When and Why Adversarial Training Improves PINNs: A Neural Tangent Kernel Perspective

Yuandong Cao[1,3*] **Chi Chiu So**[2] **Jun-Min Wang**[1] **He Wang**[3†]

[1]School of Mathematics and Statistics, Beijing Institute of Technology, China
[2]School of Professional Education and Executive Development
The Hong Kong Polytechnic University, China
[3]Department of Computer Science & UCL AI Centre, University College London, UK

Abstract

Physics-informed neural networks (PINNs) are powerful surrogates for differential equations but are notoriously difficult to train due to spectral bias, stiffness, and poor accuracy on high-frequency or multiscale solutions. Adversarial training based on generative adversarial networks (GANs) has recently gained surprisingly strong empirical results in improving the training, but the underlying mechanisms remain elusive. To this end, we propose a new analysis framework for adversarially trained PINNs, based on the key observation of how the discriminator in GANs can influence the training dynamics of PINNs. The framework first provides a much needed theoretical grounding to why and when adversarial training is effective in PINNs, then presents a unified analysis of GANs variants in such training, and finally leads to a new, practical, efficient training algorithm for PINNs. Empirical results demonstrate that our method can significantly reduce the pathology of PINNs training, thereby providing better models with superior performance, often several magnitudes more accurate than alternative methods.

1 Introduction

Partial Differential Equations (PDEs) are foundational modeling tools in science and engineering, but designing stable and efficient solvers for them is challenging, particularly for complex systems characterized by strong nonlinearity, stiffness, or multiscale structures. Neural networks recently emerged as a promising alternative for solving PDEs, where physics-informed neural networks (PINNs) have become widely used, achieving remarkable success across a broad range of applications. However, PINNs are notoriously hard to train, especially when applied to PDEs with solutions exhibiting high-frequency or multiscale features [1–3]. Recent studies have systematically analyzed these failures and attributed the pathological behavior of PINNs to multiscale interactions and spectral bias among different loss components [4, 5]. In practice, this manifests itself as a stagnation of the training loss near stiff regions of the PDE where the network prediction deviates significantly from the ground truth. A recent attempt to mitigate this pathology is to leverage adversarial training via generative adversarial networks (GANs), *i.e*. adversarial PINNs training [6–10].

GANs consist of a generator and a discriminator, trained via a minmax game [11]. When applied to PINNs, the prediction errors, *e.g*. PDE residual, boundary, or/and initial condition errors, are modeled as generated samples, while the ideal zero error is treated as the real target. In other words, training amounts to driving the empirical distribution of the prediction error toward a Dirac measure centered at zero. A growing body of work has demonstrated the empirical effectiveness of this idea, including employing vanilla GANs with hard constraints [6], Wasserstein GANs with penalty-based methods

*work done during visiting University College London
†corresponding author, he_wang@ucl.ac.uk

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

[7] for PDEs and Stochastic Differential Equations [8]. More recent research also investigates using separate discriminators for different loss terms [9] and balancing physics constraints and control objectives in PDE-constrained optimal control [10].

Despite empirical successes, the theoretical grounding of adversarial PINNs training is missing, especially in why GANs succeed or fail in some settings. This has led to inconsistent or even contradictory observations and analysis. For example, Ciftci et al. [7] and Yang et al. [8] argued that vanilla GAN training is essentially Jensen-Shannon divergence minimization under suitable assumptions, and therefore should be avoided in PINN training due to its well-known instability, but Bullwinkel et al. [6] showed that vanilla GANs can in practice drive PINNs to extremely accurate solutions. More broadly, the generated distribution of adversarially trained PINNs is often a finite Dirac mixture supported on the PDE residual samples, whereas the target distribution is a Dirac measure with singular support at zero. Under this support mismatch, the classical divergence-based theory [12, 13] provides little useful guidance and may even suggest degenerate training behaviors, which is clearly inconsistent with empirical evidence. Overall, the above discrepancies show that existing analysis which is mainly based on *static* optimal-discriminator behaviors is insufficient to explain the adversarial PINNs training. A *dynamical* perspective is required.

We draw inspiration from the gradient-flow analysis of neural networks, specifically through the lens of Neural Tangent Kernel (NTK) [14], to characterize the dynamics of adversarial PINNs training. NTKs have been employed successfully in GANs [15] to show that the study of the alternating updates between the generator and the discriminator is essential to understand the training dynamics. We generalize this analysis to adversarial PINNs training to reveal: (1) how discriminator training shapes the dynamics of PINNs, (2) how different GAN objectives induce different sample-wise weighting, and (3) how these effects interact with the generator training dynamics. These phenomena jointly explain the convergence (or the lack of) of adversarial PINNs training.

Overall, our main contributions include:

- A new unified NTK framework for analyzing adversarially trained PINNs.
- A formal analysis of adversarial PINNs training under different GAN variants.
- New findings in how the discriminator shapes the spectral dynamics of PINNs training.
- A new training algorithm with improved optimization stability and convergence behavior.

2 Preliminaries

Generative Adversarial Networks GAN consists of a generator and a discriminator. We denote the discriminator by $f(\cdot;\phi) : \mathcal{X} \to \mathbb{R}$, where ϕ denotes the discriminator parameters. The generator $g(z;\theta)$ is parameterized by θ, where $z \sim p_z$ is a latent variable. Let α_r and α_g^θ denote the real and generated data distributions on $\mathcal{X}$, respectively. GAN training can be seen as a minmax game:

$$\max_{\phi} \mathcal{L}(\phi) = \mathbb{E}_{x\sim\alpha_r}\big[P(f(x;\phi))\big] + \mathbb{E}_{x\sim\alpha_g^\theta}\big[Q(f(x;\phi))\big], \tag{1}$$

$$\min_{\theta} \mathcal{L}(\theta) = \mathbb{E}_{x\sim\alpha_g^\theta}\big[R(f(x;\phi))\big], \tag{2}$$

where $P, Q, R : \mathbb{R} \to \mathbb{R}$ are scalar functions. In particular, we consider three GAN variants:

$$\text{GAN [11]: } P(t) = \log\sigma(t),\ \ Q(t) = R(t) = \log\big(1-\sigma(t)\big),\ \ \sigma(\cdot) \text{ is the sigmoid function,} \tag{3}$$

$$\text{LSGAN [13]: } P(t) = -\frac{1}{2}(t-1)^2, \quad Q(t) = -\frac{1}{2}t^2, \quad R(t) = \frac{1}{2}(t-1)^2, \tag{4}$$

and

$$\text{WGAN/IPM[16]: } P(t) = t, \quad Q(t) = R(t) = -t. \tag{5}$$

Hard-constrained PINNs Let $\Omega \subset \mathbb{R}^d$ be the space–time domain and consider the PDE

$$\mathcal{N}(u)(x) = a(x), \quad x \in \Omega, \qquad \mathcal{B}(u)(x) = b(x), \quad x \in \partial\Omega, \tag{6}$$

where $\mathcal{N}$ is the differential operator, $a(x)$ is the source term, and $\mathcal{B}$ denotes the boundary and/or initial constraint operator. Through a proper hard-constraint parameterization [17, 18], the neural solution u_θ satisfies $\mathcal{B}(u_\theta)(x) = b(x)$ exactly. Therefore, training reduces to minimizing the interior residual $\mathcal{R}(x;\theta) := \mathcal{N}[u_\theta](x) - a(x)$. The hard-constrained PINN loss is then given by

$$\mathcal{L}_{\text{PINN}}(\theta) = \mathbb{E}_{x\sim\mu_\Omega}\big[\mathcal{R}(x;\theta)\big], \tag{7}$$

where μ_Ω denotes the sampling distribution over the interior collocation domain.

Adversarially trained PINNs Given collocation points $\{x_i\}_{i=1}^{N_r} \subset \Omega$, define residual $r_i := \mathcal{R}(x_i;\theta)$ and $\mathbf{r} := (r_1,\dots,r_{N_r})^\top \in \mathbb{R}^{N_r}$. Furthermore, we define $\hat{\mu}_\Omega := \frac{1}{N_r}\sum_{i=1}^{N_r}\delta_{x_i}$ to be the empirical collocation measure, where δ_{x_i} is the Dirac measure at x_i. Then the generated residual distribution is the pushforward empirical measure $\hat{\mu}_\theta^r := (\mathcal{R}_\theta)_\# \hat{\mu}_\Omega = \frac{1}{N_r}\sum_{i=1}^{N_r}\delta_{r_i}$, where $\mathcal{R}_\theta : x \mapsto \mathcal{R}(x;\theta)$. The ideal target residual distribution is the Dirac measure at zero, $\mu_{\text{real}} = \delta_0$. Equivalently, we define the target residual map $\mathcal{T}(x_i) := 0, i = 1,\dots,N_r$, which represents the desired zero-residual state (real data). Therefore, the adversarial PINNs training objectives become:

$$\mathcal{L}(\theta) = \frac{1}{N_r}\sum_{i=1}^{N_r} R\Big(f(\mathcal{R}(x_i;\theta);\phi)\Big), \tag{8}$$

and

$$\mathcal{L}(\phi) = \frac{1}{N_r}\sum_{i=1}^{N_r}\Big[P\Big(f(\mathcal{T}(x_i);\phi)\Big) + Q\Big(f(\mathcal{R}(x_i;\theta);\phi)\Big)\Big]. \tag{9}$$

Here P, Q, R are the same as in Eqs. (1) and (2). The objectives in Eqs. (8) and (9) will be the starting point of our analysis. Hence, the generator g is the PINN itself, while the discriminator operates on PDE residuals rather than collocation locations. Specifically, each collocation point passes through the following training pipeline:

$$x_i \xrightarrow{g} u_\theta(x_i) \xrightarrow{\mathcal{R}} r_i \xrightarrow{f} f(r_i) \xrightarrow{\text{backpropagation}} \nabla_\theta \mathcal{L}(\theta) \xrightarrow{\text{parameter update}} g. \tag{10}$$

The discriminator distinguishes these residuals from the target residual of zero, and its feedback is propagated through the residual operator to update the generator g.

Neural tangent kernel regime Our analysis is carried out in the neural tangent kernel (NTK) regime. At a high level, this regime describes sufficiently wide neural networks trained in a lazy-training manner, so that the network output can be approximated by its first-order expansion around initialization, while the associated kernel remains nearly constant during training.

We adopt the following standard assumptions:

Assumption 1 ([14]). *(i) The network is a standard architecture, such as a fully connected, convolutional, or residual network, with standard nonlinearities. (ii) The activation function is sufficiently smooth, and the network output is differentiable with respect to the parameters. (iii) Linear layers contain nonzero bias terms. (iv) The network parameters are randomly initialized with a standard width-dependent scaling. (v) The hidden widths tend to infinity.*

Under these assumptions, if $g(x;\theta)$ denotes the network output, then around initialization θ_0,

$$g(x;\theta) \approx g(x;\theta_0) + \nabla_\theta g(x;\theta_0)^\top(\theta - \theta_0), \tag{11}$$

and the NTK is defined by

$$k(x,x') = \big\langle \nabla_\theta g(x;\theta_0), \nabla_\theta g(x';\theta_0)\big\rangle. \tag{12}$$

For a finite set of training points, the output dynamics under gradient flow are then approximated by

$$\dot{\mathbf{g}}(t) \approx -K\,\nabla_{\mathbf{g}}\mathcal{L}(\mathbf{g}(t)), \tag{13}$$

where K is the NTK Gram matrix. Therefore, in the NTK regime, training can be analyzed in function space through an approximately fixed kernel rather than only through the original nonlinear parameter dynamics. A more detailed discussion of the related works and their connections to the present framework is provided in Appendix A.

