MITIGATING PROPAGATION FAILURES IN PINNS USING EVOLUTIONARY SAMPLING

Anonymous authors
Paper under double-blind review

ABSTRACT

Despite the success of physics-informed neural networks (PINNs) in approximating partial differential equations (PDEs), it is known that PINNs can sometimes fail to converge to the correct solution in problems involving complicated PDEs. This is reflected in several recent studies on characterizing and mitigating the "failure modes" of PINNs. While most of these studies have focused on balancing loss functions or adaptively tuning PDE coefficients, what is missing is a thorough understanding of the connection between failure modes of PINNs and sampling strategies used for training PINNs. In this paper, we provide a novel perspective of failure modes of PINNs by hypothesizing that the training of PINNs rely on successful "propagation" of solution from initial and/or boundary condition points to interior points. We show that PINNs with poor sampling strategies can get stuck at trivial solutions if there are *propagation failures*. We additionally demonstrate that propagation failures are characterized by highly imbalanced PDE residual fields where very high residuals are observed over very narrow regions. To mitigate propagation failures, we propose a novel *evolutionary sampling* (Evo) method that can incrementally accumulate collocation points in regions of high PDE residuals with little to no computational overhead. We provide an extension of Evo to respect the principle of causality while solving time-dependent PDEs. We theoretically analyze the behavior of Evo and empirically demonstrate its efficacy and efficiency in comparison with baselines on a variety of PDE problems.

1 INTRODUCTION

Physics-informed neural networks (PINNs) (Raissi et al., 2019) represent a seminal line of work in deep learning for solving partial differential equations (PDEs), which appear naturally in a number of domains. The basic idea of PINNs for solving a PDE is to train a neural network to minimize errors w.r.t. the solution provided at initial/boundary points of a spatio-temporal domain, as well as the PDE residuals observed over a sample of interior points, referred to as collocation points. Despite the success of PINNs, it is known that PINNs can sometimes fail to converge to the correct solution in problems involving complicated PDEs, as reflected in several recent studies on characterizing the "failure modes" of PINNs (Wang et al., 2020; 2022c; Krishnapriyan et al., 2021). Many of these failure modes are related to the susceptibility of PINNs in getting stuck at trivial solutions acting as poor local minima, due to the unique optimization challenges of PINNs. In particular, note that training PINNs is different from conventional deep learning problems as we only have access to the correct solution on the initial and/or boundary points, while for all interior points in the domain, we can only compute PDE residuals. Also note that minimizing PDE residuals does not guarantee convergence to a correct solution since there are many trivial solutions of commonly observed PDEs that show 0 residuals. While previous studies on understanding and preventing failure modes of PINNs have mainly focused on modifying network architectures or balancing loss functions during PINN training, the effect of sampling collocation points on avoiding failure modes of PINNs has been largely overlooked. Although some previous approaches have explored the effect of sampling strategies on PINN training Wang et al. (2022a); Lu et al. (2021), they either suffer from large computation costs or fail to converge to correct solutions, empirically demonstrated in our results.

In this work, we present a novel perspective of failure modes of PINNs by postulating the propagation hypothesis: "in order for PINNs to avoid converging to trivial solutions at interior points, the correct solution must be *propagated* from the initial/boundary points to the interior points." When this propagation is hindered, PINNs can get stuck at trivial solutions that are difficult to escape, referred to as the *propagation failure* mode. This hypothesis is motivated from a similar behavior observed in

numerical methods where the solution of the PDE at initial/boundary points are iteratively propagated to interior points using finite differencing schemes (LeVeque, 2007).

We show that propagation failures in PINNs are characterized by highly imbalanced PDE residual fields, where very high residuals are observed in narrow regions of the domain. Such high residual regions are not adequately sampled in the set of collocation points (which generally is kept fixed across all training iterations), making it difficult to overcome the propagation failure mode. This motivates us to develop sampling strategies that focus on selecting more collocation points from high residual regions. This is related to the idea of local-adaptive mesh refinement used in FEM (Zienkiewicz et al., 2005) to selectively refine the computational mesh in regions with higher errors.

We propose a novel *evolutionary sampling* (Evo) strategy that can accumulate collocation points in high PDE residual regions, thereby dynamically emphasizing on these skewed regions as we progress in training iterations. We also provide a causal extension of our proposed Evo algorithm (Causal Evo) that can explicitly encode the strong inductive bias of causality in propagating the solution from initial points to interior points over training iterations, when solving time-dependent PDEs. We theoretically prove the adaptive quality of Evo to accumulate points from high residual regions that persist over iterations. We empirically demonstrate the efficacy and efficiency of our proposed sampling methods in a variety of benchmark PDE problems previously studied in the PINN literature. We show the Evo and Causal Evo are able to mitigate propagation failure modes and converge to the correct solution with significantly smaller sample sizes as compared to baseline methods, while incurring negligible computational overhead. We also demonstrate the ability of Evo to solve a particularly hard PDE problem—solving 2D Eikonal equations for complex arbitrary surface geometries.

2 Background and Related Work

Physics-Informed Neural Networks (PINNs). The basic formulation of PINN (Raissi et al., 2017) is to use a neural network $f_\theta(x,t)$ to infer the forward solution u of a non-linear PDE:

$$u_t + \mathcal{N}_x[u] = 0,\ x \in \mathcal{X}, t \in [0,T];\ \ u(x,0) = h(x),\ x \in \mathcal{X};\ \ u(x,t) = g(x,t),\ t \in [0,T], x \in \partial\mathcal{X}$$

where x and t are the space and time coordinates, respectively, $\mathcal{X}$ is the spatial domain, $\partial\mathcal{X}$ is the boundary of spatial domain, and T is the time horizon. The PDE is enforced on the entire spatio-temporal domain ($\Omega = \mathcal{X} \times [0,T]$) on a set of collocation points $\{\mathbf{x_r}^i = (x_r^i, t_r^i)\}_{i=1}^{N_r}$ by computing the PDE residual ($\mathcal{R}(x,t)$) and the corresponding PDE Loss ($\mathcal{L}_r$) as follows:

