Mitigating Propagation Failures in Physics-informed Neural Networks using Retain-Resample-Release (R3) Sampling

Arka Daw [1] **Jie Bu** [1] **Sifan Wang** [2] **Paris Perdikaris** [2] **Anuj Karpatne** [1]

Abstract

Despite the success of physics-informed neural networks (PINNs) in approximating partial differential equations (PDEs), PINNs can sometimes fail to converge to the correct solution in problems involving complicated PDEs. This is reflected in several recent studies on characterizing the "failure modes" of PINNs, although a thorough understanding of the connection between PINN failure modes and sampling strategies is missing. In this paper, we provide a novel perspective of failure modes of PINNs by hypothesizing that training PINNs relies 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*, characterized by highly imbalanced PDE residual fields. To mitigate propagation failures, we propose a novel *Retain-Resample-Release sampling* (R3) algorithm that can incrementally accumulate collocation points in regions of high PDE residuals with little to no computational overhead. We provide an extension of R3 sampling to respect the principle of causality while solving time-dependent PDEs. We theoretically analyze the behavior of R3 sampling and empirically demonstrate its efficacy and efficiency in comparison with baselines on a variety of PDE problems.

1. Introduction

Our understanding of physical systems in a number of domains largely relies on our ability to solve partial differential equations (PDEs), and hence, enhancing PDE solution accuracy and computational speed can yield substantial benefits. Physics-informed neural networks (PINNs) (Raissi et al., 2019) represent a seminal line of work in deep learning for solving PDEs. The basic idea of PINNs is to train a neural network to minimize errors w.r.t. the PDE 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. Recent success with PINNs has shown significant promise across a broad range of scientific applications, from fluid dynamics (Rao et al., 2020; Zhu et al., 2021), medical imaging (Sahli Costabal et al., 2020; van Herten et al., 2022), material science (Shao et al., 2020; Lin et al., 2022), geophysics (Yang & Ma, 2021; Voytan & Sen, 2020) and climate modeling (Lutjens et al., 2021). Despite the success of PINNs, it is known that PINNs 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., 2021; 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, 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, we can only compute PDE residuals. Also, 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 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

[1]Department of Computer Science, Virginia Tech, Blacksburg, Virginia, USA [2]University of Pennsylvania, Philadelphia, Pennsylvania, USA. Correspondence to: Arka Daw <darka@vt.edu>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

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, making it difficult to adequately represent the high residual regions in the set of collocation points at every iteration. This motivates us to develop sampling strategies that dynamically focus on collocation points from high residual regions during PINN training. This is related to the idea of local-adaptive mesh refinement used in FEM (Zienkiewicz et al., 2005) to refine the computational mesh in regions with high errors.

We propose a novel *Retain-Resample-Release sampling* (R3) 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 theoretically show that R3 *retains* points from high residual regions if they persist over iterations (Retain Property) and *releases* points if they have been resolved by PINN training (Release Property), while maintaining non-zero representation of points *resampled* from a uniform distribution over the entire domain (Resample Property). We also provide a causal extension of our proposed R3 sampling algorithm (Causal R3) 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 empirically evaluate the performance of R3 in multiple benchmark PDE problems. We show the R3 and Causal R3 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.

The novel contributions of our work are as follows: (1) We provide a novel perspective for characterizing failure modes in PINNs by postulating the "Propagation Hypothesis" and empirically demonstrate how regions with high-PDE residuals lead to propagation failures in PINNs. (2) We propose a novel R3 sampling algorithm to adaptively sample collocation points in PINNs that shows superior prediction performance empirically with little to no computational overhead compared to existing methods. (3) We theoretically prove the three key properties of R3 sampling: Retain, Resample, and Release properties.

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:

$$\begin{aligned} &u_t + \mathcal{N}_x[u] = 0,\ x \in \mathcal{X}, t \in [0,T]; \\ &u(x,0) = h(x),\ x \in \mathcal{X}; \\ &u(x,t) = g(x,t),\ t \in [0,T], x \in \partial\mathcal{X} \end{aligned} \tag{1}$$

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, T is the time horizon, and $\mathcal{N}_x$ is the non-linear differential operator. 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:

$$\mathcal{R}_\theta(x,t) = \frac{\partial}{\partial t} f_\theta(x,t) - \mathcal{N}_x[f_\theta(x,t)] \tag{2}$$

$$\mathcal{L}_r(\theta) = \mathbb{E}_{\mathbf{x_r}\sim\mathcal{U}(\Omega)}[\mathcal{R}_\theta(\mathbf{x_r})^2] \approx \frac{1}{N_r}\sum_{i=1}^{N_r}[\mathcal{R}_\theta(x_r^i, t_r^i)]^2 \tag{3}$$

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., 2021) 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 another line of work, the tendency of PINNs to get stuck at trivial solutions due to poor initializations has been demonstrated theoretically in (Wong et al., 2022) and empirically in (Rohrhofer et al., 2022). 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., 2023; Lu et al., 2021; Nabian et al., 2021) or choosing higher-order L^p norms of PDE loss (Wang et al., 2022a). In another

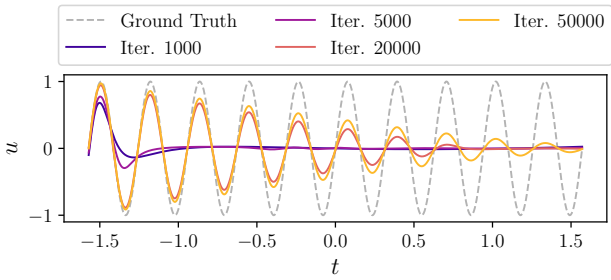


Figure 1. PINN solutions for a simple ODE: $u_{xx} + k^2 u = 0$ ($k = 20$) with the analytical solution, $u = A\sin(kx) + B\cos(kx)$. The boundary condition was set to $u(-\pi/2) = 0$, and the PINN is trained with 1000 equispaced collocation points. We can see smooth *propagation* of the correct solution from the boundary point at $x = 0$ to interior points ($x > 0$) as we increase training iterations.

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. Further, a recent study introduces the concept of "fixed points" u^* (Rohrhofer et al., 2023) which were defined as the roots of the non-linear PDE function, i.e., $\mathcal{R}[u^*] = 0$ (trivial solution $u = 0$ is a special case of fixed point). PINNs are attracted towards these "fixed points" during the initial training iterations, and can get trapped in local minimas, leading to premature convergence.

3. Propagation Hypothesis

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 G 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 H 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. Specifically, we observe that for $\beta = 50$, the PDE residual is very high at the bottom left corner, which overlaps with the region where we observe the hindrance in propagation of the PDE solution. We see similar trends of propagation failure for other values of $\beta > 10$ (see Appendix J.1). We also illustrate in Appendix J.2 that regions with high error rates contribute to the propagation failures in PINNs for another complex PDE, specifically the Kuramoto-Sivashinksy Equation (chaotic regime).

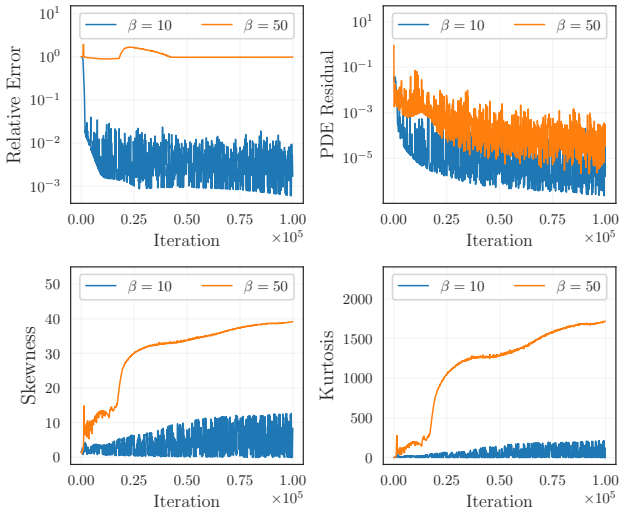


Figure 2. Demonstration of propagation failure using skewness and kurtosis while solving the convection equation with $\beta = 50$.

Remark 3.1. *"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.

Prior Works Focussing on High Residual Regions. 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., 2023; 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 (see Proposition A.1 for more details).

Limitations of Prior Work. There are two main challenges faced by the prior work 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 using a dense set of collocation points $\mathcal{P}_{dense}$ (typically with 100k $\sim$ 1M points spread uniformly across the entire domain) to locate high residual regions, such that points from high residual regions can be added to the training set every K iterations. This increases the computational cost in two ways. First, computing the PDE residuals on the entire dense set is very expensive. Second, the size of the training set keeps growing every K iterations, further increasing the training costs at later iterations. See Appendix F for a detailed analysis of the computational complexity of RAR based methods.
(2) **Difficulty selecting ideal values of k or p:** 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. Additionally, 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. We empirically demonstrate in Section 5, both sampling-based methods such as RAR and its variants

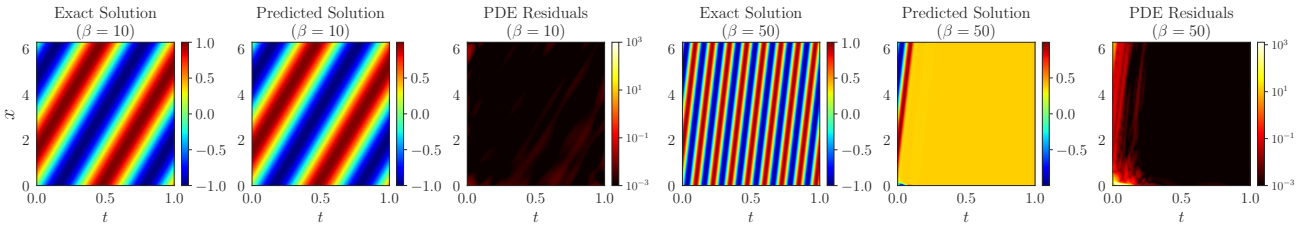


Figure 3. Demonstration of the Exact PDE solution, the predicted PDE solution, and the PDE Residual Field for the convection equation with $\beta = 10$ (first three figures) and $\beta = 50$ (last three figures) respectively.

as well as L^∞ PDE loss-based methods suffer from poor performance in converging to the correct solution for complex PDE problems due to the aforementioned limitations.

4. Proposed R3 Sampling Algorithm

We propose a novel "Retain-Resample-Release" (R3) Sampling algorithm to overcome the limitations in prior works while effectively prioritizing the high residual regions during training to mitigate propagation failures. This algorithm has three key properties which are defined as follows:
(1) ***Retain Property***: To break propagation barriers, R3 focus on *retaining* points from high residual regions such that the set of collocation points starts accumulating in these regions until the PINN training process eventually resolves them. This is similar to starting from L^2-norm and dynamically increasing the order of L^p norm if high residual regions persist over iterations, in contrast to using a fixed value of p as is done in prior works.
(2) ***Resample Property***: At every iteration, R3 ensures that the set of collocation points contains non-zero support of points resampled from a uniform distribution over the entire domain Ω. This is done so that the collocation points do not collapse solely to high residual regions, which is one of the limitations of using L^∞ loss.
(3) ***Release Property***: Upon sufficient minimization of a high residual region through PINN training, collocation points that were once accumulated from the region are *released*, such that R3 can focus on minimizing other high residual regions in later iterations. Note that if the retained points are not released, the set of collocation points would keep growing, thus increasing the computational costs and biasing the sampling to these regions even at later iterations (which is one of the limitations of RAR-based methods).

Along with satisfying the above properties, the R3 algorithm also incurs little to no computational overhead in sampling collocation points from high residual regions, in contrast to prior works. Specifically, we are 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.

Algorithm 1 shows the pseudo-code of our proposed R3 sampling strategy. 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 update the population to the next iteration, we first construct the "retained population" $\mathcal{P}_i^r$ comprising of points from $\mathcal{P}_i$ falling in high residual regions. Specifically, we define the "residual function" $\mathcal{F}(\mathbf{x_r})$ for every collocation point $\mathbf{x_r}$ as the absolute value of the PDE residual of $\mathbf{x_r}$, i.e., $\mathcal{F}(\mathbf{x_r}) = |\mathcal{R}(\mathbf{x_r})|$. We compute τ_i as the expected value of residual function over all points in $\mathcal{P}_i$. Points in $\mathcal{P}_i$ with residual function values greater than τ_i are then considered to be part of the retained population $\mathcal{P}_i^r$, i.e., $\mathcal{P}_i^r \leftarrow \{\mathbf{x}_r^j : \mathcal{F}(\mathbf{x}_r^j) > \tau_i\}$. The remainder of collocation points in $\mathcal{P}_i$ with $\mathcal{F}(\mathbf{x}_r^j) \leq \tau_i$ are dropped and replaced with points 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 and re-sampled population are then merged to generate the population for the next iteration, $\mathcal{P}_{i+1}$. Note that the size of the population at every iteration is constant (equal to N_r). We schematically show the dynamics of collocation points in R3 Algorithm over training iterations in Figure 4.

Algorithm 1 Proposed R3 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 residual function 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 $\mathcal{P}_i^s \leftarrow \{\mathbf{x_r}^j : \mathbf{x_r}^j \sim \mathcal{U}(\Omega)\}$, s.t. $|\mathcal{P}_i^s| + |\mathcal{P}_i^r| = N_r$
7: Merge the two populations $\mathcal{P}_{i+1} \leftarrow \mathcal{P}_i^r \cup \mathcal{P}_i^s$
8: **end for**

4.1. Theoretical Validations of R3 Sampling properties

A unique feature of our proposed R3 sampling algorithm is that while prior works have only addressed a subset of the three sampling properties: Retain, Resample, and Release, we are able to satisfy all of them while being computationally efficient (i.e., without maintaining a dense set of

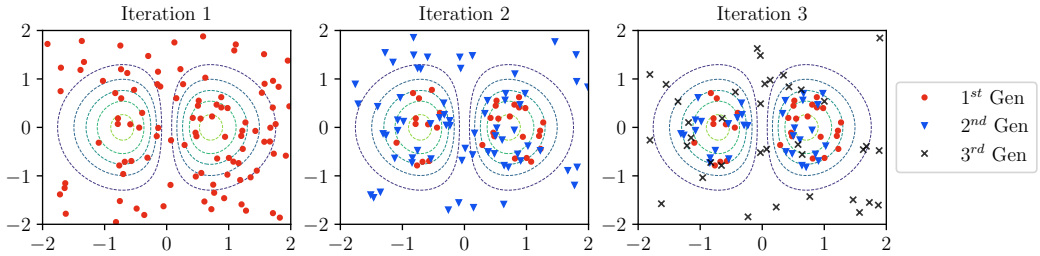


Figure 4. Schematic to describe our proposed R3 sampling algorithm, where collocation points are incrementally accumulated in regions with high PDE residuals (shown as contour lines).

collocation points). See Table 2 in the Appendix for a comparison of our R3 sampling algorithm with prior works. In the following, we theoretically verify the ability of R3 to satisfy the three sampling properties.

(1) **Retain Property:** Theorem 4.1 shows that for a fixed $\mathcal{F}(\mathbf{x})$ (e.g., when θ is kept constant), the expectation of the retained population in R3 sampling becomes maximum (equal to L^{∞}) when the number of iterations approaches ∞.

Theorem 4.1 (Accumulation Dynamics Theorem)**.** *Let $\mathcal{F}_{\theta}(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}^+$ be a fixed real-valued k-Lipschitz continuous objective function optimized using the R3 Sampling algorithm. Then, the expectation of the retained population $\mathbb{E}_{\mathbf{x}\in\mathcal{P}^r}[\mathcal{F}(\mathbf{x})] \geq \max_{\mathbf{x}} \mathcal{F}(\mathbf{x}) - k\epsilon$ as iteration $i \rightarrow \infty$, for any arbitrarily small $\epsilon > 0$.*

The proof of Theorem 4.1 can be found in Appendix C.1. This demonstrates the *Retain property* of R3 sampling as points from high residual regions would keep accumulating in the retained population and make its expectation maximal if the residual function is kept fixed. Note that since the PINN optimizer is also minimizing the residuals at every iteration, we would not expect the residual function to be fixed unless a high residual region persists over a long number of iterations. In fact, points from a high residual region would keep on accumulating until they are resolved by the PINN optimizer and thus eventually released from $\mathcal{P}^r$.

(2) **Resample Property:** Theorem 4.2 states that the size of the resampled population $\mathcal{P}^s$ is always greater zero. As a result, there is always some background density of collocation points sampled from a uniform distribution, preventing the R3 Sampling algorithm from collapsing to high residual regions.

Theorem 4.2 (Non-Empty Theorem)**.** *For any population $\mathcal{P}$ generated at an arbitrary iteration of R3 sampling, the re-sampled population is always non-empty, i.e., $|\mathcal{P}^s| > 0$.*

The proof of Theorem 4.2 can be found in Appendix C.1 stated as a part of Lemma C.3.

(3) **Release Property:** Let us define that for an arbitrary iteration i, a collocation point $\mathbf{x_r} \in \mathcal{P}_i$ is "*sufficiently minimized*" if its residual function is less than τ_i, i.e., $\mathcal{R}_{\theta}(\mathbf{x_r}) \leq \mathbb{E}_{\mathbf{x_r}\in\mathcal{P}_i}[\mathcal{R}_{\theta}(\mathbf{x_r})] = \tau_i$ (where τ_i is the threshold at the i-th iteration). Then, by definition, $\mathbf{x_r}$ will belong to the "non-retained population", i.e., it will be "released" from the population $\mathcal{P}_i$ and replaced by a new point re-sampled from the uniform distribution. Thus, R3 sampling satisfies the "*Release Property*" by design.

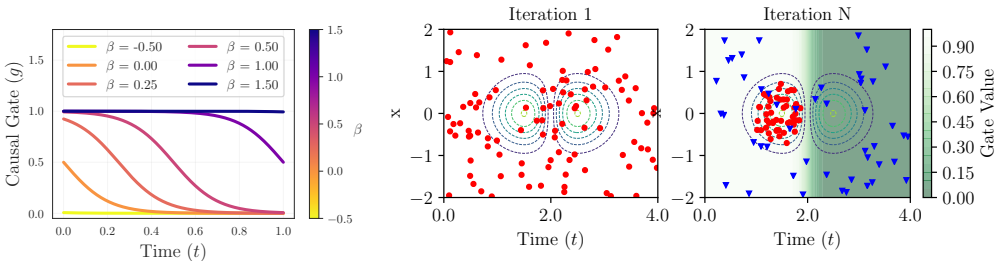
Connections to Numerical Methods: Note that R3 sampling shares a similar motivation as *local-adaptive mesh refinement* methods developed for Finite Element Methods (FEM) (Zienkiewicz et al., 2005), where the goal is to preferentially refine the computational mesh used in numerical methods based on localization 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).

