$ and $(t, r) \in [0, T] \times \{R\}$, where initial and final boundary conditions are enforced. In addition, the spatial positions at $t = 0$ are also kept fixed so as not to affect the calculation of the temporal initial constraints. The entire process is described in detail, step by step, in Algorithm 1.

3.3. Model architecture

Throughout this work, several novel methodologies recently introduced in the literature have been considered with the aim of establishing a comparative analysis among them. At the same time, an attempt has been made to introduce a slightly adapted approach of our own, tailored to the specific physical problem under study. This neural architecture is illustrated in the diagram of figure 2, and, for the sake of simplicity in terminology, we shall refer to it as ModPINN—as an alias for “modified PINN”—throughout the remainder of the text. In general terms, this and any other physics-informed model presented in this work take as input the spacetime coordinates (t, r) , process them according to the chosen architecture, and output the set of physical variables, $(\phi, \alpha, C)_\Theta$. These variables are then used to compute the PDEs and, consequently, the loss metric. Although most of the physics of the problem is encapsulated in the proposed cost function (24), the neural architecture has also been shown to be a fundamental component in terms of performance, as it can usually be informed by physical principles as well. Consequently, it is worth highlighting the main aspects of our proposed methodology.

Embedding. One of the most well-known issues affecting PINNs is their spectral bias, which often leads to higher frequencies being either unresolved or incorrectly captured [20, 46]. Since many physical problems involve multiscale solutions or sharp gradients, understanding and mitigating this bias has become a central priority

in recent research efforts (e.g., see [47, 48] and references therein). In our particular case, we will confront this problem directly, since the formation of black holes due to gravitational collapse leads to sharp gradients in the physical variables, becoming of utmost importance to address the issue particularly in scenarios near criticality and in supercritical regimes. In this context, Gaussian embeddings [19] project the physical input coordinates into a higher-dimensional space, thus alleviating the problem and facilitating the resolution of high frequencies. Other types of coordinate transformation, such as radial basis functions or Chebyshev polynomials [49], can also be considered. In our specific context, the embedding module constitutes a core element of the architecture, integrating multiple techniques designed to enhance the analytical capabilities of the model. It comprises two parallel and independent paths, which we describe in the following.

- **Upper branch.** The input coordinates provided, (t, r) , will lie within the domain $[0, T] \times [0, R]$, where T and R denote the maximal temporal and spatial extents of the sampling, respectively, which are arbitrary depending on the problem at hand. Since arbitrarily large values are typically impractical for this class of neural networks, the initial step is to normalize the coordinates according to

$$(\tilde{t}, \tilde{r}) = \left(\frac{t - t_{\text{center}}}{t_{\text{scale}}}, \frac{r - r_{\text{center}}}{r_{\text{scale}}} \right) \in [-1, 1]. \quad (31)$$

In this way, the coordinates are recast into a numerically well-conditioned interval for the PINN, defining the temporal center and scale as $t_{\text{center}} = (t_{\text{max}} + t_{\text{min}})/2 = T/2$ and $t_{\text{scale}} = (t_{\text{max}} - t_{\text{min}})/2 = T/2$; the spatial variable is treated analogously. Once normalized, the coordinates are passed through a “warp” block. Conceptually, this block applies a function of the form

$$g_d(\mathbf{x}) = a_d \mathbf{x}_d + b_d + \sum_{k=1}^K c_{k,d} \tanh(s_{k,d}(\mathbf{x}_d - \nu_{k,d})), \quad (32)$$

i.e. an affine transformation, $ax + b$, augmented with a superposition of K independent bump-like deformations (scaled and shifted hyperbolic tangents). Because all parameters

(a, b, c_k, s_k, ν_k) are trainable, each bump introduces a smooth local distortion around its center ν_k . By combining multiple bumps, the transformation can selectively stretch or compress different regions of the coordinate axes. Here, $d \in \{t, r\}$ refers to the coordinate dimension and $\mathbf{x}$ to the coordinate tensor. This transformation is particularly valuable for solving critical scenarios in which a black hole forms from gravitational collapse and both sharp and smooth features coexist within the same domain. By adapting the effective resolution of the coordinates, this block yields a smooth, infinitely differentiable, and fully adaptive representation in regions with sharp gradients, especially close to the origin. Beyond these transformations, the normalized coordinates are also used to generate some low-order polynomials such as $\tilde{t}^2$, $\tilde{r}^2$ or $\tilde{t}\tilde{r}$, which are concatenated to the previous features, thereby giving more information to the model, which may help overcome spectral bias [50].

- **Lower branch.** Simultaneously, before normalizing the coordinates, a dictionary of Gaussian radial basis functions $\{\varphi_i(t, r)\}_{i=1}^M$ is constructed. This dictionary comprises M different individual functions, each defined as

$$\varphi_i(\mathbf{x}) = \exp \left[-\frac{1}{2} \sum_{d \in \{t, r\}} \left(\frac{x_d - \mu_{i,d}}{l_{i,d}} \right)^2 \right], \quad i = 1, \dots, M, \quad (33)$$

and centered at a learnable location $\mu_i = (\mu_{i,t}, \mu_{i,r})$ with learnable anisotropic scales $l_i > 0$ for both time and space. Because all parameters are trainable, the model learns *where* and *how wide* to place these Gaussian characteristics according to the physical cost function. By stacking M of such functions, the embedding yields a nonlinear, high-dimensional, and smooth representation of the original input space. This enriched map captures localized structure and facilitates the detection of regions with sharp gradients, such as those taking place near formation of black holes.

Finally, the features produced by both branches are concatenated and projected through two consecutive quadratic layers to generate the output of the block.

Quadratic residual layers. Motivated by the promising results obtained by our numerical experiments, we propose using quadratic residual layers [22] in place of simple linear layers. These implement the straightforward operation

$$z = (\mathbf{W}_2 x) \circ (\mathbf{W}_1 x) + \mathbf{W}_1 x + b, \quad (34)$$

prior to nonlinear activation, introducing two sets of trainable weights, $\mathbf{W}_1$ and $\mathbf{W}_2$. Therefore, for problems where linear layers suffice to capture the functional complexity of the target function, one would expect $\mathbf{W}_2 \rightarrow 0$. Conversely, when the model takes advantage of the second-order components, these weights will be non-negligible. Here, “ $\circ$ ” stands for the element-wise matrix multiplication (a.k.a. Hadamard product).

A central point behind the motivation is that many physical mappings are nonlinear even before the activation function. Equipping the model with a second-order interaction allows it to natively express numerous multiplicative couplings (e.g., products of variables and their derivatives) that may appear directly in the PDEs. In our particular equations, many terms are at least of second order in the variables—for example, products such as $\alpha\Phi$ or $\alpha\Pi$ —and quadratic contributions in energy densities like $X^2 + Y^2$. QRes pre-activations can naturally handle these structures, reducing the number of factorizations that linear layers would otherwise need to learn, typically improving general efficiency and mitigating residual stiffness. For resolving criticality, the extremely sharp gradients observed in the physical variables near the black hole threshold demand high model expressivity. These layers can be particularly advantageous in this regime, offering greater resolving power without a blow-up in the

number of parameters.

Adaptive activation functions. Activation functions are essential in neural networks to endow models with nonlinearity, allowing them to address general problems rather than degenerate into simple successive linear combinations of the input. Depending on the chosen activation, these functions can also be used to constrain the physical ranges of the variables of interest. In our specific setting, the final set of physical variables comprises the field ϕ , the lapse function α , and the compactness parameter C . The field ϕ is not confined to any certain interval, so no activation is necessary for this output (the identity function is used). On the other hand, both α and C are physically restricted to the range $[0, 1]$; in particular, values of $C \rightarrow 1$ represent the emergence of a black hole singularity.

Beyond constraining physics to the admissible domain of the problem at hand, activation functions also play a fundamental methodological role. Their choice can be critical to the performance of DL models and, in particular, of PINNs when solving physical scenarios. The influence of activation functions on model performance has attracted considerable attention in the recent literature. Some studies [51] analyze their effects on relatively simple problems, while others [52] design problem-specific activations tailored to the physics; an approach that, while potentially effective, may introduce excessive dependence on the problem. A widely shared conclusion is that no single “default” activation function fits all the problems in PINNs, outperforming all others in different fields. Consequently, the choice of activation is highly problem-specific, as some authors [53] propose employing a set of basis activation functions, allowing the model itself to select the certain subset from which it can benefit the most.