$$\mathcal{R}_\theta(x,t) = \frac{\partial}{\partial t} f_\theta(x,t) - \mathcal{N}_x[f_\theta(x,t)] \tag{1}$$

$$\mathcal{L}_r(\theta) = \mathbb{E}_{\mathbf{x_r} \sim \mathcal{U}(\Omega)}[\mathcal{R}_\theta(\mathbf{x_r})^2] \approx \frac{1}{N_r}\sum_{i=1}^{N_r}[\mathcal{R}_\theta(x_r^i, t_r^i)]^2 \tag{2}$$

where $\mathcal{L}_r$ is the expectation of the squared PDE Residuals over collocation points sampled from a uniform distribution $\mathcal{U}$. PINNs approximate the solution of the PDE by optimizing the following overall loss function $\mathcal{L} = \lambda_r\mathcal{L}_r(\theta) + \lambda_{bc}\mathcal{L}_{bc}(\theta) + \lambda_{ic}\mathcal{L}_{ic}(\theta)$, where $\mathcal{L}_{ic}$ and $\mathcal{L}_{bc}$ are the mean squared loss on the initial and boundary data respectively, and $\lambda_r, \lambda_{ic}, \lambda_{bc}$ are hyperparameters that control the interplay between the different loss terms. Although PINNs can be applied to inverse problems, i.e., to estimate PDE parameters from observations, we only focus on forward problems in this paper.

Prior Work on Characterizing Failure Modes of PINNs. Despite the popularity of PINNs in approximating PDEs, several works have emphasized the presence of failure modes while training PINNs. One early work (Wang et al., 2020) demonstrated that imbalance in the gradients of multiple loss terms could lead to poor convergence of PINNs, motivating the development of Adaptive PINNs. Another recent development (Wang et al., 2022c) made use of the Neural Tangent Kernel (NTK) theory to indicate that the different convergence rates of the loss terms can lead to training instabilities. Large values of PDE coefficients have also been connected to possible failure modes in PINNs (Krishnapriyan et al., 2021). In all these works, the effect of sampling collocation points on PINN failure modes has largely been overlooked. Although some recent works have explored strategies to grow the representation of collocation points with high residuals, either by modifying the sampling procedure Wu et al. (2022); Lu et al. (2021); Nabian et al. (2021) or choosing higher-order L^p norms of PDE loss Wang et al. (2022a), they either require a prohibitively dense set of collocation points and hence are computationally expensive, or suffer from poor convergence to the correct solution

as empirically demonstrated in Section 5. In another recent line of work on Causal PINNs (Wang et al., 2022b), it was shown that traditional approaches for training PINNs can violate the principle of causality for time-dependent PDEs. Hence, they proposed an explicit way of incorporating the causal structure in the training procedure. However, this solution can only be applied to time-dependent PDEs, and as we demonstrate empirically in Section 5, also requires large sample sizes.

3 Propagation Hypothesis: A New Perspective of Failure Modes in PINNs

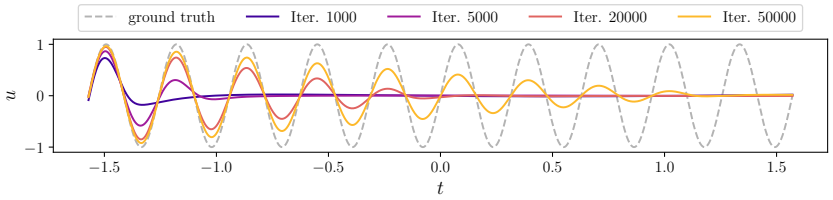


Figure 1: PINN solutions for a simple ODE: $u_{xx} + k^2 u = 0$ ($k = 20$) with the analytical solution, $u = A\sin(kx) + B\cos(kx)$. We can see smooth *propagation* of the correct solution from the boundary point at $x = 0$ to interior points ($x > 0$) as we increase training iterations.

What is Unique About Training PINNs? Training PINNs presents fundamentally different optimization challenges than those encountered in conventional deep learning problems. In a conventional supervised learning problem, the correct solution for every training sample is known and the training samples are considered representative of the test samples such that the trained model can easily be extrapolated on closely situated test samples. However, in the context of PINNs, we only have access to the "correct" solution of the PDE on the initial and/or boundary points, while not having any labels for the interior points in the spatio-temporal domain Ω. Note that the interior points in Ω can be quite far away from the initial/boundary points, making extrapolation difficult. Further, training PINNs involves minimizing the PDE residuals over a set of collocation points sampled from Ω. However, minimizing PDE residuals alone is not sufficient to ensure convergence to the correct solution, since there may exist many trivial solutions of a PDE showing very small residuals. For example, $u(x, t) = 0$ is a trivial solution for any homogeneous PDE, which a neural network is likely to get stuck at in the absence of correct solution at initial/boundary points. Another unique aspect of training PINNs is that minimizing PDE residuals requires computing the gradients of the output w.r.t. (x, t) (e.g., u_x and u_t). Hence, the solution at a collocation point is affected by the solutions at nearby points leading to local propagation of solutions.

Propagation Hypothesis. In light of the unique properties of PINNs, we postulate that in order for PINNs to converge to the "correct" solution, the correct solution must propagate from the initial and/or boundary points to the interior points as we progress in training iterations. We draw inspiration for this hypothesis from a similar behavior observed in numerical methods for solving PDEs, where the solution of the PDE at initial/boundary points are iteratively propagated to interior points using finite differencing schemes (LeVeque, 2007). Figure 1 demonstrates the propagation hypothesis of PINNs for a simple ordinary differential equation (ODE).