Connections to Evolutionary Algorithms: Our proposed R3 sampling strategy also shares similarity with evolutionary algorithms that are used for modeling biological evolution (Eiben et al., 2003). The residual function $\mathcal{F}$ is equivalent to the fitness function defined for typical evolutionary algorithms. At each iteration, we retain the points with the largest fitness (i.e., points corresponding to the highest PDE residuals), which is similar to the "survival of the fittest" concept commonly found in evolutionary algorithms.

4.2. Causal Extension of R3 Sampling (Causal R3)

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 R3 (*Causal R3*) 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.

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 por-



(a) Causal Gate. (b) Causally R3 Sampling

Figure 5. Causal R3 uses a time-dependent causal gate for computing PDE loss and for sampling.

tion 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 D. 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 R3 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 residual function as $\mathcal{F}(\mathbf{x_r}) = |\mathcal{R}(\mathbf{x_r})| * g(t_r)$. Figure 5(b) represents a schematic describing the causally biased R3 sampling described in Section 4.2 and the causal gate g that is updated every iteration.

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.

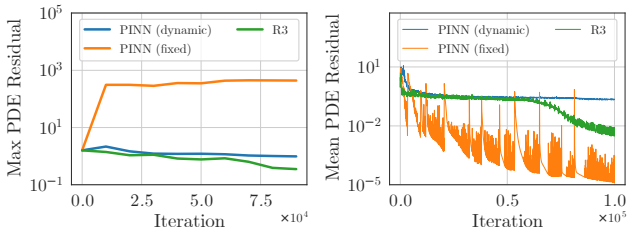


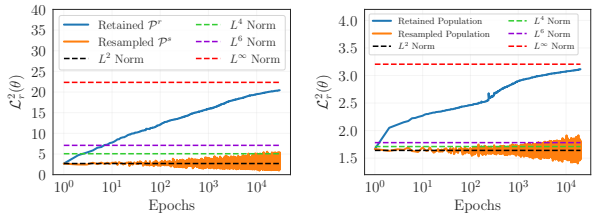
Figure 6. Comparison of Max PDE Residuals, and Mean PDE Residuals over training iterations for PINN-fixed, PINN-dynamic, and R3 Sampling for convection equation with $\beta = 50$.

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. For PINNs and CausalPINNs (Wang et al., 2022b) we have defined two separate baselines, (1) fixed: which uses a fixed set of uniformly sampled collocation points, (2) dynamic: a simple modification where the collocation points are dynamically sampled from a uniform distribution every iteration (see Appendix B for more details). We introduce another baseline L^∞ where we sample top N_r collocation points at every iteration from a dense set $\mathcal{P}_{dense}$ to approximate L^∞ norm. The other baselines have been listed in Table 1. 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 I. All of our codes and datasets are available here [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

[1] https://github.com/arkadaw9/r3_sampling_icml2023



(a) Auckley Function (b) Michaelewicz Function

Figure 7. Demonstration of accumulation behavior of R3 sampling on two sample objective functions using the PDE loss of $\mathcal{P}^r$ going from L^2 to L^∞ with iterations for two sample objective function.

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. 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 (R3 and Causal R3) consistently show the lowest errors across all PDEs. Figure 5 shows that R3 and PINN-dynamic are indeed able to mitigate *propagation failures* for convection ($\beta = 50$), by maintaining low values of max PDE residuals across all iterations, in contrast to PINN-fixed. Note that R3 is further able to reduce the mean PDE residuals better than PINN-dynamic. Additional visualizations of the evolution of samples in R3, Caual R3, and RAR-based methods across iterations are provided in Appendix J. Sensitivity of RAR-based methods to hyper-parameters is provided in Appendix J.5.

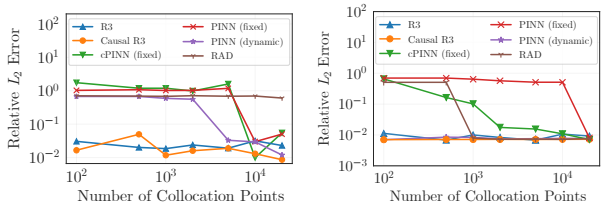
Accumulation Behavior of R3 Sampling: To demonstrate the ability of R3 Sampling to accumulate high residual points in the retained population $\mathcal{P}^r$ (or equivalently, focus on higher-order L^p norms of PDE loss), we consider optimizing two fixed objective functions: the Auckley function and Michaelewicz Function in Figure 7 (see Appendix K.1 and K.7 for details of these 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 behavior of R3 sampling as theoretically stated in Theorem 4.1. Addition visualizations of the dynamics of R3 sampling for a number of test optimization functions are provided in Appendix K.

Sampling Efficiency: Figure 8 shows the scalability of R3 sampling to smaller sample sizes of collocation points, N_r. Though all the baselines demonstrate similar performances when N_r is large ($> 10K$), only R3 and Causal R3 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 R3 sampling and Causal R3 show faster convergence speed than baseline methods for both convection and Allen Cahn equations (see Appendix J.4 for details). Further, in Figure 9 we compare sample efficiency of CPINN, R3 sampling, and Causal R3 on the Kuramoto-Sivashinsky (KS) equation (regular case) (for details refer to Appendix G). We can see that both R3 and Causal R3 show improvements over CPINN when the number of collocation points is small ($N_r = 128$). As the number of collocation points is increased, Causal R3 shows

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 (Krishnapriyan et al., 2021)	$63.2 \pm 9.89\%$	$2.65 \pm 1.44\%$	$48.9 \pm 7.44\%$	$31.5 \pm 16.6\%$	-
CPINN (fixed) (Wang et al., 2022b)	$138.8 \pm 11.0\%$	$138.8 \pm 11.0\%$	$106.5 \pm 10.5\%$	$106.5 \pm 10.5\%$	$48.7 \pm 19.6\%$
CPINN (dynamic) (Wang et al., 2022b)	$52.2 \pm 43.6\%$	$23.8 \pm 45.1\%$	$79.0 \pm 5.11\%$	$73.2 \pm 8.36\%$	$1.5 \pm 0.75\%$
RAR-G (Lu et al., 2021)	$10.5 \pm 5.67\%$	$2.66 \pm 1.41\%$	$65.7 \pm 17.0\%$	$43.1 \pm 28.9\%$	$25.1 \pm 23.2\%$
RAD (Nabian et al., 2021)	$3.35 \pm 2.02\%$	$1.85 \pm 1.90\%$	$66.0 \pm 1.55\%$	$64.1 \pm 1.98\%$	$0.78 \pm 0.05\%$
RAR-D (Wu et al., 2023)	$67.1 \pm 4.28\%$	$32.0 \pm 25.8\%$	$82.9 \pm 5.99\%$	$75.3 \pm 9.58\%$	$51.6 \pm 0.41\%$
L^∞	$66.6 \pm 2.35\%$	$41.2 \pm 27.9\%$	$76.6 \pm 1.04\%$	$75.8 \pm 1.01\%$	$1.65 \pm 1.36\%$
R3 (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 R3 (ours)	$2.12 \pm 0.67\%$	$\mathbf{0.75 \pm 0.12}\%$	$\mathbf{5.99 \pm 5.25}\%$	$2.28 \pm 0.76\%$	$\mathbf{0.71 \pm 0.007}\%$

better performance than R3, as it incorporates an additional prior of causality along with satisfying the three properties of R3. Additional results on other cases of KS Equations including chaotic behavior are provided in Appendix J.9.



(a) Convection Equation ($\beta = 50$, Iter: 300k) (b) Allen Cahn Equation (Iter: 200k)

Figure 8. Demonstrating Sample Efficiency of R3 sampling and Causal R3 at low values of N_r.

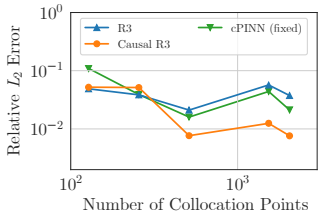


Figure 9. Comparison of R3, Causal R3 and CausalPINNs on KS Equation (regular) with varying number of collocation points.

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 G for details of the Eikonal equation. The primary difficulty in solving Eikonal equation comes from determining the sign of the field (interior or exterior) in regions with rich details. We compare the performance of different baseline methods with respect to the ground-truth (GT) solution obtained from numerical methods for two complex surface geometries in Figure 10. 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. The results show that PINN-fixed shows poor performance across both geometries, while PINN-dynamic is able to capture most of the outline of the solutions with a few details missing. On the other hand, R3 sampling is able to capture even the fine details of the SDF for both geometries and thus show better reconstruction quality. We can see that mIOU of R3 sampling is significantly higher than baselines. See Appendix Section J.10 for more discussion and visualizations.

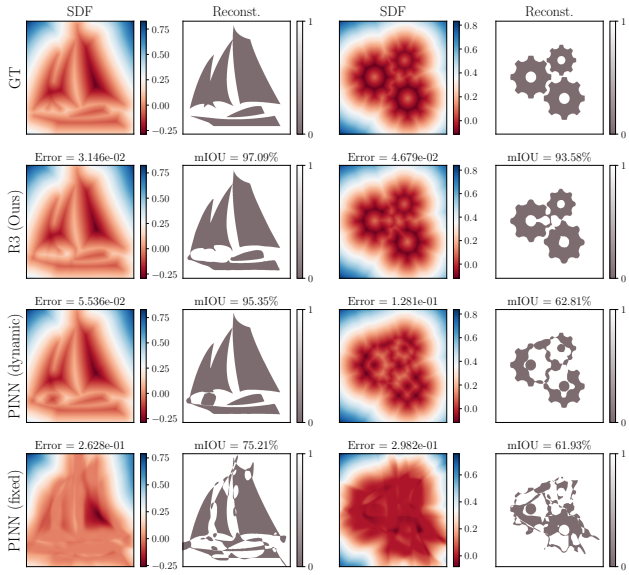


Figure 10. Solving Eikonal equation for signed distance field (SDF). The color of the heatmap represents the values of the SDF. The gray region shows the negative values of SDF that represents the interior points in the reconstructed geometry from predicted SDF.

6. Conclusions And Future Work Directions

We present a novel perspective for identifying failure modes in PINNs named "*propagation failures*." and develop a novel *R3 sampling* algorithm to mitigate propagation failures. R3 empirically demonstrates better performance on a variety of benchmark PDEs. Future work can focus on building the theoretical foundations of the propagation hypothesis and studying the interplay between minimizing PDE loss and sampling from high residual regions. Building on the success of established sampling techniques such as importance sampling in neural implicit representation learning problems, future research can focus on investigating the potential applicability of our proposed R3 sampling strategy in these problems.

7. Acknowledgement

We gratefully acknowledge the support of the National Science Foundation (NSF) under Grant #2213550 and #2107332, which provided funding for this research. We would also like to acknowledge the support of Virginia Tech Advanced Research Computing (ARC) for providing us with necessary resources for our study.

References

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

Kokoska, S. and Zwillinger, D. *CRC standard probability and statistics tables and formulae*. Crc Press, 2000.

Krishnapriyan, A., Gholami, A., Zhe, S., Kirby, R., and Mahoney, M. W. Characterizing possible failure modes in physics-informed neural networks. *Advances in Neural Information Processing Systems*, 34, 2021.

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

Lin, Y.-S., Pun, G. P. P., and Mishin, Y. Development of a physically-informed neural network interatomic potential for tantalum. *Computational Materials Science*, 205: 111180, 2022.

Lu, L., Meng, X., Mao, Z., and Karniadakis, G. E. Deepxde: A deep learning library for solving differential equations. *SIAM Review*, 63(1):208–228, 2021.

Lutjens, B., Crawford, C. H., Veillette, M., and Newman, D. Pce-pinns: Physics-informed neural networks for uncertainty propagation in ocean modeling. *arXiv preprint arXiv:2105.02939*, 2021.

Nabian, M. A., Gladstone, R. J., and Meidani, H. Efficient training of physics-informed neural networks via importance sampling. *Computer-Aided Civil and Infrastructure Engineering*, 36(8):962–977, 2021.

Raissi, M., Perdikaris, P., and Karniadakis, G. E. Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations. *arXiv preprint arXiv:1711.10561*, 2017.

Raissi, M., Perdikaris, P., and Karniadakis, G. E. 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.

Rao, C., Sun, H., and Liu, Y. Physics-informed deep learning for incompressible laminar flows. *Theoretical and Applied Mechanics Letters*, 10(3):207–212, 2020.

Rohrhofer, F. M., Posch, S., Gößnitzer, C., and Geiger, B. C. Understanding the difficulty of training physics-informed neural networks on dynamical systems. *arXiv preprint arXiv:2203.13648*, 2022.

Rohrhofer, F. M., Posch, S., Gößnitzer, C., and Geiger, B. C. On the role of fixed points of dynamical systems in training physics-informed neural networks. *Transactions on Machine Learning Research*, 2023.

Sahli Costabal, F., Yang, Y., Perdikaris, P., Hurtado, D. E., and Kuhl, E. Physics-informed neural networks for cardiac activation mapping. *Frontiers in Physics*, 8:42, 2020.

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

Shao, Y., Hellstrom, M., Mitev, P. D., Knijff, L., and Zhang, C. Pinn: A python library for building atomic neural networks of molecules and materials. *Journal of chemical information and modeling*, 60(3):1184–1193, 2020.

van Herten, R. L., Chiribiri, A., Breeuwer, M., Veta, M., and Scannell, C. M. Physics-informed neural networks for myocardial perfusion mri quantification. *Medical Image Analysis*, 78:102399, 2022.

Voytan, D. and Sen, M. K. Wave propagation with physics informed neural networks. In *SEG International Exposition and Annual Meeting*. OnePetro, 2020.

Wang, C., Li, S., He, D., and Wang, L. Is l2 physics informed loss always suitable for training physics informed neural network? *Advances in Neural Information Processing Systems*, 35, 2022a.

Wang, S., Teng, Y., and Perdikaris, P. Understanding and mitigating gradient flow pathologies in physics-informed neural networks. *SIAM Journal on Scientific Computing*, 43(5):A3055–A3081, 2021.

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

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

Wong, J. C., Ooi, C., Gupta, A., and Ong, Y.-S. Learning in sinusoidal spaces with physics-informed neural networks. *IEEE Transactions on Artificial Intelligence*, 2022.

Wu, C., Zhu, M., Tan, Q., Kartha, Y., and Lu, L. A comprehensive study of non-adaptive and residual-based adaptive sampling for physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 403:115671, 2023.

Yang, F. and Ma, J. Revisit geophysical imaging in a new view of physics-informed generative adversarial learning. *arXiv preprint arXiv:2109.11452*, 2021.

Zhu, Q., Liu, Z., and Yan, J. Machine learning for metal additive manufacturing: predicting temperature and melt pool fluid dynamics using physics-informed neural networks. *Computational Mechanics*, 67:619–635, 2021.

Zienkiewicz, O. C., Taylor, R. L., and Zhu, J. Z. *The finite element method: its basis and fundamentals*. Elsevier, 2005.

A. Connections Between L^p Norm and Sampling

In this section, we provide connections between adaptively sampling collocation points from a distribution $q(\mathbf{x_r}) \propto |\mathcal{R}_\theta(\mathbf{x_r})|^k$ (Wu et al., 2023) and using L^p norm of the PDE loss (Wang et al., 2022a).

Theorem A.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, $\mathcal{L}_r^2(\mathcal{Q}^k) = \frac{1}{Z^{1/2}V^{-1/2}}\Big(\mathcal{L}_r^{k+2}(\mathcal{U})\Big)^{(k+2)/2}$, where Z is a normalization constant, defined as $Z = \int_{\mathbf{x}_r \in \Omega} |\mathcal{R}_\theta(\mathbf{x_r})|^k d\mathbf{x_r}$ and V is the volume of the domain Ω.*

Proof. The expectation of L^p PDE Loss for collocation points sampled from a uniform distribution $\mathcal{U}(\Omega) : \mathbf{x_r} \sim p(\mathbf{x_r})$ can be defined as follows [2]:

$$\begin{aligned}\mathcal{L}_r^p(\mathcal{U}) &= \Big(\mathbb{E}_{\mathbf{x_r}\sim\mathcal{U}(\Omega)}|\mathcal{R}_\theta(\mathbf{x_r})|^p\Big)^{1/p} \\ &= \Big(\int p(\mathbf{x_r})|\mathcal{R}_\theta(\mathbf{x_r})|^p d\mathbf{x_r}\Big)^{1/p}\end{aligned} \tag{4}$$

Note that for a uniform distribution, $p(\mathbf{x_r}) = \frac{1}{V}$, where V is the volume of the domain Ω, i.e., $V = \prod_{i=1}^n \Big(\text{supp}(x_i) - \text{inf}(x_i)\Big)$ with supp(.) and inf(.) being the supremum and infimum operators, and x_i is the i-th dimension of $\mathbf{x_r}$ (e.g., the space dimension x or the time dimension t).

Now, let us consider the case where we are interested in sampling from an alternate distribution $\mathcal{Q}^k(\Omega) : \mathbf{x_r} \sim q(\mathbf{x_r})$, where $q(\mathbf{x_r}) \propto |\mathcal{R}_\theta(\mathbf{x_r})|^k$ while using the L^2 PDE Loss (the most standard loss formulation used in PINNs). The sampling function of $\mathcal{Q}^k(\Omega)$ can be defined as follows:

$$q(\mathbf{x_r}) = \frac{|\mathcal{R}_\theta(\mathbf{x_r})|^k}{Z} \tag{5}$$

where Z is the normalizing constant, i.e., $Z = \int_{\mathbf{x}_r \in \Omega} |\mathcal{R}_\theta(\mathbf{x_r})|^k d\mathbf{x_r}$.