In our setting, because the weight mappings of the hidden layers are quadratic, it is crucial to employ activation functions that act in a bounded range. Using unbounded activations such as SiLU or ReLU could introduce numerical instabilities by producing arbitrarily large values, particularly during the early stages of training. As noted before, a main concern in the field of PINN methodologies is mitigating their spectral bias, which often causes these models to be unable to resolve problems featuring rapid oscillations and sharp gradients, which is indeed something that we will encounter when solving gravitational collapses. Among the available strategies to counteract this effect, one is to provide the model with an additional degree of freedom by including trainable activation functions that the network can adapt to its needs (see, e.g., [54] and references therein). These functions can be made trainable by introducing values that the network learns, with the specific parameterization depending on both the chosen function and the problem at hand. The parameters may be global, layer-specific, or even neuron-specific. In our case, the trainable activation for the hidden layers will be a hyperbolic tangent,

$$\sigma_{\text{hidden},i}(x) := \tanh(\psi_i x + \xi_i), \quad (35)$$

where $\{\psi_i, \xi_i\}$ are independent parameters for the i -th hidden layer: ψ_i controls the slope while ξ_i represents the offset. As noted, among (ϕ, α, C) the field $\phi(t, r)$ is not constrained, but α and C are physically restricted to the interval $[0, 1]$. Therefore, the most natural choice for these two is a (trainable) sigmoid activation

$$\sigma_{\text{output}}(x) := \text{sigmoid}(\psi_{\text{output}} x + \xi_{\text{output}}). \quad (36)$$

The set $\{\psi_{\text{output}}, \xi_{\text{output}}\}$ can be shared or distinct for the variables; in all our experiments, however, we will treat them as different. One might also consider a learnable scaling parameter that multiplies the entire function; however, this can be counterproductive, as it may incorrectly modify the admissible physical ranges of the variables.

For the gravitational collapse problems that we aim to tackle, activation functions—particularly those in the hidden layers—are not a central concern. However, making them trainable can provide the model with an additional degree of flexibility, even if they are not decisive. Relative to a vanilla PINN, the modification is modest: we introduce only two extra parameters per hidden layer and four at

the output (two for each trainable head), resulting in a negligible computational overhead. That said, care is needed: an excessive number of additional parameters can induce numerical instabilities and substantially slow training without yielding better solutions.

4. Results

In this section, we propose several initial-value problems for the scalar field $\phi(t, r)$, each leading to an independent solution and encompassing both the subcritical and supercritical regimes of gravitational collapse. We solve these problems using a range of recent methodologies from the literature—the ModPINN proposed in the previous section is included—to enable a systematic comparison across different approaches. In summary, this section is organized as follows: First, we introduce and discuss the problems to be solved and the metric used. Second, we outline the general parameter settings used for both the ModPINN and the comparison models, i.e., the setup of the methodologies and the technical details of the training process. Third, we report performance across all models and cases. Fourth, we present learning curves for the principal models. Fifth, we assess the robustness of our model with respect to the architectural choices and sample size. Sixth, we conduct a focused study of the critical regime and compare the methods performing the best. Finally, we visualize the resulting physical variables over the sampled spacetime.

Problem setup. As noted in the introduction, the seminal findings and conclusions of [1] are remarkable for their generality and universality. The study examines several families of initial conditions for the scalar field and concludes, for instance, that the exponent γ in the black hole mass law (3) appears to be a universal quantity, independent of the initial field configuration. Instead, its dependence is directly related to the deviation of a control parameter p from its critical value p^* . Despite the well-established universality of critical phenomena in gravitational collapse, in our experiments we fix the initial field profile to the “family (a)” (a.k.a. $\phi_0(r)$ from now on) configuration from [1], expressed as

$$\phi(0, r) = \phi_0 r^3 \exp\left(-\frac{(r - r_0)^q}{\delta q}\right). \quad (37)$$

This field profile corresponds to a compact Gaussian pulse for which the scaling law involving the mass of the black hole, M_{BH} , is known to persist well beyond the asymptotic limit $p \rightarrow p^*$ [4]. Specifically, the profile is fixed by the set of parameters $\{\phi_0, r_0, \delta, q\}$. Among these, the parameter of interest that we vary towards criticality is the initial amplitude ϕ_0 ; from now on, “ p ” will be identified with ϕ_0 , and the critical value “ p^* ” identified as the critical amplitude. The remaining parameters are fixed for all tests: $r_0 = 0$, $q = 2$, and $\delta = \sqrt{2}\sigma$ with $\sigma = 1$.

This profile constitutes the only prior information that is available before any numerical solution—whether via DL or FDMs—is obtained. However, the initial profiles for the other variables, $\alpha(t, r)$ and $C(t, r)$, are also required for the PINN to learn. These are calculated by setting $t = 0$ in the field equations (5–7) and solving numerically as a preprocessing step. It should be noted that, once the initial pulse is specified, the spatial derivative of the field, $\frac{\partial \phi}{\partial r}$, is straightforwardly determined, while its temporal derivative is not. Because the field evolution equations (5) are second order, the latter must be imposed. Several choices are possible; for example, $\frac{\partial \phi}{\partial t} = 0$ (time-symmetric initial data, $\Pi = 0$), or $\Pi = \Phi$ (purely ingoing field). In this work, we adopt $\Pi = 0$ for all problems and numerical solutions considered. This choice is purely pragmatic: for mathematical simplicity—both in the DL model and in generating the numerical solution—we set $\Pi = 0$ without entailing any loss of generality for the study. Regarding the boundary conditions, we impose $C(t, 0) = 0$ (the so-called elementary flatness at the origin). For $\alpha(t, 0)$, there is gauge freedom: $\alpha(t, 0)$ may be chosen arbitrarily and $\alpha(t, r)$ can be rescaled by a constant, so that at $r = R$ one may enforce $\alpha(t, R) = \frac{1}{\alpha(t, R)}$. In our

setting, we adopt derivative-based boundary conditions (23) instead of Dirichlet conditions, since fixing these variables to constant values might introduce problem-specific dependence. Moreover, since our main objective is to resolve criticality in gravitational collapse using DL-based methods, the chosen scenario is sufficient: it exhibits the relevant interesting behavior and we have a reasonably accurate estimate of the critical parameter, enabling us to distinguish subcritical from supercritical regimes. This setup therefore supports a comprehensive study of the problem, despite the fact that the space of admissible initial field profiles is theoretically unlimited.

Regarding the evaluation metric, we employ the relative mean squared error,

$$l_2 = \sum_{i \in \{\phi, \alpha, C\}} \frac{\sqrt{\sum_{(t,r) \in [0,T] \times \Omega} |\mathcal{F}_\Theta^i(t, r) - \mathcal{F}_{\text{GT}}^i(t, r)|^2}}{\sqrt{\sum_{(t,r) \in [0,T] \times \Omega} |\mathcal{F}_{\text{GT}}^i(t, r)|^2}}, \quad (38)$$

between the prediction of the model, denoted as $\mathcal{F}_\Theta$, and the ground truth solution, $\mathcal{F}_{\text{GT}}$. In particular, this metric is simply the sum of the relative errors l_2 for each output variable, so one may also define variable-wise errors $l_2^\phi, l_2^\alpha, l_2^C$, which can be useful for dissecting performance.

General configuration. As outlined in the methodology, numerous parameters and design choices can be specified regarding the configuration of our PINN model (ModPINN). Therefore, it is necessary to define a “general configuration”, which will be used in all the cases studied unless otherwise stated.

- **Loss function.** Regarding the loss computation, we set the weights in (22) as follows: $\lambda_{je} = 10^3$ for the initial conditions; $\lambda_{\mathcal{B}C, \text{left}} = \lambda_{\mathcal{B}C, \text{right}} = 1$ for the boundary conditions at $r = 0$ and $r = R$; and $\lambda_i = 1$ for all the field equations, that is, the PDE residuals. Although several studies conclude that dynamically adapting these weights to the loss of each term can be beneficial, we opt to avoid unnecessary additional complexity and keep them fixed, with $\lambda_{je} \gg \{\lambda_i\}$ to ensure accurate learning of the initial conditions, which largely determine the physical solution. Regarding the causality and spatial enforcement terms, we use $(\epsilon_t, \epsilon_r) = (50, 10)$. These values are neither excessively large nor particularly small, leading to a necessary, though nontrivial modification of the vanilla loss function.
- **Spacetime sampling.** With respect to the sampled spacetime domain (t, r) , several configurations may be considered. In general, the number of points does not need to be large; a standard grid of $N_t \times N_r = 128 \times 128$ is typically sufficient. Lighter (64×64) and heavier (256×256) grids are also examined, particularly to assess how the sampling density affects the model performance. For adaptivity and point movement, we adopt a modest value $\Lambda = 1$ in (28), which allows light repositioning of the points.
- **Neural architecture.** Regarding the network layout, we employ quadratic-weight layers, stacking six hidden layers with 50 neurons each after embedding. For the embedding itself, the upper branch uses $K = 32$ hyperbolic-tangent bumps for each input coordinate, while the lower branch includes a dictionary of $M = 32$ Gaussian radial basis functions. These choices allow us to exceed the simplicity of vanilla PINNs without imposing substantial additional cost. Finally, after concatenating both branches, we use a 50-neuron hidden layer to project the combined features to the output of the block. Regarding the random initialization of trainable parameters in the linear layers, we adopt the default setting in PyTorch, namely the Kaiming He uniform initialization scheme [55]. For the function $g_d(\mathbf{x})$, the parameters a_d and b_d , which are shared across all bumps, are initially set to 1 and 0, respectively. The $\{s_{k,d}\}$ are initialized to 1 or all k , while the $\{\nu_{k,d}\}$ are uniformly initialized over the

interval $[-1, 1]$ partitioned into k segments. Similarly, for the lower branch, the set $\{\mu_{i,d}\}$ is sampled randomly from a uniform distribution over $[-1, 1]$, while all elements of $\{l_{i,d}\}$ are initially assigned to 1. Finally, for the trainable activation functions in (36) and (35), the parameter sets $\{\psi_i\}$, ψ_{output} and $\{\xi_i\}$, ξ_{output} are set to 1 and 0, respectively, for all hidden layers and for both physical outputs.