3 The Unreasonable Successes in Adversarial PINNs Training

In vanilla GAN, it is known that the generator's objective is, up to constants, minimizing the Jensen–Shannon divergence between the generated $\hat{\mu}_\theta^r$ and the target δ_0, under proper assumptions. Similar conclusions also arise from other analysis based on *e.g.* KL-type [12] and χ^2-type [13] criteria. All analyses point to one phenomenon: *when the supports of the generated and target distributions are nearly disjoint and the discriminator becomes nearly optimal, the generator receives a degenerate*

training signal. This issue is particularly severe in adversarial PINNs. At the beginning of training, the residual samples $\{r_i\}_{i=1}^{N_r}$ are typically far from zero, so the support of $\hat{\mu}_\theta^r$ is almost disjoint from that of δ_0. Theoretically, one would expect the training signal for the generator to saturate or become uninformative. In reality, existing adversarial PINNs methods often perform well. This discrepancy indicates that the classical divergence-minimization framing does not directly apply to PINNs, as it assumes the generator is always trained against an optimal discriminator, whereas practical adversarial PINNs training uses a sub-optimal discriminator. As a result, the generator is driven not by the gradient of a fixed divergence alone, but also by the gradient field induced by a transient discriminator state.

As a motivating example, we show how the generator–discriminator update scheme can significantly influence the training, using the two-dimensional Laplace equation:

$$\Delta u(x, y) = 0, \qquad (x, y) \in [0, 1]^2, \tag{14}$$

with boundary conditions

$$u(x, 0) = 0, \;\; u(0, y) = 0, \;\; u(1, y) = 0, \;\; u(x, 1) = \frac{1}{2}\cosh(\pi)\sin(\pi x)\big(e^{\pi y} - e^{-\pi y}\big), \tag{15}$$

and a hard-constrained PINNs. We vary the generator–discriminator update ratio by:

$$G : D = 1000{:}1,\; 100{:}1,\; \ldots,\; 1{:}1,\; \ldots,\; 1{:}100,\; 1{:}1000,$$

and show the training dynamics in Fig. 1.

Fig. 1 shows that both over-trained discriminators and over-trained generators can harm adversarial PINNs optimization. When D is updated too many times relative to G, it may become nearly optimal for the current residual distribution, causing the learning signal passed to G to become saturated, weak, or poorly conditioned. Conversely, when G is updated too many times relative to D, the discriminator remains under-trained and cannot provide a reliable gradient. In both cases, the imbalance leads to deteriorated training and validation errors. Therefore, the U-shaped curve in Fig. 1 provides direct evidence that the success of adversarial PINNs is governed by finite-step alternating dynamics between G and D, rather than by the static optimal-discriminator alone. The best performance appears in an intermediate regime where D is informative but not over-trained, and G is updated sufficiently but does not outrun the discriminator.

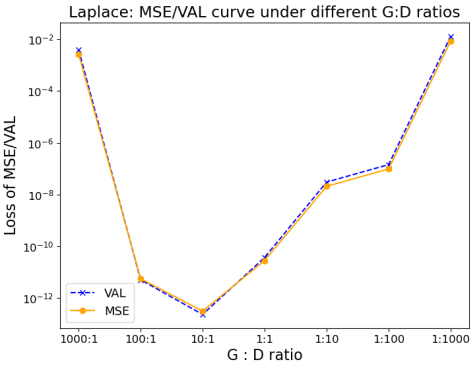


Figure 1: Laplace equation: final converged training and validation errors under different generator-to-discriminator update ratios (G:D).

These observations naturally raise two questions: how can we formally characterize such finite-step alternating dynamics, and how can we use this characterization to choose or control the generator–discriminator update schedule? Addressing these questions is the main goal of the following analysis.

4 NTK Analysis of Adversarially Trained PINNs

We establish a dynamical analysis of the alternating optimization of adversarially trained PINNs. To this end, we first formulate the discriminator-induced residual-space gradient field, then show how it influences the training of the generator.

4.1 Discriminator-induced gradient field in the NTK regime

Given a fixed generator with θ and the objective in Eq. (9), the discriminator gradient flow is:

$$\dot{\phi}(t) = \frac{1}{N_r}\sum_{i=1}^{N_r}\left[P'\big(f(\mathcal{T}(x_i);\phi(t))\big)\,\frac{\partial f(\mathcal{T}(x_i);\phi(t))}{\partial \phi} + Q'\big(f(r_i;\phi(t))\big)\,\frac{\partial f(r_i;\phi(t))}{\partial \phi}\right]. \tag{16}$$

If we define the discriminator NTK as:

$$k_t^D(x,y) := \left\langle \frac{\partial f(x;\phi(t))}{\partial \phi(t)}, \frac{\partial f(y;\phi(t))}{\partial \phi(t)} \right\rangle, \tag{17}$$

then NTK theory implies that $k_t^D(x,y) \approx k^D(x,y)$ during training. Therefore, for any residual-space input x, the discriminator admits the function-space evolution:

$$\frac{d}{dt} f(x;\phi(t)) = \frac{1}{N_r} \sum_{i=1}^{N_r} \Big[P'\big(f(\mathcal{T}(x_i);\phi(t))\big)\, k^D(x,\mathcal{T}(x_i)) + Q'\big(f(r_i;\phi(t))\big)\, k^D(x,r_i) \Big]. \tag{18}$$

NTK regularity implies that $k^D(\cdot, y)$ is differentiable with respect to its first argument, so $f(\cdot;\phi(t))$ is differentiable w.r.t. the input; see Appendix B.

Lemma 1 (Input-gradient representation of the discriminator [15])**.** *Under Assumptions 1 the discriminator input gradient satisfies*

$$\partial_x f(x;\phi(t)) = \partial_x f_0(x) + \int_0^t \frac{1}{N_r} \sum_{i=1}^{N_r} \Big[P'\big(f(\mathcal{T}(x_i);\phi(s))\big)\, \partial_x k^D(x,\mathcal{T}(x_i)) + Q'\big(f(r_i;\phi(s))\big)\, \partial_x k^D(x,r_i) \Big] ds, \tag{19}$$

where $f_0(x) := f(x;\phi(0))$.

Remark 1. *Lemma 1 describes the evolution of the discriminator gradient field w.r.t. the input during training. It shows that* $\partial_r f(r;\phi(t))$ *is governed by the derivatives of the discriminator objective and the derivatives of the discriminator NTK kernel* k^D *w.r.t. the input. Therefore, the residual-space gradient field produced by the discriminator is not arbitrary, but is structured by the discriminator architecture, objective, and tangent kernel. This discriminator gradient field will later enter the generator dynamics through the sample-wise weighting (see Appendix B). The corresponding NTK-based discriminator dynamics for vanilla GAN, LSGAN, and IPM-type GANs are given in Appendix B.1–B.3.*

4.2 Generator residual dynamics and objective-dependent weighting

For the generator, given the generator objective in Eq. (8), the generator gradient flow is:

$$\dot{\theta}(t) = -\frac{\partial \mathcal{L}(\theta(t))}{\partial \theta(t)}. \tag{20}$$

We further define the generator residual NTK:

$$(K_{rr}^G)_{ij}(t) = \left\langle \frac{\partial \mathcal{R}(x_i;\theta(t))}{\partial \theta(t)}, \frac{\partial \mathcal{R}(x_j;\theta(t))}{\partial \theta(t)} \right\rangle \tag{21}$$

and a discriminator-induced weighting vector

$$\boldsymbol{\gamma}(t) := \big[\gamma_1(t), \ldots, \gamma_{N_r}(t)\big]^\top, \tag{22}$$

where

$$\gamma_i(t) = -\frac{1}{N_r} R'\Big(f(r_i(t);\phi(t))\Big)\, \partial_r f(r_i(t);\phi(t)), \qquad i = 1, \ldots, N_r. \tag{23}$$

Proposition 1 (Residual dynamics under adversarial training)**.** *Under the generator gradient flow* (20)*, the residual vector evolves according to*

$$\dot{\mathbf{r}}(t) = K_{rr}^G(t)\, \boldsymbol{\gamma}(t). \tag{24}$$

Remark 2. *Equation* (24) *shows that adversarial PINNs training takes the form of a* preconditioned residual descent*. The generator residual NTK* $K_{rr}^G(t)$ *determines how a weighted residual signal is propagated across collocation points, whereas the discriminator enters through the sample-wise weighting vector* $\boldsymbol{\gamma}(t)$*. In other words, the discriminator reshapes the residual descent field through its input gradient rather than through its scalar output alone. Detailed derivations of the generator–discriminator update dynamics, together with the corresponding propositions and theorems, are provided in Appendix B.4.*

Remark 3. *The NTK-based generator dynamics above are derived for the hard-constrained setting with the residual itself as the discriminator input. When the discriminator input is changed from the residual to squared residual, the dynamics become fundamentally different and take a closed-loop multiplicative form. The detailed derivation of the dynamics based on squared residual input and the corresponding residual-energy law under hard constraints is provided in Appendix D.*

Remark 4. *Beyond the hard-constrained setting, Appendix E extends the analysis to soft-constrained adversarial PINNs, where we derive the coupled residual and energy dynamics for interior, boundary, and initial-condition violations, and further motivate our research.*

Remark 5. *More broadly, we emphasize that the present framework is not limited to standard GAN variants. Existing adversarially trained PINNs such as SA-PINN [19], LA-PINN[9], and WAN[20] are all special cases of our analysis with the discriminator optimization restricted to structured function classes; the corresponding derivations are given in Appendix F. This perspective also reveals a much broader and still underexplored design space of adversarial PINNs methods, which we summarize in Appendix G.*

4.3 Residual-energy Law, Spectral Interpretation, and Practical Optimization

To characterize the optimization effect of the discriminator-induced weighting $\boldsymbol{\gamma}(t)$ (Eq. (22)), we define a residual energy $E(t) := \frac{1}{2}\|\mathbf{r}(t)\|^2$.

Theorem 1 (Residual-energy law under adversarial training)**.** *Under the residual dynamics in Eq.* (24)*, the residual energy satisfies*

$$\frac{d}{dt}E(t) = \mathbf{r}(t)^\top K_{rr}^G(t)\boldsymbol{\gamma}(t) = S(t), \tag{25}$$

where $S(t)$ is the instantaneous residual-energy variation rate induced by the current discriminator state. In particular,

$$S(t) < 0 \Longrightarrow \frac{d}{dt}E(t) < 0, \qquad S(t) > 0 \Longrightarrow \frac{d}{dt}E(t) > 0. \tag{26}$$

Theorem 1 shows that the first-order effect of adversarial interaction on the generator optimization is fully characterized by the scalar quantity $S(t)$. To interpret this quantity from a modal viewpoint, we consider the constant-NTK regime $K_{rr}^G(t) \approx K_{rr}^G$, decompose it as

$$K_{rr}^G = U\Lambda U^\top, \qquad \Lambda = \mathrm{Diag}(\lambda_1, \ldots, \lambda_{N_r}), \qquad \lambda_j \geq 0, \tag{27}$$

and define $\widetilde{\mathbf{r}}(t) := U^\top \mathbf{r}(t), \widetilde{\boldsymbol{\gamma}}(t) := U^\top \boldsymbol{\gamma}(t)$. Then we obtain

$$S(t) = \mathbf{r}(t)^\top K_{rr}^G \boldsymbol{\gamma}(t) = \widetilde{\mathbf{r}}(t)^\top \Lambda \widetilde{\boldsymbol{\gamma}}(t) = \sum_{j=1}^{N_r} \lambda_j \, \widetilde{r}_j(t) \, \widetilde{\gamma}_j(t). \tag{28}$$

Remark 6 (Energy decay rate and spectral analysis)**.** *By Eqs.* (25) *and* (28)*, the decay of $E(t)$ is jointly determined by the generator spectrum $\{\lambda_j\}$, the residual amplitudes $\widetilde{r}_j(t)$, and the discriminator-induced weights $\widetilde{\gamma}_j(t)$. As training proceeds, modes with large λ_js decay fast, then the energy decay increasingly depends on whether $\widetilde{\gamma}_j(t)$ remains aligned with the dominant modes. If it aligns with already decayed modes thereby underweighting the others, the energy decay slows down.*

Remark 7 (Representative failures)**.** *The analysis explains several representative failures in adversarial PINNs training: (i)* energy increase*, where $S(t) > 0$; (ii)* vanishing discriminator gradients*, where $\boldsymbol{\gamma}(t) \approx 0$; and (iii)* noisy or misaligned weighting*, where $\widetilde{\gamma}_j(t)$ becomes unstable or ineffective.*

More detailed analysis and justifications for Remark 6 and 7 are in Appendix C.