Hence, the L^2 PDE Loss for collocation points sampled from $\mathcal{Q}^k(\Omega)$ can be defined as:

$$\begin{aligned}\mathcal{L}_r^2(\mathcal{Q}^k) &= \Big(\mathbb{E}_{\mathbf{x_r}\sim\mathcal{Q}^k(\Omega)}|\mathcal{R}_\theta(\mathbf{x_r})|^2\Big)^{1/2} \\ &= \Big(\int q(\mathbf{x_r})|\mathcal{R}_\theta(\mathbf{x_r})|^2 d\mathbf{x_r}\Big)^{1/2} \\ &= \Big(\int \frac{|\mathcal{R}_\theta(\mathbf{x_r})|^k}{Z}|\mathcal{R}_\theta(\mathbf{x_r})|^2 d\mathbf{x_r}\Big)^{1/2} && \text{(From Equation 5)} \\ &= \frac{1}{Z^{1/2}}\Big(\int |\mathcal{R}_\theta(\mathbf{x_r})|^{k+2} d\mathbf{x_r}\Big)^{1/2} \\ &= \frac{1}{Z^{1/2}}\Big(\int \frac{p(\mathbf{x_r})}{p(\mathbf{x_r})}|\mathcal{R}_\theta(\mathbf{x_r})|^{k+2} d\mathbf{x_r}\Big)^{1/2} \\ &= \frac{1}{Z^{1/2}V^{-1/2}}\Big(\int p(\mathbf{x_r})|\mathcal{R}_\theta(\mathbf{x_r})|^{k+2} d\mathbf{x_r}\Big)^{1/2}, \qquad \because \mathbf{x_r} \sim \mathcal{U}(\Omega) \implies p(\mathbf{x_r}) = \frac{1}{V} \\ &= \frac{1}{Z^{1/2}V^{-1/2}}\Big(\mathbb{E}_{\mathbf{x_r}\sim\mathcal{U}(\Omega)}|\mathcal{R}_\theta(\mathbf{x_r})|^{k+2}\Big)^{1/2} \\ &= \frac{1}{Z^{1/2}V^{-1/2}}\Big(\mathcal{L}_r^{k+2}(\mathcal{U})\Big)^{(k+2)/2} && \text{(From Equation 4, with } p = k+2)\end{aligned} \tag{6}$$

□

[2] We assume that the batch size/number of collocation points used to compute the PDE Loss tends to infinity, i.e., $N_r \to \infty$. This allows us to analyze the behavior of the continuous PDE loss function $\mathcal{L}_r(\theta)$ as $N_r \to \infty$

Theorem A.1 suggests that sampling collocation points from a distribution $q(\mathbf{x_r}) \propto |\mathcal{R}_\theta(\mathbf{x_r})|^k$ and L^p norm of the PDE loss are related to each other by a scaling term $\frac{1}{Z^{1/2}V^{-1/2}}$. However, since this scaling term is variable in nature as Z depends on the neural network parameters θ, optimizing $\mathcal{L}_r^p(\mathcal{U})$ is not directly equivalent to optimizing $\mathcal{L}_r^2(\mathcal{Q}^k)$ (or a power thereof).

B. Analyzing PINN-Dynamic: A Strong Baseline

A very simple baseline for mitigating propagation failures is to *dynamically* sample a random set of N_r collocation points from a uniform distribution at every iteration, independently of previous iterations. To see how this simple sampling strategy can help in sampling points from high PDE residual regions, let us consider partitioning the entire input domain Ω into two subsets: regions with high PDE residuals, Ω_{high}, and regions with low PDE residuals Ω_{low}. If the high PDE residual regions are imbalanced because of propagation failure, we would expect that the area of Ω_{high} (A_{high}) is significantly smaller than the area of Ω_{low} (A_{low}), i.e., $A_{high} \ll A_{low}$. Let us compute the probability of sampling at least one collocation point from Ω_{high} across all iterations of PINN training. If we use a *fixed* set of collocation points sampled from a uniform distribution at every iteration, this probability will be equal to $A_{high}/(A_{high} + A_{low})$, which can be very small. Hence, we are likely to under-represent points from Ω_{high} in the training process and thus get stuck in trivial solutions with high residual regions around its boundaries, especially when N_r is low.

On the other hand, if we perform *dynamic* sampling, the probability of picking at least one point from Ω_{high} across all iterations will be equal to $1 - (1 - A_{high}/(A_{high} + A_{low}))^N$, where N is the number of iterations for which the high residual region A_{high} persists during the training of PINN. Note that $N << N_{train}$ where N_{train} is the total number of training iterations for the PINN. We can see that when N is large, this probability approaches 1, indicating that across all iterations of PINN training, we would have likely sampled a point from Ω_{high} in at least one iteration, and used it to minimize its PDE residual. As we empirically demonstrate later in Section 5, dynamic sampling is indeed able to control the skewness of PDE residuals compared to fixed sampling, and thus act as a strong baseline for mitigating propagation failures.

However, note that even if we use dynamic sampling, the contribution of points from Ω_{high} in the overall PDE residual loss computed at any iteration is still low. In particular, since the probability of sampling points from Ω_{high} at any iteration is equal to $A_{high}/(A_{high} + A_{low})$, the expected PDE residual loss computed over all collocation points will be equal to

$$\mathbb{E}_\Omega[\mathcal{L}_r(\theta)] = \mathbb{E}_{\Omega_{high}}[\mathcal{L}_r(\theta)] \times \frac{A_{high}}{A_{high} + A_{low}} + \mathbb{E}_{\Omega_{low}}[\mathcal{L}_r(\theta)] \times \frac{A_{low}}{A_{high} + A_{low}} \tag{7}$$

Since $A_{high} \ll A_{low}$, the gradient update of θ at every epoch will be dominated by the low PDE residuals observed over points from Ω_{low}, leading to slow propagation of information from initial/boundary points to interior points.

C. Analysis of R3 Sampling

In this section, we analyze the dynamic behavior (or evolution) of the collocation points for our proposed *Retain-Resample-Release Sampling* (R3) approach over the iterations.

C.1. Retain Property of R3.

In this section, we provide the proof of Theorem 4.1 presented in the main paper.

Definition C.1 (Objective Function)**.** Let $\mathcal{F}_\theta(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^+$ be an arbitrary positive real-valued k-Lipschitz continuous function, where θ denotes the neural network parameters. When θ is fixed, the function $\mathcal{F}(\mathbf{x})$ does not vary with iterations, representing a fixed objective function.

Let $\mathbf{X}^* = \{\mathbf{x}_i^* : \mathcal{F}(\mathbf{x}_i^*) = \max_\mathbf{x} \mathcal{F}(\mathbf{x}) \;\; \forall\, i \in [n]\}$ be the set of points where the objective function $\mathcal{F}$ is maximal.

Now, let us define an ϵ-neighborhood around each of point $\mathbf{x}_i^* \in \mathbf{x}^*$ as $\mathcal{N}_\epsilon(\mathbf{x}_i)$ such that $||\mathbf{x}_i - \mathbf{x}_i^*|| \leq \epsilon$ for any arbitrarily small $\epsilon > 0$ and for all $\mathbf{x}_i \in \mathcal{N}_\epsilon(\mathbf{x}_i)$ with $i \in [n]$.

Let us also assume that the objection function $\mathcal{F}$ is k-Lipschitz continuous. Then the following is true:

$$|\mathcal{F}(\mathbf{x}_i^*) - \mathcal{F}(\mathbf{x}_i)| \leq k\epsilon \qquad \forall\, \mathbf{x}_i \in \mathcal{N}_\epsilon(\mathbf{x}_i) \;\;\&\;\; i \in [n] \tag{8}$$

$$\implies \mathcal{F}(\mathbf{x}_i^*) - \mathcal{F}(\mathbf{x}_i) \leq k\epsilon \qquad \because \mathcal{F}(\mathbf{x}_i^*) = \max_\mathbf{x} \mathcal{F}(\mathbf{x}) \tag{9}$$

Definition C.2 (ϵ-maximal Neighborhood)**.** Let $\mathcal{F}^* = \max_{\mathbf{x}} \mathcal{F}(\mathbf{x})$ be the maximal value of the objective function $\mathcal{F}(\mathbf{x})$ (Definition C.1). Then an ϵ-maximal Neighborhood $\mathcal{N}_\epsilon^\infty$ can be defined as: $\mathcal{N}_\epsilon^\infty = \mathcal{N}_\epsilon(\mathbf{x}_0) \cup \mathcal{N}_\epsilon(\mathbf{x}_1) \cup ... \cup \mathcal{N}_\epsilon(\mathbf{x}_n)$ such that any point $\mathbf{x}$ sampled from $\mathcal{N}_\epsilon^\infty$ would have $\mathcal{F}^* - \mathcal{F}(\mathbf{x}) \leq k\epsilon \;\; \forall \; \mathbf{x} \in \mathcal{N}_\epsilon^\infty$ and for any arbitrarily small $\epsilon > 0$.

Note that since the volume of $\mathcal{N}_\epsilon^\infty$ is greater than 0, the probability of sampling any $\mathbf{x} \in \mathcal{N}_\epsilon^\infty$ from a uniform distribution $\mathcal{U}(\mathbf{x})$ is greater than 0.

Lemma C.3 (Population Properties)**.** *For any population $\mathcal{P}$ generated at some iteration of R3 sampling, optimizing a given objective function $\mathcal{F}(\mathbf{x})$ (Definition C.1), the following properties are always true:*

1. *The re-sampled population is always non-empty, i.e., $|\mathcal{P}^s| > 0$*
2. *The size of the retained population is always less than the total population size, i.e., $|\mathcal{P}^r| < |\mathcal{P}|$*
3. *The size of the retained population is zero, i.e., $|\mathcal{P}^r| = 0$, if and only if $\mathcal{F}(\mathbf{x}) = c, \forall \mathbf{x} \in \mathcal{P}$.*

Proof. The threshold τ for the R3 Sampling can be computed as $\tau = \frac{1}{|\mathcal{P}|} \sum_{\mathbf{x} \in \mathcal{P}} \mathcal{F}(\mathbf{x})$.

The retained population is defined as: $\mathcal{P}^r \leftarrow \{\mathbf{x} : \mathcal{F}(\mathbf{x}) > \tau \;\; \forall \mathbf{x} \in \mathcal{P}\}$,

Similarly, the non-retained population can be defined as: $\overline{\mathcal{P}^r} \leftarrow \{\mathbf{x} : \mathcal{F}(\mathbf{x}) \leq \tau \;\; \forall \mathbf{x} \in \mathcal{P}\}$.

Proof of Property 1: For any arbitrary set of real numbers, there always exists some element in the set that is less than or equal to the mean. Hence, the size of the non-retained population is always non-zero as there always exists some point $\mathbf{x} \in \mathcal{P}$ such that $\mathcal{F}(\mathbf{x}) \leq \tau$. Thus, $|\overline{\mathcal{P}^r}| > 0$.

Now by definition, since the re-sampled population $\mathcal{P}^s$ replaces the non-retained population at every iteration, $|\mathcal{P}^s| = |\overline{\mathcal{P}^r}|$. Hence, $|\mathcal{P}^s| > 0$, i.e., the size of the resampled population is always non-zero.

Proof of Property 2: By definition, $|\overline{\mathcal{P}^r}| + |\mathcal{P}^r| = |\mathcal{P}|$ (where $|\mathcal{P}|$ is the total size of the population and is always constant). Since, $|\overline{\mathcal{P}^r}| > 0$, we can say that $|\mathcal{P}^r| < |\mathcal{P}|$, i.e., the size of the retained population can never be equal to the entire population size $|\mathcal{P}|$.

Proof of Property 3: Let us consider the case where $\mathcal{F}(\mathbf{x}) = c, \forall \mathbf{x} \in \mathcal{P}$ (where c is some constant), i.e., the value of the function is constant at all of the points $\mathbf{x} \in \mathcal{P}$. In this case, the mean of the population $\mathcal{P}$, which is equal to the threshold, τ will be equal to c. This condition would lead to the entire population to be re-sampled as all element $\mathbf{x} \in \mathcal{P}$ would satisfy the condition to belong in the non-retained population. Note that the constant function $\mathcal{F}(\mathbf{x}) = c$ is the only case where all of the elements are less than or equal to the mean. Otherwise, there would always be at least one element greater than the mean, resulting in a non-zero size of the retained population. □

Lemma C.4 (Entry Condition)**.** *If a point $\mathbf{x}_m$ is sampled from $\mathcal{N}_\epsilon^\infty$ at any arbitrary iteration m, then it will always enter the retained population $\mathcal{P}_m^r$ unless $\mathbb{E}_{\mathbf{x} \in \mathcal{P}_m^r}[\mathcal{F}(\mathbf{x})] > \mathcal{F}^* - k\epsilon$.*

Proof. The condition for any arbitrary point $\mathbf{x}_m$ to enter the retained population $\mathcal{P}_m^r$ at any arbitrary iteration m is given by the following:

$$\mathcal{F}(\mathbf{x}_m) > \tau_m = \mathbb{E}_{\mathbf{x} \in \mathcal{P}_m}[\mathcal{F}(\mathbf{x})]. \qquad \text{(By definition of the threshold } \tau_m\text{)} \tag{10}$$

Now, if the point $\mathbf{x}_m$ is sampled from $\mathcal{N}_\epsilon^\infty$, then $\mathcal{F}(\mathbf{x}_m) \geq \mathcal{F}^* - k\epsilon$ (from Definition C.2). Hence, for $\mathbf{x}_m$ to enter the retained population $\mathcal{P}_m^r$, we need to ensure that $\mathcal{F}^* - k\epsilon > \tau_m$.

Let us consider the case where $\mathbf{x}_m$ is not able to enter the retained population. In such a case, we will have the following inequality:

$$\mathcal{F}^* - k\epsilon < \tau_m. \tag{11}$$

It is also easy to show from the definition of retained population that the threshold τ is always less than the expectation of the retained population:

$$\tau_m \leq \mathbb{E}_{\mathbf{x} \in \mathcal{P}_m^r}[\mathcal{F}(\mathbf{x})] \tag{12}$$

From Equations 11 and 12, we get,

$$\begin{aligned} &\mathcal{F}^* - k\epsilon < \tau_m \leq \mathbb{E}_{\mathbf{x}\in\mathcal{P}_m^r}[\mathcal{F}(\mathbf{x})] \\ \implies &\mathbb{E}_{\mathbf{x}\in\mathcal{P}_m^r}[\mathcal{F}(\mathbf{x})] > \mathcal{F}^* - k\epsilon \end{aligned} \tag{13}$$

We have thus proved that $\mathbf{x}_m$ will not be able to enter the retained population $\mathcal{P}_m^r$ only if $\mathbb{E}_{\mathbf{x}\in\mathcal{P}_m^r}[\mathcal{F}(\mathbf{x})] > \mathcal{F}^* - k\epsilon$, which suggests that the expectation of the retained population is already close to $\mathcal{F}^*$, for any arbitrarily small $\epsilon > 0$. On the other hand, if $\mathbb{E}_{\mathbf{x}\in\mathcal{P}_m^r}[\mathcal{F}(\mathbf{x})] \leq \mathcal{F}^* - k\epsilon$, we would necessarily add $\mathbf{x}_m$ □

Lemma C.5 (Exit Condition)**.** *A point $\mathbf{x}_m$ sampled from $\mathcal{N}_\epsilon^\infty$ that entered the retained population at any arbitrary iteration m, can exit the retained population $\mathcal{P}_n^r$ at an arbitrary iteration n (such that $n > m$) only if $\mathbb{E}_{\mathbf{x}\in\mathcal{P}_n^r}[\mathcal{F}(\mathbf{x})] \geq \mathcal{F}^* - k\epsilon$.*

Proof. The generic condition for any arbitrary point $\mathbf{x}$ to exit the retained population $\mathcal{P}_n^r$ at iteration n is given by: $\mathcal{F}(\mathbf{x}) \leq \tau_n = \mathbb{E}_{\mathbf{x}\in\mathcal{P}_n}[\mathcal{F}(\mathbf{x})]$ (By definition of the threshold τ).

Since a point $\mathbf{x}_m$ that was originally sampled from $\mathcal{N}_\epsilon^\infty$ will have $\mathcal{F}(\mathbf{x}_m) \geq \mathcal{F}^* - k\epsilon$ (from Definition C.2), we can use this inequality in the generic exit condition shown above to get,

$$\mathcal{F}^* - k\epsilon \leq \tau_n \leq \mathbb{E}_{\mathbf{x}\in\mathcal{P}_n^r}[\mathcal{F}(\mathbf{x})] \tag{14}$$

Hence, the point $\mathbf{x}_m$ can exit the retained population at iteration n only if $\mathbb{E}_{\mathbf{x}\in\mathcal{P}_n^r}[\mathcal{F}(\mathbf{x})] \geq \mathcal{F}^* - k\epsilon$. □

Theorem C.6 (Accumulation Dynamics Theorem)**.** *Let $\mathcal{F}_\theta(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^+$ be a fixed real-valued k-Lipschitz continuous objective function optimized using the R3 Sampling algorithm. Then, the expectation of the retained population $\mathbb{E}_{\mathbf{x}\in\mathcal{P}^r}[\mathcal{F}(\mathbf{x})] \geq \max_{\mathbf{x}} \mathcal{F}(\mathbf{x}) - k\epsilon$ as iteration $i \to \infty$, for any arbitrarily small $\epsilon > 0$.*

Proof. We prove this theorem by contradiction. For the sake of contradiction, let us assume that as iterations $i \to \infty$, the expectation of the retained population $\mathbb{E}_{\mathbf{x}\in\mathcal{P}^r}[\mathcal{F}(\mathbf{x})] < \max_{\mathbf{x}} \mathcal{F}(\mathbf{x}) - k\epsilon$, for any arbitrarily small $\epsilon > 0$. We can then make the following two remarks.

Entry of collocation points: Note that the probability of sampling $\mathbf{x}$ from $\mathcal{N}_\epsilon^\infty$ is non-zero because the size of the re-sampled population is non-zero, i.e., $|\mathcal{P}^s| > 0$ (proved in Lemma C.3). Also, since we have assumed $\mathbb{E}_{\mathbf{x}\in\mathcal{P}^r}[\mathcal{F}(\mathbf{x})] < \mathcal{F}^* - k\epsilon$, we can use the Entry condition proved in Lemma C.4 to arrive at the conclusion that a point from $\mathcal{N}_\epsilon^\infty$ will always be able to enter the retained population.

Exit of collocation points: Similarly, a point $\mathbf{x}$ that belongs in the ϵ-maximal neighborhood and is part of the retained population $\mathcal{P}^r$ will not be able to escape the retained population as we have asssumed $\mathbb{E}_{\mathbf{x}\in\mathcal{P}^r}[\mathcal{F}(\mathbf{x})] < \mathcal{F}^* - k\epsilon$ (using the Exit condition proved in Lemmas C.5).

From the above two remarks, we can see that points would keep accumulating indefinitely in the retained population if our initial assumption (for the sake of contradiction) is true. However, since the total size of the population $|\mathcal{P}|$ is bounded, the size of the retained population $|\mathcal{P}^r|$ cannot grow indefinitely. We have thus arrived at a contradiction suggesting our assumption is incorrect. Hence, as iterations $i \to \infty$, the expectation of the retained population $\mathbb{E}_{\mathbf{x}\in\mathcal{P}^r}[\mathcal{F}(\mathbf{x})] \geq \max_{\mathbf{x}} \mathcal{F}(\mathbf{x}) - k\epsilon$, for any arbitrarily small $\epsilon > 0$.