- **Training details.** The optimizer employed in all training runs and experiments has been Shampoo with Adam in the Preconditioner’s eigenbases (SOAP) [56], a recently introduced method in the optimization literature. SOAP has achieved state-of-the-art results in general tasks such as language modeling. More interesting to our work, recent studies [57] have applied SOAP to solve various PDE problems with PINN-type models, reporting highly competitive performance and even surpassing commonly used baselines such as Adam or AdamW. Nevertheless, this optimizer is effectively of (quasi) second-order, which typically means slightly slower training per iteration; however, this is countered by faster and more accurate convergence. Unless otherwise specified, we train for 100,000 epochs with a fixed learning rate of 10^{-4} and normalize the gradient norm to unity. If overfitting is detected—manifested as degradation in the physics loss—we retain the best-performing checkpoint for each case, the selected models being the ones reported in the analyses.

Numerical comparisons. The field equations (5-7) are straightforwardly solvable in subcritical regimes, i.e., in the absence of black holes, where a vanilla PINN is typically enough. However, as the control parameter p (here, the initial amplitude ϕ_0) approaches its critical value p^* , the solution develops increasingly sharp and localized gradients (in both space and time). This regime demands a more exhaustive analysis and a modeling specifically tailored to handle such extreme behaviors. Given the inherent difficulty of modeling gravitational collapses, we refrain from relying on a single model. Instead, we compare solutions produced by several PINN-based approaches proposed in recent literature, encompassing methodologies such as the KANs [23], the PirateNets [18] or the PINN “FP64” [58] which refers to a vanilla PINN where double-precision floating-point format has been considered. Other methodologies—such as PINNs-Former [24], which is based on attention, and the recent PINN-Mamba [25], which introduces subsequence modeling via State Space Models (SSMs)—were also considered but excluded from the final study due to their training and computational requirements being substantially higher while yielding performances similar to, or even worse than, other simpler models. All training was performed in PyTorch on an NVIDIA A100 (40GB) GPU.

The final results for all models considered and four distinct initial-condition cases are reported in Table 1. From previous numerical studies [1, 4], we know with high accuracy that the critical value of the parameter ϕ_0 in the proposed problem occurs near $\phi_0 = 0.1125$, which we adopt as the critical scenario of gravitational collapse. Consequently, we examine two subcritical settings ($\phi_0 = 0.1, 0.10625$) and one supercritical setting ($\phi_0 = 0.125$). The exact same loss function, with parameters $(\epsilon_1, \epsilon_r) = (50, 10)$, is used for all models, as is the adaptive spacetime sampling prescribed by our methodology with $\Lambda = 1$. Therefore, the purpose has been to compare neural architectures to handle the gravitational collapse while keeping all other factors fixed. Regarding model-specific settings, we note that, where feasible, all architectures employ six hidden layers with 50 neurons each—except PirateNet, which is organized in blocks rather than layers, and KAN, which uses three layers with five nodes each. Where possible, the hyperbolic tangent is used as an activation function for the hidden layers. Finally, for the proposed ModPINN, the previously introduced standard configuration is adopted.

For each case and each model, we have conducted 10 independent training runs, randomizing the seed—and thus the initial parameters—in every instance, thereby generating some statistics.

Table 1. Best achieved values of the relative total l_2 error (38) after 100,000 epochs across state-of-the-art models for the scalar-field initial-value problems. The best results for each case are shown in bold.

Model	ϕ_0			
	0.1	0.10625	0.1125	0.125
PINN	0.119 ± 0.009	0.160 ± 0.013	0.694 ± 0.210	0.709 ± 0.340
PINN “FP64” [58]	0.114 ± 0.005	0.165 ± 0.010	0.710 ± 0.240	0.863 ± 0.610
sf-PINN [21]	0.113 ± 0.006	0.138 ± 0.017	0.960 ± 0.290	0.855 ± 0.320
QRes [22]	0.111 ± 0.009	0.132 ± 0.008	0.817 ± 0.203	0.333 ± 0.160
KAN [23]	0.456 ± 0.250	0.498 ± 0.320	1.888 ± 0.190	2.107 ± 0.210
PirateNet [18]	0.133 ± 0.009	0.151 ± 0.030	0.698 ± 0.240	0.566 ± 0.180
ModPINN	0.114 ± 0.006	0.152 ± 0.015	0.588 ± 0.060	0.297 ± 0.090

We report the mean performance alongside the standard deviation. Inspecting the results, we observe that the critical and supercritical cases exhibit the largest variability across most models, highlighting the difficulty in solving these scenarios. For the critical case, ModPINN achieves the best performance in both mean accuracy and variability, indicating greater stability than the alternatives for this scenario. The same conclusion holds for the supercritical case with $\phi_0 = 0.125$. In contrast, in the two subcritical settings, the purely quadratic PINN (QRes) leads to the strongest results; however, as noted, these cases are easily solved by a vanilla PINN due to their relative simplicity and smoothness, so the outcome and the similarity across models are generally accepted. It is therefore expected that QRes ranks among the strongest models: quadratic-weight layers help mitigate the frequency bias typical of PINNs. These layers also aid in resolving sharp localized gradients, as evidenced by the proposed ModPINN. However, for $\phi_0 = 0.1125$ and 0.125 , QRes is not among the top performers. The principal reason is that, while quadratic layers can substantially benefit such problems, they may also induce numerical instabilities. Therefore, implementing the proposed embedding—which normalizes input coordinates to numerically manageable values and generates additional features—indeed improves performance on criticality and gravitational collapse cases. These are precisely the regimes of greatest interest as they reveal the major physical phenomena.

For the remaining models, performance in subcritical cases is generally consistent across architectures, with the exception of the KAN considered, which is slightly behind the others. This behavior may be a consequence of its chosen architectural hyperparameters, as we tried to find a balance between computational cost and accuracy, since it is known that KANs tend to train more slowly. It should be noted that the results reported here reflect a single configuration of the methodology beyond architecture alone. With a careful study of hyperparameter tuning, better outcomes could likely be achieved for all methodologies, particularly those that perform poorly in this comparison.

Training curves. As noted earlier, the relative quadratic errors l_2 defined in (38) can be decomposed into a sum of components, $l_2 = l_2^\phi + l_2^\alpha + l_2^r$. This decomposition allows us to assess the performance of each variable individually, rather than relying solely on the total sum, which may lead to masked deficiencies in training. In figure 3, we present the learning curves for these metrics across all considered problems, focusing on the four best performing models. For each variable, the error scale is kept fixed in all different scenarios. The first conclusion is that the cases showing gravitational collapse, especially the critical case ($\phi_0 = 0.1125$), are by far the most challenging to train, achieving higher errors in general and evolving within a much narrower range. In general, all models achieve comparable performance in this case. It is worth highlighting that, given the construction of the l_2 metric, small variations can translate into substantial improvements in the target variable—particularly in regions exhibiting sharp changes over small (t, r) intervals. Moreover, even though most models converge faster than the vanilla PINN, none stands out as decisively superior—except ModPINN, which yields an average total error of 0.588, compared with 0.694 and 0.698 for

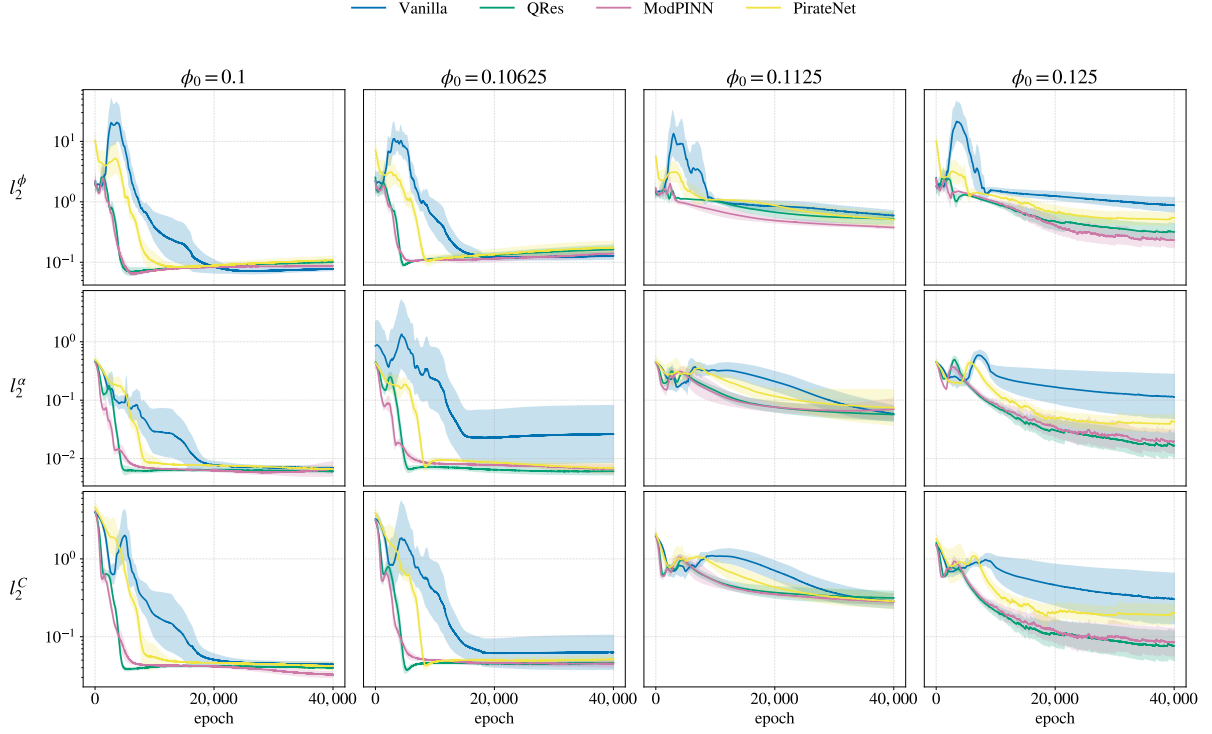


Figure 3. Convergence of the relative l_2 errors for each physical variable (row by row) over training, across different models and initial-value problems. Beyond approximately 40,000 epochs, performance gains become barely minimal.