To systematically address the above possible failures, one could explicitly control individual modes, where modes would need to be monitored during training and the discriminator feedback would need to be adjusted. However, a straightforward solution from Eq. (25) is to ensure the continuous decrease of $E(t)$, *i.e.* $S(t) < 0$. Therefore, we propose a practical approach below.

We denote by θ^m the generator parameters at the beginning of the m-th iteration, and define residual

$$\mathbf{r}^m := [\mathcal{R}(x_1; \theta^m), \ldots, \mathcal{R}(x_{N_r}; \theta^m)]^\top. \tag{29}$$

After a discriminator substep, we denote the frozen discriminator by $f^{m+\frac{1}{2}}$, and define

$$\gamma_i^m = -\frac{1}{N_r} R'\big(f^{m+\frac{1}{2}}(r_i^m)\big)\,\partial_r f^{m+\frac{1}{2}}(r_i^m), \qquad \boldsymbol{\gamma}^m = [\gamma_1^m, \dots, \gamma_{N_r}^m]^\top. \tag{30}$$

We also let $K_{rr}^{G,m}$ denote the NTK at θ^m.

Proposition 2 (One-step discrete residual-energy variation)**.** *A first-order generator update with frozen discriminator yields*

$$\mathbf{r}^{m+1} = \mathbf{r}^m + \eta_G K_{rr}^{G,m} \boldsymbol{\gamma}^m + \mathcal{O}(\eta_G^2).$$

Consequently, if $E^m := \frac{1}{2}\|\mathbf{r}^m\|^2$*, then*

$$E^{m+1} - E^m = \eta_G (\mathbf{r}^m)^\top K_{rr}^{G,m} \boldsymbol{\gamma}^m + \mathcal{O}(\eta_G^2).$$

Algorithm 1 Adaptive Alternating Adversarial PINNs training with Rollback

Require: Initial parameters θ^0, ϕ^0, discriminator budget T_D, generator budget T_G

1: **for** $m = 0, 1, 2, \dots$ **do**

2: Build a fixed evaluation set and compute

$$\mathbf{r}^m = [\mathcal{R}(x_1; \theta^m), \dots, \mathcal{R}(x_{N_r}; \theta^m)]^\top, \qquad E^m = \frac{1}{2}\|\mathbf{r}^m\|^2.$$

3: **Discriminator:** with θ^m frozen, run up to T_D discriminator updates; after each step compute

$$S^{m,t} = (\mathbf{r}^m)^\top K_{rr}^{G,m} \boldsymbol{\gamma}^{m,t}.$$

Keep the discriminator state attaining the smallest score, and set it as $\phi^{m+\frac{1}{2}}$.

4: **Generator:** with $\phi^{m+\frac{1}{2}}$ frozen, run up to T_G generator updates; after each step compute

$$E^{m,t} = \frac{1}{2}\|\mathbf{r}^{m,t}\|^2.$$

Keep the generator state attaining the smallest residual energy, and set it as θ^{m+1}.

5: **end for**

From Proposition 2, we derive Algorithm 1, which is essentially a dynamic update scheme for both the generator and the discriminator. Its key principle is that the discriminator should be retained only when it can induce a residual weighting that is able to improve the generator's ability to reduce the residual. At the same time, the generator update should be conducted only when the residual energy actually decreases. In this way, the discriminator acts not as an always accepted adversary but as an adaptive controller to balance the residual modes for continuous residual-energy decay.

5 Experiments

5.1 Experimental setup

We compare DEQGAN [6], GAN [11], LSGAN [13], and WGAN-GP [16] with their variants enhanced versions GAN-RB, LSGAN-RB, and WGAN-GP-RB, where "RB" denotes our rollback strategy. The benchmark suite includes the Poisson, Laplace, viscous Burgers, Reaction–Diffusion, and Klein–Gordon equations, covering linear and nonlinear, stationary and time-dependent PDEs.

For fairness, all methods share the same generator architecture, while the discriminator architecture is chosen according to the adversarial objective. Except for DEQGAN, which follows its original tuning-dependent protocol, all methods are trained under the same default setting. We report training and validation errors as the main accuracy metrics, and additionally monitor the residual energy and the first-order quantity $r^\top K_{rr}^G \gamma$. Full implementation details are deferred to Appendix H.1.

5.2 Mechanistic validation: when and why adversarial training improves PINNs

We revisit DEQGAN under different generator–discriminator update ratios (Fig. 1) to clarify a central question of this paper: *when* adversarial training improves PINNs, and *why*. Rather than merely

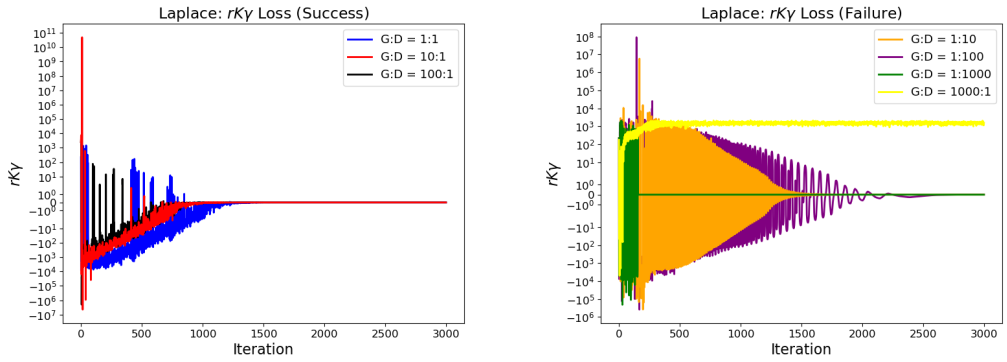


Figure 2: Left: successful training regimes under balanced or moderately imbalanced $G : D$ ratios. Right: three representative failure modes caused by extreme update imbalance.

Table 1: Statistics of the positive part of $r^\top K_{rr}^G \gamma$ for the three successful update ratios (Fig. 2 Left). Smaller values indicate fewer or weaker violations of the energy descent requirement.

$G : D$	Positive ratio	Positive mean	Max value	L^1 violation	L^2 violation
$1 : 1$	0.0220	18.1644	322.8066	1180.6849	190919.5174
$10 : 1$	**0.0075**	**0.1429**	**1.8219**	**3.1449**	**4.0658**
$100 : 1$	0.0661	1.0747	75.9468	209.5676	10192.1785

comparing schedules, we use the first-order quantity $r^\top K_{rr}^G \gamma$ to distinguish successful alternating regimes from failed ones.

A clear pattern emerges from Fig. 2. Fig. 2 Left shows three ratios $G : D = 100:1$, $10:1$, and $1:1$ which are successful regimes because their $r^\top K_{rr}^G \gamma$ curves remain predominantly negative. This indicates that, on average, the discriminator-induced signal is aligned with residual descent after modulation by the generator kernel. However, the behaviors of the three are not identical. Table 1 reports aggregated statistics of the positive part of $r^\top K_{rr}^G \gamma$, including the positive ratio, positive mean, maximum value, and L^1/L^2 violation measures. $G : D = 10:1$ yields the smallest positive ratio and the weakest violation of the energy descent requirement, indicating the best overall performance. By contrast, although $G : D = 1:1$ has a smaller positive ratio than $100:1$, its other indicators are much larger. So, successful adversarial training depends not only on whether $r^\top K_{rr}^G \gamma$ is mostly negative, but also on how strongly it deviates from the energy descent regime when violations occur.

Fig. 2 Right illustrates three failure modes described in Remark 7. $G : D = 1000:1$ corresponds to persistent misalignment: $r^\top K_{rr}^G \gamma$ remains almost always positive between 10^2 and 10^3, indicating that the discriminator is updated too slowly to provide an effective gradient. Also, $G : D = 1:1000$ corresponds to discriminator overtraining: the curve rapidly collapses to zero after roughly 200 steps, showing that many residual samples receive nearly zero gradients, consistent with the classical analysis of vanishing generator gradients under an overly strong discriminator. Next, the intermediate discriminator-dominant ratios $G : D = 1:10$ and $1:100$ correspond to unstable weighting: $r^\top K_{rr}^G \gamma$ oscillates repeatedly across positive and negative regions, indicating that the effective alignment direction after modulation by K_{rr}^G is unstable.

Overall, these results support our main claims. Adversarial training can improve PINNs but only when the alternating generator–discriminator dynamics induce a weighting field that is sufficiently aligned with the residual descent. $r^\top K_{rr}^G \gamma$ is useful not only as a strict per-step certificate, but also as an aggregated indicator that separates successful regimes from qualitatively different failure modes.

5.3 Comparison on Diverse PDEs

We next compare the proposed rollback strategy with existing adversarial PINNs baselines on the benchmark suite above. The main results are summarized in Table 2, while detailed per-equation curves are in Appendix H.2.

Table 2: Comparison of adversarial PINNs methods on diverse benchmark equations. Each entry is reported as "Validation Error / Training Error".

Equation	DEQGAN	WGAN-GP	WGAN-GP + RB	LSGAN	LSGAN + RB	GAN	GAN + RB
Poisson	9.78e-13 / 9.75e-13	1.33e-04 / 1.33e-04	7.20e-05 / 7.18e-05	1.02e-08 / 1.01e-08	**5.86e-13 / 5.84e-13**	4.80e-07 / 4.79e-07	8.43e-12 / 8.47e-12
Laplace	3.65e-11 / 2.79e-11	9.04e-06 / 6.30e-06	6.51e-05 / 4.29e-05	5.86e-10 / 4.25e-10	**3.49e-12 / 2.74e-12**	8.12e-06 / 5.58e-06	2.39e-11 / 1.69e-11
Viscous-Burgers	6.49e-06 / **1.84e-04**	7.36e-02 / 7.38e-02	1.84e-01 / 1.84e-01	2.35e-02 / 2.37e-02	4.34e-05 / 3.49e-04	6.48e-04 / 7.78e-04	**2.57e-06** / 3.87e-04
Reaction-Diffusion	2.92e-09 / 2.45e-09	2.13e-02 / 1.79e-02	7.15e-02 / 5.96e-02	8.16e-07 / 7.09e-07	1.18e-09 / 9.24e-10	9.81e-08 / 8.19e-08	**1.40e-13 / 1.08e-13**
Klein-Gordon	3.79e-06 / 3.16e-06	1.62e-02 / 1.35e-02	2.39e-04 / 1.78e-04	3.91e-03 / 3.23e-03	5.45e-05 / 4.36e-05	1.65e-02 / 1.40e-02	**4.75e-10 / 4.02e-10**

Four observations are most important. First, LSGAN is already a strong adversarial PINNs baseline, and even the classical GAN performs well under practical alternating optimization, which again suggests that the standard optimal-discriminator divergence picture is insufficient. Second, rollback is particularly effective for GAN and LSGAN: GAN-RB and LSGAN-RB consistently outperform their non-RB counterparts and achieve the best overall results on this benchmark suite. Third, the benefit is not uniform across objectives: rollback is weaker for WGAN-GP, and can even hurt performance on some problems, likely because the gradient penalty in WGAN-GP already regularizes the discriminator and the witness-style objective is less aligned with the proposed first-order criterion. Finally, LSGAN has rarely been explored in adversarial PINNs training so far but appears particularly promising, suggesting substantial room to revisit classical adversarial objectives from the perspective of practical alternating dynamics rather than nominal divergence form alone.