□

From Theorem 4.1, we can prove a continuous accumulation of collocation points from the ϵ-maximal neighborhood until the expectation of the retained population is close to the maximum point (i.e., reaches L^∞), thus exhibiting the ***Retain Property*** described in Section 4. Although this theorem assumes that the objective function $\mathcal{F}(\mathbf{x})$ (or in the context of PINNs, the absolute residual values, $\mathcal{R}_\theta(\mathbf{x_r})$) is constant, this theorem is still valid when $\mathcal{R}_\theta(\mathbf{x_r})$ is gradually changing with the highest error regions (defined using our ϵ-maximal neighborhood) persisting over iterations. Under such conditions, the theorem states that the retained population $\mathcal{P}^r$ would always accumulate points from the ϵ-maximal neighborhood, thereby adaptively increasing their contribution to the overall PDE residual loss and eventually resulting in their minimization.

C.2. Release of Collocation Points from High PDE Residual Regions.

Our definition of the ***Release Property*** states that the distribution of collocation points should revert back to its original form by releasing the accumulated points in the high PDE residual regions once they are "*sufficiently minimized*". Let us define that for an arbitrary collocation point $\mathbf{x_r}^i \in \mathcal{P}$, "*sufficient minimization*" of the PDE is achieved if $\mathcal{R}_\theta(\mathbf{x_r}^i) \leq \mathbb{E}_{\mathbf{x_r} \in \mathcal{P}}[\mathcal{R}_\theta(\mathbf{x_r})] = \tau$ (where τ is the threshold used by R3 sampling). Then, by definition, such points will belong to the "non-retained population" and will be immediately replaced by the re-sampled population. Since we generate the re-sampled population $\mathcal{P}^s$ from a uniform distribution, these "*sufficiently minimized*" collocation points are replaced with a uniform density. Thus, R3 sampling satisfies the "*Release Property*".

D. Additional Details for Causal R3 Sampling

D.1. Preventing Abrupt Causal Gate Movement.

The shift parameter γ of the causal gate is updated every iteration using the following scheme: $\gamma_{i+1} = \gamma_i + \eta_g e^{-\epsilon \mathcal{L}_r^g(\theta)}$, where η_g is the learning rate that controls how fast the gate should propagate and ϵ denotes tolerance that controls how low the PDE loss needs to be before the gate shifts to the right, and i denotes the ith iteration. Typically, in our experiments we set the learning rate to 1e-3. Thus, for example, if the expectation of $e^{-\epsilon \mathcal{L}_r^g(\theta)}$ over 1000 iterations is 0.1, then γ would change by a value of 0.1 after 1000 iterations (since $\gamma_{i+N} \approx \gamma_i + \eta_g * N * \mathbb{E}[e^{-\epsilon \mathcal{L}_r^g(\theta)}]$). Additionally, note that, for a typical "tanh" causal gate, the operating range of γ values vary from -0.5 to 1.5. However, if the loss is very small ($\mathcal{L}_r^g(\theta) \to 0$), the magnitude of the update $e^{-\epsilon \mathcal{L}_r^g(\theta)} \to 1$, i.e., leads to an abrupt change in the causal gate. Thus, to prevent an abrupt gate movement due to large magnitude update, we employ a magnitude clipping scheme (similar to gradient clipping in conventional ML) as follows: $\gamma_{i+1} = \gamma_i + \eta_g \min(e^{-\epsilon \mathcal{L}_r^g(\theta)}, \Delta_{max})$, where Δ_{max} is the maximum allowed magnitude of update. Typically, for our experiments we keep $\Delta_{max} = 0.1$. Note, that Δ_{max} needs to be carefully chosen depending on the gate learning rate η_g.

D.2. Choice of Other Gate Functions.

The gate function g to enforce the principle of causality is not limited to the "tanh" gate presented in Section 4.2 of the main paper. Any arbitrary function can be used for a causal gate as long as it obeys the following criteria:

1. **Continuous Time Property**: The function g should be continuous in time, such that it can be evaluated at any arbitrary time t.
2. **Monotonic Property**: The value of gate g at time $t + \Delta t$ should be less than the value of the gate at time t, i.e., $g(t + \Delta t) \leq g(t)$. In other words, g should be a monotonically decreasing function,
3. **Shift Property**: The gate function should be parameterized using a shift parameter γ, such that $g_\gamma(t) < g_{\gamma+\delta}(t)$, where $\delta > 0$, i.e., by increasing the value of the shift parameter the gate value of any arbitrary time should increase.

An alternate choice of a causal gate is using a composition of ReLU and $\tanh$ functions: $g = ReLU(-\tanh(\alpha(t - \gamma))$ (as shown in Figure 11. We can see that by using ReLU, this alternate gate function provides a stricter thresholding of gate values to 0 after a cutoff value of time. The effect of this strict thresholding on the incorporation of causality in training PINNs can be studied in future analyses. In our current analysis, we simply used the $\tanh$ gate function for all our experiments.

E. Comparison of Baselines

In this Section, we compare the different baseline methods w.r.t. the motivating properties. See Table 2.

F. Computational Complexity Analysis of R3 Sampling vs its baselines

In this section, we aim to provide a comprehensive comparison of the computational complexity of R3 Sampling and its baselines. It is well-known that the cost of computing the PDE residuals using automatic-differentiation during training amounts is reasonably large, especially when the PDE contains higher order gradients that require repeated backward passes through the computational graphs used by standard deep learning packages like PyTorch/Tensorflow. Thus, in this section

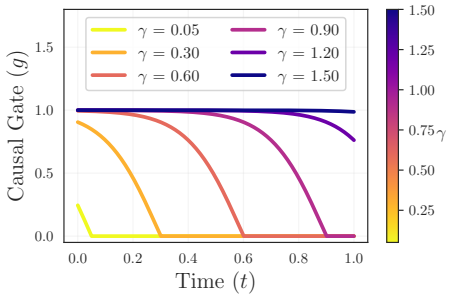


Figure 11. ReLU-tanh Causal Gate

Table 2. Table comparing baseline methods in terms of their ability to comply with the motivating properties of this work.

	Retain Property	**Release Property**	**Resample Property**	**Computational Efficiency**	**Causality**
RAR-G (Lu et al., 2021)	✓		✓		
RAD (Nabian et al., 2021)	✓	✓	✓		
RAR-D (Wu et al., 2023)	✓		✓		
$L-\infty$ (Wang et al., 2022a)	✓	✓		✓	
CausalPINN (Wang et al., 2022b)			✓	✓	✓
R3 (ours)	✓	✓	✓	✓	
Causal R3 (ours)	✓	✓	✓	✓	✓

we would mainly focus on comparing the number of PDE residual computations that each algorithm makes during training. We can quantify the effect of this difference on computational costs as follows. Let us first define the Notations that we are going to use for the analysis:

Notations for Computational Analysis

C	Computational Cost to evaluate the PDE residual on a single collocation point.
N	Total Number of Training Iterations
$\lvert\mathcal{P}\rvert$	Total Number of Collocation Points/Population Size (Also referred to as the initial set of collocation points for RAR based methods)
$\lvert\mathcal{P}_{dense}\rvert$	Auxiliary Set of dense points [For RAR-G,RAD, and RAR-D], where $\lvert\mathcal{P}_{dense}\rvert >> \lvert\mathcal{P}\rvert$
K	Resampling Period [For RAR-G,RAD, and RAR-D]
M	Number of additional collocation points added to the initial set

Next, we present the computational cost to run each method separately.

PINN/R3: The cost of computing the PDE residual at an arbitrary epoch i: $C_{R3}(i) = C|\mathcal{P}|$. Thus, the overall computational cost for N iterations is $C_{R3} = NC|\mathcal{P}|$.

RAR-G/RAD/RAR-D: For RAR-based methods, the initial set of collocation points $\mathcal{P}$ keeps growing as M new test points are added every K iterations. Thus, at an iteration i, the size of the collocation point set $|\mathcal{P}_i| = |\mathcal{P}_0| + m\lfloor i/K \rfloor = |\mathcal{P}| + M\lfloor i/K \rfloor$ (to simplify notations, let $|\mathcal{P}_0| = |\mathcal{P}|$, i.e., all of the methods start with the same number of initial collocation points).

Thus, the cost to compute the PDE residuals on this training set $\mathcal{P}$ is:

$$\begin{aligned} C_{RAR}^{train} &= KC|\mathcal{P}| + KC(|\mathcal{P}| + M) + KC(|\mathcal{P}| + 2M) + ... + KC(|\mathcal{P}| + M\left\lfloor \frac{N}{K} \right\rfloor) \\ &= KC|\mathcal{P}|\left(\left\lfloor \frac{N}{K} \right\rfloor + 1 + \frac{M}{2|\mathcal{P}|}\left\lfloor \frac{N}{K} \right\rfloor \left(\left\lfloor \frac{N}{K} \right\rfloor + 1 \right) \right) \end{aligned} \tag{15}$$

There is also an additional cost of evaluating the PDE residuals on the dense set C_{RAR}^{dense} to select these M points from high PDE residual region.

$$C_{RAR}^{dense} = C|\mathcal{P}_{dense}|\left\lfloor \frac{N}{K} \right\rfloor \tag{16}$$

Therefore, the overall cost for RAR-based methods is: $C_{RAR} = C_{RAR}^{dense} + C_{RAR}^{train}$

Comparing the cost between R3 and RAR: Assuming that the total number of epochs N is divisible by the resampling period K, which is true for most practical scenarios.

$$\begin{aligned} C_{RAR} &= NC|\mathcal{P}| + KC|\mathcal{P}| + \frac{MNC}{2}\left(\frac{N}{K} + 1\right) + C|\mathcal{P}_{dense}|\frac{N}{K} \\ C_{RAR} &= C_{R3} + KC|\mathcal{P}| + \frac{MNC}{2}\left(\frac{N}{K} + 1\right) + C|\mathcal{P}_{dense}|\frac{N}{K} \\ C_{RAR} - C_{R3} &= KC|\mathcal{P}| + \frac{MNC}{2}\left(\frac{N}{K} + 1\right) + C|\mathcal{P}_{dense}|\frac{N}{K} \end{aligned} \tag{17}$$

Thus, we can see that the difference in the computational cost can quickly grow depending on the choice of RAR setting. Also note that, since $|\mathcal{P}_{dense}| >> |\mathcal{P}|$, the additional cost of $C|\mathcal{P}_{dense}|\frac{N}{K}$ is significant, especially if we want to re-sample/re-evaluate the adaptive sampling frequently (i.e., for small values of K).

G. Details of Partial Differential Equations Used in this Work

G.1. Convection Equation

We considered a 1D-convection equation that is commonly used to model transport phenomenon, described as follows:

$$\frac{\partial u}{\partial t} + \beta \frac{\partial u}{\partial x} = 0, \;\; x \in [0, 2\pi], t \in [0, 1] \tag{18}$$

$$u(x, 0) = h(x) \tag{19}$$

$$u(0, t) = u(2\pi, t) \tag{20}$$

where β is the convection coefficient and h(x) is the initial condition. For our case studies, we used a constant setting of $h(x) = \sin(x)$ with periodic boundary conditions in all our experiments, while varying the value of β in different case studies.

G.2. Allen-Cahn Equation

We considered a 1D - Allen Cahn equation that is used to describe the process of phase-separation in multi-component alloy systems as follows:

$$\frac{\partial u}{\partial t} - 0.0001\frac{\partial^2 u}{\partial x^2} + 5u^3 - 5u = 0, \;\; x \in [-1, 1], t \in [0, 1] \tag{21}$$

$$u(x, 0) = x^2 \cos(\pi x) \tag{22}$$

$$u(t, -1) = u(t, 1) \tag{23}$$

$$\frac{\partial u}{\partial t}\Big|_{x=-1} = \frac{\partial u}{\partial t}\Big|_{x=1} \tag{24}$$

G.3. Eikonal Equation

We formulate the Eiknonal equation for signed distance function (SDF) calculation as:

$$|\nabla u| = 1, \qquad x, t \in [-1, 1] \tag{25}$$

$$u(x_s) = 0, \qquad x_s \in \mathcal{S} \tag{26}$$

$$u(x, -1), u(x, 1), u(-1, y), u(1, y) > 0 \tag{27}$$

where $\mathcal{S}$ is zero contour set of the SDF. In training PINN, we use the zero contour constraint as initial condition loss and positive boundary constraint as boundary loss (see Table 3 for details of loss balancing).

G.4. Kuramoto–Sivashinsky Equation

We use 1-D Kuramoto–Sivashinsky equation from CausalPINN (Wang et al., 2022b):

$$\frac{\partial u}{\partial t} + \alpha u \frac{\partial u}{\partial x} + \beta \frac{\partial^2 u}{\partial x^2} + \gamma \frac{\partial^4 u}{\partial x^4} = 0, \tag{28}$$

subject to periodic boundary conditions and an initial condition

$$u(0, x) = u_0(x) \tag{29}$$

The parameter α, β, γ controls the dynamical behavior of the equation. We use the same configurations as the CausalPINN: $\alpha = 5, \beta = 0.5, \gamma = 0.005$ for regular settings, and $\alpha = 100/16, \beta = 100/162, \gamma = 100/164$ for chaotic behaviors.

H. Details on Skewness and Kurtosis Metrics

Skewness and kurtosis are two basic metrics used in statistics to characterize the properties of a distribution of values $\{Y_i\}_{i=1}^N$. A high value of Skewness indicates lack of symmetry in the distribution, i.e., the distribution of values to the left and to the right of the center point of the distribution are not identical. On the other hand, a high value of Kurtosis indicates the presence of a heavy-tail, i.e., there are more values farther away from the center of the distribution relative to a Normal distribution. In our implementation using *scipy*, we used the adjusted Fisher-Pearson coefficient of skewness and Fisher's definition of kurtosis, as defined below.

Skewness: For univariate data $Y_1, Y_2, ..., Y_N$, the formula of skewness is

$$\text{skewness} = \frac{\sqrt{N(N-1)}}{N-2} \times \frac{\sum_{i=1}^N (Y_i - \bar{Y})^3 / N}{s^3}, \tag{30}$$

where $\bar{Y}$ is the sample mean of the distribution and s is the standard deviation. For any symmetric distribution (e.g., Normal distribution), the skewness is equal to zero. A positive value of skewness means there are more points to the right of the center point of the distribution than there are to the left. Similarly, a negative value of skewness means there are more points to the left of the center point than there are to the right. In our use-case, a large positive value of skewness of the PDE residuals indicates that there are some asymmetrically high PDE residual values to the right.

Kurtosis: Kurtosis is the fourth central moment divided by the square of the variance after subtracting 3, defined as follows:

$$\text{kurtosis} = \frac{\sum_{i=1}^N (Y_i - \bar{Y})^4 / N}{s^4} - 3 \tag{31}$$

For a Normal distribution, Kurtosis is equal to 0. A positive value of Kurtosis indicates that there are more values in the tails of the distribution than what is expected from a Normal distribution. On the other hand, a negative value of Kurtosis indicates that there are lesser values in the tails of the distribution relative to a Normal distribution. In our use-case, a large positive value of Kurtosis of the PDE residuals indicates that there are some high PDE residual values occurring in very narrow regions of the spatio-temporal domain, that are being picked up as the heavy-tails of the distribution.

I. Hyper-parameter Settings and Implementation Details

The hyper-parameter settings for the different baseline methods for every benchmark PDE are provided in Table 3. Note that we used the same network architecture and other hyper-parameter settings across all baseline method implementations for the same PDE. In this table, the column on 'r/ic/bc' represents the setting of the $\lambda_r, \lambda_{ic}, \lambda_{bc}$ hyper-parameters that are used to weight the different loss terms in the overall learning objective of PINNs. Table 3 also lists the type of Optimizer, learning rate (lr), and learning rate scheduler (lr.scheduler) used across all baselines for every PDE. For the Eikonal equation, we used the same modified multi-layer perceptron (MLP) architecture as the one proposed in (Wang et al., 2021). Additionally, for the Causal R3 Sampling method, we used the following hyper-parameter settings across all PDEs: $\alpha = 5$, learning rate of the gate $\eta_g = 1e-3$, tolerance $\epsilon = 20$, initial value of $\beta = -0.5$, and $\Delta_{max} = 0.1$. The number of iterations (and the corresponding PDE coefficients for the Convection Equation) are provided in Section 5 of the main paper.

Table 3. Hyper-parameter settings for different baseline methods for every benchmark PDE

PDE	Method	Architecture	Periodic Encoding	r/ic/bc	Optimizer/ lr	lr. scheduler	RAR Hyperparams k/m/N/$\lvert\mathcal{S}\rvert$
Convection	PINN, cPINN, R3, Causal R3	50 x 4 (MLP)	No	1/100/100	Adam/ 1e-3	StepLR rate=0.9 steps=5000	N/A
	Curr. Reg.	50 x 4 (MLP)	No	1/1/1	Adam/ 1e-4	No	N/A
	RAR-G RAD RAR-D	50 x 4 (MLP)	No	1/100/100	Adam/ 1e-3	StepLR rate=0.9 steps=5000	-/1/100/100000 1/1/100/100000 1/1/100/100000
Allen Cahn	PINN, cPINN, R3, Causal R3	128x4 (MLP)	Yes	1/100/100	Adam/ 1e-3	StepLR rate=0.9 steps=5000	N/A
	RAR-G RAD RAR-D	128x4 (MLP)	Yes	1/100/100	Adam/ 1e-3	StepLR rate=0.9 steps=5000	-/1/100/100000 1/1/100/100000 1/1/100/100000
Eikonal	PINN, R3	128x4 (Wang et al., 2021) (modified MLP)	No	1/500/10	Adam/ 1e-3	StepLR rate=0.9 steps=5000	N/A

Hardware Implementation Details: We trained each of our models on one Nvidia Titan RTX 24GB GPU.

J. Additional Discussion of Results

J.1. Visualizing Propagation Failure for Different Settings of β

In Figure 2 of the main paper, we demonstrated the phenomenon of propagation failure for convection equation with $\beta = 50$, which was characterized by large values of Skewness and Kurtosis in the PDE residual fields for a large number of iterations (or epochs), and a simultaneous stagnation in the relative error values even though the mean PDE residual kept on decreasing. Here, in Figure 12, we show that the same phenomenon can be observed for other large values of $\beta > 10$, namely, $\beta = 30, 50, 70$. We can see that the relative errors for all these three cases remains high even though the PDE residual loss keeps on decreasing with iterations. We can also see that the absolute values of skewness and kurtosis increase as we increase β, indicating higher risks of propagation failure. In fact, for $\beta = 30$, we can even see that the epoch that marks an abrupt increase in skewness and kurtosis (around 50K iterations) also shows a sudden increase in the relative error

at the same epoch, highlighting the connection between imbalanced PDE residuals and the phenomenon of propagation failure.