the next best (the vanilla and PirateNet, respectively) and with variability roughly three times smaller. In principle, given the difficulty of the critical case, likely due to a sharp and hard-to-navigate loss landscape, we might expect independent trainings to yield slightly different results. However, as Table 1 has shown, the variability is concerning across models, on the order of $\sim 30\%$ relative to the mean, except for ModPINN, where it is only $\sim 10\%$. This indicates a more stable model that may admit further optimization and potentially even better results with a more exhaustive preconditioning of the hyperparameters. For the supercritical case, $\phi_0 = 0.125$, it seems that all models identify black hole formation more clearly as the dominant result, as evidenced by the errors being uniformly lower than those of the vanilla PINN. In addition, the learning curves suggest that the problem becomes less difficult with a substantially wider range of progress during training. Finally, for the subcritical cases, all models give similar solutions, except for $\alpha(t, r)$ produced by the vanilla PINN at $\phi_0 = 0.10625$. We also observe that every alternative model converges faster—roughly in half the training time—representing a clear advantage. However, in general, the main conclusion is that all methods perform reasonably well due to the relative “ease” of these problems: smooth gradients make them straightforward for any of the models to solve.

It is important to note that these learning curves correspond to the training runs reported in table 1; thus, all methodological settings are the same, including the spatiotemporal sampling size $N_t \times N_r = 128 \times 128$. Several factors could be adjusted to improve results: increasing sampling density, fine-tuning ϵ_t and ϵ_r , and / or refining the weights assigned to the initial and boundary conditions. For the critical case in particular, a more detailed analysis is needed, which we will present in the remainder of the paper.

Robustness of ModPINN. Up to this point, we have reviewed a set of recent models in the literature, but it is worth putting more focus on our own proposal, which we have denoted as ModPINN. In essence, this architecture is no more than a modified PINN that replaces linear layers with quadratic ones, and whose main strength is an embedding introduced before the layer concatenation. As we have shown, this embedding leads to improved results for the supercritical

case and, specifically, for the critical case $\phi_0 = 0.1125$, improving average performance and reducing variability. Taking into account that this model is the one giving the strongest results for gravitational collapse, it is necessary to carry out a more thorough examination of how various pre-training design choices influence its performance. Among these factors, we include the sampling size (that is, the number of spatiotemporal sampling points, $N_t \times N_r$) and the scale of the neural architecture, in particular the number of quadratic layers and the number of neurons per layer. Another parameter of interest that we examine here is the strength of the temporal causality, ϵ_t , while ϵ_r will be constant and fixed at 10 for this analysis.

Figure 4 reports the achieved l_2 as a function of $\epsilon_t \in \{0, 0.1, 1, 10, 100\}$. Each row of sub-figures corresponds to one sampling size— 64×64 , 128×128 and 256×256 —and within each sub-figure we display the errors for three different network sizes. Each column therefore corresponds to each initial-condition problem. By varying ϵ_t , we move the spectrum from a “vanilla” loss with respect to temporal causality to one that strongly enforces it at $\epsilon_t = 100$. Regarding sampling sizes, we consider grids from $64 \times 64 = 4,096$ to $256 \times 256 = 65,536$. Although the latter may appear large, we note that the original finite-difference simulations [1] employed on the order of 500,000 grid points. In principle, then, the sampling size could be increased substantially; however, doing so would likely require batching strategies to feed the network, probably leading to higher computation times that are out of the main scope of this study. Several key insights can be concluded from this figure.

- The 10×75 architecture—the deepest and widest configuration—yields the most stable overall results across all cases, much like the largest sampling size, 256×256 , which also exhibits the greatest stability. Consequently, the model clearly benefits from larger sampling grids and higher network capacities, providing robustness independent of the problem considered. In contrast, smaller networks display greater variability as ϵ_t changes. This behavior is a clear indication of the inherent physical complexity of the problem.
- In general, the motivation behind the enforcement of temporal causality is to prevent PINNs from “looking ahead” in time, that

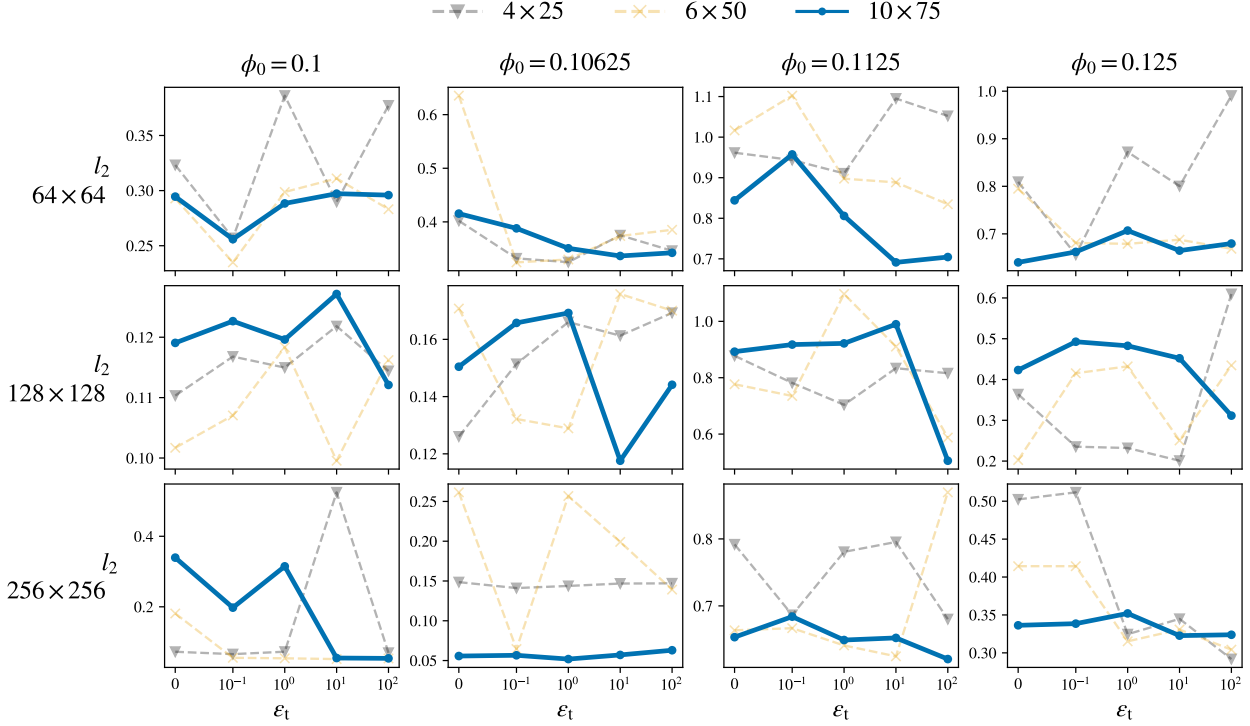


Figure 4. Relative total l_2 error of the proposed ModPINN for the considered scalar-field initial-value problems (by column) as a function of the hyperparameter ϵ_t , which modulates the strength of the temporal causality. Each row corresponds to a different spatiotemporal grid resolution (t, r) , and each sub-figure reports results for three network sizes.

is, from resolving late-time points before early-time ones [45]. This approach can be helpful in problems with a strong dependence on initial conditions, where the evolution of the system is fully determined by them. Consequently, it is reasonable to expect that in a smooth, easy-to-solve case like $\phi_0 = 0.1$, causality enforcement will have limited impact, which is essentially what we observe. For this problem, there is no clear trend in l_2 as ϵ_t increases, because the solutions with $\epsilon_t = 0$ are generally good and any observed improvements are not substantial.

- Following the same logic, for the critical case $\phi_0 = 0.1125$, we observe a stronger dependence on ϵ_t , although the errors are generally higher due to the steep highly localized gradients mentioned. We see that the model clearly benefits from enforcing causality, particularly for the 128×128 case, where the error quickly drops for $\epsilon_t = 10^2$. Regarding the $\phi_0 = 0.125$ scenario—specifically with 256×256 points and a 10×75 network—the solution is markedly stable, showing only minor sensitivity to ϵ_t . By contrast, when smaller networks are used, the improvement becomes appreciable.