An important advantage of our method is that it avoids expensive hyperparameter search. Except for DEQGAN, all methods are trained with the same default learning rate 10^{-3} and default optimizer settings. By contrast, DEQGAN relies on exhaustively tuning both generator/discriminator hyperparameters. Therefore, our method provides a more efficient route to strong PINNs performance. In addition, the rollback strategy can be used as a plugin in existing adversarial PINNs methods, also demonstrated by additional ablations in the appendix H.3: (i) robustness under modified boundary conditions; (ii) compatibility with DEQGAN, and (iii) controlled comparisons under a fixed $G : D = 20 : 20$ update budget. These results indicate that the observed gains come from the rollback mechanism itself rather than merely from extra update steps. Our code is publicly available.[3]

6 Related work

Wang et al. [5] studied PINNs training from the NTK perspective and explained optimization difficulties such as spectral bias and unbalanced convergence. However, their analysis is restricted to gradient descent on a fixed PINNs loss. By contrast, we study adversarial PINNs training, where the generator residual dynamics are continuously modulated by discriminator feedback. Classical GAN theory explains training through min–max optimization and divergence minimization [13, 12], but this static picture does not explain why adversarial PINNs can still train successfully when the generated residual distribution and the zero target are initially far apart. Franceschi et al. [15] later analyzed GANs from an NTK viewpoint, but their setting is standard generative modeling rather than PDE residual learning. Our work instead studies adversarial PINNs from a residual-dynamics viewpoint. Ciftci and Hackl [7] proposed a physics-informed GAN framework and reported strong empirical performance, while Bullwinkel et al. [6] showed that adversarial objectives can improve differential equation solvers. However, these works mainly demonstrate that adversarial training can work well in practice, rather than explaining when and why it works. In contrast, we develop a residual-dynamics framework for adversarial PINNs and, based on it, derive a practical first-order rollback strategy. To the best of our knowledge, this is the first work to provide a unified theoretical framework for analyzing when and why adversarial training improves PINNs.

7 Limitations, Conclusions, and Future Work

There are limitations to our analysis. The spectral results rely on a fixed residual NTK K_{rr}, which provides a local approximation to the training dynamics. For finite-width networks, particularly when solving nonlinear PDEs, the residual NTK $K_{rr}^{(t)}$ may evolve as the network parameters change. The fixed-NTK assumption is used only for theoretical analysis, whereas the practical rollback algorithm recomputes quantities involving the current residual NTK at each iteration. Consequently, the rollback

[3] https://github.com/realcrane/PINN-GAN

criterion does not require the kernel to remain constant throughout training. Nevertheless, the fixed-NTK analysis may not fully explain the behavior observed when the kernel changes substantially. Training can also oscillate under some PDEs and training settings despite rollback; identifying the mechanisms behind these oscillations requires further investigation.

In summary, we have proposed a new analysis framework to explain the success of adversarially trained PINNs. The framework is rigorous in providing the much needed theoretical grounding, general in unifying the analysis of existing PINNs training, and practical by presenting effective and efficient training guidance. It reconciles the conflicts between the theories in GANs analysis and their empirically successful applications in training PINNs.

In future work, we will move away from the constant-NTK approximation and derive a more refined update strategy based on an evolving NTK spectrum. We include detailed discussions and limitations in Appendix I.

Acknowledgments

This work was supported in part by the UK Research and Innovation AIRR Innovator Award (0261-5654-9320-1) and the National Natural Science Foundation of China under Grants 92471108 and 12131008.

References

[1] Olga Fuks and Hamdi A Tchelepi. Limitations of physics informed machine learning for nonlinear two-phase transport in porous media. *Journal of Machine Learning for Modeling and Computing*, 1(1), 2020.

[2] Maziar Raissi. Deep hidden physics models: Deep learning of nonlinear partial differential equations. *Journal of Machine Learning Research*, 19(25):1–24, 2018.

[3] Yinhao Zhu, Nicholas Zabaras, Phaedon-Stelios Koutsourelakis, and Paris Perdikaris. Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data. *Journal of computational physics*, 394:56–81, 2019.

[4] Sifan Wang, Yujun Teng, and Paris Perdikaris. Understanding and mitigating gradient flow pathologies in physics-informed neural networks. *SIAM Journal on Scientific Computing*, 43 (5):A3055–A3081, 2021.

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

[6] Blake Bullwinkel, Dylan Randle, Pavlos Protopapas, and David Sondak. Deqgan: Learning the loss function for pinns with generative adversarial networks. *arXiv preprint arXiv:2209.07081*, 2022.

[7] Kerem Ciftci and Klaus Hackl. A physics-informed gan framework based on model-free data-driven computational mechanics. *Computer Methods in Applied Mechanics and Engineering*, 424:116907, 2024.

[8] Liu Yang, Dongkun Zhang, and George Em Karniadakis. Physics-informed generative adversarial networks for stochastic differential equations. *SIAM Journal on Scientific Computing*, 42(1): A292–A317, 2020.

[9] Yanjie Song, He Wang, He Yang, Maria Luisa Taccari, and Xiaohui Chen. Loss-attentional physics-informed neural networks. *Journal of Computational Physics*, 501:112781, 2024.

[10] Yuandong Cao, Chi Chiu So, Yifan Dai, Siu Pang Yung, and Jun-Min Wang. Adversarial physics-informed neural networks with hard constraints for optimal control of pdes. *Journal of Computational Physics*, page 114307, 2025.

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[12] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. *Advances in neural information processing systems*, 29, 2016.

[13] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2794–2802, 2017.

[14] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

[15] Jean-Yves Franceschi, Emmanuel De Bézenac, Ibrahim Ayed, Mickaël Chen, Sylvain Lamprier, and Patrick Gallinari. A neural tangent kernel perspective of gans. In *International Conference on Machine Learning*, pages 6660–6704. PMLR, 2022.

[16] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. Pmlr, 2017.

[17] Isaac E Lagaris, Aristidis Likas, and Dimitrios I Fotiadis. Artificial neural networks for solving ordinary and partial differential equations. *IEEE transactions on neural networks*, 9(5): 987–1000, 1998.

[18] Cedric Flamant, Pavlos Protopapas, and David Sondak. Solving differential equations using neural network solution bundles. *arXiv preprint arXiv:2006.14372*, 2020.

[19] Levi D McClenny and Ulisses M Braga-Neto. Self-adaptive physics-informed neural networks. *Journal of Computational Physics*, 474:111722, 2023.

[20] Yaohua Zang, Gang Bao, Xiaojing Ye, and Haomin Zhou. Weak adversarial networks for high-dimensional partial differential equations. *Journal of Computational Physics*, 411:109409, 2020.

[21] Weiye Gan, Yicheng Li, Qian Lin, and Zuoqiang Shi. Neural tangent kernel of neural networks with loss informed by differential operators. *arXiv preprint arXiv:2503.11029*, 2025.

[22] Zijian Zhou and Zhenya Yan. Is the neural tangent kernel of pinns deep learning general partial differential equations always convergent? *Physica D: Nonlinear Phenomena*, 457:133987, 2024.

[23] Luís Carvalho, João L Costa, José Mourão, and Gonçalo Oliveira. The positivity of the neural tangent kernel. *SIAM Journal on Mathematics of Data Science*, 7(2):495–515, 2025.

Appendix Contents

A Preliminary Knowledge **15**

A.1 Generative adversarial networks 15

A.2 Physics-informed neural network 15

A.3 Solving Partial Differential Equations via Adversarially Trained PINNs 16

A.4 Background on the neural tangent kernel regime 17

B Detailed derivations of discriminator and generator dynamics **18**

B.1 Discriminator dynamics in parameter space 19

B.2 Functional-space notation for discriminator training 20

B.3 Discriminator analysis under alternating training: an NTK interpretation in residual space 21

B.3.1 The IPM case: kernel witness function on residual samples 22

B.3.2 The LSGAN case: kernel-smoothed regression target for our choice of labels 23

B.3.3 The vanilla GAN case: kernel-smoothed nonlinear classifier in residual space 25

B.4 Generator residual dynamics 27

B.4.1 Specialization to common adversarial objectives 29

B.4.2 Interpretation in our framework 30

B.5 The effect of the PDE operator on the NTK spectrum 30

C Analysis of the residual-energy decay **31**

C.1 Residual-energy law and constant-kernel approximation 31

C.2 Spectral factorization of the aligned dynamics 32

C.3 Modal view and slower decay under weaker alignment 33

C.4 Failure modes: ascent and stagnation 35

C.5 Discrete-time interpretation for the alternating algorithm 36

D Different Dynamics Induced by Residual Input and Squared-Residual Input **36**

D.1 Generator dynamics with squared-residual input 37

D.2 Spectral interpretation 38

D.3 Interpretation and implications 42

E Soft-Constrained PINNs under Adversarial Training **42**

E.1 Problem setup and violation channels 42

E.2 Adversarial formulation with three discriminators 43

E.3 Generator gradient flow and channel-wise adversarial weights 44

E.4 Coupled violation dynamics and block kernel structure 45

E.5 Diagonal-weight representation 46

E.6 Energy dynamics 47

E.7 Constant-kernel regime and NTK interpretation 47

E.8 Discriminator-side functional dynamics 48

F Existing Adaptive and Weak-Adversarial PINNs Methods as Constrained-Discriminator Special Cases 48

F.1 General constrained-discriminator formulation 49
F.2 SA-PINN as a monotone masked discriminator 49
F.3 LA-PINN as a linear attentional discriminator 50
F.4 Weak-adversarial methods as projection-type discriminators 51
F.5 Unified special-case theorem 51

G A broader design space for adversarial PINNs training / Future work 52

G.1 Dimension I: residual input versus squared-residual input 53
G.2 Dimension II: discriminator objective and functional class 53
G.3 Why SA-PINN, LA-PINN, and WAN each have distinctive advantages 54
G.4 A two-dimensional taxonomy of adversarial PINNs training strategies 54
G.5 Summary 54

H Experiments 55

H.1 Experimental setup details 55
H.1.1 Compared methods 55
H.1.2 Benchmark equations 55
H.1.3 Network architectures 56
H.1.4 Optimization settings 56
H.1.5 Training protocol 57
H.1.6 Sampling Strategy and Training–Validation Relationship 57
H.1.7 Evaluation metrics 58
H.1.8 Relation to the main text 58
H.2 Benchmark equations Comparison 58
H.2.1 Laplace 58
H.2.2 Poisson 60
H.2.3 Reaction-Difussion 61
H.2.4 Viscous-Burgers 62
H.2.5 Klein-Gordon 63
H.3 Ablation experiment 64
H.3.1 Laplace equation (Under different boundary conditions) 64
H.3.2 Compatibility with DEQGAN 66
H.3.3 Does rollback help beyond increasing the update budget? 68

I Contributions, Limitations, and Future Work 71

J Two-Dimensional Heatmap Comparison of GAN-Based Solutions 73

A Preliminary Knowledge

A.1 Generative adversarial networks

Generative Adversarial Networks (GANs) provide a powerful framework for learning probability distributions through an adversarial game between a generator and a discriminator. In this work, we adopt a unified functional perspective that allows several popular GAN variants to be represented under a common formulation.

Let α_r denote the real data distribution and α_g^θ denote the generated distribution induced by the generator $g(z;\theta)$, where $z \sim p_z$ is a latent variable. The discriminator is represented by a function $f(\cdot\,;\phi) : \mathcal{X} \to \mathbb{R}$. Here, θ and ϕ denote the trainable parameters of the generator and discriminator networks, respectively.

We consider the following general adversarial objective:

$$\max_{\phi} \mathcal{L}(\phi) = \mathbb{E}_{x\sim\alpha_r}\big[P(f(x;\phi))\big] + \mathbb{E}_{x\sim\alpha_g^\theta}\big[Q(f(x;\phi))\big] \tag{31}$$

$$\min_{\theta} \mathcal{L}(\theta) = \mathbb{E}_{x\sim\alpha_g^\theta}\big[R(f(x;\phi))\big] \tag{32}$$

where $P(\cdot)$, $Q(\cdot)$ and $R(\cdot)$ are scalar-valued functions that determine the specific adversarial divergence or discrepancy being minimized between α_r and α_g^θ. This formulation encompasses a broad family of GAN objectives depending on the choice of P, Q and R.