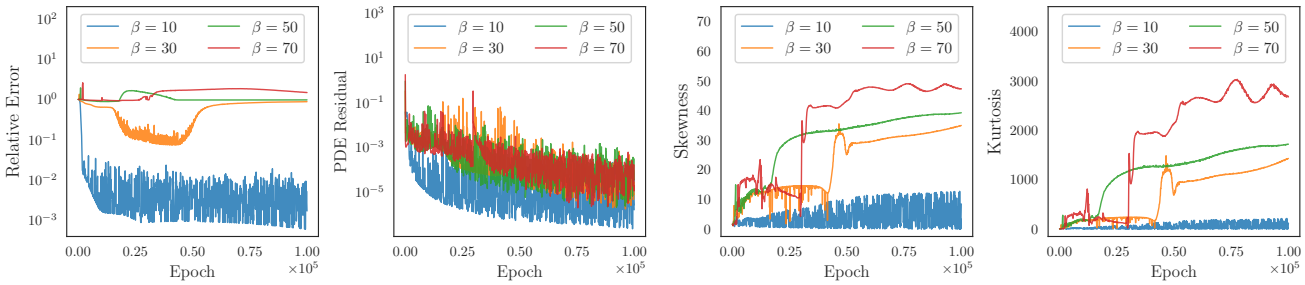


Figure 12. Demonstration of Propagation Failure for Different Settings of β ($\beta = 10, 30, 50, 70$)

J.2. Visualizing Propagation Failure for Kuramoto-Shivashinksy Equation

Figure 13 is similar to Figure 2, and demonstrates that during the training of conventional PINNs although the PDE residuals steadily decrease with iterations, the skewness and kurtosis quickly become very large. This behavior suggests the presence of small regions with very high PDE residuals, which we refer to as the "propagation failure" of PINNs. Figure 14 shows the Ground-truth solution, the predicted PDE field and the PDE Residuals after 100k iterations. It can be seen that there exists small regions or sharp boundaries of high error regions which we hypothesise is the effect for propagation failure.

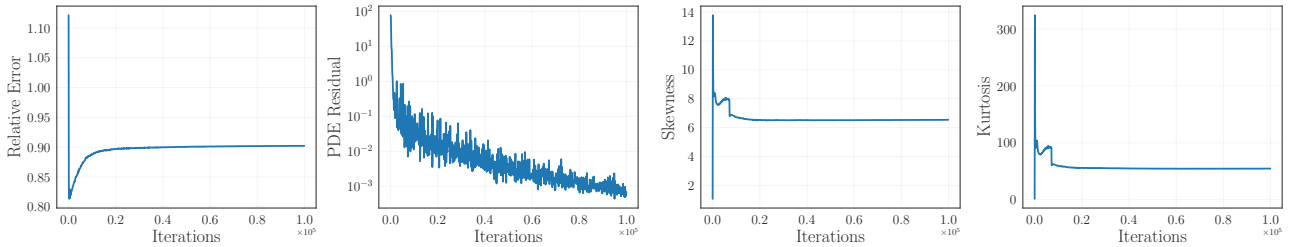


Figure 13. Demonstration of Propagation Failure for Kuramoto-Shivashinksky Equation.

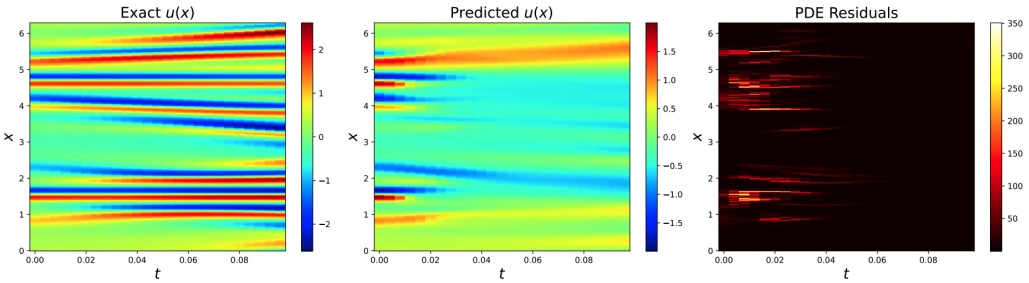


Figure 14. Visualization of High PDE Residuals due to Propagation Failure for Kuramoto-Shivashinksky Equation (Chaotic).

J.3. Additional Results for Sample Efficiency of R3 Sampling

We provide the results for the sample efficiency results for the Convection Equation and Allen Cahn Equations in Table 4 and 5 respectively, averaged across 3 random runs.

J.4. Convergence Speed of R3 Sampling

Figure 15 shows that Causal R3 is able to converge faster to low error solutions than all other baseline methods for both convection and Allen Cahn equations. This shows the importance of respecting causality along with focusing on high residual

Table 4. Relative $\mathcal{L}_2$ errors (in %) of comparative methods for Convection Equation with $\beta = 50$ for different number of collocation points N_s averaged across 3 random runs.

Method	N_s **= 100**	N_s **= 500**	N_s **= 1000**	N_s **= 2000**	N_s **= 5000**	N_s **= 10000**	N_s **= 20000**
PINN (fixed)	103.0 ± 0.97	121.0 ± 13.1	108.0 ± 11.5	114.0 ± 22.8	135.0 ± 15.4	7.04 ± 7.81	3.35 ± 1.91
PINN (dynamic)	66.2 ± 4.87	51.5 ± 26.8	52.2 ± 17.8	63.0 ± 7.22	4.93 ± 1.59	5.90 ± 3.17	3.14 ± 1.76
CausalPINN	89.0 ± 9.49	66.9 ± 11.2	72.5 ± 3.82	43.4 ± 37.1	44.9 ± 36.3	61.5 ± 13.8	61.8 ± 6.77
RAD	83.4 ± 11.3	28.9 ± 34.9	67.1 ± 1.57	51.0 ± 29.9	23.5 ± 32.1	23.5 ± 32.1	23.5 ± 32.1
R3	4.27 ± 2.11	2.22 ± 1.26	1.47 ± 0.45	1.93 ± 0.66	3.27 ± 2.70	2.77 ± 1.52	1.96 ± 0.33
CausalR3	5.54 ± 3.52	4.59 ± 2.87	1.14 ± 0.11	5.69 ± 4.10	6.51 ± 5.50	4.11 ± 2.86	1.52 ± 9.69

Table 5. Relative $\mathcal{L}_2$ errors (in %) of comparative methods for Allen Cahn Equation for different number of collocation points N_s averaged across 3 random runs.

Method	N_s **= 100**	N_s **= 500**	N_s **= 1000**	N_s **= 2000**	N_s **= 5000**	N_s **= 10000**	N_s **= 20000**
PINN (fixed)	70.4 ± 5.49	70.1 ± 0.28	62.4 ± 1.80	56.0 ± 3.70	76.4 ± 21.9	51.2 ± 0.02	5.11 ± 4.32
PINN (dynamic)	2.88 ± 3.67	2.93 ± 2.22	3.69 ± 5.00	0.78 ± 0.10	0.83 ± 0.16	0.82 ± 0.04	0.78 ± 0.02
CausalPINN	75.4 ± 9.43	22.9 ± 6.85	23.6 ± 21.2	5.50 ± 3.32	2.53 ± 1.23	1.90 ± 1.68	0.79 ± 0.10
RAD	50.9 ± 0.13	21.3 ± 23.2	5.10 ± 4.23	1.08 ± 0.40	1.92 ± 2.06	0.82 ± 0.11	0.73 ± 0.02
R3	0.86 ± 0.22	0.69 ± 0.03	0.81 ± 0.16	0.75 ± 0.07	0.69 ± 0.01	0.81 ± 0.22	0.81 ± 0.10
CausalR3	0.70 ± 0.01	0.69 ± 0.04	0.71 ± 0.01	0.69 ± 0.02	0.71 ± 0.01	0.71 ± 0.03	0.72 ± 0.02

regions to ensure fast propagation of correct solution from initial/boundary points to interior points. While cPINN-fixed and PINN-dynamic do not converge for convection ($N_r = 1K$), we can see that they both converge to lower errors compared to PINN-fixed for Allen Cahn and for convection when N_r is large ($20K$).

J.5. Sensitivity of RAR-based Methods

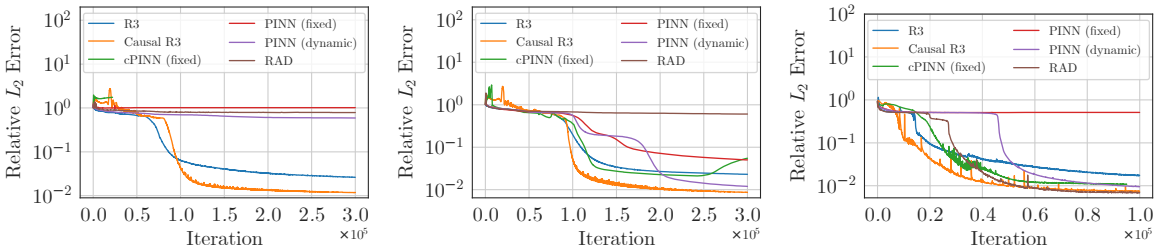
Figures 16 shows the sensitivity of two RAR-based methods: RAR-G and RAR-D respectively on different values of the resampling period K and the number of collocation points m added from the dense set $\mathcal{P}_{dense}$. Note that although the size of the initial set of collocation points $|\mathcal{P}|$ was same for each of these experiments, the final size of the collocation points vary depending on the choice of K (the final size of the collocation points increases as k decreases) and m (the final size of the collocation points increases with m). We essentially observe that adding more collocation points almost always improves the performance of these RAR-based methods.

J.6. Visualizing the Evolution of Collocation Points in R3 sampling

Figure 17 shows the evolution of collocation points and PDE residual maps of R3 sampling as we progress in training iterations for the convection equation with $\beta = 50$. We can see that the retained population of R3 sampling at every iteration (shown in red) selectively focuses on high PDE residual regions, while the re-sampled population (shown in blue) are generated from a uniform distribution. By increasing the contribution of high residual regions in the computation of the PDE loss, we can see that R3 sampling is able to reduce the PDE loss over iterations without admitting high imbalance, thus mitigating the propagation failure mode, in contrast to conventional PINNs.

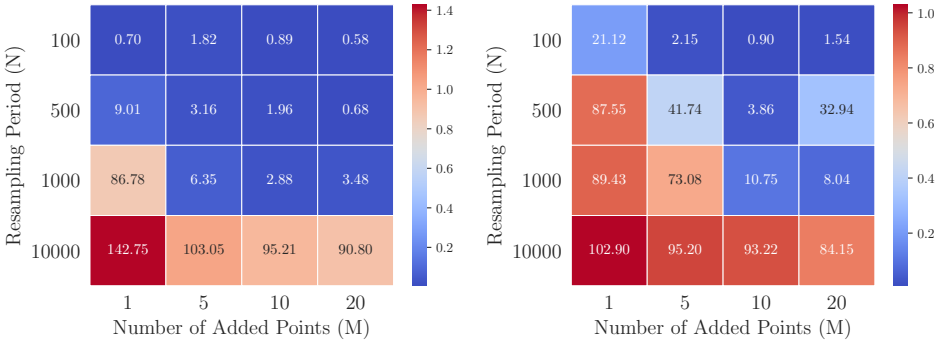
J.7. Visualizing the Evolution of Collocation Points in Causal R3

Figure 18 shows the evolution of collocation points and PDE residuals of Causal R3, along with the dynamics of the Causal Gate function. We can see that the retained population at every iteration (shown in red) strictly adheres to the principle of causality such that the collocation points are sampled from later times only when the PDE residuals at earlier times have been minimized. This is also reflected in the movement of the causal gate function where the gate values are close to 1 for only small portions of time domain at intermediate epochs. At 90K iterations, we can see that the causal gate values are close to 1 for all values of time, indicating that the entire time domain is now revealed for training PINN to converge to the



(a) Convection ($\beta = 50$, $N_r = 1$k) (b) Convection ($\beta = 50$, $N_r = 20$k) (c) Allen Cahn ($N_r = 10$k)

Figure 15. Comparing convergence speeds of baselines w.r.t. R3 sampling and Causal R3.



(a) RAR-G (b) RAR-D

Figure 16. Sensitivity of RAR-based Methods on different values of K and M (for a fixed $|\mathcal{P}_{dense}| = 100k$ on Convection Equation with $\beta = 30$.) The numbers in the heatmap denote the % Relative $\mathcal{L}_2$ Error.

correct solution.

J.8. Visualizing the Dynamics of Collocation Points for RAR-Based Methods

Figures 19, 20, annd 21 shows the evolution of collocation points and PDE residuals of RAR-G, RAD, and RAR-D, respectively. We can see that all three RAR-based methods are failing to converge to the correct solution even after 100K iterations, demonstrating their inabillity to mitigate propagation failures.

J.9. Kuramoto-Sivashinsky (KS) Equations

We used three additional experiments on the Kuramoto-Sivashinsky (KS) Equations (one for a regular relatively simple case and the remaining two exhibiting chaotic behavior). Please note that these equations are particularly more complex, especially the chaotic cases where a small change in the state of the solution can result in very large errors downstream in time. Thus, for chaotic domains, the successful propagation of solution from the initial and boundary conditions is critical to guarantee convergence. We would also like to highlight that the computational cost of these experiments are significantly higher. We used the exact same hyper-parameter settings as those provided in CPINN except the sample size, which was varied from 128 to 2048 in the KS-regular case, and the number of training iterations, which was kept as 300k in our proposed approaches while CPINN was allowed to use about 1 M maximum number of iterations with early stopping. Our method on average takes 50-60% less time than CPINN because of the significantly smaller number of iterations.

Table 6 compares the performance of CPINN, R3, and Causal R3 on the three KS-Equations cases as was used in the original CPINN paper. Please note that for these experiments, we used the exact same hyper-parameter settings as the original CPINN, thus a large number of collocation points were used (2048 for the regular case and 8192 for the two chaotic regimes for each time-window). We can observe that on the regular case, R3 sampling performs similarly to CPINN, while Causal R3 is significantly better than both. However, in the first chaotic case, CPINN is slightly better than both R3 sampling and Causal R3. Finally, in the much more chaotic regime for the KS-Equation (extended case), we find that all of the methods struggle to obtain a high fidelity solution of the field. However, R3 sampling and Causal R3 are somewhat better

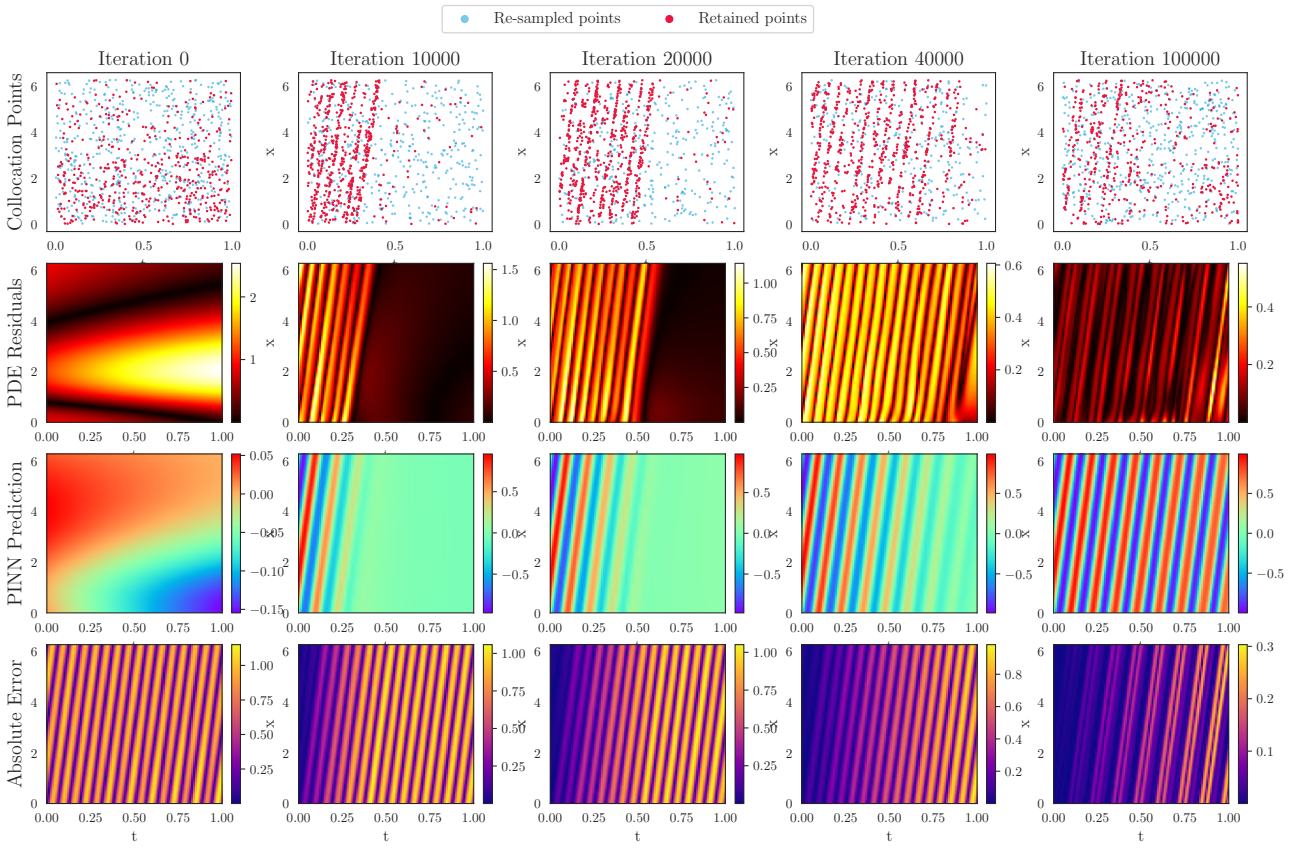


Figure 17. Demonstrating the propagation of information from the initial/boundary points to the interior points for R3 Sampling on Convection Equation($\beta = 50$)

than CPINNs. Hence, we can comment that R3 sampling and Causal R3 have comparable performance to CPINN on their benchmark settings. Additional visualizations of the solutions of comparative methods on the three KS equations cases are provided in Figures 22, 23, and 24.

J.10. Additional Discussion and Visualization for the Eikonal Equation

We chose to solve 2D Eikonal Equations for complex arbitrary surface geometries as they represent particularly hard PDE problems that are susceptible to PINN failure modes. In these problems, we are given the zero contours of the equation on the boundaries (representing the outline of the 2D object), which can take arbitrary shapes. The goal is to correctly propagate the boundary conditions to obtain the unique target solution where the interior is negative and the exterior is positive. Here, any small error in propagation from the boundaries can lead to cascading errors such that a large segment of the predicted field can have opposite signs compared to the ground-truth, even though their PDE residuals are close to 0. Since R3 sampling is explicitly designed to break propagation barriers and thus enable easy transmission of the solution from the boundary to the interior/exterior points, we can see that it shows significantly better performance. On the other hand, PINN (fixed) and PINN (dynamic) struggle to converge to the correct solution especially for complex geometries (e.g., the 'gear') because of the inherent challenge in sampling an adequate number of points from arbitrary shaped object boundaries exhibiting highly imbalanced residuals.