These results demonstrate that the model benefits from large sampling grids as well as deep and wide network architectures. While adaptive sampling can indeed be helpful when the number of points is relatively small, in general a denser spacetime regime helps the model to solve the task better. Moreover, enforcing causality generally improves stability, and it is especially beneficial in cases where strongly tuning the initial conditions is critical, as demonstrated for $\phi_0 = 0.1125$.

Critical regime. At this point, it is important to focus on the analysis of the critical case for this family of initial data. This scenario, defined by $\phi_0 = 0.1125$, represents the boundary between dispersive fields and black hole formation. Accordingly, accurately resolving this regime is of particular interest for the community, paying special attention to the models that have yielded the most promising results: the QRes model, the PirateNet, the surprisingly effective vanilla PINN, and, finally, the ModPINN proposed in this review.

In figure 5 (top), we show the mean relative l^2 error of each physical variable, together with its variability between several random trains, for all models considered in the critical case. In general, the lapse function $\alpha(t, r)$ is the best resolved quantity, having modest variability and slightly better performance in the vanilla PINN. This behavior can be attributed to its generally smooth profile, which does not present sharp gradients, making it easier to be approximated by any model. Regarding the other two variables, ModPINN achieves better performance than other state-of-the-art proposals, with our main claim focused on model stability. Across multiple independent trainings with random initializations, we observe that the convergence of our model is substantially more robust. Near the threshold of collapse, it is natural to expect a sharp loss landscape, such that small differences during optimization can lead to relatively different outcomes. However, ModPINN successfully attenuates this variability by yielding a more deterministic methodology for these problems in particular.

In the two bottom panels, we depict how the l^2 errors vary with the hyperparameter Λ , which controls the “strength” with which spacetime points (t, r) are moved towards regions exhibiting higher values of the normalized physical loss, as defined in (28). The lower-left panel reports this magnitude for the two subcritical cases ($\phi_0 = 0.1, 0.10625$), while the lower-right panel presents the corresponding results for the critical and supercritical cases ($\phi_0 = 0.1125, 0.125$). We observe that as Λ increases within the range considered, the error shows a clear decreasing trend for the subcritical scenarios, with a particular reduction in variability at $\Lambda = 10$ for the case $\phi_0 = 0.1$. For the critical scenario $\phi_0 = 0.1125$, the trend is also downward from $\Lambda = 0$ to 1, with the error dropping from approximately ~ 1.0 to ~ 0.6 . This constitutes a substantial improvement that should not be eclipsed by the use of a broader scale relative to the other panel. Regarding $\Lambda = 10$ we observe a sudden increase in error for this case, indicating that the solutions deteriorate when this parameter is overly enlarged. This behavior is somewhat expected: the physics lies near the boundary of criticality, and the collocation points are required and used within a highly localized region of spacetime mainly determined by the

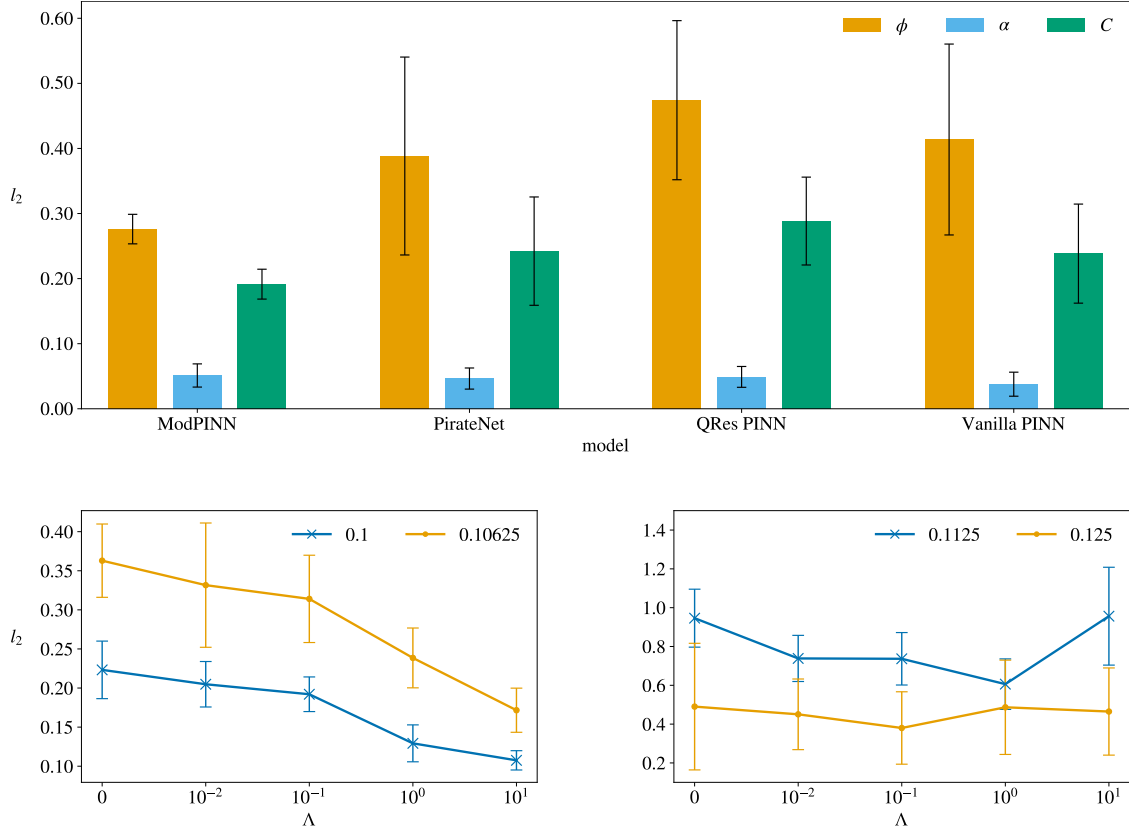


Figure 5. Top: l_2 errors for each variable, together with their variability across ten independent runs, shown for different models considered. Bottom: total l_2 errors as a function of the adaptive-sampling strength Λ , evaluated for each initial-condition problem. Only the ModPINN model is considered.

black hole formation. Excessive and rapid relocation of the points can move some of them into undesirable regions, thereby devaluing their utility and, overall, leading to worse solutions. Finally, for the supercritical case with $\phi_0 = 0.125$, the trend is essentially flat. It is worth noting that the model under consideration here is ModPINN, which—consistent with table 1—provides the best performance for this regime with quite low variability. Consequently, this scenario is already well resolved without additional techniques, as the model alone effectively minimizes the loss. It is therefore unsurprising that the solution does not improve substantially when adaptive sampling is introduced. By contrast, performing the same analysis with models such as the vanilla PINN—where this case is less accurately captured—might reveal a decreasing trend under the said sampling methodology.

Evolution of the physical variables. Figure 6 displays the physical variables—the scalar field ϕ , the lapse function α , and the compactness parameter C —over the spacetime domain $(t, r) \in [0, 10] \times [0, 10]$. All four initial-condition scenarios are shown. By columns, we present the ground truth solution, followed by the predictions from ModPINN and from the vanilla PINN, respectively. Perhaps the most interesting feature is the parameter C , which approaches 1.0 only in regions where an apparent horizon arises. As seen for $\phi_0 = 0.1125$ and 0.125, the coordinate singularity emerges at times $t \gtrsim 5.0$ with r values near the origin. As ϕ_0 increases beyond 0.1125, the black hole seems to shift to $r > 0$. In both scenarios, the region where $C \sim 1$ is narrowly localized in r , a structure that ModPINN reproduces notably well. Although the vanilla PINN can detect this feature as well, it moves the coordinate singularity towards smaller r , yielding an overall inferior result. However, when using only the loss function (22) together with the adaptive sampling, the vanilla PINN is still able to identify the black hole with relatively strong performance. Regarding the other variables, we find that they are generally well resolved across all cases and for both models. The expected physical behavior is recovered: $\alpha \rightarrow 1$ (i.e., the spacetime tends towards flatness) in dispersive regions, while $\alpha \rightarrow 0$ precisely where

the coordinate singularity forms. This is mirrored in the scalar field, which exhibits sharply localized and strongly negative values in r when $C \rightarrow 1$. These limiting behaviors are captured thanks to the boundary conditions (16) which enforce the correct asymptotics at distances sufficiently far from the source of perturbation.

5. Conclusions

The gravitational collapse of a massless scalar field remains a demanding benchmark for numerical methods in numerical relativity, as it exhibits critical behavior at the boundary between dispersion and black hole formation. The original investigations conducted by Choptuik [1, 4] tackled the solution of the field equations using finite-difference methods. Those equations govern the evolution of a massless scalar field ϕ in GR, minimally coupled to the spacetime metric $g_{\mu\nu}$, and, provided that the initial data is sufficiently strong, black hole formation can occur in the model. The studies of [1, 4] probed the existence of a critical value, p^* , for the control parameter fully describing the initial scalar field distribution. As $p \rightarrow p^*$, the mass of the black hole, M_{BH} , scales with the deviation $|p - p^*|$ via a universal exponent γ . Additional phenomena, such as self-similarity and solution echoing, were also observed.