Vanilla GAN. The vanilla GAN objective corresponds to the minimization of the Jensenv–Shannon divergence between the real and generated distributions. This is recovered by choosing

$$P(t) = \log\sigma(t), \quad Q(t) = R(t) = \log\big(1-\sigma(t)\big), \tag{33}$$

where $\sigma(\cdot)$ denotes the sigmoid function.

Least-Squares GAN (LSGAN). The least-squares GAN replaces the logistic loss with a quadratic penalty, which improves gradient stability and alleviates saturation. This corresponds to choosing

$$P(t) = -\frac{1}{2}(t-1)^2, \quad Q(t) = -\frac{1}{2}t^2, \quad R(t) = \frac{1}{2}(t-1)^2. \tag{34}$$

Integral Probability Metric (IPM) GANs. IPM-based GANs measure distribution discrepancy via a supremum over a constrained function class $\mathcal{F}$. In this setting, the adversarial objective becomes

$$\min_{\theta} \max_{f(x;\phi)\in\mathcal{F}} \mathbb{E}_{x\sim\alpha_r} f(x;\phi) - \mathbb{E}_{x\sim\alpha_g^\theta} f(x;\phi). \tag{35}$$

This formulation is recovered from 31-32 by choosing

$$P(t) = t, \quad Q(t) = R(t) = -t. \tag{36}$$

together with structural constraints on f, such as Lipschitz continuity. Prominent examples include Wasserstein GANs (WGAN) and Wasserstein GANs with gradient penalty (WGAN-GP). At this point, f is often referred to as a critic in the literature. For consistency and ease of understanding, we will uniformly refer to it as the discriminator throughout this paper.

A.2 Physics-informed neural network

Physics-Informed Neural Networks (PINNs) provide a framework for solving partial differential equations (PDEs) by embedding governing physical laws into neural network training. Instead of relying solely on labeled data, PINNs incorporate PDE constraints through automatic differentiation.

Let $\Omega \subset \mathbb{R}^d$ denote a spatial–temporal domain with boundary $\partial\Omega$. We consider a general partial differential equation (PDE) expressed in operator form

$$\mathcal{N}(u)(x) = a(x), \quad x \in \Omega, \tag{37}$$

where $u : \Omega \to \mathbb{R}$ represents the unknown solution field, $\mathcal{N}$ denotes a differential operator involving spatial and/or temporal derivatives of u, and $a(x)$ is a prescribed source or forcing term defined on Ω. The governing equation is complemented by boundary and/or initial conditions given by

$$\mathcal{B}(u)(x) = b(x), \quad x \in \partial\Omega, \tag{38}$$

where $\mathcal{B}$ is a boundary or initial operator specifying constraints on the solution, such as Dirichlet, Neumann, or Robin conditions, and $b(x)$ denotes the corresponding prescribed data on the boundary or initial manifold. In physics-informed neural networks (PINNs), the unknown solution $u(x)$ is approximated by a neural network parameterized by θ, namely $u_\theta(x) \approx u(x)$, allowing the differential operators $\mathcal{N}$ and $\mathcal{B}$ to be evaluated through automatic differentiation during the training process.

Physics-informed loss construction. Let μ_Ω denote a sampling measure over the interior domain Ω, and $\mu_{\partial\Omega}$ denote a sampling measure over the boundary $\partial\Omega$. The PINNs objective can be written as

$$\mathcal{L}_{PINN}(\theta) = \lambda_r \, \mathbb{E}_{x\sim\mu_\Omega} \left[\mathcal{N}[u_\theta](x) - a(x)\right] + \lambda_b \, \mathbb{E}_{x\sim\mu_{\partial\Omega}} \left[\mathcal{B}[u_\theta](x) - b(x)\right]. \tag{39}$$

The expectations are approximated in practice via Monte Carlo sampling over collocation and boundary points.

Hard constraint enforcement. Besides penalty-based enforcement, physical constraints can also be incorporated directly into the network architecture through hard-constrained parameterizations. Instead of penalizing boundary or initial condition violations, the solution ansatz is parameterized as

$$u_\theta(x) = \mathcal{G}(x) + \mathcal{H}(x)\, \tilde{u}_\theta(x), \tag{40}$$

where $\mathcal{G}(x)$ satisfies the prescribed boundary or initial conditions, and $\mathcal{H}(x)$ is a function that vanishes on $\partial\Omega$. The neural network $\tilde{u}_\theta(x)$ represents the unconstrained component of the solution.

By construction, the resulting approximation $u_\theta(x)$ satisfies the constraints exactly for all parameter values θ. Consequently, the training objective reduces to minimizing the interior PDE residual

$$\mathcal{L}_{PINN}(\theta) = \mathbb{E}_{x\sim\mu_\Omega} \left[\mathcal{N}[u_\theta](x) - a(x)\right]. \tag{41}$$

Hard constraint formulations often improve training stability and reduce sensitivity to loss weighting parameters, although they may introduce additional complexity in constructing suitable constraint functions. For clarity and conciseness in the theoretical derivation and exposition, we focus primarily on the hard-constrained formulation of PINNs (41) in the subsequent analysis.

A.3 Solving Partial Differential Equations via Adversarially Trained PINNs

To mitigate imbalance issues in physics-informed learning, we introduce an adversarial training framework for PINNs based on the vanilla GAN formulation.

Let the PDE residual evaluated at collocation point $\{x_i\}_{i=1}^{N_r}$ be defined as $\mathcal{R}(x_i;\theta) := \mathcal{N}[u_\theta](x_i) - a(x_i)$ and $\mathcal{T}(x_i) := 0$. Here, $\mathcal{R}(\cdot;\theta)$ denotes the PDE residual function induced by the generator parameterized by θ, and is interpreted as the source of generated data in the adversarial framework. In contrast, $\mathcal{T}$ denotes the target residual map corresponding to the ideal zero-residual state. The collocation points $\{x_i\}_{i=1}^{N_r} \subset \Omega$ are sampled from the interior sampling measure μ_Ω, where N_r is the number of residual collocation points and Ω is the interior domain of the PDE. For notational convenience, define

$$r_i := \mathcal{R}(x_i;\theta), \qquad i = 1, \ldots, N_r, \tag{42}$$

and collect these residual values into the residual vector

$$\mathbf{r} := (r_1, r_2, \ldots, r_{N_r})^\top \in \mathbb{R}^{N_r}. \tag{43}$$

Let

$$\hat{\mu}_\Omega := \frac{1}{N_r} \sum_{i=1}^{N_r} \delta_{x_i} \tag{44}$$

denote the empirical measure associated with the collocation points, where δ_{x_i} is the Dirac measure centered at x_i. Then the generated residual distribution is given by the pushforward empirical measure

$$\hat{\mu}_\theta^r := (\mathcal{R}_\theta)_{\#}\hat{\mu}_\Omega = \frac{1}{N_r} \sum_{i=1}^{N_r} \delta_{r_i}, \tag{45}$$

where $\mathcal{R}_\theta : x \mapsto \mathcal{R}(x;\theta)$ denotes the residual mapping, and $(\mathcal{R}_\theta)_{\#}\hat{\mu}_\Omega$ denotes the pushforward of the empirical collocation measure under this map. Therefore, the generated data distribution can be viewed as a finite Dirac mixture supported on the residual samples $\{r_i\}_{i=1}^{N_r}$, which serves as a discrete approximation of the pushforward measure $(\mathcal{R}_\theta)_{\#}\mu_\Omega$. In contrast, the real data distribution associated with the ideal PDE-satisfying state is concentrated at zero and is thus represented by the Dirac measure

$$\mu_{\text{real}} = \delta_0. \tag{46}$$

Adversarial objective. We formulate the adversarial training between the solution network u_θ and the discriminator network $f(\cdot;\phi)$ within a general adversarial learning framework. Let $\mathcal{R}(x;\theta)$ denote the generated PDE residual induced by the solution network and $\mathcal{T}(x)$ represent the target residual distribution. The generator (solution network) is trained to minimize

$$\mathcal{L}(\theta) = \frac{1}{N_r}\sum_{i=1}^{N_r} R\Bigg(f\Big(\mathcal{R}\left(x_i;\theta\right);\phi\Big)\Bigg), \tag{47}$$

while the discriminator aims to distinguish target samples from generated residuals by maximizing

$$\mathcal{L}(\phi) = \frac{1}{N_r}\sum_{i=1}^{N_r}\left[P\Bigg(f\Big(\mathcal{T}\left(x_i\right);\phi\Big)\Bigg) + Q\Bigg(f\Big(\mathcal{R}\left(x_i;\theta\right);\phi\Big)\Bigg)\right]. \tag{48}$$

where $P(\cdot)$, $Q(\cdot)$, and $R(\cdot)$ are scalar loss functions defined in (33), (34) and (36). Different choices of these functions recover various GAN formulations.

In the general setting, the real data correspond to the zero-residual target. For the original GAN and LSGAN formulations, the discriminator is trained to output 1 when the input equals the real data (i.e., $\mathcal{T}(x_i) = 0$) and 0 when the input corresponds to generated residual samples ($\mathcal{R}(x_i;\theta)$). In contrast, under the WGAN formulation, the discriminator (critic) is trained to assign larger values to the real data (zero residual) and smaller values to generated residual samples, without restricting its output to the interval $[0, 1]$. Meanwhile, the generator seeks to reduce the PDE residual magnitude so that the generated samples become indistinguishable from the zero-residual target, thereby enforcing the governing equation.

A.4 Background on the neural tangent kernel regime

This appendix recalls the standard assumptions and conclusions of the neural tangent kernel (NTK) regime used in our analysis. The goal is not to rederive NTK theory from scratch, but to state clearly the conditions under which later kernel-based generator and discriminator dynamics are justified. The presentation follows the standard NTK viewpoint established for wide neural networks, where the network output becomes approximately linearized around initialization and the corresponding kernel remains nearly constant during training [14].

Let $g(x;\theta)$ denote a neural network with input $x \in \mathcal{X}$ and parameters $\theta \in \mathbb{R}^P$. We are interested in the regime where gradient-based training of g can be described directly at the level of its output function, rather than only through the nonlinear parameter trajectory.

Assumption 2 (Network architecture)**.** *The network is a standard architecture, such as a fully connected, convolutional, or residual network, with standard hidden layers and affine transformations.*

Assumption 3 (Activation regularity)**.** *The activation function is standard and sufficiently smooth. The activation can be any standard function: tanh, softplus, sigmoid, Gaussian, etc.*

Assumption 4 (Nondegenerate bias)**.** *Linear layers contain nonzero bias terms, which is standard in NTK formulations and helps avoid degenerate kernels in certain settings.*

Assumption 5 (Initialization)**.** *The network parameters are randomly initialized with a standard width-dependent scaling. In the usual NTK parameterization, the preactivations remain of order one as the hidden widths grow, which ensures a nontrivial infinite-width limit. Under such initialization, the network output at initialization converges in law to a Gaussian process.*

Assumption 6 (Infinite-width limit)**.** *The hidden widths $n_1, \ldots, n_{L-1}$ tend to infinity. In this regime, fluctuations due to finite width vanish, and both the network output at initialization and the NTK admit deterministic limiting descriptions.*

Under these assumptions, the network output admits the first-order approximation

$$g(x;\theta(t)) \approx g(x;\theta_0) + \nabla_\theta g(x;\theta_0)^\top \big(\theta(t) - \theta_0\big). \tag{49}$$

This approximation is the basis of the NTK viewpoint: the nonlinear network is replaced by a linearized model around initialization, while the training dynamics are encoded by a kernel.