In Figure 25, we show the evolution of the solutions of comparative methods for the 'gear' case over iterations. We can see that R3 sampling is able to resolve the high residual regions better than the baselines, and thus encounter less incorrect "sign flips" compared to the ground-truth, even in the early iterations of PINN training.

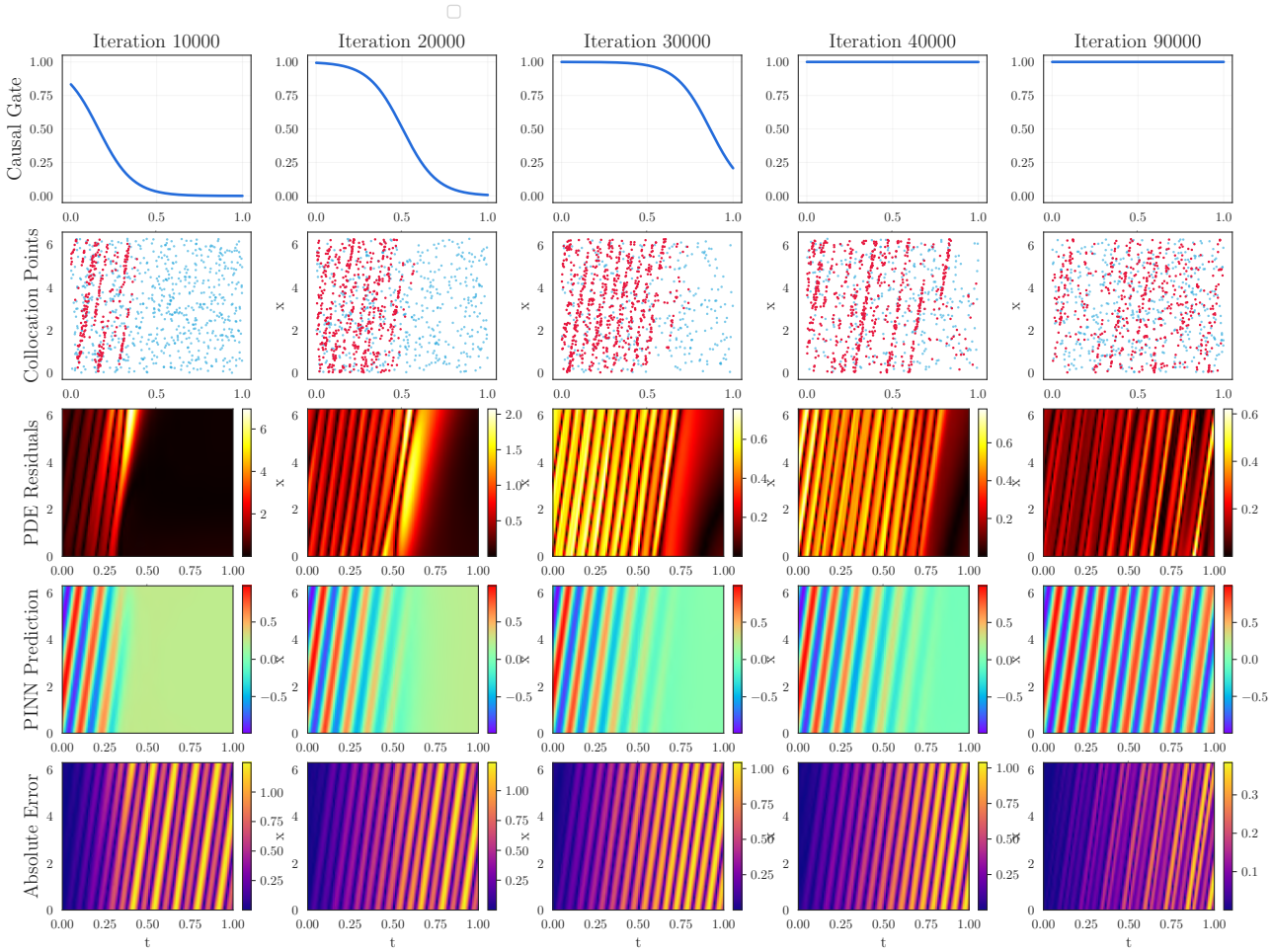


Figure 18. Demonstrating the propagation of information from the initial/boundary points to the interior points for Causal R3 on Convection Equation($\beta = 50$)

K. Optimization Characteristics of R3 Sampling on Test Optimization Functions

In this section, we demonstrate the ability of our proposed R3 Sampling Algorithm to find global minimas on various test optimization functions. We will also provide other characterizations of our proposed R3 Sampling algorithm.

K.1. Auckley Function

The two-dimensional form of the Auckley function has multiple local maximas in the near-flat region of the function and one large peak at the center.

$$f(x) = a\exp\left(-b\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right) + \exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(cx_i)\right) + a + \exp(1) \tag{32}$$

K.2. Bohachevsky Function

The two-dimensional form of the Bohachevsky function is a bowl shaped function having one global maxima.

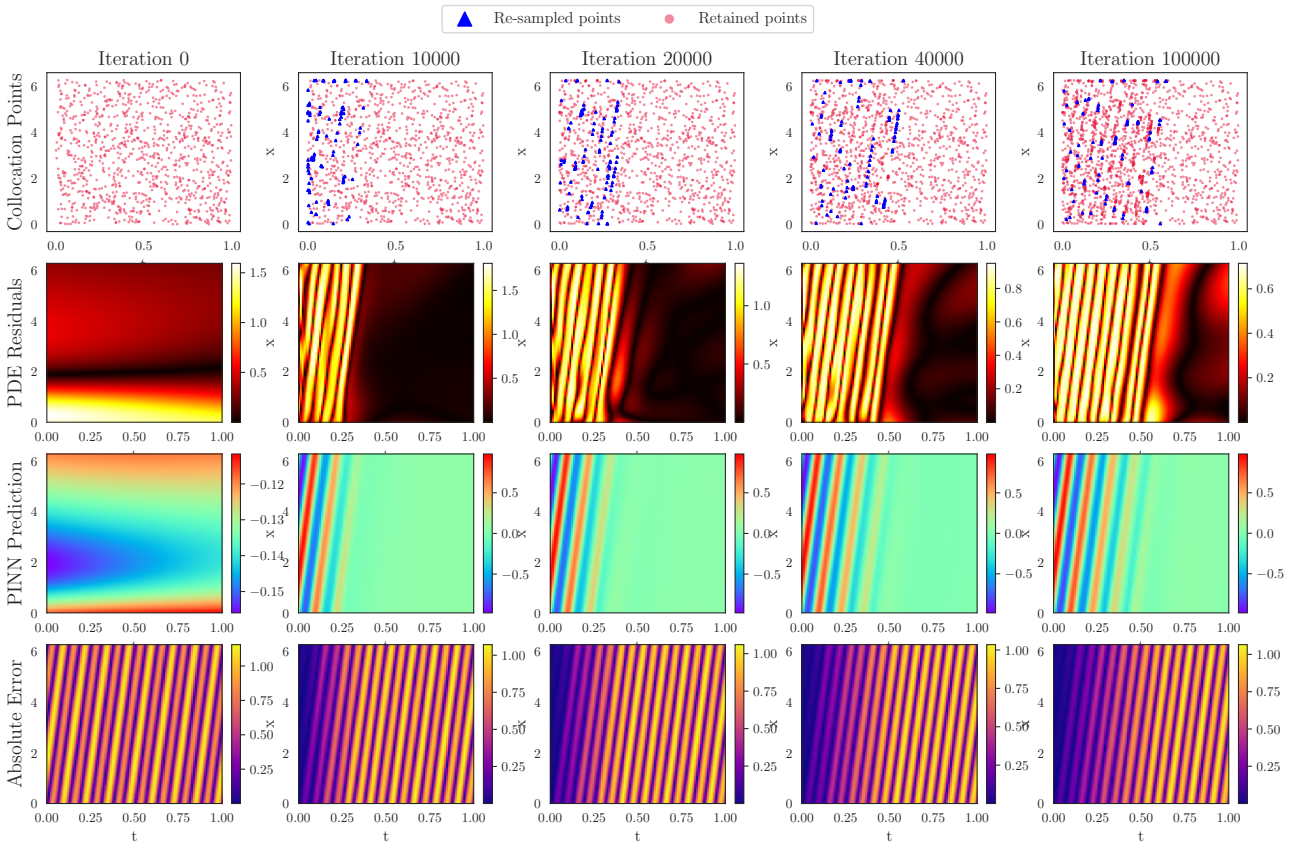


Figure 19. Demonstrating the dynamic changes in the collocation points for RAR-G on Convection Equation($\beta = 50$)

$$f(x, y) = -x^2 - 2y^2 + 0.3\cos(3\pi x) + 0.4cos(4\pi y) - 0.7 \tag{33}$$

K.3. Drop-Wave Function

The two-dimensional form of the Drop-Wave function which is multimodal and highly complex.

$$f(x, y) = \frac{1 + \cos(12\sqrt{x^2 + y^2})}{0.5(x^2 + y^2) + 2} \tag{34}$$

K.4. Egg-Holder Function

The two-dimensional form of the Egg-Holder function is highly complex function that is difficult to optimize because of the presence of multiple local maximas.

$$f(x, y) = (y + 47)\sin(\sqrt{|y + \frac{x}{2} + 47|}) + x\sin(\sqrt{|x - (y + 47)|}) \tag{35}$$

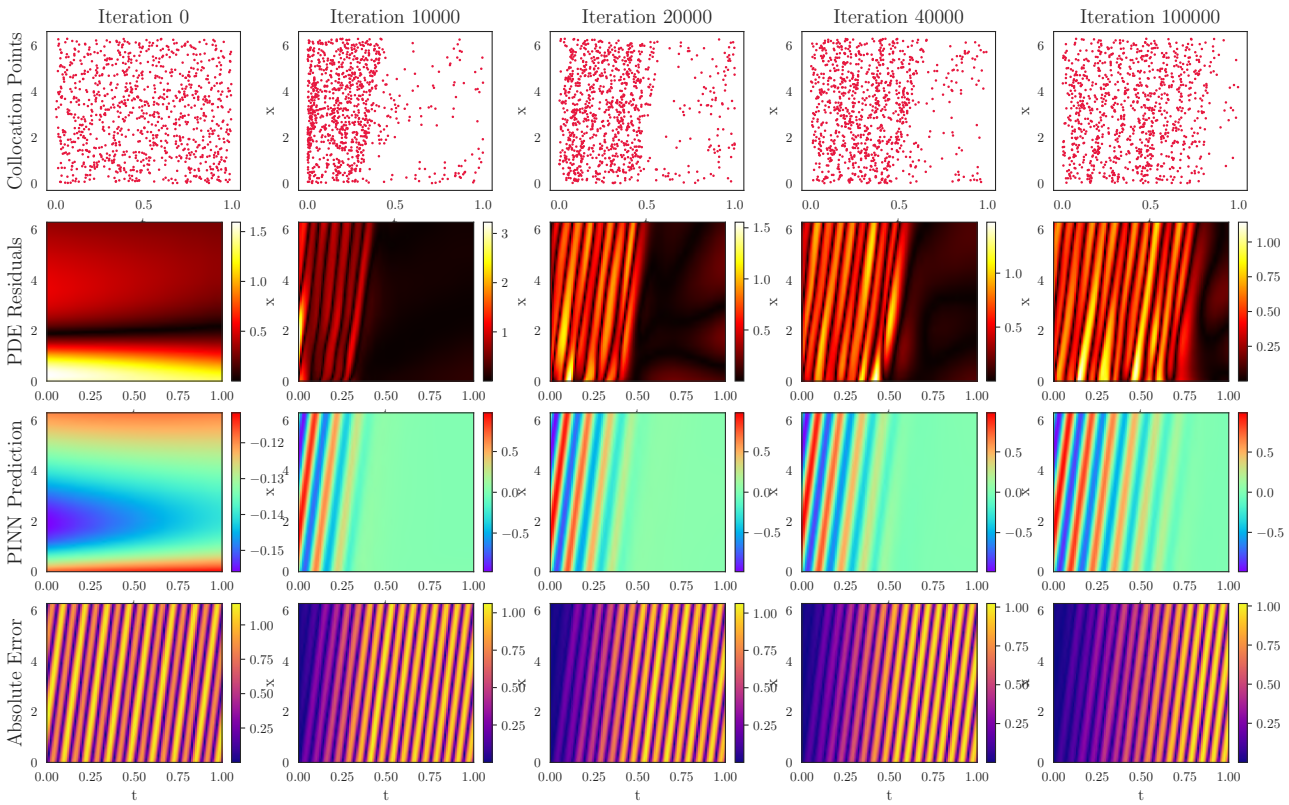


Figure 20. Demonstrating the dynamic changes in the collocation points for RAD on Convection Equation($\beta = 50$)

K.5. Holder-Table Function

The two-dimensional form of the Holder-Table function has many local maximas, but has 4 global maximas at the four corners.

$$f(x, y) = |\sin(x)\cos(y)\exp\left(|1 - \frac{\sqrt{x^2 + y^2}}{\pi}|\right)| \tag{36}$$

K.6. Bukin Function

The two-dimensional form of the Bukin function has many local maximas, all of which lie on a ridge.

$$f(x, y) = -100\sqrt{|y - 0.01x^2|} - 0.01|x + 10| \tag{37}$$

K.7. Michalewicz Function

The two-dimensional form of the Michalewicz function has multiple ridges and valleys which are very steep.

$$f(x) = \sum_{i=1}^{d} \sin(x_i)\sin^{2m}\left(\frac{ix_i^2}{\pi}\right) \tag{38}$$

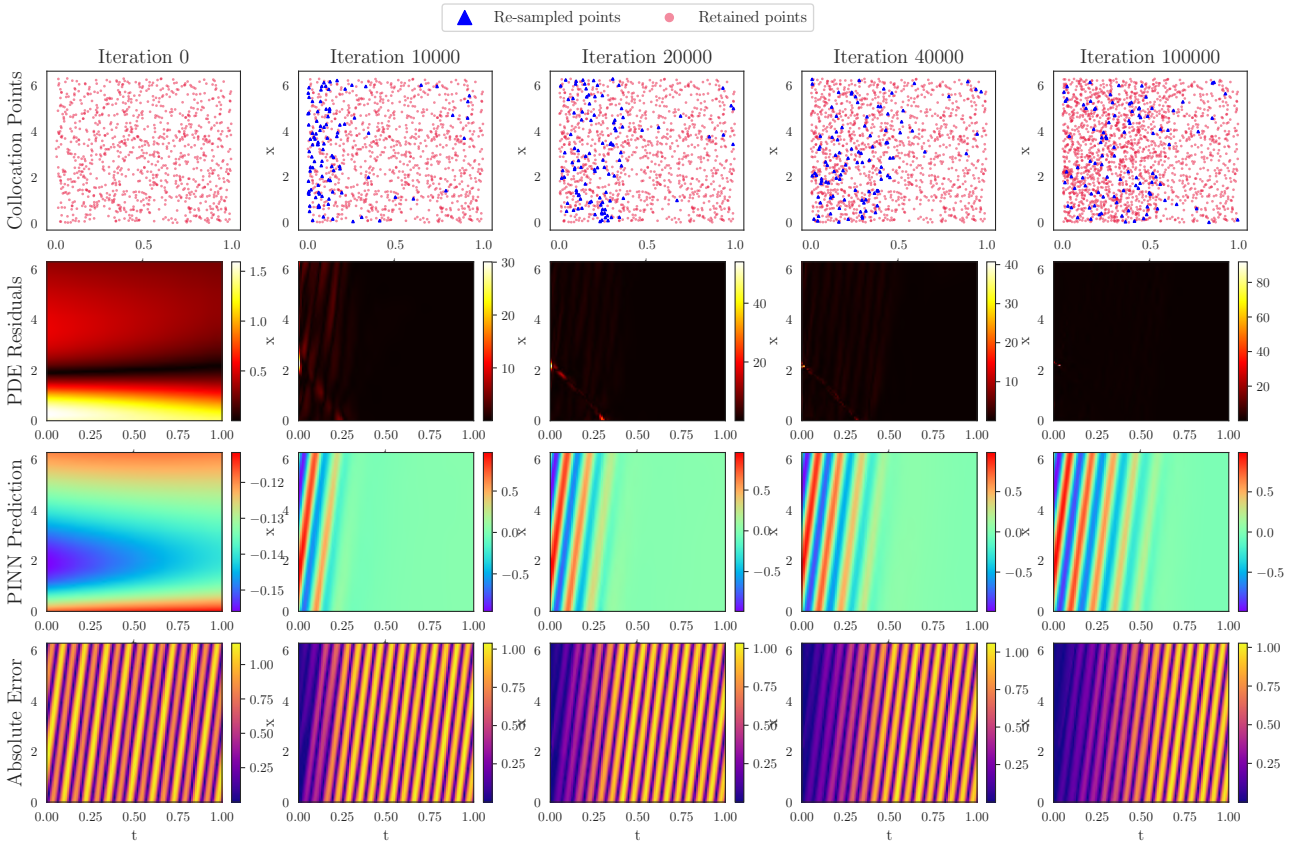


Figure 21. Demonstrating the dynamic changes in the collocation points for RAR-D on Convection Equation($\beta = 50$)

Table 6. Relative $\mathcal{L}_2$ errors (in %) of CausalPINN, R3 sampling and Causal R3 over the three different KS-Equation Benchmarks (Wang et al., 2022b).

	CausalPINN	R3 (ours)	Causal R3 (ours)
Regular	2.120%	3.740%	**0.761**%
Chaotic	**3.272**%	6.924%	7.630%
Chaotic - Extended	52.66%	**29.26**%	33.50%

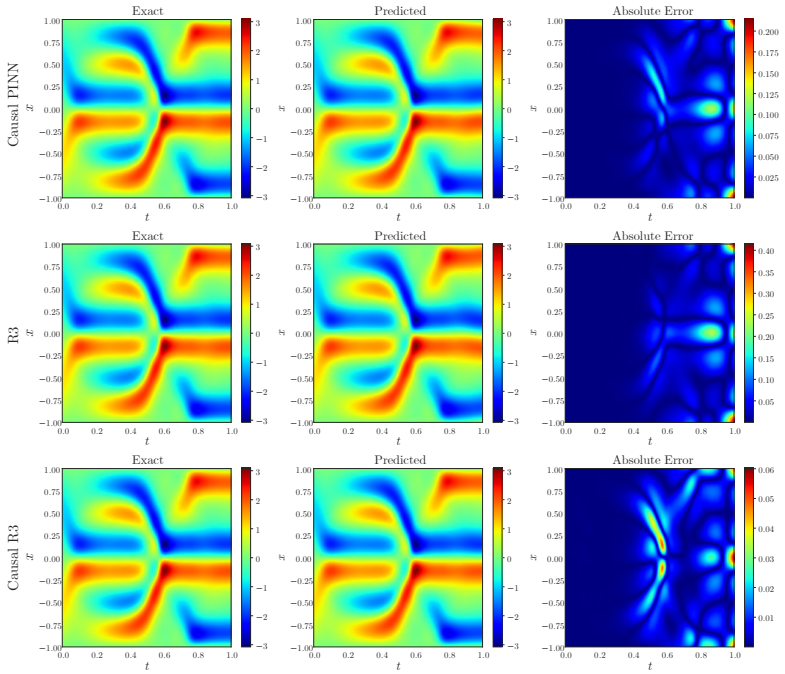


Figure 22. Visualization of the R3 sampling, Causal R3 and CausalPINN on KS-Equation (Regular Case).