The work reported in this paper has revisited this challenging physical problem, focusing on solutions at and around the critical point. Our primary aim has been to demonstrate that neural network-based methodologies, such as Physics-Informed Neural Networks (PINNs), can serve as a viable and useful alternative to finite differences. To this end, we have considered four different initial-value problems for the field, using the amplitude of the initial kernel, ϕ_0 , as the control parameter, with the critical case included among them. We have addressed these cases using several recent methodologies from the literature and introduced our own, termed ModPINN, which has been tailored to this class of scenarios, including several novel aspects from recent works such as coordinate embedding, quadratic neural layers,

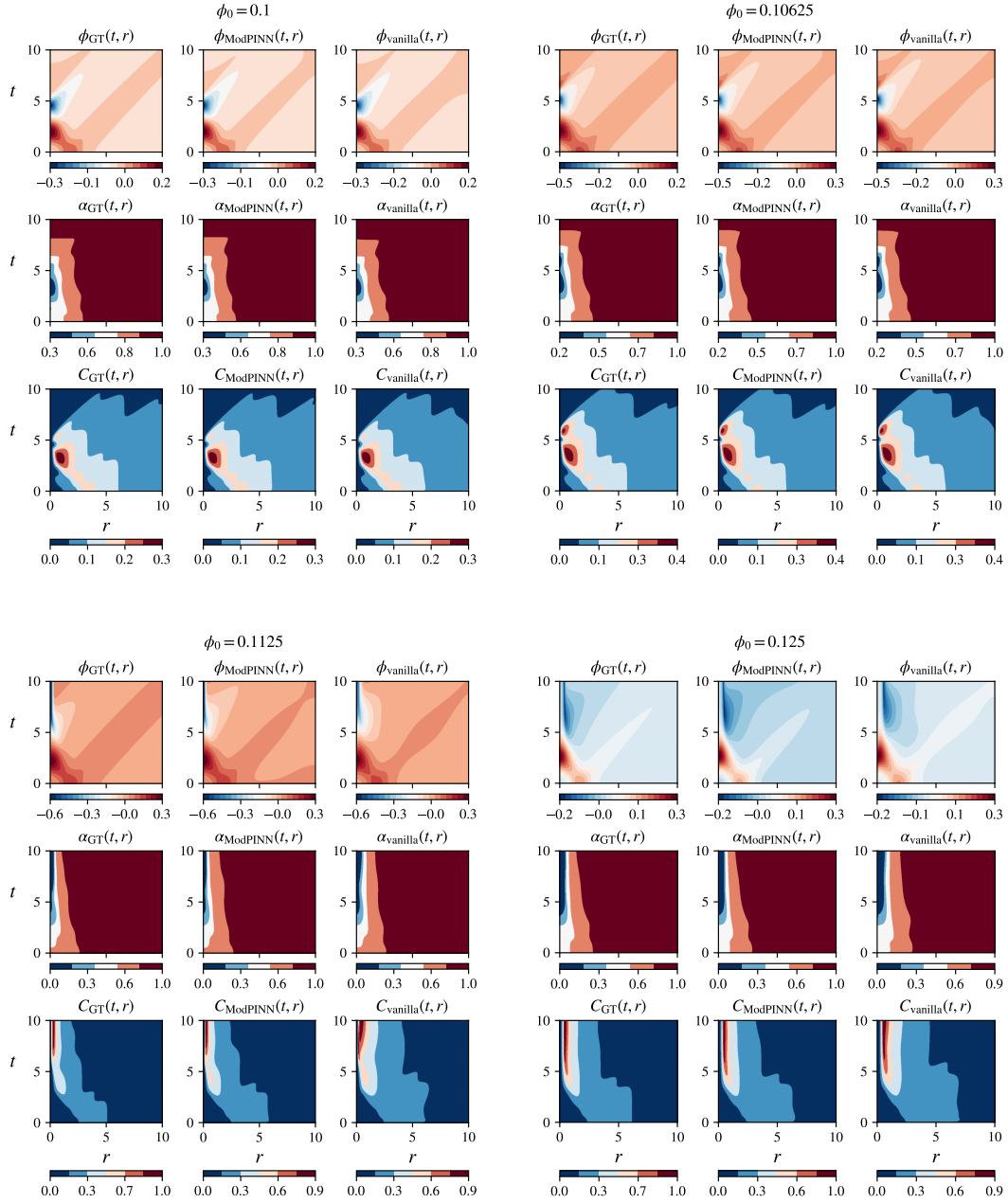


Figure 6. Two-dimensional visualizations of the physical variables (ϕ, α, C) depicted over the sampled spacetime for each initial-condition case, $\phi_0 \in \{0.1, 0.10625, 0.1125, 0.125\}$. For each problem, the columns—from left to right—display the “ground truth” solution obtained via finite differences, the solution produced by the ModPINN, and the solution from a vanilla PINN, respectively. All cases employ 128×128 sampling points and a standard configuration for both models.

and learnable activation functions. Our model delivers slightly better critical solutions with lower variability than other state-of-the-art approaches, even though all of them have led to very competitive predictions. Furthermore, by implementing a relatively simple algorithm that moves the spacetime sampling points (t, r) , we treat them as particles subject to a “force” with a prescribed acceleration modulated via the Λ hyperparameter in (28), which indicates how quickly points migrate to regions with higher physical loss. The key consequence is that we are able to work with comparatively small sets, unlike traditional numerical methods that resolve black holes by brute force—highly densifying the mesh in spacetime regions that are known a priori to be most challenging.

Irrespective of the specific models used, much of the substantive effort resides in the design of the physical loss. The boundary conditions for $r \gg 0$ introduced in (16), are formulated in terms of the derivatives of the physical variables, whose enforcement leads to the asymptotic behaviors that these variables are expected to have

far from the source of perturbation. These conditions, as well as the mechanisms of temporal and spatial enforcement, have proved to be crucial and, in some sense, help to emulate the adaptive refinements of finite-difference schemes without incurring additional computational effort. In the analysis displayed in figure 4, we have examined the robustness of ModPINN with respect to several factors, including the density of sampling points and the width and depth of the network. We have found that wider and deeper architectures indeed facilitate better solutions, and they also perform relatively well on comparatively large spatiotemporal grids such as 256×256 points—still small in comparison with those used in finite-difference schemes. Overall, these configurations exhibit greater robustness and smoother performance across independent trials. While further analysis could be considered, this allows us to highlight a notable advantage of PINNs: aside from hardware constraints, there is substantial flexibility to scale model size and sampling, and the resulting $N_t \times N_r$ grids in deep learning setups are orders of magnitude smaller

than those typically required by finite differences.

Across all models considered for the target problems, we can conclude that physics-informed deep-learning approaches are indeed capable of resolving gravitational collapse driven by a massless scalar field in a spherically symmetric, time-dependent spacetime. By judiciously selecting activation functions to restrict variables to their physically feasible ranges and by formulating boundary conditions in derivative form, we obtain solutions close to the ground truth. While there is still room for improvement, the results indicate that these models are progressing in the right direction. Nevertheless, a PINN can be seen as an alternative (or surrogate) to a finite-difference numerical solver. Broadly speaking, PINN-type models employ the governing PDEs as components of a constructed loss function—alongside additional information—to solve the physical problem at hand. Consequently, once the initial scalar field is specified, the full evolution of ϕ , α , and C over (t, r) is determined. Thus the problem becomes deterministic; gravitational collapse may or may not occur depending on that initial configuration. Therefore, this imposes certain limitations when investigating aspects such as the universality of the exponent γ in the scaling law of the black hole mass M_{BH} near criticality, as well as the generality of the proportionality constant c_f with respect to the family of the initial field. To examine these aspects—and to study the self-similarity of solutions as $p \rightarrow p^*$ —we would need access to a broad set of initial condition sets and their full corresponding solutions. Therefore, addressing these questions will require a different point of view—one that allows us to traverse families of initial conditions with full flexibility and to generate data with relative ease. In this sense, the present project paves the way for future studies that can probe the aforementioned issues, extend to alternative spacetime metrics and matter models, and tackle more complex scenarios in which finite-difference methods could be either inapplicable or impractical due to prohibitive computational costs.

6. Acknowledgments

This work has been partially funded by the Valencian Government grant with Reference Number CIAICO/2024/111; the Spanish Ministry of Economic Affairs and Digital Transformation through the QUANTUM ENIA project call-QuantumSpain project, and the European Union through the Recovery, Transformation and Resilience Plan—NextGenerationEU within the framework of the Digital Spain 2025 Agenda. This work has been also supported by the Spanish Agencia Estatal de Investigación (grant PID2024-159689NB-C21) funded by MCIN/AEI/10.13039/501100011033 and ERDF A way of making Europe, by the Prometeo excellence programme grant CIPROM/2022/49 funded by the Generalitat Valenciana, and by the European Horizon Europe staff exchange (SE) programme HORIZON-MSCA2021-SE-01 Grant No. NewFunFiCO-101086251. MWC is supported by NSERC. R. RdA is supported by PID2020-113644GB-I00 from the Spanish Ministerio de Ciencia e Innovación and by PROMETEO/2022/69 funded by the Generalitat Valenciana.