The corresponding neural tangent kernel is defined by

$$k(x,x') := \langle \nabla_\theta g(x;\theta_0), \nabla_\theta g(x';\theta_0) \rangle . \tag{50}$$

For a finite set of points $\{x_i\}_{i=1}^N$, the associated NTK Gram matrix is

$$K = \big[k(x_i,x_j)\big]_{i,j=1}^N. \tag{51}$$

The central NTK conclusions can then be summarized as follows.

Theorem 2 (Standard NTK regime [14])**.** *Under Assumptions 2–6, the following hold:*

1. ***Gaussian-process initialization.*** *The random function $g(\cdot;\theta_0)$ converges in law to a Gaussian process.*
2. ***Deterministic limiting NTK.*** *The random NTK $k(x,x';\theta_0)$ converges to a deterministic limiting kernel $k_\infty(x,x')$.*
3. ***Kernel stability during training.*** *Along training, the NTK remains approximately constant:*
$$k(x,x';\theta(t)) \approx k(x,x';\theta_0) \approx k_\infty(x,x'). \tag{52}$$
4. ***Output dynamics in function space.*** *If $\mathbf{g}(t) = [g(x_1;\theta(t)),\dots,g(x_N;\theta(t))]^\top$ denotes the vector of network outputs on training points and $\mathcal{L}(\mathbf{g})$ is the loss viewed as a function of the outputs, then under gradient flow one has*
$$\dot{\mathbf{g}}(t) \approx -K\,\nabla_{\mathbf{g}}\mathcal{L}(\mathbf{g}(t)), \tag{53}$$
where K is the limiting NTK Gram matrix.

Interpretation. The theorem states that, in the wide-network regime, training can be analyzed directly in function space. The nonlinear parameter dynamics are replaced by a kernel-preconditioned gradient flow on the outputs. This is precisely why the NTK framework is useful for studying residual dynamics, modewise convergence, and kernel-induced weighting mechanisms.

Hence the convergence rate is governed by the spectrum of the kernel matrix K: components associated with large eigenvalues decay faster, while components associated with small eigenvalues decay more slowly. This spectral viewpoint is one of the key motivations for using NTK analysis in training-dynamics problems.

Scope of the approximation. The NTK regime is an asymptotic and mechanism-level approximation. Its justification relies on wide hidden layers, standard smooth parameterization, and small parameter drift. Therefore, it is not a complete description of all finite-width training phenomena. Nevertheless, it provides a mathematically tractable and often accurate description of the early and intermediate training dynamics of overparameterized neural networks, which is why it serves as the theoretical basis for the kernel-based analysis in this work.

Relevance for the present paper. In the sequel, this NTK viewpoint will be applied separately to the generator and the discriminator. On the generator side, it yields a kernel description of residual evolution. On the discriminator side, it motivates a function-space flow under alternating training. The resulting interaction between these two kernel-induced dynamics is the basis of our later residual-space analysis.

B Detailed derivations of discriminator and generator dynamics

In this appendix, we provide the detailed derivations of the discriminator and generator dynamics used in the main text. Throughout, we follow the notation introduced in Section 2. Recall that the adversarial objective is written in the unified form

$$\mathcal{L}_D(\phi) = \frac{1}{N_r}\sum_{i=1}^{N_r}\Big[P\big(f(\mathcal{T}(x_i);\phi)\big) + Q\big(f(\mathcal{R}(x_i;\theta);\phi)\big)\Big], \tag{54}$$

for the discriminator, and

$$\mathcal{L}_G(\theta) = \frac{1}{N_r}\sum_{i=1}^{N_r} R\big(f(\mathcal{R}(x_i;\theta);\phi)\big) \tag{55}$$

for the generator.

Under alternating optimization, gradient ascent is applied to ϕ and gradient descent is applied to θ. The corresponding gradients are

$$\frac{\partial \mathcal{L}_G(\theta)}{\partial \theta} = \frac{1}{N_r}\sum_{i=1}^{N_r} R'\Big(f(\mathcal{R}(x_i;\theta);\phi)\Big)\,\frac{\partial f(\mathcal{R}(x_i;\theta);\phi)}{\partial \mathcal{R}(x_i;\theta)}\,\frac{\partial \mathcal{R}(x_i;\theta)}{\partial \theta(t)}, \tag{56}$$

and

$$\frac{\partial \mathcal{L}_D(\phi)}{\partial \phi} = \frac{1}{N_r}\sum_{i=1}^{N_r}\left[P'\Big(f(\mathcal{T}(x_i);\phi)\Big)\frac{\partial f(\mathcal{T}(x_i);\phi)}{\partial \phi(t)} + Q'\Big(f(\mathcal{R}(x_i;\theta);\phi)\Big)\frac{\partial f(\mathcal{R}(x_i;\theta);\phi)}{\partial \phi(t)}\right]. \tag{57}$$

Hence the discrete parameter updates are

$$\theta(n+1) = \theta(n) - a(n)\frac{\partial \mathcal{L}_G(\theta)}{\partial \theta}, \tag{58}$$

$$\phi(n+1) = \phi(n) + b(n)\frac{\partial \mathcal{L}_D(\phi)}{\partial \phi}, \tag{59}$$

where $a(n)$ and $b(n)$ denote the generator and discriminator learning rates, respectively. Passing formally to the continuous-time limit yields the gradient-flow system

$$\dot{\phi}(t) = \frac{\partial \mathcal{L}_D(\phi(t))}{\partial \phi(t)}, \tag{60}$$

$$\dot{\theta}(t) = -\frac{\partial \mathcal{L}_G(\theta(t))}{\partial \theta(t)}. \tag{61}$$

B.1 Discriminator dynamics in parameter space

We first derive the discriminator evolution. For a fixed generator state θ, we have defined the residual samples

$$r_i := \mathcal{R}(x_i;\theta), \qquad i = 1,\dots,N_r. \tag{62}$$

Then (60) becomes

$$\dot{\phi}(t) = \frac{1}{N_r}\sum_{i=1}^{N_r}\left[P'\big(f(\mathcal{T}(x_i);\phi(t))\big)\,\frac{\partial f(\mathcal{T}(x_i);\phi(t))}{\partial \phi(t)} + Q'\big(f(r_i;\phi(t))\big)\,\frac{\partial f(r_i;\phi(t))}{\partial \phi(t)}\right]. \tag{63}$$

This is the discriminator gradient flow in parameter space.

Let

$$k_t^D(x,y) := \left\langle \frac{\partial f(x;\phi(t))}{\partial \phi(t)}, \frac{\partial f(y;\phi(t))}{\partial \phi(t)} \right\rangle \tag{64}$$

denote the discriminator NTK. Under the standard Assumptions 2-6, this kernel remains approximately constant during training:

$$k_t^D(x,y) \approx k^D(x,y). \tag{65}$$

For any input x in the discriminator input space, the time derivative of the discriminator output is

$$\begin{aligned}
\frac{d}{dt}f(x;\phi(t)) &= \left\langle \frac{\partial f(x;\phi(t))}{\partial \phi(t)}, \dot{\phi}(t) \right\rangle \\
&= \left\langle \frac{\partial f(x;\phi(t))}{\partial \phi(t)}, \frac{1}{N_r}\sum_{i=1}^{N_r}\Big[P'\big(f(\mathcal{T}(x_i);\phi(t))\big)\frac{\partial f(\mathcal{T}(x_i);\phi(t))}{\partial \phi(t)} + Q'\big(f(r_i;\phi(t))\big)\frac{\partial f(r_i;\phi(t))}{\partial \phi(t)}\Big] \right\rangle \\
&= \frac{1}{N_r}\sum_{i=1}^{N_r}\Big[P'\big(f(\mathcal{T}(x_i);\phi(t))\big)\,k_t^D\big(x,\mathcal{T}(x_i)\big) + Q'\big(f(r_i;\phi(t))\big)\,k_t^D(x,r_i)\Big].
\end{aligned} \tag{66}$$

Under the constant-NTK approximation, this reduces to

$$\frac{d}{dt}f(x;\phi(t)) = \frac{1}{N_r}\sum_{i=1}^{N_r}\Big[P'\big(f(\mathcal{T}(x_i);\phi(t))\big)\,k^D\big(x,\mathcal{T}(x_i)\big) + Q'\big(f(r_i;\phi(t))\big)\,k^D(x,r_i)\Big]. \quad (67)$$

In particular, evaluating at $x = r_j$ yields

$$\frac{d}{dt}f(r_j;\phi(t)) = \frac{1}{N_r}\sum_{i=1}^{N_r}\Big[P'\big(f(\mathcal{T}(x_i);\phi(t))\big)\,k^D\big(r_j,\mathcal{T}(x_i)\big) + Q'\big(f(r_i;\phi(t))\big)\,k^D(r_j,r_i)\Big]. \quad (68)$$

Integrating (67) from 0 to t, we obtain the representation

$$f(x;\phi(t)) = f_0(x) + \int_0^t \frac{1}{N_r}\sum_{i=1}^{N_r}\Big[P'\big(f(\mathcal{T}(x_i);\phi(s))\big)\,k^D\big(x,\mathcal{T}(x_i)\big) + Q'\big(f(r_i;\phi(s))\big)\,k^D(x,r_i)\Big]\,ds, \quad (69)$$

where $f_0(x) := f(x;\phi(0))$.

B.2 Functional-space notation for discriminator training

Before deriving the discriminator dynamics in function space, we briefly introduce the notation that will be used in the following subsection. The purpose of this step is to rewrite the discriminator update not only at the parameter level ϕ, but also at the level of the discriminator function $f(\cdot;\phi)$ acting on the residual domain.

Residual-domain training measure. For a fixed generator state θ, the discriminator is trained on two types of samples in the residual domain:

- the real target samples $\mathcal{T}(x_i) = 0$,
- the generated residual samples $\mathcal{R}(x_i;\theta) = r_i$.

Accordingly, we introduce the empirical training measure on the residual domain,

$$\hat{\gamma}_\theta^r := \frac{1}{2N_r}\sum_{i=1}^{N_r}\delta_0 + \frac{1}{2N_r}\sum_{i=1}^{N_r}\delta_{r_i}, \qquad r_i = \mathcal{R}(x_i;\theta). \quad (70)$$

Thus, $\hat{\gamma}_\theta^r$ is the empirical measure supported on the real zero state and the generated residual samples. It plays the role of the discriminator-side training distribution in residual space.

Discriminator as a function on residual space. Instead of only viewing the discriminator through its parameter vector ϕ, we may also regard it as a time-dependent function

$$f_t(\cdot) := f(\cdot;\phi(t)), \quad (71)$$

defined on the residual variable $r \in \mathbb{R}$. In this viewpoint, the discriminator training flow (60) induces an evolution equation for the function f_t itself.

Functional gradient. Let $\mathcal{L}_D(f)$ denote the discriminator objective viewed as a functional of the discriminator function f. Its gradient with respect to the empirical measure $\hat{\gamma}_\theta^r$, denoted by

$$\nabla_{\hat{\gamma}_\theta^r}\mathcal{L}_D(f),$$

is the function on the support of $\hat{\gamma}_\theta^r$ such that, for every perturbation $\psi \in L^2(\hat{\gamma}_\theta^r)$,

$$\frac{d}{d\varepsilon}\mathcal{L}_D(f+\varepsilon\psi)\Big|_{\varepsilon=0} = \Big\langle \nabla_{\hat{\gamma}_\theta^r}\mathcal{L}_D(f),\,\psi\Big\rangle_{L^2(\hat{\gamma}_\theta^r)}. \quad (72)$$

In other words, $\nabla_{\hat{\gamma}_\theta^r}\mathcal{L}_D(f)$ is the discriminator-side functional gradient evaluated on the empirical residual training measure.

Discriminator NTK and RKHS. Assume now that the discriminator is in the infinite-width or constant-NTK regime, and let $k^D(r, r')$ denote its Neural Tangent Kernel on the residual domain. Associated with k^D and the empirical measure $\hat{\gamma}_\theta^r$, one defines the reproducing kernel Hilbert space (RKHS), denoted by

$$\mathcal{H}_{k^D}^{\hat{\gamma}_\theta^r},$$

which is the function space generated by the kernel sections $k^D(r, \cdot)$ over points r in the support of $\hat{\gamma}_\theta^r$.