Propagation *Failure*: Why It Happens and How to Diagnose? As a corollary of the propagation hypothesis, PINNs can fail to converge to the correct solution if the solution at initial/boundary points is *unable to propagate* to interior points during the training process. We call this phenomenon the "*propagation failure*" mode of PINNs. This is likely to happen if some collocation points start converging to trivial solutions before the correct solution from initial/boundary points is able to reach them. Such collocation points would also propagate their trivial solutions to nearby interior points, leading to a cascading effect in the learning of trivial solutions over large regions of Ω and further hindering the propagation of the correct solution from initial/boundary points.

To *diagnose* propagation failures, note that the PDE residuals are expected to be low over both types of regions: regions that have converged to the correct solution and regions that have converged to trivial solutions. However, the boundaries of these two types of regions would show a sharp discontinuity in solutions, leading to very high PDE residuals in very narrow regions. A similar phenomenon is observed in numerical methods where sharp high-error regions disrupt the evolution of the PDE solution at surrounding regions, leading to cascading of errors. We use the imbalance of high PDE residual regions as a diagnosis tool for characterizing propagation failure modes in PINNs.

To demonstrate propagation failure, let us consider an example PDE for the convection equation: $\frac{\partial u}{\partial t} + \beta \frac{\partial u}{\partial x} = 0, u(x, 0) = h(x)$, where β is the convection coefficient and $h(x)$ is the initial condition (see Appendix **??** for details about this PDE). In a previous work (Krishnapriyan et al., 2021), it has been shown that PINNs fail to converge for this PDE for $\beta > 10$. We experiment with two cases, $\beta = 10$ and $\beta = 50$, in Figure 2. We can see that the PDE loss steadily decreases with training iterations for both these cases, but the relative error w.r.t. the ground-truth solution only decreases for $\beta = 10$, while for $\beta = 50$, it remains flat. This suggests that for $\beta = 50$, PINN is likely getting stuck at a trivial solution that shows low PDE residuals but high errors. To diagnose this failure mode, we plot two additional metrics in Figure 2 to measure the imbalance in high PDE residual regions: Fisher-Pearson's coefficient of Skewness (Kokoska & Zwillinger, 2000) and Fisher's Kurtosis (Kokoska & Zwillinger, 2000) (see Appendix **??** for computation details). High Skewness indicates lack of symmetry in the distribution of PDE residuals while high Kurtosis indicates the presence of a heavy-tail. For $\beta = 10$, we can see that both Skewness and Kurtosis are relatively small across all iterations, indicating absence of imbalance in the residual field. However, for $\beta = 50$, both these metrics shoot up significantly as the training progresses, which indicates the formation of very high residuals in very narrow regions—a characteristic feature of the propagation failure mode. Figure 3 confirms that this indeed is the case by visualizing the PINN solution and PDE residual maps. We see similar trends of propagation failure for other values of $\beta > 10$ (see Appendix **??**).

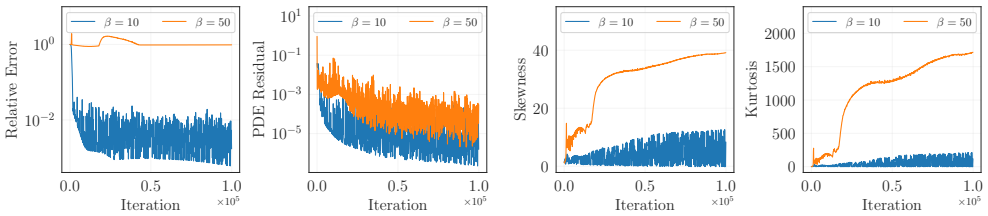


Figure 2: Demonstration of propagation failure while solving the convection equation with $\beta = 50$.

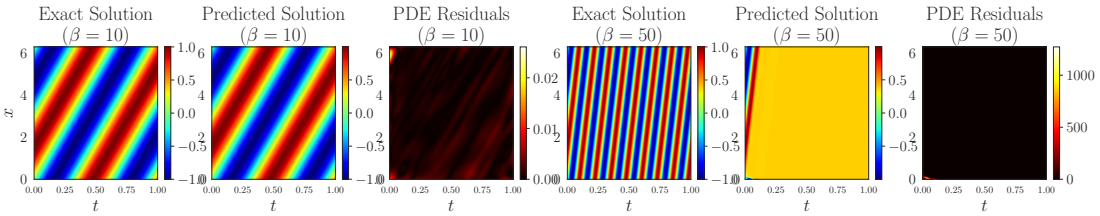


Figure 3: Demonstration of the Exact PDE solution, the predicted PDE solution, and the PDE Residual Field for the convection equation with $\beta = 10$ and $\beta = 50$ respectively.

Potential Remedies for Mitigating Propagation Failures: Since *propagation failures* in PINNs are characterized by skewed regions with very high residuals acting as "barriers" in the flow of information, algorithms that can selectively focus on high residual regions during training can potentially break such barriers and thus offer some respite. Recently, a number of algorithms have been proposed to increase the importance of collocation points from high residual regions in PINN training, which can be broadly grouped into two categories. The first category includes methods that alter the sampling strategy such that a point $\mathbf{x_r}$ is picked up as a collocation point with probability proportional to its residual, i.e., $p(\mathbf{x_r}) \propto |\mathcal{R}(\mathbf{x_r})|^k$, where $k \geq 1$. This includes residual-based adaptive refinement (RAR) methods (Lu et al., 2021) and its variants Wu et al. (2022); Nabian et al. (2021), where a dense set of collocation points $\mathcal{P}_{dense}$ is maintained to approximate the continuous residual field $\mathcal{R}(\mathbf{x})$, and points with high residuals are regularly added from $\mathcal{P}_{dense}$ to the set of collocation points every K iterations according to a sampling function. A second line of work was recently proposed in Wang et al. (2022a), where higher-order L^p norms of the PDE loss (e.g., L^∞) were advocated to be used in the training of PINNs in contrast to the standard practice of using L^2 norms, to ensure stability of learned solutions in control problems involving high-dimensional PDEs. Note that by using higher-order L^p norms, we are effectively increasing the importance of collocation points from high residual regions in the PDE loss, thereby having a similar effect as increasing their representation in the sample of collocation points. Theorem 3.1 below shows that sampling collocation points with $p(\mathbf{x_r}) \propto |\mathcal{R}(\mathbf{x_r})|^k$ is exactly equivalent to modifying the L^p norm of the PDE loss, thus providing a bridge between these two categories of existing methods.