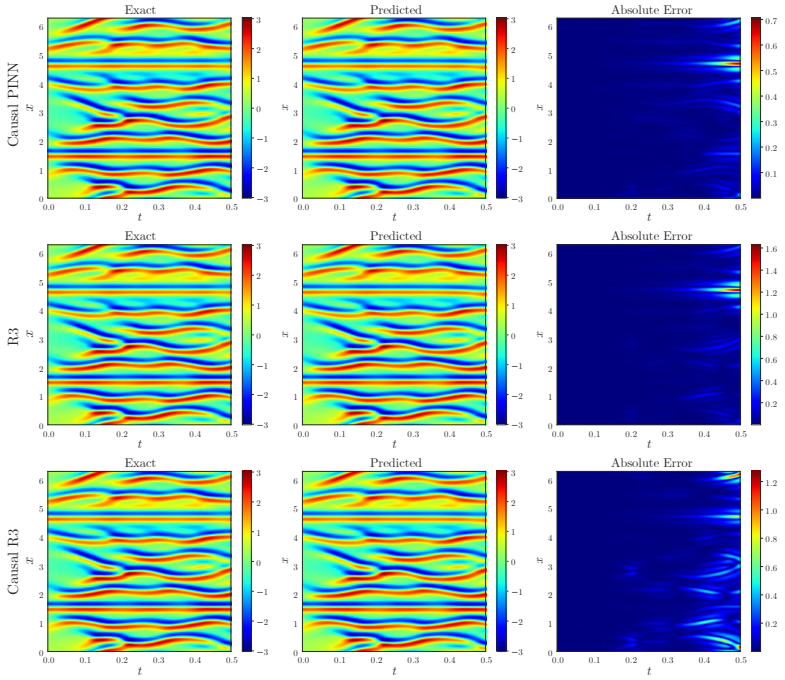


Figure 23. Visualization of the R3, Causal R3 and CausalPINN on KS-Equation (Chaotic Case).

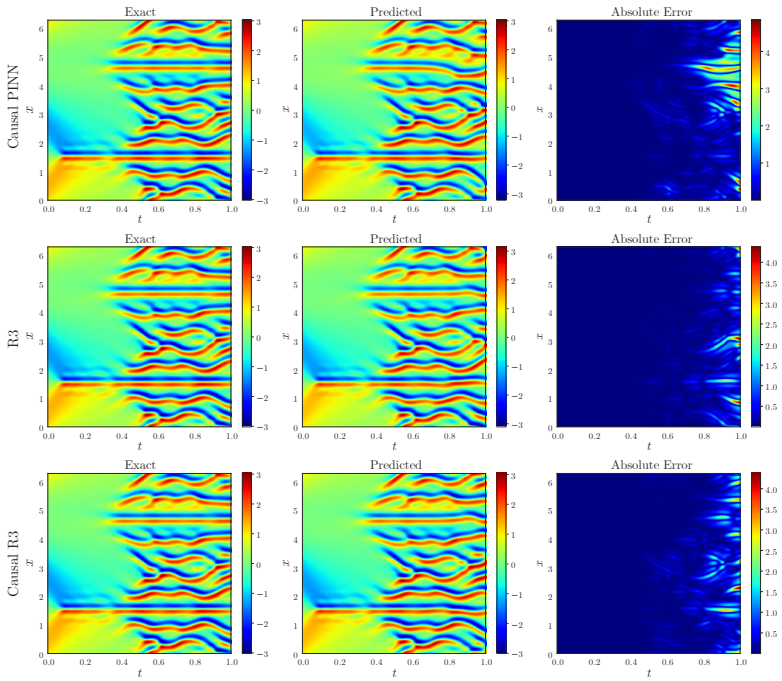


Figure 24. Visualization of the R3, Causal R3 and CausalPINN on KS-Equation (Extended Chaotic Case).

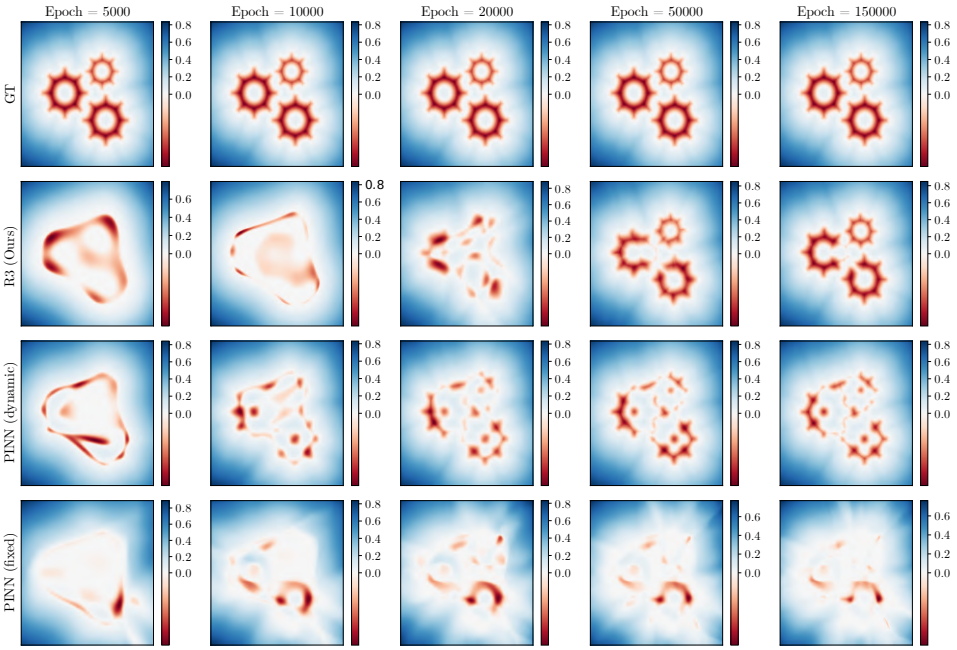


Figure 25. The predicted solutions of Eikonal equation (Figure 10) at different iterations during training.

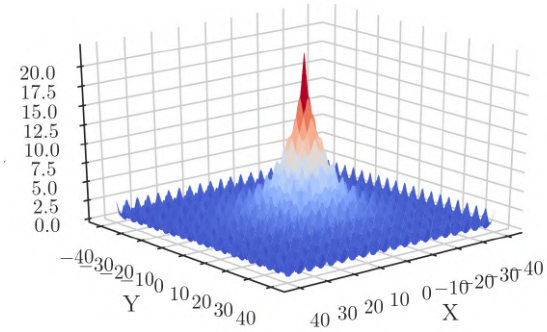


Figure 26. Surface Plot of the 2-D Auckley Function

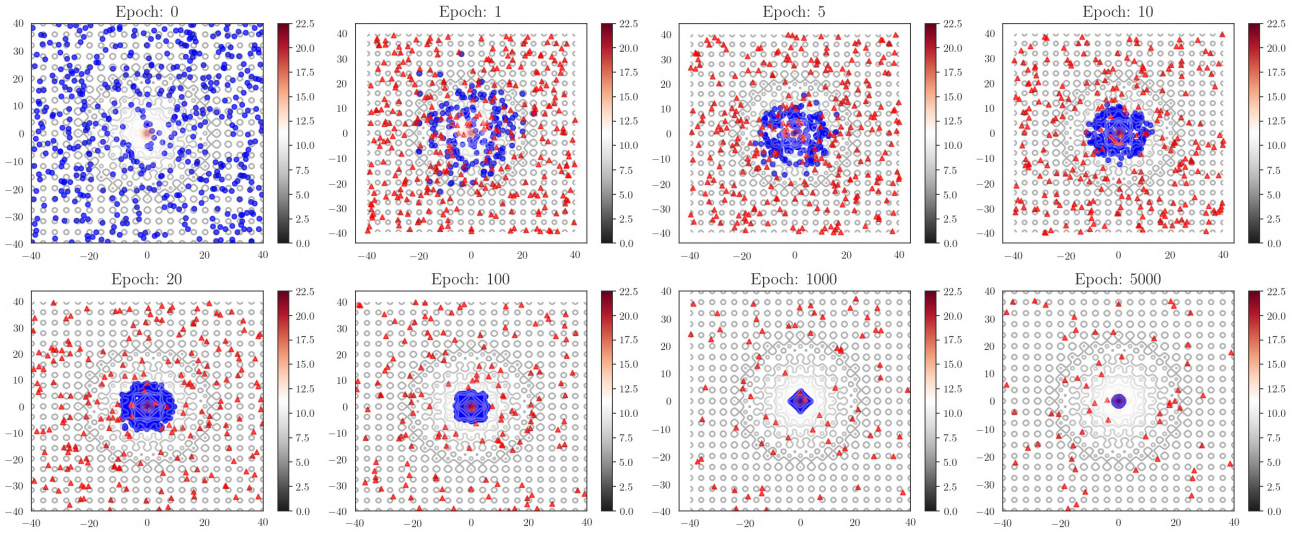


Figure 27. Demonstrating the evolution of the randomly initialized points while optimizing the Auckley function. The red triangles represent the re-sampled population at that epoch, and the blue dots represent the retained population at that epoch. The contour function of the objective function is shown in the background.

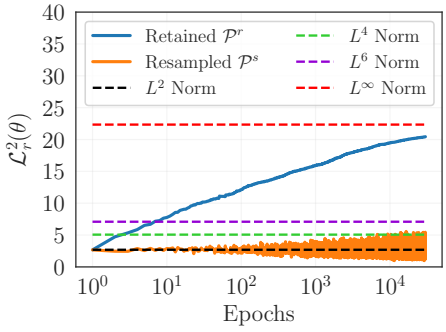


Figure 28. Illustrating the dynamic behavior of the R3 Sampling algorithm on Auckley Function using the L^2 Physics-informed Loss computed on the retained and re-sampled populations. The horizontal lines represent the L^p Physics-informed Loss on a dense set of uniformly sampled collocation points (where $p = 2, 4, 6, \infty$).

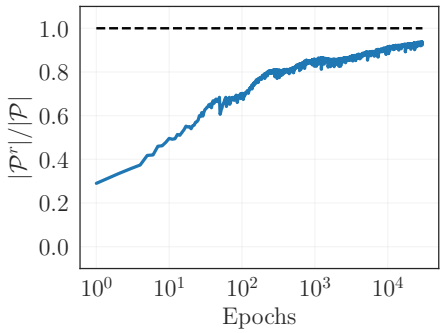


Figure 29. Demonstrating the dynamic evolution of the retained population size over epochs on the Auckley Function.

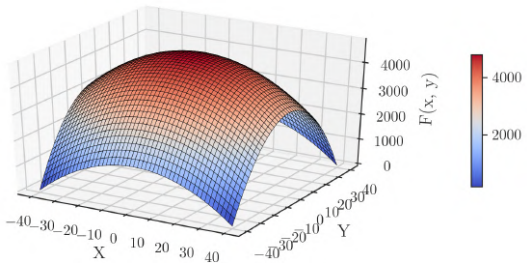


Figure 30. Surface Plot of the 2-D Bohachevsky Function

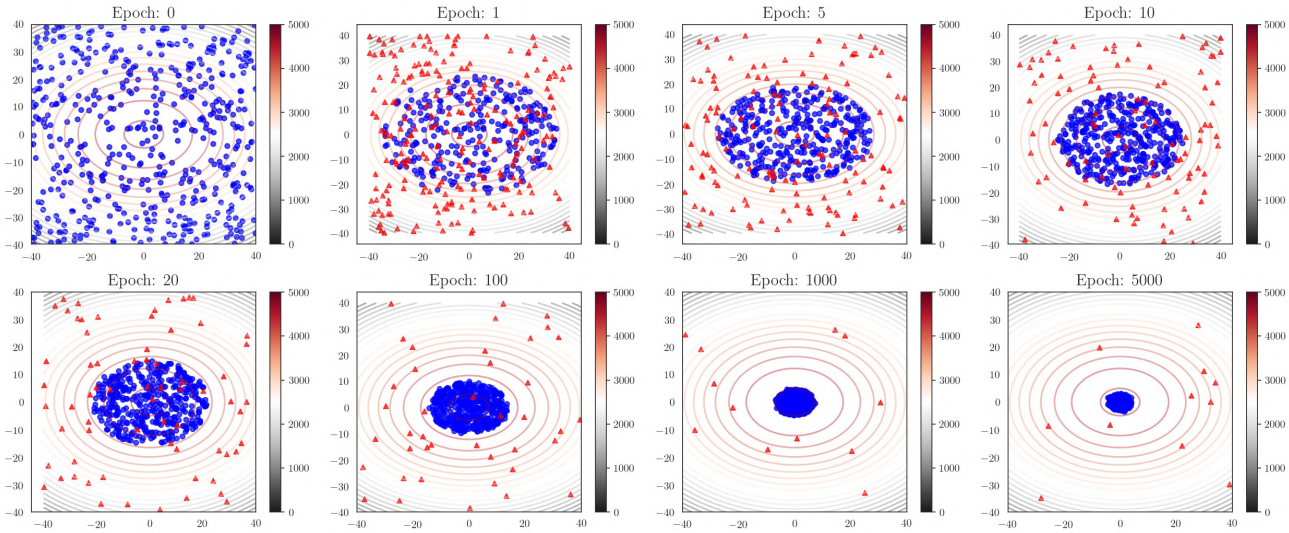


Figure 31. Demonstrating the evolution of the randomly initialized points while optimizing the Bohachevsky function. The red triangles represent the re-sampled population at that epoch, and the blue dots represent the retained population at that epoch. The contour function of the objective function is shown in the background.

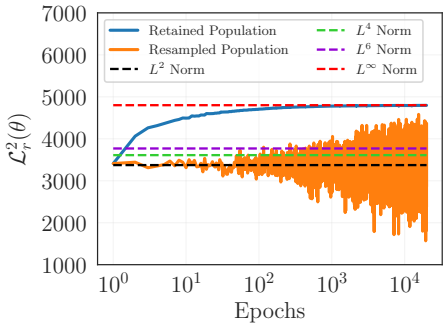


Figure 32. Illustrating the dynamic behavior of the R3 Sampling algorithm on Bohachevsky Function using the L^2 Physics-informed Loss computed on the retained and re-sampled populations. The horizontal lines represent the L^p Physics-informed Loss on a dense set of uniformly sampled collocation points (where $p = 2, 4, 6, \infty$).

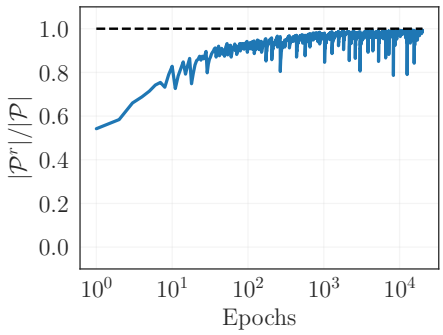


Figure 33. Demonstrating the dynamic evolution of the retained population size over epochs on the Bohachevsky Function.

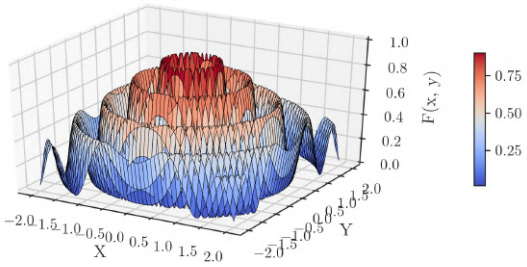


Figure 34. Surface Plot of the 2-D Drop-Wave Function

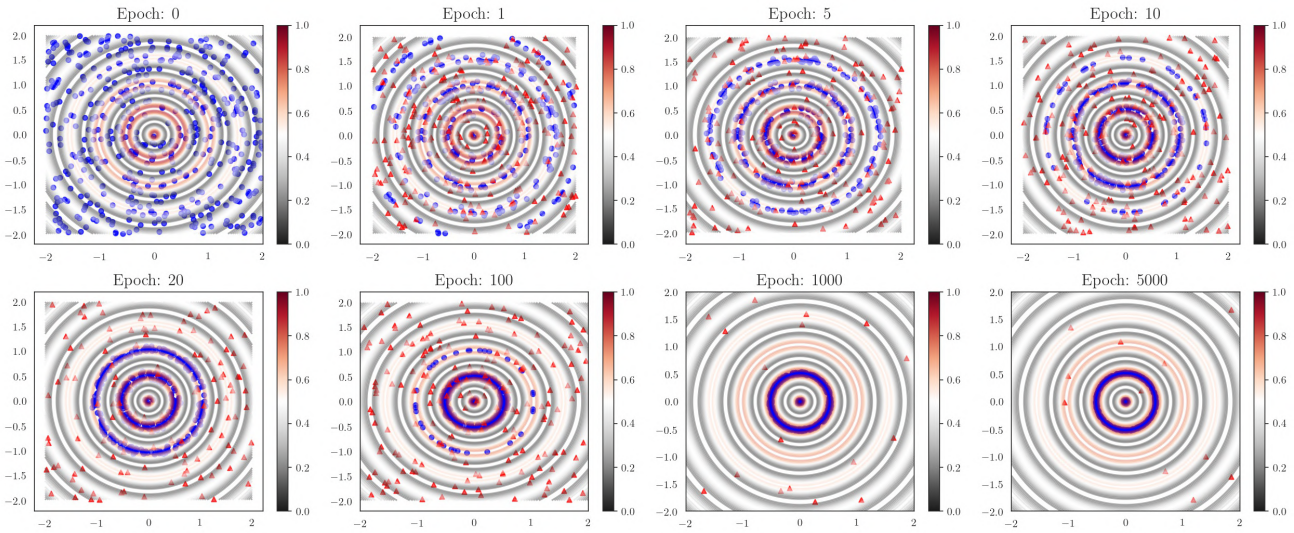


Figure 35. Demonstrating the evolution of the randomly initialized points while optimizing the Drop-Wave function. The red triangles represent the re-sampled population at that epoch, and the blue dots represent the retained population at that epoch. The contour function of the objective function is shown in the background.