References

- [1] Matthew W. Choptuik. “Universality and scaling in gravitational collapse of a massless scalar field”. In: *Phys. Rev. Lett.* 70 (1 Jan. 1993), pp. 9–12. DOI: 10.1103/PhysRevLett.70.9. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.70.9>.
- [2] Carsten Gundlach and José M. Martín-García. “Critical Phenomena in Gravitational Collapse”. In: *Living Reviews in Relativity* 10.1 (2007), p. 5. ISSN: 1433-8351. DOI: 10.12942/lrr-2007-5. URL: <https://doi.org/10.12942/lrr-2007-5>.
- [3] Tim Taves and Gabor Kunstatter. “Higher dimensional Choptuik scaling in Painleve-Gullstrand coordinates”. In: *Phys. Rev. D* 84 (4 Aug. 2011), p. 044034. DOI: 10.1103/PhysRevD.84.044034. URL: <https://link.aps.org/doi/10.1103/PhysRevD.84.044034>.
- [4] M. W. Choptuik. *The (Unstable) Threshold of Black Hole Formation*. 1998. arXiv: gr-qc/9803075 [gr-qc]. URL: <https://arxiv.org/abs/gr-qc/9803075>.
- [5] David Garfinkle, Curt Cutler, and G. Comer Duncan. “Choptuik scaling in six dimensions”. In: *Phys. Rev. D* 60 (10 Oct. 1999), p. 104007. DOI: 10.1103/PhysRevD.60.104007. URL: <https://link.aps.org/doi/10.1103/PhysRevD.60.104007>.
- [6] Evgeny Sorkin and Yonatan Oren. “Choptuik’s scaling in higher dimensions”. In: *Phys. Rev. D* 71 (12 June 2005), p. 124005. DOI: 10.1103/PhysRevD.71.124005. URL: <https://link.aps.org/doi/10.1103/PhysRevD.71.124005>.
- [7] Erik Jiménez-Vázquez and Miguel Alcubierre. “Critical gravitational collapse of a massive complex scalar field”. In: *Phys. Rev. D* 106 (4 Aug. 2022), p. 044071. DOI: 10.1103/PhysRevD.106.044071. URL: <https://link.aps.org/doi/10.1103/PhysRevD.106.044071>.
- [8] Carsten Gundlach, Thomas W. Baumgarte, and David Hilditch. “Critical phenomena in gravitational collapse with two competing massless matter fields”. In: *Phys. Rev. D* 100 (10 Nov. 2019), p. 104010. DOI: 10.1103/PhysRevD.100.104010. URL: <https://link.aps.org/doi/10.1103/PhysRevD.100.104010>.
- [9] Florencia Benítez et al. “Critical Collapse of a Scalar Field in Semiclassical Loop Quantum Gravity”. In: *Phys. Rev. Lett.* 124 (7 Feb. 2020), p. 071301. DOI: 10.1103/PhysRevLett.124.071301. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.124.071301>.
- [10] Rong-Gen Cai, Li-Wei Ji, and Run-Qiu Yang. “On the Critical Behavior of Gapped Gravitational Collapse in Confined Spacetime”. In: *Communications in Theoretical Physics* 68.1 (July 2017), p. 67. DOI: 10.1088/0253-6102/68/1/67. URL: <https://dx.doi.org/10.1088/0253-6102/68/1/67>.
- [11] Marc Finzi et al. *A Stable and Scalable Method for Solving Initial Value PDEs with Neural Networks*. 2023. arXiv: 2304.14994 [cs.LG]. URL: <https://arxiv.org/abs/2304.14994>.
- [12] Hatefi, Ehsan, Hatefi, Armin, and López-Sastre, Roberto J. “Analysis of black hole solutions in parabolic class using neural networks”. In: *Eur. Phys. J. C* 83.7 (2023), p. 623. DOI: 10.1140/epjc/s10052-023-11781-8. URL: <https://doi.org/10.1140/epjc/s10052-023-11781-8>.
- [13] Armin Hatefi and Ehsan Hatefi. “Sequential Monte Carlo with cross-validated neural networks for complexity of hyperbolic black hole solutions in 4D”. In: *The European Physical Journal C* 83.11 (Nov. 2023), p. 1083. ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-023-12284-2. URL: <https://doi.org/10.1140/epjc/s10052-023-12284-2>.
- [14] Armin Hatefi, Ehsan Hatefi, and Roberto J. Lopez-Sastre. “Neural networks assisted Metropolis-Hastings for Bayesian estimation of critical exponent on elliptic black hole solution in 4D using quantum perturbation theory”. In: *Journal of Cosmology and Astroparticle Physics* 2024.09 (Sept. 2024), p. 015. DOI: 10.1088/1475-7516/2024/09/015. URL: <https://dx.doi.org/10.1088/1475-7516/2024/09/015>.
- [15] 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”. In: *Journal of Computational Physics* 378 (2019), pp. 686–707. ISSN: 0021-9991. DOI: <https://doi.org/10.1016/j.jcp.2018.10.045>.
- [16] Allan Pinkus. “Approximation theory of the MLP model in neural networks”. In: *Acta Numerica* 8 (1999), pp. 143–195. DOI: 10.1017/S0962492900002919.

- [17] Yifan Wang and Linlin Zhong. “NAS-PINN: Neural architecture search-guided physics-informed neural network for solving PDEs”. In: *Journal of Computational Physics* 496 (2024), p. 112603. ISSN: 0021-9991. DOI: <https://doi.org/10.1016/j.jcp.2023.112603>. URL: <https://www.sciencedirect.com/science/article/pii/S0021999123006988>.
- [18] Sifan Wang et al. “PirateNets: Physics-informed Deep Learning with Residual Adaptive Networks”. In: *Journal of Machine Learning Research* 25.402 (2024), pp. 1–51. URL: <http://jmlr.org/papers/v25/24-0313.html>.
- [19] Matthew Tancik et al. “Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 7537–7547. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/55053683268957697aa39fba6f231c68-Paper.pdf.
- [20] Sifan Wang, Hanwen Wang, and Paris Perdikaris. “On the eigenvector bias of Fourier feature networks: From regression to solving multi-scale PDEs with physics-informed neural networks”. In: *Computer Methods in Applied Mechanics and Engineering* 384 (2021), p. 113938. ISSN: 0045-7825. DOI: <https://doi.org/10.1016/j.cma.2021.113938>. URL: <https://www.sciencedirect.com/science/article/pii/S0045782521002759>.
- [21] Jian Cheng Wong et al. “Learning in Sinusoidal Spaces With Physics-Informed Neural Networks”. In: *IEEE Transactions on Artificial Intelligence* 5.3 (2024), pp. 985–1000. DOI: [10.1109/TAI.2022.3192362](https://doi.org/10.1109/TAI.2022.3192362).
- [22] Jie Bu and Anuj Karpatne. *Quadratic Residual Networks: A New Class of Neural Networks for Solving Forward and Inverse Problems in Physics Involving PDEs*. 2021. arXiv: [2101.08366](https://arxiv.org/abs/2101.08366) [cs.LG]. URL: <https://arxiv.org/abs/2101.08366>.
- [23] Ziming Liu et al. *KAN: Kolmogorov-Arnold Networks*. 2025. arXiv: [2404.19756](https://arxiv.org/abs/2404.19756) [cs.LG]. URL: <https://arxiv.org/abs/2404.19756>.
- [24] Zhiyuan Zhao, Xueying Ding, and B. Aditya Prakash. *PINNs-Former: A Transformer-Based Framework For Physics-Informed Neural Networks*. 2024. arXiv: [2307.11833](https://arxiv.org/abs/2307.11833) [cs.CE]. URL: <https://arxiv.org/abs/2307.11833>.
- [25] Chenhui Xu et al. *Sub-Sequential Physics-Informed Learning with State Space Model*. 2025. arXiv: [2502.00318](https://arxiv.org/abs/2502.00318) [cs.LG]. URL: <https://arxiv.org/abs/2502.00318>.
- [26] Albert Gu and Tri Dao. *Mamba: Linear-Time Sequence Modeling with Selective State Spaces*. 2024. arXiv: [2312.00752](https://arxiv.org/abs/2312.00752) [cs.LG]. URL: <https://arxiv.org/abs/2312.00752>.
- [27] I.M Sobol’. “On the distribution of points in a cube and the approximate evaluation of integrals”. In: *USSR Computational Mathematics and Mathematical Physics* 7.4 (1967), pp. 86–112. ISSN: 0041-5553. DOI: [https://doi.org/10.1016/0041-5553\(67\)90144-9](https://doi.org/10.1016/0041-5553(67)90144-9).
- [28] Michael Stein. “Large sample properties of simulations using Latin hypercube sampling”. In: *Technometrics* 29.2 (1987), pp. 143–151.
- [29] Michael D. McKay, Richard J. Beckman, and William J. Conover. “A comparison of three methods for selecting values of input variables in the analysis of output from a computer code”. In: *Technometrics* 42.1 (2000), pp. 55–61.
- [30] Vittorio Bauduin, Salvatore Cuomo, and Vincenzo Schiano Di Cola. “Impact of collocation point sampling techniques on PINN performance in groundwater flow predictions”. In: *Journal of Computational Mathematics and Data Science* 14 (2025), p. 100107. ISSN: 2772-4158. DOI: <https://doi.org/10.1016/j.jcmds.2024.100107>. URL: <https://www.sciencedirect.com/science/article/pii/S277241582400018X>.
- [31] Jose Florido et al. “Investigating Guiding Information for Adaptive Collocation Point Sampling in PINNs”. In: *Computational Science – ICCS 2024*. Springer Nature Switzerland, 2024, pp. 323–337. ISBN: 9783031637599. DOI: [10.1007/978-3-031-63759-9_36](https://doi.org/10.1007/978-3-031-63759-9_36). URL: http://dx.doi.org/10.1007/978-3-031-63759-9_36.
- [32] Chenxi Wu et al. “A comprehensive study of non-adaptive and residual-based adaptive sampling for physics-informed neural networks”. In: *Computer Methods in Applied Mechanics and Engineering* 403 (2023), p. 115671. ISSN: 0045-7825. DOI: <https://doi.org/10.1016/j.cma.2022.115671>. URL: <https://www.sciencedirect.com/science/article/pii/S0045782522006260>.
- [33] Mohammad Amin Nabian, Rini Jasmine Gladstone, and Hadi Meidani. “Efficient training of physics-informed neural networks via importance sampling”. In: *Computer-Aided Civil and Infrastructure Engineering* 36.8 (Apr. 2021), pp. 962–977. ISSN: 1467-8667. DOI: [10.1111/mice.12685](https://doi.org/10.1111/mice.12685). URL: <http://dx.doi.org/10.1111/mice.12685>.
- [34] Xiaotian Song et al. “Physics-Informed Neural Networks with Generalized Residual-Based Adaptive Sampling”. In: *ICIC* (2). 2024, pp. 320–332. URL: https://doi.org/10.1007/978-981-97-5581-3_26.
- [35] Wei Peng et al. *RANG: A Residual-based Adaptive Node Generation Method for Physics-Informed Neural Networks*. 2022. arXiv: [2205.01051](https://arxiv.org/abs/2205.01051) [cs.LG]. URL: <https://arxiv.org/abs/2205.01051>.
- [36] Hyuk Lee and In Seok Kang. “Neural algorithm for solving differential equations”. In: *Journal of Computational Physics* 91.1 (1990), pp. 110–131. ISSN: 0021-9991. DOI: [https://doi.org/10.1016/0021-9991\(90\)90007-N](https://doi.org/10.1016/0021-9991(90)90007-N). URL: <https://www.sciencedirect.com/science/article/pii/002199919090007N>.
- [37] Dimitris C. Psychogios and Lyle H. Ungar. “A hybrid neural network-first principles approach to process modeling”. In: *AIChE Journal* 38.10 (1992), pp. 1499–1511. DOI: <https://doi.org/10.1002/aic.690381003>. eprint: <https://aiche.onlinelibrary.wiley.com/doi/pdf/10.1002/aic.690381003>. URL: <https://aiche.onlinelibrary.wiley.com/doi/abs/10.1002/aic.690381003>.
- [38] M. W. M. G. Dissanayake and N. Phan-Thien. “Neural-network-based approximations for solving partial differential equations”. In: *Communications in Numerical Methods in Engineering* 10.3 (1994), pp. 195–201. DOI: <https://doi.org/10.1002/cnm.1640100303>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/cnm.1640100303>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/cnm.1640100303>.
- [39] Atılım Güneş Baydin et al. “Automatic Differentiation in Machine Learning: a Survey”. In: *Journal of Machine Learning Research* 18 (2018), pp. 1–43. DOI: [10.5555/3122009.3242010](https://doi.org/10.5555/3122009.3242010). URL: <https://www.jmlr.org/papers/volume18/17-468/17-468.pdf>.
- [40] Salvatore Cuomo et al. “Scientific Machine Learning Through Physics-Informed Neural Networks: Where we are and What’s Next”. In: *Journal of Scientific Computing* 92.3 (July 26, 2022), p. 88. ISSN: 1573-7691. DOI: [10.1007/s10915-022-01939-z](https://doi.org/10.1007/s10915-022-01939-z). URL: <https://doi.org/10.1007/s10915-022-01939-z>.
- [41] Ameya D. Jagtap, Kenji Kawaguchi, and George Em Karniadakis. “Locally adaptive activation functions with slope recovery for deep and physics-informed neural networks”. In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 476.2239 (July 2020). Epub 2020 Jul 15, p. 20200334. DOI: [10.1098/rspa.2020.0334](https://doi.org/10.1098/rspa.2020.0334).