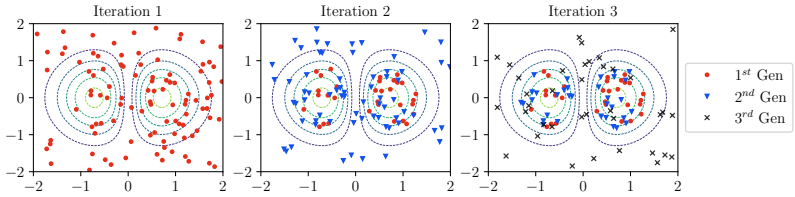


Figure 4: Schematic to describe our proposed evolutionary sampling algorithm, where collocation points are incrementally accumulated in regions with high PDE residuals (shown as contour lines).

Theorem 3.1. *For $p \geq 2$, let $\mathcal{L}_r^p(\mathcal{U})$ denote the expected L^p PDE Loss computed on collocation points sampled from a uniform distribution, $\mathcal{U}(\Omega)$. Similarly, for $k \geq 0$, let $\mathcal{L}_r^2(\mathcal{Q}^k)$ denote the expected L^2 PDE Loss computed on collocation points sampled from an alternate distribution $\mathcal{Q}^k(\Omega) : \mathbf{x_r} \sim q(\mathbf{x_r})$, where $q(\mathbf{x_r}) \propto |\mathcal{R}_\theta(\mathbf{x_r})|^k$. Then, $\mathcal{L}_r^2(\mathcal{Q}^k) = \frac{1}{Z^{1/2}V^{1/2}}\left(\mathcal{L}_r^{k+2}(\mathcal{U})\right)^{(k+2)/2}$, where Z is a normalization constant and V is the volume of the domain Ω.*

The Proof of Theorem 3.1 can be found in Appendix **??**.

Challenges with Potential Remedies. There are two main challenges faced by the potential remedies described above that limit their effectiveness in mitigating propagation failures in PINNs. (1) **High computational complexity:** Sampling methods such as RAR and its variants require maintaining a dense set of collocation points $\mathcal{P}_{dense}$ (typically with 100K $\sim$ 1M points) and computing PDE residuals at every point in $\mathcal{P}_{dense}$ every K iterations. This introduces a severe computational overhead for adding new points, which also necessitates the resampling period K to be high (See Appendix **??** for a detailed analysis of the computational complexity of RAR based methods). (2) **Poor Prediction Performance:** As we empirically demonstrate in Section 5, both sampling-based methods such as RAR and its variants as well as L^∞ PDE loss-based methods suffer from poor performance in converging to the correct solution for complex PDE problems. This can be attributed to several reasons. First, while increasing the value of k in sampling-based methods and p in L^p norm-based methods influences a greater skew in the sample set towards points with higher residuals, the optimal values of k or p are generally unknown for an arbitrary PDE. Second, choosing L^∞ loss (or equivalently, only sampling collocation points with highest residuals) may not ideal for the training dynamics of PINNs, as it can lead to oscillatory behavior between different peaks of the PDE residual landscape, while forgetting to retain the solution over other regions. Instead, we need a way to gradually accumulate points from high residual regions in the collocation set while having non-zero representation of points from other regions.

4 Proposed Approach: Evolutionary Sampling (Evo)

Before we present our proposed sampling approach, in the following we first describe the *four salient properties* that we expect to see in an "ideal" sampling algorithm to mitigate propagation failures.

• ***Gradual Accumulation Property***: To break propagation barriers, the set of collocation points should evolve over every iteration such that they start from a uniform distribution and *gradually* accumulate in regions with high residuals until the PINN training process eventually resolves them in later iterations. This is analogous to starting with an L^2 loss (equivalent to uniform distribution) and gradually increasing the order of L^p norm if high residual regions persist over iterations.

• ***Sample Release Property***: Upon sufficient minimization of a high residual region through PINN training, collocation points that were once accumulated from the region needs to be *released*, such that we can focus on minimizing other high residual regions in later iterations.

• ***Uniform Background Property***: At every iteration, the set of collocation points should contain non-zero support of points sampled from a uniform distribution over the entire domain Ω, such that the collocation points do not collapse solely to high residual regions.

• ***Marginal Computational Overhead***: While satisfying the above properties, we should incur little to no computational overhead in sampling collocation points from high residual regions. Specifically, we should be able to add points from high residual regions without maintaining a dense set of collocation points, $\mathcal{P}_{dense}$, and by only observing the residuals over a small set of N_r points at every iteration.

To satisfy all of the above properties, we present a novel sampling strategy termed *Evolutionary Sampling (Evo)* that is inspired by algorithms used for modeling biological evolution (Eiben et al.,

Algorithm 1 Proposed Evolutionary Sampling Algorithm For PINN

1: Sample the initial population $\mathcal{P}_0$ of N_r collocations point $\mathcal{P}_0 \leftarrow \{\mathbf{x_r}\}_{i=1}^{N_r}$ from a uniform distribution ${\mathbf{x_r}}^i \sim \mathcal{U}(\Omega)$, where Ω is the input domain ($\Omega = [0, T] \times \mathcal{X}$).
2: **for** i = 0 to max_iterations - 1 **do**
3: Compute the fitness of collocation points $\mathbf{x_r} \in \mathcal{P}_i$ as $\mathcal{F}(\mathbf{x_r}) = |\mathcal{R}(\mathbf{x_r})|$.
4: Compute the threshold $\tau_i = \frac{1}{N_r} \sum_{j=1}^{N_r} \mathcal{F}({\mathbf{x_r}}^j)$
5: Select the retained population $\mathcal{P}_i^r$ such that $\mathcal{P}_i^r \leftarrow \{{\mathbf{x_r}}^j : \mathcal{F}({\mathbf{x_r}}^j) > \tau_{it}\}$
6: Generate the re-sampled population $\mathcal{P}_i^s \leftarrow \{{\mathbf{x_r}}^j : {\mathbf{x_r}}^j \sim \mathcal{U}(\Omega)\}$
7: Merge the two populations $\mathcal{P}_{i+1} \leftarrow \mathcal{P}_i^r \cup \mathcal{P}_i^s$
8: **end for**

2003). The key idea behind Evo is to gradually *evolve* the population of collocation points at every iteration by retaining points with high residuals from the current population and re-sampling new points from a uniform distribution. Figure 4 schematically shows the accumulation of collocation points from high residual regions in Evo over training iterations.