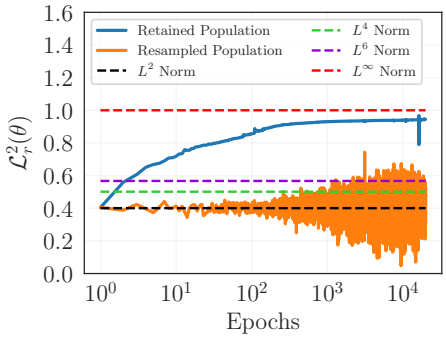


Figure 36. Illustrating the dynamic behavior of the R3 Sampling algorithm on Drop-Wave Function using the L^2 Physics-informed Loss computed on the retained and re-sampled populations. The horizontal lines represent the L^p Physics-informed Loss on a dense set of uniformly sampled collocation points (where $p = 2, 4, 6, \infty$).

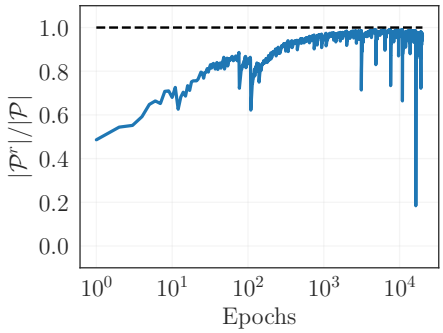


Figure 37. Demonstrating the dynamic evolution of the retained population size over epochs on the Drop-Wave Function.

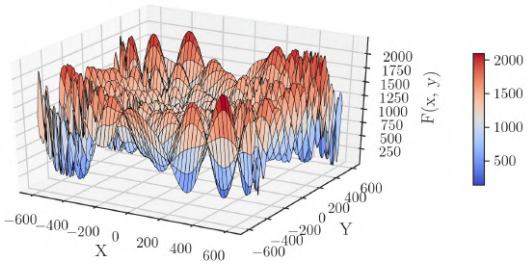


Figure 38. Surface Plot of the 2-D Egg-Holder Function

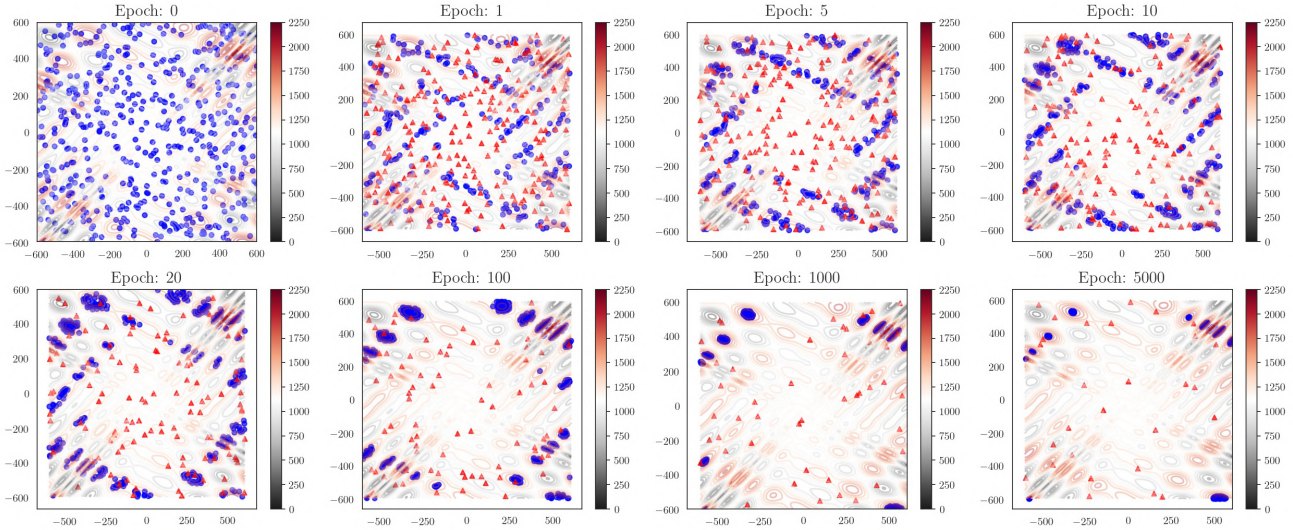


Figure 39. Demonstrating the evolution of the randomly initialized points while optimizing the Egg-Holder function. The red triangles represent the re-sampled population at that epoch, and the blue dots represent the retained population at that epoch. The contour function of the objective function is shown in the background.

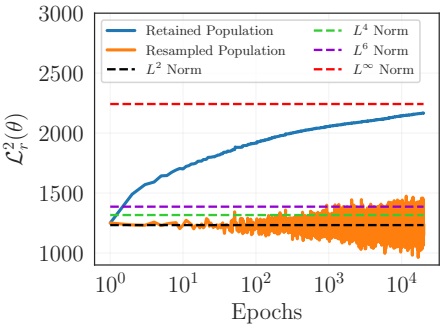


Figure 40. Illustrating the dynamic behavior of the R3 Sampling algorithm on Egg-Holder Function using the L^2 Physics-informed Loss computed on the retained and re-sampled populations. The horizontal lines represent the L^p Physics-informed Loss on a dense set of uniformly sampled collocation points (where $p = 2, 4, 6, \infty$).

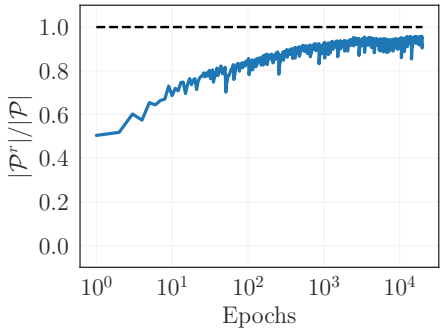


Figure 41. Demonstrating the dynamic evolution of the retained population size over epochs on the Egg-Holder Function.

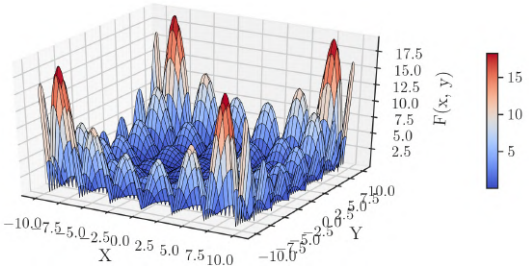


Figure 42. Surface Plot of the 2-D Holder-Table Function

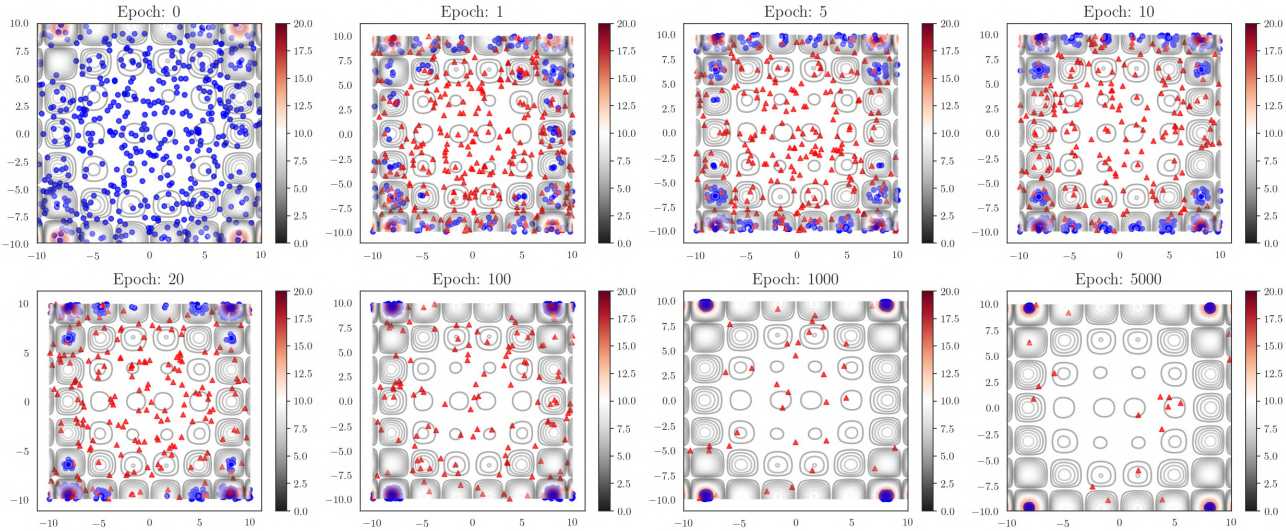


Figure 43. Demonstrating the evolution of the randomly initialized points while optimizing the Holder-Table function. The red triangles represent the re-sampled population at that epoch, and the blue dots represent the retained population at that epoch. The contour function of the objective function is shown in the background.

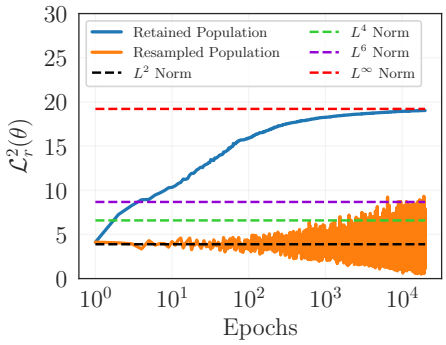


Figure 44. Illustrating the dynamic behavior of the R3 Sampling algorithm on Holder-Table Function using the L^2 Physics-informed Loss computed on the retained and re-sampled populations. The horizontal lines represent the L^p Physics-informed Loss on a dense set of uniformly sampled collocation points (where $p = 2, 4, 6, \infty$).

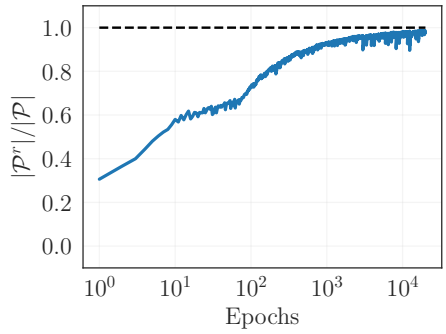


Figure 45. Demonstrating the dynamic evolution of the retained population size over epochs on the Holder-Table Function.

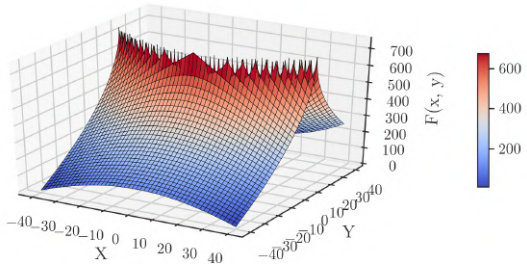


Figure 46. Surface Plot of the 2-D Bukin Function

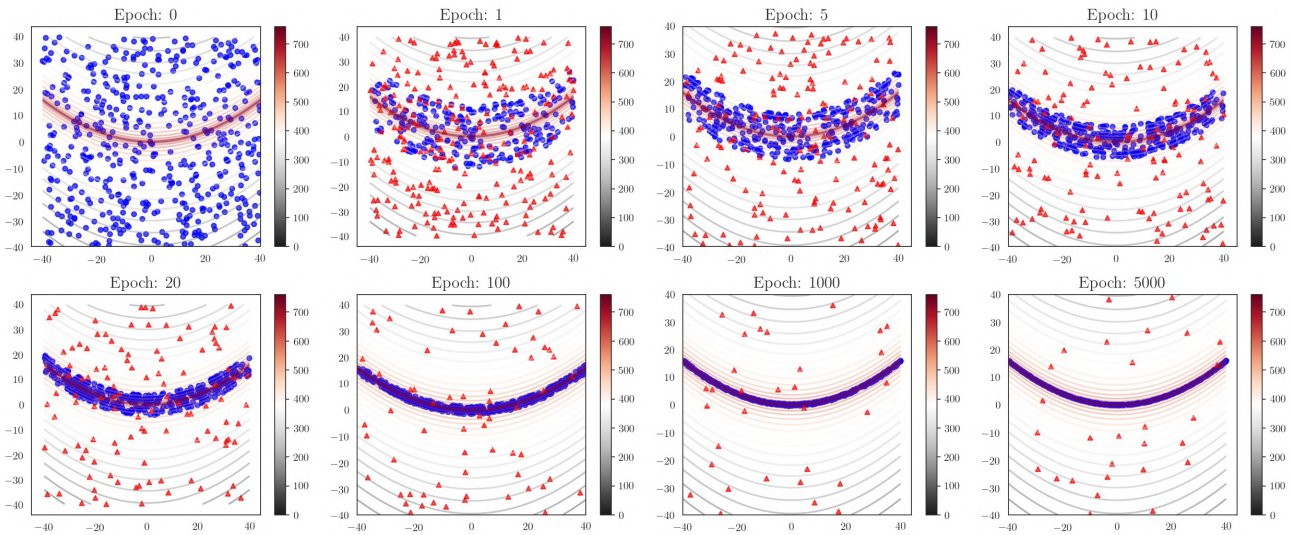


Figure 47. Demonstrating the evolution of the randomly initialized points while optimizing the Bukin function. The red triangles represent the re-sampled population at that epoch, and the blue dots represent the retained population at that epoch. The contour function of the objective function is shown in the background.

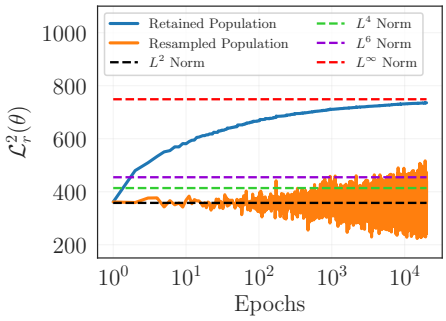


Figure 48. Illustrating the dynamic behavior of the R3 Sampling algorithm on Bukin Function using the L^2 Physics-informed Loss computed on the retained and re-sampled populations. The horizontal lines represent the L^p Physics-informed Loss on a dense set of uniformly sampled collocation points (where $p = 2, 4, 6, \infty$).

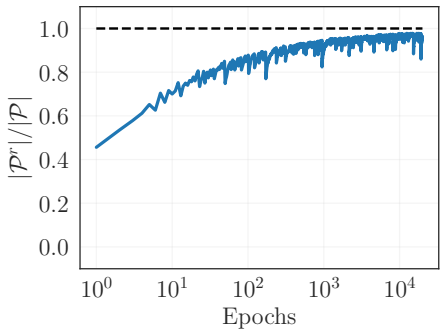


Figure 49. Demonstrating the dynamic evolution of the retained population size over epochs on the Bukin Function.

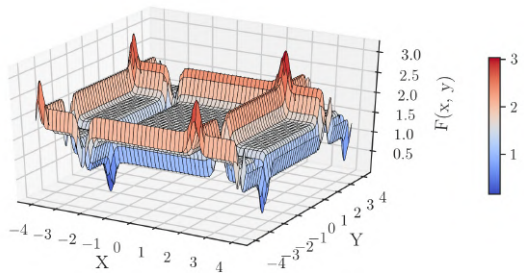


Figure 50. Surface Plot of the 2-D Michalewicz Function

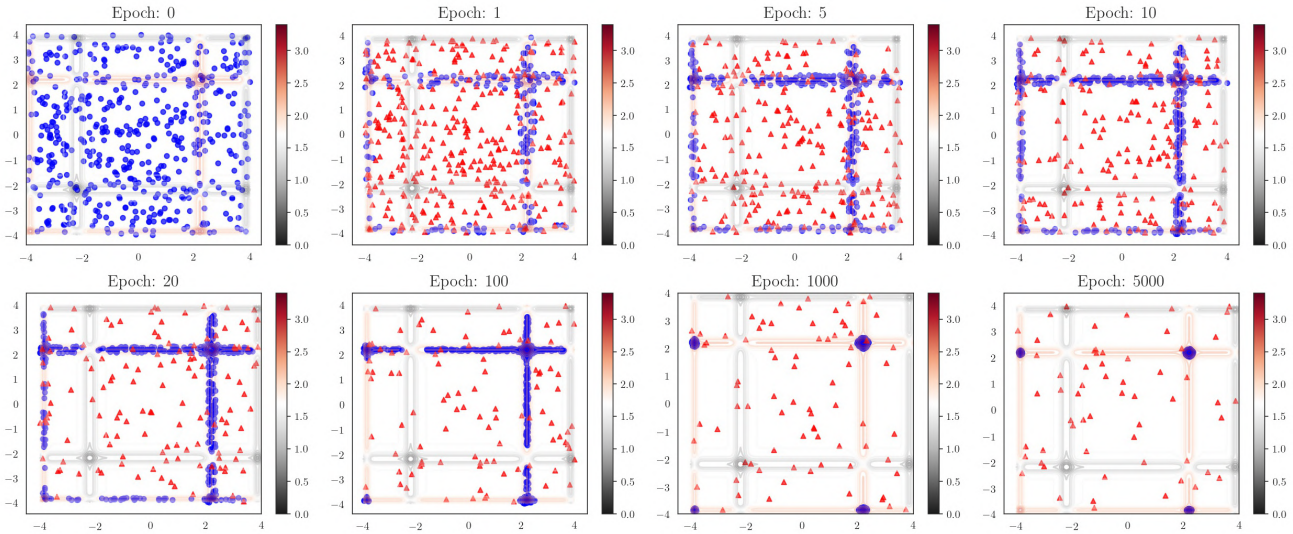


Figure 51. Demonstrating the evolution of the randomly initialized points while optimizing the Michalewicz function. The red triangles represent the re-sampled population at that epoch, and the blue dots represent the retained population at that epoch. The contour function of the objective function is shown in the background.

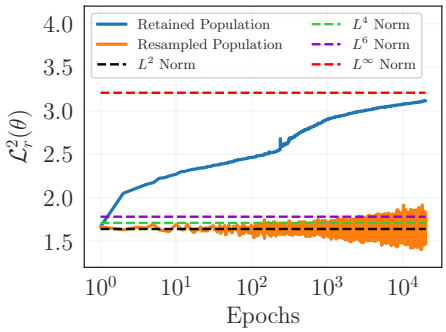


Figure 52. Illustrating the dynamic behavior of the R3 Sampling algorithm on Michalewicz Function using the L^2 Physics-informed Loss computed on the retained and re-sampled populations. The horizontal lines represent the L^p Physics-informed Loss on a dense set of uniformly sampled collocation points (where $p = 2, 4, 6, \infty$).

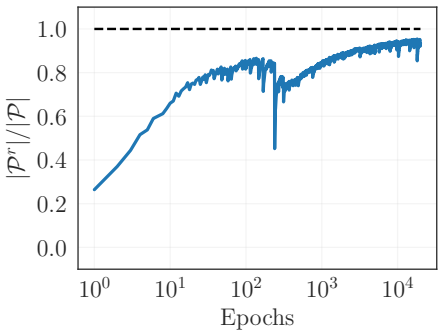


Figure 53. Demonstrating the dynamic evolution of the retained population size over epochs on the Michalewicz Function.