- [42] Amirhossein Khademi and Steven Dufour. “Physics-informed neural networks with trainable sinusoidal activation functions for approximating the solutions of the Navier-Stokes equations”. In: *Computer Physics Communications* 314 (2025), p. 109672. ISSN: 0010-4655. DOI: <https://doi.org/10.1016/j.cpc.2025.109672>. URL: <https://www.sciencedirect.com/science/article/pii/S0010465525001742>.
- [43] Chuwei Wang et al. *Is L^2 Physics-Informed Loss Always Suitable for Training Physics-Informed Neural Network?* 2022. arXiv: 2206.02016 [cs.LG]. URL: <https://arxiv.org/abs/2206.02016>.
- [44] Pauli Virtanen et al. “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python”. In: *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.
- [45] Sifan Wang, Shyam Sankaran, and Paris Perdikaris. “Respecting causality for training physics-informed neural networks”. In: *Computer Methods in Applied Mechanics and Engineering* 421 (2024), p. 116813. ISSN: 0045-7825. DOI: <https://doi.org/10.1016/j.cma.2024.116813>. URL: <https://www.sciencedirect.com/science/article/pii/S0045782524000690>.
- [46] Nasim Rahaman et al. “On the Spectral Bias of Neural Networks”. In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, Sept. 2019, pp. 5301–5310. URL: <https://proceedings.mlr.press/v97/rahaman19a.html>.
- [47] Mayank Deshpande et al. *Investigations on convergence behaviour of Physics Informed Neural Networks across spectral ranges and derivative orders*. 2023. arXiv: 2301.02790 [cs.LG]. URL: <https://arxiv.org/abs/2301.02790>.
- [48] Xiong Xiong et al. *Separated-Variable Spectral Neural Networks: A Physics-Informed Learning Approach for High-Frequency PDEs*. 2025. arXiv: 2508.00628 [cs.LG]. URL: <https://arxiv.org/abs/2508.00628>.
- [49] Yixin Huang et al. “Chebyshev spectral approximation-based physics-informed neural network for solving higher-order non-linear differential equations”. In: *Engineering with Computers* 41.2 (Apr. 2025), pp. 1191–1210. ISSN: 1435-5663. DOI: 10.1007/s00366-024-02073-0. URL: <https://doi.org/10.1007/s00366-024-02073-0>.
- [50] Shaghayegh Fazliani, Zachary Frangella, and Madeleine Udell. *Enhancing Physics-Informed Neural Networks Through Feature Engineering*. 2025. arXiv: 2502.07209 [cs.LG]. URL: <https://arxiv.org/abs/2502.07209>.
- [51] Paweł Maczuga and Maciej Paszyński. “Influence of Activation Functions on the Convergence of Physics-Informed Neural Networks for 1D Wave Equation”. In: *Computational Science – ICCS 2023*. Ed. by Jiří Mikýška et al. Cham: Springer Nature Switzerland, 2023, pp. 74–88. ISBN: 978-3-031-35995-8.
- [52] Jassem Abbasi and Pål Østebø Andersen. “Physical activation functions (PAFs): An approach for more efficient induction of physics into physics-informed neural networks (PINNs)”. In: *Neurocomputing* 608 (Dec. 2024), p. 128352. ISSN: 0925-2312. DOI: 10.1016/j.neucom.2024.128352. URL: <http://dx.doi.org/10.1016/j.neucom.2024.128352>.
- [53] Honghui Wang Honghui Wang et al. “Learning Specialized Activation Functions for Physics-Informed Neural Networks”. In: *Communications in Computational Physics* 34.4 (Jan. 2023), pp. 869–906. ISSN: 1815-2406. DOI: 10.4208/cicp.oa-2023-0058. URL: <http://dx.doi.org/10.4208/cicp.OA-2023-0058>.
- [54] Afrah Farea and Mustafa Serdar Celebi. “Learnable activation functions in physics-informed neural networks for solving partial differential equations”. In: *Computer Physics Communications* 315 (2025), p. 109753. ISSN: 0010-4655. DOI: <https://doi.org/10.1016/j.cpc.2025.109753>. URL: <https://www.sciencedirect.com/science/article/pii/S0010465525002553>.
- [55] Kaiming He et al. “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification”. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 1026–1034. DOI: 10.1109/ICCV.2015.123.
- [56] Nikhil Vyas et al. *SOAP: Improving and Stabilizing Shampoo using Adam*. 2025. arXiv: 2409.11321 [cs.LG]. URL: <https://arxiv.org/abs/2409.11321>.
- [57] Sifan Wang et al. *Gradient Alignment in Physics-informed Neural Networks: A Second-Order Optimization Perspective*. 2025. arXiv: 2502.00604 [cs.LG]. URL: <https://arxiv.org/abs/2502.00604>.
- [58] Chenhui Xu et al. *FP64 is All You Need: Rethinking Failure Modes in Physics-Informed Neural Networks*. 2025. arXiv: 2505.10949 [cs.LG]. URL: <https://arxiv.org/abs/2505.10949>.