Algorithm 1 shows the pseudo-code of our proposed evolutionary sampling strategy. Analogous to evolutionary algorithms developed in optimization literature (Simon, 2013), we introduce a notion of "fitness" $\mathcal{F}(\mathbf{x_r})$ for every collocation point $\mathbf{x_r}$ such that points with higher fitness are allowed to survive in the next iteration. Specifically, we define $\mathcal{F}(\mathbf{x_r})$ as the absolute value of PDE residual of $\mathbf{x_r}$, i.e., $\mathcal{F}(\mathbf{x_r}) = |\mathcal{R}(\mathbf{x_r})|$. At iteration 0, we start with an initial population $\mathcal{P}_0$ of N_r points sampled from a uniform distribution. At iteration i, in order to evolve the population to the next iteration, we first construct the "retained population" $\mathcal{P}_i^r$ comprising of points from $\mathcal{P}_i$ with fitness values greater than τ_i, i.e., $\mathcal{P}_i^r \leftarrow \{\mathbf{x}_r^j : \mathcal{F}(\mathbf{x}_r^j) > \tau_i\}$, where τ_i is equal to the expectation of fitness values over all points in $\mathcal{P}_i$. The remainder of collocation points in $\mathcal{P}_i$ are re-sampled from a uniform distribution, thus constructing the "re-sampled population" $\mathcal{P}_i^s \leftarrow \{{\mathbf{x_r}}^j : {\mathbf{x_r}}^j \sim \mathcal{U}(\Omega)\}$. The retained population and the re-sampled population are then merged to generate the population for the next iteration, $\mathcal{P}_{i+1}$.

Analysis of Evo: Theorem 4.1 shows that for a fixed objective function, the expectation of the retained population in Evo becomes maximum (equal to L^∞) when the number of iterations approaches ∞.

Theorem 4.1 (Accumulation Dynamics Theorem)**.** *Let $f_\theta(x) : \mathbb{R}^n \to \mathbb{R}^+$ be an arbitrary positive real-valued k-Lipschitz continuous objective function optimized using the Evolutionary Sampling algorithm with fixed θ. Then, the expectation of the retain population $\mathbb{E}_{x \in \mathcal{P}^r}[f(x)] \geq \max_x f(x) - k\epsilon$ as iteration $i \to \infty$, for any arbitrarily small $\epsilon > 0$.*

The proof of Theorem 4.1 can be found in Appendix **??**. This demonstrates the *gradual accumulation property* of Evo as points from a high residual region would keep accumulating in the retained population and make its expectation maximal if the region persists over iterations. Such points would eventually be released from $\mathcal{P}^r$ when their residual values are sufficiently minimized through PINN training. Also note that Evo always maintains some collocation points from a uniform distribution i.e., the re-sampled population $\mathcal{P}^s$ is always non-empty. Theoretical proofs of the *sample release property* and the *uniform background property* of Evo are provided in Appendix **??**. We also provide details of the computational complexity of Evo in comparison with baseline methods in Appendix **??**, showing that Evo incurs *marginal computational overhead*. Note that Evo shares a similar motivation as *local-adaptive mesh refinement* methods developed for Finite Element Methods (FEM) (Zienkiewicz et al., 2005), where the goal is to preferentially refine the computational mesh used in numerical methods based on localization of the errors. It is also related to the idea of *boosting* in ensemble learning where training samples with larger errors are assigned higher weights of being picked in the next epoch, to increasingly focus on high error regions (Schapire, 2003).

4.1 Causal Extension of Evolutionary Sampling (Causal Evo)

In problems with time-dependent PDEs, a strong prior dictating the propagation of solution is the *principle of causality*, where the solution of the PDE needs to be well-approximated at time t before moving to time $t + \Delta t$. To incorporate this prior guidance, we present a Causal Extension of Evo (*Causal Evo*) that includes two modifications: (1) we develop a causal formulation of the PDE loss $\mathcal{L}_r$ that pays attention to the temporal evolution of PDE solutions over iterations, and (2) we develop a causally biased sampling scheme that respects the causal structure while sampling collocation points. We describe both these modifications in the following.

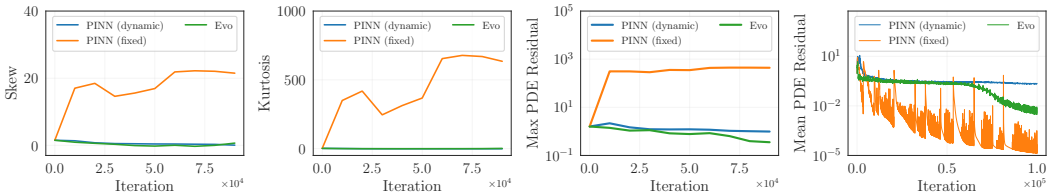


Figure 5: Comparison of Skewness, Kurtosis, Max PDE Residuals, and Mean PDE Residuals over training iterations for PINN-fixed, PINN-dynamic, and Evo for convection equation with $\beta = 50$.