Kernel integral operator. The corresponding kernel integral operator

$$T_{k^D, \hat{\gamma}_\theta^r} : L^2(\hat{\gamma}_\theta^r) \to \mathcal{H}_{k^D}^{\hat{\gamma}_\theta^r}$$

is defined by

$$\big(T_{k^D, \hat{\gamma}_\theta^r} h\big)(r) := \int k^D(r, \tilde{r})\, h(\tilde{r})\, d\hat{\gamma}_\theta^r(\tilde{r}). \tag{73}$$

Since $\hat{\gamma}_\theta^r$ is empirical, this becomes

$$\big(T_{k^D, \hat{\gamma}_\theta^r} h\big)(r) = \frac{1}{2N_r} \sum_{i=1}^{N_r} k^D(r, 0)\, h(0) + \frac{1}{2N_r} \sum_{i=1}^{N_r} k^D(r, r_i)\, h(r_i). \tag{74}$$

Therefore, $T_{k^D, \hat{\gamma}_\theta^r}$ maps a function defined only on the discrete training support into a kernel-smoothed function on the whole residual domain.

Interpretation. The key point is that the discriminator loss is only evaluated on the empirical support $\{0, r_1, \ldots, r_{N_r}\}$, whereas the generator requires the input derivative $\partial_r f_t(r)$ of the trained discriminator as a genuine function of the residual variable. The kernel operator $T_{k^D, \hat{\gamma}_\theta^r}$ provides precisely this extension: it turns the empirical discriminator training signal into a smooth function on residual space, shaped by the architecture-dependent NTK k^D.

With these notations in place, the next subsection rewrites discriminator training as a functional gradient flow in residual space.

B.3 Discriminator analysis under alternating training: an NTK interpretation in residual space

We adapt the discriminator-side NTK viewpoint of alternating GAN training to our residual-space formulation of adversarial PINNs. Concretely, the general NTK function-space flow of the discriminator is adapted from the discriminator-side NTK framework of Franceschi et al. [15], while our contribution here is to rewrite and interpret it in the present residual setting, where the generated distribution is the empirical residual measure $\hat{\mu}_\theta^r$ and the real distribution is the zero state δ_0. This residual-space reinterpretation is what allows the discriminator flow to be connected directly to the generator-side residual dynamics studied in the main text. With the functional-space notation introduced above, the discriminator training dynamics under a fixed generator state can be written as

$$\partial_t f_t = T_{k^D, \hat{\gamma}_\theta^r}\Big(\nabla_{\hat{\gamma}_\theta^r} \mathcal{L}_D(f_t)\Big). \tag{75}$$

This equation shows that, under alternating optimization, the discriminator should not be viewed as an arbitrary optimizer on the empirical residual samples, but rather as a finite-time function-space flow shaped jointly by the empirical residual training measure $\hat{\gamma}_\theta^r$ and the discriminator NTK k^D. In particular, the discriminator architecture enters generator training through the kernel operator in (75), and therefore directly affects the gradient field induced on residual space.

A key consequence of (75) is that the trained discriminator is uniquely determined by its initial state and admits the representation

$$f_t = f_0 + T_{k^D, \hat{\gamma}_\theta^r}\left(\int_0^t \nabla_{\hat{\gamma}_\theta^r} \mathcal{L}_D(f_s)\, ds\right). \tag{76}$$

Hence the increment $f_t - f_0$ lies in the RKHS generated by k^D, which means that discriminator training amounts to adding to the initial network a kernel-smoothed correction determined by the

empirical residual samples and the real zero state. This observation is important in our setting because the discriminator loss is only evaluated on the discrete support $\{0, r_1, \dots, r_{N_r}\}$, whereas the generator dynamics require the input derivative $\partial_r f_t(r)$ as a function defined on the whole residual domain.

Under standard architectural assumptions, the discriminator NTK is differentiable, and therefore so is the trained discriminator f_t. As a result, the derivative field $\partial_r f_t(r)$ appearing in the generator-side residual dynamics is well-defined. This resolves the gradient-indeterminacy issue that would arise if one treated the empirical discriminator as an unconstrained optimum only defined on a discrete support. In our framework, the discriminator derivative is instead a finite-time, NTK-smoothed quantity determined by both the sample distribution and the discriminator architecture.

This interpretation fits directly into the generator dynamics developed in the main text (24). Once the discriminator is frozen after its substep, the residual evolution of the generator takes the form

$$\dot{\mathbf{r}} = K_{rr}^{G}(\theta)\, \Gamma^{D}\, \mathbf{1}, \tag{77}$$

where the diagonal entries of Γ^D depend on (23):

$$\gamma_i = -\frac{1}{N_r}\, R'\big(f_t(r_i)\big)\, \partial_r f_t(r_i).$$

Therefore, the discriminator-induced weighting in adversarial PINNs is not produced by an abstract optimal critic, but by the finite-time residual-space vector field generated by (75). From this viewpoint, the discriminator plays the role of an architecture-dependent smoothing and weighting operator acting on the empirical residual distribution.

B.3.1 The IPM case: kernel witness function on residual samples

We now reinterpret the general discriminator-side NTK framework of [15] in our residual-space setting. The key point here is that the discriminator behaves as a kernel witness function measuring the discrepancy between the empirical residual distribution and the zero target. This makes the resulting generator-side weighting especially explicit.

Proposition 3 (IPM discriminator flow as a kernel witness function on residual space)**.** *Fix the generated residual distribution $\hat{\mu}_\theta^r$ and the target δ_0. In the IPM case, let k^D denote the discriminator kernel. Then:*

1. *The discriminator evolves linearly in function space as*

$$f_t = f_0 + t\, f_\theta^*, \tag{78}$$

   where f_θ^* is the unnormalized MMD witness function associated with k^D.
2. *In residual-space notation, the witness function is given by*

$$f_\theta^*(r) = -\mathbb{E}_{\tilde{r}\sim\hat{\mu}_\theta^r}\big[k^D(\tilde{r}, r)\big] + k^D(0, r), \tag{79}$$

   or equivalently

$$f_\theta^*(r) = -\frac{1}{N_r}\sum_{i=1}^{N_r} k^D(r_i, r) + k^D(0, r). \tag{80}$$

3. *The induced generator-side sample weighting is*

$$\gamma_i^{\mathrm{IPM}} = \frac{1}{N_r}\, \partial_r f_t(r_i) = \frac{1}{N_r}\, \partial_r f_0(r_i) + \frac{t}{N_r}\, \partial_r f_\theta^*(r_i). \tag{81}$$

Proof. In the IPM setting, the discriminator-side NTK dynamics admit an explicit linear form in function space. For fixed $\hat{\mu}_\theta^r$ and target δ_0, the discriminator moves in the direction of the kernel witness function associated with the discrepancy between the generated residual distribution and the zero target. Therefore,

$$f_t = f_0 + t\, f_\theta^*, \tag{82}$$

which proves (78).

By definition, the corresponding unnormalized witness function is

$$f_\theta^*(r) = -\mathbb{E}_{\tilde{r}\sim\hat{\mu}_\theta^r}\big[k^D(\tilde{r}, r)\big] + k^D(0, r), \tag{83}$$

which proves (79). Since

$$\hat{\mu}_\theta^r = \frac{1}{N_r}\sum_{i=1}^{N_r}\delta_{r_i},$$

the empirical representation (80) follows immediately:

$$f_\theta^*(r) = -\frac{1}{N_r}\sum_{i=1}^{N_r} k^D(r_i, r) + k^D(0, r). \tag{84}$$

Finally, in the IPM-GAN case, the generator-side weighting is given by

$$\gamma_i^{\mathrm{IPM}} = \frac{1}{N_r}\,\partial_r f_t(r_i). \tag{85}$$

Substituting $f_t = f_0 + t\, f_\theta^*$ yields

$$\gamma_i^{\mathrm{IPM}} = \frac{1}{N_r}\,\partial_r f_0(r_i) + \frac{t}{N_r}\,\partial_r f_\theta^*(r_i), \tag{86}$$

which proves (81). □

Interpretation. Proposition 3 shows that, in the IPM case, the discriminator is most naturally interpreted as a kernel witness function on residual space. Unlike the LSGAN discriminator, which relaxes toward a regression target, or the vanilla GAN discriminator, which behaves as a nonlinear kernel classifier, the IPM discriminator directly represents the discrepancy between the empirical residual distribution and the zero target through the kernel k^D.

This makes the generator-side effect especially transparent: the discriminator induces a residual-space gradient field through $\partial_r f_t(r_i)$, and this field is precisely the derivative of the witness function. Therefore, the IPM discriminator contributes not merely a notion of sample realism, but an explicit sample-dependent preconditioning field for residual descent. In the language of this paper, this is the clearest example of how discriminator training reshapes generator dynamics through residual-space weighting.

B.3.2 The LSGAN case: kernel-smoothed regression target for our choice of labels

We next specialize the discriminator-side NTK framework to our LSGAN formulation in residual space. The main point here is not to introduce a new NTK tool, but to reinterpret the discriminator dynamics under our labeling convention: generated residual samples are assigned target value 0, while the real zero state is assigned target value 1. Under this choice, the discriminator becomes a kernel-smoothed regression field on residual space, whose derivative then induces the generator-side residual weighting.

Proposition 4 (LSGAN discriminator flow as kernel-smoothed regression on residual space)**.** *Consider the discriminator objective*

$$\mathcal{L}_D^{\mathrm{LS}}(f) = -\frac{1}{2}\,\mathbb{E}_{r\sim\hat{\mu}_\theta^r}\big[f(r)^2\big] - \frac{1}{2}\,\mathbb{E}_{y\sim\delta_0}\big[(f(y)-1)^2\big]. \tag{87}$$

Let

$$\hat{\gamma}_\theta^r = \frac{1}{2}\hat{\mu}_\theta^r + \frac{1}{2}\delta_0, \tag{88}$$

and define the regression target

$$\rho_\theta := \frac{d\delta_0}{d(\delta_0 + \hat{\mu}_\theta^r)}. \tag{89}$$

Then the following statements hold:

1. *The $L^2(\hat{\gamma}_\theta^r)$ functional gradient of $\mathcal{L}_D^{\mathrm{LS}}$ is*

$$\nabla_{\hat{\gamma}_\theta^r}\mathcal{L}_D^{\mathrm{LS}}(f) = 2(\rho_\theta - f). \tag{90}$$

2. *Consequently, the discriminator NTK flow is*

$$\partial_t f_t = 2\, T_{k^D,\hat{\gamma}_\theta^r}(\rho_\theta - f_t). \tag{91}$$

3. The resulting function-space dynamics are linear, and the solution is

$$f_t = \exp\big(-2t\, T_{k^D,\hat{\gamma}_\theta^r}\big)(f_0 - \rho_\theta) + \rho_\theta. \tag{92}$$

Equivalently, if

$$\varphi_t(x) := e^{-2tx} - 1, \tag{93}$$

then

$$f_t = f_0 + \varphi_t\big(T_{k^D,\hat{\gamma}_\theta^r}\big)(f_0 - \rho_\theta). \tag{94}$$

4. The induced generator-side sample weighting is

$$\gamma_i^{\mathrm{LSGAN}} = -\frac{1}{N_r}\big(f_t(r_i) - 1\big)\,\partial_r f_t(r_i). \tag{95}$$

Proof. Starting from (87), we rewrite the discriminator loss with respect to the empirical measure $\hat{\gamma}_\theta^r$. Since

$$\hat{\gamma}_\theta^r = \frac{1}{2}\hat{\mu}_\theta^r + \frac{1}{2}\delta_0,$$

we may express $\mathcal{L}_D^{\mathrm{LS}}$ as

$$\mathcal{L}_D^{\mathrm{LS}}(f) = \int \left[-\frac{1}{2}\frac{d\hat{\mu}_\theta^r}{d\hat{\gamma}_\theta^r} f^2 - \frac{1}{2}\frac{d\delta_0}{d\hat{\gamma}_\theta^r}(f-1)^2\right] d\hat{\gamma}_\theta^r. \tag{96}$$

Taking the functional derivative with respect to f gives

$$\begin{aligned}\nabla_{\hat{\gamma}_\theta^r}\mathcal{L}_D^{\mathrm{LS}}(f) &= -\frac{d\hat{\mu}_\theta^r}{d\hat{\gamma}_\theta^r} f - \frac{d\delta_0}{d\hat{\gamma}_\theta^r}(f-1) \\ &= -\left(\frac{d\hat{\mu}_\theta^r}{d\hat{\gamma}_\theta^r} + \frac{d\delta_0}{d\hat{\gamma}_\theta^r}\right) f + \frac{d\delta_0}{d\hat{\gamma}_\theta^r}.\end{aligned} \tag{97}$$

Since

$$\hat{\gamma}_\theta^r = \frac{1}{2}(\hat{\mu}_\theta^r + \delta_0),$$

we have

$$\frac{d\hat{\mu}_\theta^r}{d\hat{\gamma}_\theta^r} + \frac{d\delta_0}{d\hat{\gamma}_\theta^r} = 2, \qquad \frac{d\delta_0}{d\hat{\gamma}_\theta^r} = 2\rho_\theta.$$

Therefore,

$$\nabla_{\hat{\gamma}_\theta^r}\mathcal{L}_D^{\mathrm{LS}}(f) = -2f + 2\rho_\theta = 2(\rho_\theta - f), \tag{98}$$

which proves (90).