Causal Formulation of PDE Loss. The key idea here is to utilize a simple time-dependent gate function $g(t)$ that can explicitly enforce causality by revealing only a portion of the entire time-domain to PINN training. Specifically, we introduce a continuous gate function $g(t) = (1-\tanh(\alpha(\tilde{t}-\gamma)))/2$, where γ is the scalar shift parameter that controls the fraction of time that is revealed to the model, $\alpha = 5$ is a constant scalar parameter that determines the steepness of the gate, and $\tilde{t}$ is the normalized time, i.e., $\tilde{t} = t/T$. Example causal gates for different settings of γ are provided in Appendix **??**. We use $g(t)$ to obtain a causally-weighted PDE residual loss as $\mathcal{L}_r^g(\theta) = \frac{1}{N_r}\sum_{i=1}^{N_r}[\mathcal{R}(x_r^i, t_r^i)]^2 * g(t_r^i)$. We initially start with a small value of the shift parameter ($\gamma = -0.5$), which essentially only reveals a very small portion of the time domain, and then gradually increase γ during training to reveal more portions of the time domain. For $\gamma \geq 1.5$, the entire time domain is revealed.

Causally Biased Sampling. We bias the sampling strategy in Evo such that it not only favors the selection of collocation points from high residual regions but also accounts for the causal gate values at every iteration. In particular, we modify the fitness function as $\mathcal{F}(\mathbf{x_r}) = |\mathcal{R}(\mathbf{x_r})| * g(t_r)$. A schematic illustration of causally biased sampling is provided in Appendix **??**.

How to Update γ? Ideally, at some iteration i, we would like increase γ_i at the next iteration *only if* the PDE residuals at iteration i are low. Otherwise, γ_i should remain in its place until the PDE residuals under the current gate are minimized. To achieve this behaviour, we propose the following update scheme for γ: $\gamma_{i+1} = \gamma_i + \eta_g e^{-\epsilon\mathcal{L}_r^g(\theta)}$, where η_g is the learning rate and ϵ denotes tolerance that controls how low the PDE loss needs to be before the gate shifts to the right. Since the update in γ is inversely proportional to the causally-weighted PDE loss $\mathcal{L}_r^g$, the gate will shift slowly if the PDE residuals are large. Also note that increasing γ also increases the value of $g(t)$ for all collocation points, thus increasing the causally-weighted PDE loss and slowing down gate movement. Upon convergence, γ attains a large value such that the entire time domain is revealed.

Table 1: Relative $\mathcal{L}_2$ errors (in %) of comparative methods over benchmark PDEs with $N_r = 1000$.

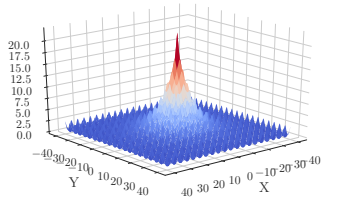
	Convection ($\beta = 30$)		Convection ($\beta = 50$)		Allen Cahn
Epochs.	100k	300k	150k	300k	200k
PINN (fixed)	$107.5 \pm 10.9\%$	$107.5 \pm 10.7\%$	$108.5 \pm 6.38\%$	$108.7 \pm 6.59\%$	$69.4 \pm 4.02\%$
PINN (dynamic)	$2.81 \pm 1.45\%$	$1.35 \pm 0.59\%$	$24.2 \pm 23.2\%$	$56.9 \pm 9.08\%$	$0.77 \pm 0.06\%$
Curr Reg	$63.2 \pm 9.89\%$	$2.65 \pm 1.44\%$	$48.9 \pm 7.44\%$	$31.5 \pm 16.6\%$	-
CPINN (fixed)	$138.8 \pm 11.0\%$	$138.8 \pm 11.0\%$	$106.5 \pm 10.5\%$	$106.5 \pm 10.5\%$	$48.7 \pm 19.6\%$
CPINN (dynamic)	$52.2 \pm 43.6\%$	$23.8 \pm 45.1\%$	$79.0 \pm 5.11\%$	$73.2 \pm 8.36\%$	$1.5 \pm 0.75\%$
RAR-G	$10.5 \pm 5.67\%$	$2.66 \pm 1.41\%$	$65.7 \pm 17.0\%$	$43.1 \pm 28.9\%$	$25.1 \pm 23.2\%$
RAD	$3.35 \pm 2.02\%$	$1.85 \pm 1.90\%$	$66.0 \pm 1.55\%$	$64.1 \pm 1.98\%$	$0.78 \pm 0.05\%$
RAR-D	$67.1 \pm 4.28\%$	$32.0 \pm 25.8\%$	$82.9 \pm 5.99\%$	$75.3 \pm 9.58\%$	$51.6 \pm 0.41\%$
L^∞	$66.6 \pm 2.35\%$	$41.2 \pm 27.9\%$	$76.6 \pm 1.04\%$	$75.8 \pm 1.01\%$	$1.65 \pm 1.36\%$
Evo. (ours)	$\mathbf{1.51 \pm 0.26}\%$	$0.78 \pm 0.18\%$	$6.03 \pm 6.99\%$	$\mathbf{1.98 \pm 0.72}\%$	$0.83 \pm 0.15\%$
Causal Evo. (ours)	$2.12 \pm 0.67\%$	$\mathbf{0.75 \pm 0.12}\%$	$\mathbf{5.99 \pm 5.25}\%$	$2.28 \pm 0.76\%$	$\mathbf{0.71 \pm 0.007}\%$

5 Results

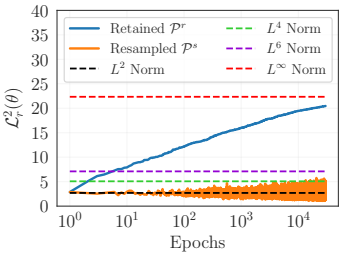
Experiment Setup. We perform experiments over three benchmark PDEs that have been used in existing literature to study failure modes of PINNs. In particular, we consider two time-dependent PDEs: convection equation (with $\beta = 30$ and $\beta = 50$) and Allen Cahn equation, and one time-independent PDE: the Eikonal equation for solving signed distance fields for varying input geometries. We consider the following baseline methods: **(1) PINN-fixed** (conventional PINN using a fixed set of uniformly sampled collocation points), **(2) PINN-dynamic** (a simple baseline where collocation

points are dynamically sampled from a uniform distribution every iteration, see Appendix for more details), **(3) Curr. Reg.** (curriculum regularization method proposed in (Krishnapriyan et al., 2021)), **(4) cPINN-fixed** (causal PINNs proposed in (Wang et al., 2022b) with fixed sampling), **(5) cPINN-dynamic** (cPINN with dynamic sampling), **(6) RAR-G** (Residual-based Adaptive Refinement strategy proposed in (Lu et al., 2021)), **(7) RAD** (Residual-based Adaptive Distribution originally proposed in (Nabian et al., 2021) and later generalized in Wu et al. (2022)), **(8) RAR-D** (RAR with a sampling Distribution proposed in Wu et al. (2022)), **(9)** L^∞ (Sampling top N_r collocation points at every iteration from a dense set $\mathcal{P}_{dense}$ to approximate L^∞ norm). For every benchmark PDE, we use the same neural network architecture and hyper-parameter settings across all baselines and our proposed methods, wherever possible. Details about the PDEs, experiment setups, and hyper-parameter settings are provided in Appendix **??**. All code and datasets used in this paper are available at [1]

Comparing Prediction Performance. Table 1 shows the relative $\mathcal{L}_2$ errors (over 5 random seeds) of PDE solutions obtained by comparative methods w.r.t. ground-truth solutions for different time-dependent PDEs when N_r is set to 1K. We particularly chose a small value of N_r to study the effect of small sample size on PINN performance (note that the original formulations of baseline methods used very high N_r). We can see that while PINN-fixed fails to converge for convection ($\beta = 30$) and Allen Cahn equations (admitting very high errors), PINN-dynamic shows significantly lower errors. This demonstrates that even a very simple baseline of dynamic sampling can be effective in mitigating propagation failures. However, for complex PDEs such as convection ($\beta = 50$), PINN-dynamic is still not able to converge to low errors. We also see that cPINN-fixed shows high errors across all PDEs when $N_r = 1000$. This is likely because the small size of collocation samples are insufficient for cPINNs to converge to the correct solution. As we show later, cPINN indeed is able to converge to the correct solution when N_r is large. Performing dynamic sampling with cPINN shows some reduction in errors, but it is still not sufficient for convection ($\beta = 50$) case. All other baseline methods including Curr Reg, RAR-based methods, and L^∞ fail to converge on most PDEs and show worse performance than even the simple baseline of PINN-dynamic. On the other hand, our proposed approaches (Evo and Causal Evo) consistently show the lowest errors across all PDEs. Figure 5 shows that Evo and PINN-dynamic are indeed able to mitigate *propagation failures* for convection ($\beta = 50$), by maintaining low values of Skewness, Kurtosis, and Max PDE residuals across all iterations, in contrast to PINN-fixed. Additional visualizations of the evolution of collocation samples in Evo, Caual Evo, and RAR-based methods across iterations are provided in Appendix **??**. Sensitivity of RAR-based methods to hyper-parameters is provided in Appendix **??**.



(a) Surface Plot of Auckley Function



(b) Dynamics of PDE loss for Evo

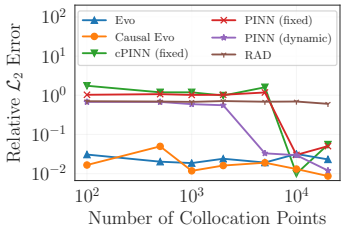
Figure 6: PDE loss of $\mathcal{P}^r$ going from L^2 to L^∞ with iterations for a sample objective function.

Adaptive Nature of Evo PDE loss: To demonstrate the ability of Evo to accumulate high residual points in the retained population $\mathcal{P}^r$ (or equivalently, focus on higher-order L^p norms of PDE loss), we consider optimizing a fixed objective function: the Auckley function in Figure 6 (see Appendix **??** for details of this function). We can see that at iteration 1, the expected loss over $\mathcal{P}^r$ is equal to the L^2 norm of PDE loss over the entire domains. As training progresses, the expected loss over $\mathcal{P}^r$ quickly reaches higher-order L^p norms, and approaches L^∞ at very large iterations. This confirms the gradual accumulation property of Evo as theoretically stated in Theorem 4.1. Addition visualizations of the dynamics of Evo for a number of test optimization functions are provided in Appendix **??**.

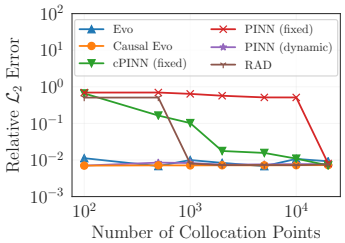
Sampling Efficiency: Figure 7 shows the scalability of Evo to smaller sample sizes of collocation points, N_r. Though all the baselines demonstrate similar performances when N_r is large ($> 10K$), only Evo and Causal Evo manage to maintain low errors even for very small values of $N_r = 100$, showing two orders of magnitude improvement in sampling efficiency. Note that the sample size N_r is directly related to the compute and memory requirements of training PINNs. We also show that Evo and Causal Evo show faster convergence speed than baseline methods for both convection and Allen Cahn equations (see Appendix **??** for details).

[1] https://www.dropbox.com/sh/45gec7qvgiutz2x/AABz4aFJ1IfMIxY11OkLbJ9Oa?dl=0.

Solving Eikonal Equations. Given the equation of a surface geometry in a 2D-space, $u(x_s, y_s) = 0$, the Eikonal equation is a time-independent PDE used to solve for the *signed distance field*(SDF), $u(x, y)$, which has negative values inside the surface and positive values outside the surface. See Appendix **??** for details of the Eikonal equation. We compare the performance of different baseline methods with respect to the ground-truth (GT) solution obtained from numerical methods for three complex surface geometries in Figure 8. The primary difficulty in solving Eikonal equation comes from determining the sign of the field (interior or exterior) in regions with rich details. We also plot the reconstructed geometry of the predicted solutions to demonstrate the real-world application of solving this PDE, e.g., in downstream real-time graphics rendering. The quality of reconstructed geometries are quantitatively evaluated using the mean Intersection-Over-Union (mIOU) metric that measures the degree of overlap between the predicted geometry and GT. The results show that PINN-fixed shows poor performance across all three geometries, while PINN-dynamic is able to capture most of the outline of the solutions with a few details missing for difficult geometries like "sailboat" and "gear". On the other hand, Evo is able to capture even the fine details of the SDF for all three complex geometries and thus show better reconstruction quality. We can see that mIOU of Evo is significantly higher than baselines for "sailboat" and "gear".



(a) Convection ($\beta = 50$, Iter: 300k)



(b) Allen Cahn (Iter: 200k)

Figure 7: Sample Efficiency of Evo and Causal Evo at low N_r.

6 Conclusions And Future Work Directions

We present a novel perspective for identifying a failure modes in PINNs named "*propagation failure*" which are typically characterized by very narrow regions with very high PDE residuals. We develop a novel *evolutionary sampling* algorithm where the collocation points gradually evolve over the training iterations to prioritize high density regions while maintaining a background distribution of unifomly sampled points. From our experiments, we demonstrate better performance on a variety of benchmark PDEs. Future work can focus on theoretically understanding the interplay between minimizing PDE loss and sampling from high residual regions on PINN performance.

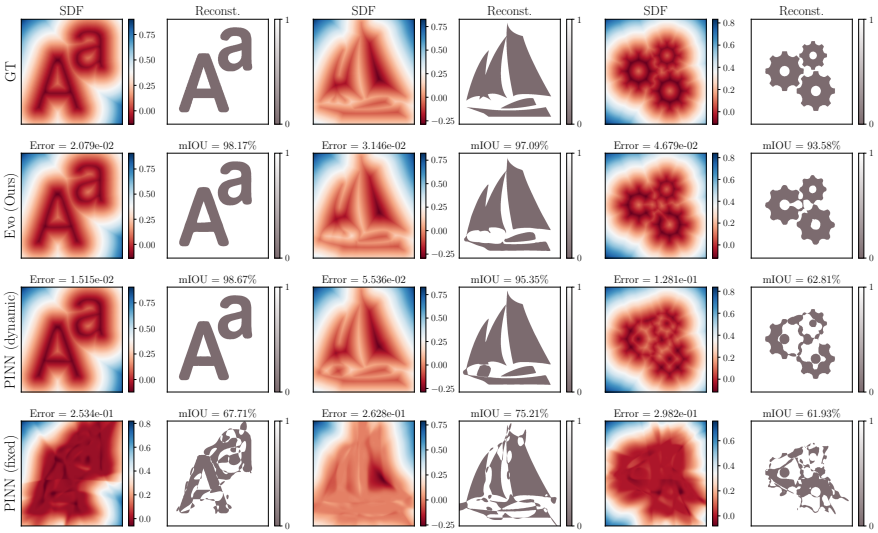


Figure 8: Solving Eikonal equation for signed distance field (SDF). The color of the heatmap represents the values of the SDF. The gray region shows the negative values of SDF that represents the interior points in the reconstructed geometry from predicted SDF.

REFERENCES

Agoston E Eiben, James E Smith, et al. *Introduction to evolutionary computing*, volume 53. Springer, 2003.

Stephen Kokoska and Daniel Zwillinger. *CRC standard probability and statistics tables and formulae*. Crc Press, 2000.

Aditi Krishnapriyan, Amir Gholami, Shandian Zhe, Robert Kirby, and Michael W Mahoney. Characterizing possible failure modes in physics-informed neural networks. *Advances in Neural Information Processing Systems*, 34, 2021.

Randall J LeVeque. *Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems*. SIAM, 2007.

Lu Lu, Xuhui Meng, Zhiping Mao, and George Em Karniadakis. Deepxde: A deep learning library for solving differential equations. *SIAM Review*, 63(1):208–228, 2021.

Mohammad Amin Nabian, Rini Jasmine Gladstone, and Hadi Meidani. Efficient training of physics-informed neural networks via importance sampling. *Computer-Aided Civil and Infrastructure Engineering*, 36(8):962–977, 2021.

Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations. *arXiv preprint arXiv:1711.10561*, 2017.

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

Robert E Schapire. The boosting approach to machine learning: An overview. *Nonlinear estimation and classification*, pp. 149–171, 2003.

Dan Simon. *Evolutionary optimization algorithms*. John Wiley & Sons, 2013.

Chuwei Wang, Shanda Li, Di He, and Liwei Wang. Is l^2 physics-informed loss always suitable for training physics-informed neural network? *arXiv preprint arXiv:2206.02016*, 2022a.

Sifan Wang, Yujun Teng, and Paris Perdikaris. Understanding and mitigating gradient pathologies in physics-informed neural networks. *arXiv preprint arXiv:2001.04536*, 2020.

Sifan Wang, Shyam Sankaran, and Paris Perdikaris. Respecting causality is all you need for training physics-informed neural networks. *arXiv preprint arXiv:2203.07404*, 2022b.

Sifan Wang, Xinling Yu, and Paris Perdikaris. When and why pinns fail to train: A neural tangent kernel perspective. *Journal of Computational Physics*, 449:110768, 2022c.

Chenxi Wu, Min Zhu, Qinyang Tan, Yadhu Kartha, and Lu Lu. A comprehensive study of non-adaptive and residual-based adaptive sampling for physics-informed neural networks. *arXiv preprint arXiv:2207.10289*, 2022.

Olek C Zienkiewicz, Robert L Taylor, and Jian Z Zhu. *The finite element method: its basis and fundamentals*. Elsevier, 2005.