Substituting this gradient into the general discriminator function-space flow yields

$$\partial_t f_t = T_{k^D,\hat{\gamma}_\theta^r}\Big(\nabla_{\hat{\gamma}_\theta^r}\mathcal{L}_D^{\mathrm{LS}}(f_t)\Big) = 2\, T_{k^D,\hat{\gamma}_\theta^r}(\rho_\theta - f_t), \tag{99}$$

which proves (91).

Since this is a linear differential equation in function space, its solution is

$$f_t = \exp\big(-2t\, T_{k^D,\hat{\gamma}_\theta^r}\big)(f_0 - \rho_\theta) + \rho_\theta, \tag{100}$$

which proves (92). The equivalent representation (94) follows immediately from the definition $\varphi_t(x) = e^{-2tx} - 1$.

Finally, under our generator-side convention, the induced sample-wise weighting is

$$\gamma_i^{\mathrm{LSGAN}} = -\frac{1}{N_r}\big(f_t(r_i) - 1\big)\,\partial_r f_t(r_i), \tag{101}$$

which proves (95). □

Interpretation. Proposition 4 shows that, under our labeling convention, the LSGAN discriminator is best understood not as an abstract optimal critic, but as a kernel-smoothed regression function on residual space. The target ρ_θ encodes the desired regression rule induced by the real zero state and the generated residual samples. When the supports of δ_0 and $\hat{\mu}_\theta^r$ are disjoint, this rule reduces to the usual binary labeling convention: the real zero state is assigned value 1, while generated residual samples are assigned value 0. The NTK flow then drives the discriminator toward this regression target in function space, up to the null space of the kernel operator.

This interpretation is useful for adversarial PINNs because the generator does not only need discriminator values on the empirical support, but also requires the derivative field $\partial_r f_t(r)$. The kernel operator $T_{k^D,\hat{\gamma}_\theta^r}$ extends the target regression signal into a smooth function on the whole residual domain, so that $\partial_r f_t(r_i)$ is well-defined at finite training time. In this sense, the discriminator acts as a kernel-smoothed regression field whose derivative modulates the generator residual dynamics.

Compared with the IPM case, where the discriminator behaves as a witness function, the LSGAN discriminator instead behaves like a regression map relaxing toward a target field. Therefore, the essential difference is not only the nominal loss form, but the type of residual-space structure induced on the generator: witness-type geometry in the IPM case versus regression-type weighting in the LSGAN case.

Remark 8 (On the target density and convergence)**.** *The target $\rho_\theta = d\delta_0/d(\delta_0 + \hat{\mu}_\theta^r)$ should be interpreted as a Radon–Nikodym density with respect to the combined residual measure, rather than always as a pointwise binary label. In particular, if some generated residual samples satisfy $r_i = 0$, then the supports of δ_0 and $\hat{\mu}_\theta^r$ overlap, and the value of ρ_θ at zero reflects the relative mass of the real zero state and the generated residual samples located at zero. Thus, the simple $1/0$ labeling interpretation is exact only in the support-disjoint case.*

Moreover, the formula

$$f_t = \exp\big(-2t\, T_{k^D,\hat{\gamma}_\theta^r}\big)(f_0 - \rho_\theta) + \rho_\theta$$

should be understood as relaxation toward ρ_θ modulo the null space of the kernel operator. If $T_{k^D,\hat{\gamma}_\theta^r}$ is injective on the empirical support, then the discriminator converges exponentially to ρ_θ on that support. Otherwise, components of $f_0 - \rho_\theta$ lying in the null space of $T_{k^D,\hat{\gamma}_\theta^r}$ are not dissipated by the flow.

B.3.3 The vanilla GAN case: kernel-smoothed nonlinear classifier in residual space

We finally rewrite the vanilla GAN discriminator flow in the same residual-space language, so that it can be compared directly with the IPM and LSGAN cases from the viewpoint of generator-side residual weighting. In contrast to the linear regression structure of LSGAN, the key new feature here is the sigmoid nonlinearity, which turns the discriminator dynamics into a nonlinear kernel classification flow.

Proposition 5 (Vanilla GAN discriminator flow on residual space)**.** *Let the discriminator output be*

$$D_t(r) := \sigma(f_t(r)), \tag{102}$$

where $f_t(r)$ is the pre-sigmoid score. Consider the discriminator objective

$$\mathcal{L}_D^{\mathrm{GAN}}(f) = \mathbb{E}_{y\sim\delta_0}\big[\log D(y)\big] + \mathbb{E}_{r\sim\hat{\mu}_\theta^r}\big[\log(1 - D(r))\big]. \tag{103}$$

Let

$$\hat{\gamma}_\theta^r = \frac{1}{2}\hat{\mu}_\theta^r + \frac{1}{2}\delta_0, \tag{104}$$

and define

$$\rho_\theta := \frac{d\delta_0}{d(\delta_0 + \hat{\mu}_\theta^r)}. \tag{105}$$

Then:

1. *The $L^2(\hat{\gamma}_\theta^r)$ functional gradient of $\mathcal{L}_D^{\mathrm{GAN}}$ is*

$$\nabla_{\hat{\gamma}_\theta^r}\mathcal{L}_D^{\mathrm{GAN}}(f) = 2\big(\rho_\theta - \sigma(f)\big). \tag{106}$$

2. *Consequently, the discriminator NTK flow is*

$$\partial_t f_t = 2\, T_{k^D,\hat{\gamma}_\theta^r}\big(\rho_\theta - \sigma(f_t)\big). \tag{107}$$

3. *If f_∞ is a stationary point of* (107)*, then*

$$T_{k^D,\hat{\gamma}_\theta^r}\big(\rho_\theta - \sigma(f_\infty)\big) = 0. \tag{108}$$

In particular, if the kernel operator is injective on the empirical support, then

$$\sigma(f_\infty) = \rho_\theta \qquad \textit{on } \operatorname{supp}(\hat{\gamma}_\theta^r). \tag{109}$$

4. *Near a stationary point f_∞, writing $f_t = f_\infty + h_t$ with $\|h_t\| \ll 1$, one has the first-order approximation*

$$\partial_t h_t = -2\, T_{k^D,\hat{\gamma}_\theta^r}\big(\sigma'(f_\infty)h_t\big) + \mathcal{O}(\|h_t\|^2). \tag{110}$$

5. *The induced generator-side sample weighting is*

$$\gamma_i^{\mathrm{GAN}} = \frac{1}{N_r}\,\frac{1}{1 - D_t(r_i)}\,\partial_r D_t(r_i) = \frac{1}{N_r}\,D_t(r_i)\,\partial_r f_t(r_i). \tag{111}$$

Proof. Rewriting (103) over the empirical measure $\hat{\gamma}_\theta^r$, we have

$$\mathcal{L}_D^{\mathrm{GAN}}(f) = \int \left[\frac{d\delta_0}{d\hat{\gamma}_\theta^r}\log\sigma(f) + \frac{d\hat{\mu}_\theta^r}{d\hat{\gamma}_\theta^r}\log(1-\sigma(f))\right] d\hat{\gamma}_\theta^r. \tag{112}$$

Differentiating with respect to f yields

$$\begin{aligned}\nabla_{\hat{\gamma}_\theta^r}\mathcal{L}_D^{\mathrm{GAN}}(f) &= \frac{d\delta_0}{d\hat{\gamma}_\theta^r}\frac{\sigma'(f)}{\sigma(f)} - \frac{d\hat{\mu}_\theta^r}{d\hat{\gamma}_\theta^r}\frac{\sigma'(f)}{1-\sigma(f)} \\ &= \frac{d\delta_0}{d\hat{\gamma}_\theta^r}(1-\sigma(f)) - \frac{d\hat{\mu}_\theta^r}{d\hat{\gamma}_\theta^r}\sigma(f).\end{aligned} \tag{113}$$

Since

$$\hat{\gamma}_\theta^r = \frac{1}{2}(\hat{\mu}_\theta^r + \delta_0), \qquad \frac{d\delta_0}{d\hat{\gamma}_\theta^r} = 2\rho_\theta, \qquad \frac{d\hat{\mu}_\theta^r}{d\hat{\gamma}_\theta^r} = 2(1-\rho_\theta),$$

we obtain

$$\nabla_{\hat{\gamma}_\theta^r}\mathcal{L}_D^{\mathrm{GAN}}(f) = 2\rho_\theta - 2\sigma(f) = 2(\rho_\theta - \sigma(f)), \tag{114}$$

which proves (106). Substituting this into the general discriminator NTK flow gives (107).

The equilibrium condition follows immediately by setting $\partial_t f_t = 0$ in (107), which gives

$$T_{k^D,\hat{\gamma}_\theta^r}\big(\rho_\theta - \sigma(f_\infty)\big) = 0.$$

If the kernel operator is injective on the empirical support, then (109) follows.

For the local linearization, write $f_t = f_\infty + h_t$ and use the Taylor expansion

$$\sigma(f_\infty + h_t) = \sigma(f_\infty) + \sigma'(f_\infty)h_t + \mathcal{O}(\|h_t\|^2).$$

Substituting this into (107) and using the equilibrium relation yields (110).

Finally, for the minimax generator objective,

$$\gamma_i^{\mathrm{GAN}} = -\frac{1}{N_r}R'\big(f_t(r_i)\big)\,\partial_r f_t(r_i). \tag{115}$$

Equivalently, writing the discriminator output directly as $D_t(r) = \sigma(f_t(r))$, one obtains

$$\gamma_i^{\mathrm{GAN}} = \frac{1}{N_r}\,\frac{1}{1 - D_t(r_i)}\,\partial_r D_t(r_i). \tag{116}$$

Using the chain rule

$$\partial_r D_t(r_i) = D_t(r_i)(1 - D_t(r_i))\,\partial_r f_t(r_i),$$

this becomes

$$\gamma_i^{\mathrm{GAN}} = \frac{1}{N_r}\,D_t(r_i)\,\partial_r f_t(r_i), \tag{117}$$

which proves (111). □

Interpretation. Proposition 5 shows that, in residual space, the vanilla GAN discriminator is best understood as a kernel-smoothed nonlinear classifier rather than as a linear regression field or a witness function. The target ρ_θ still separates the real zero residual from the generated residual samples, but the sigmoid nonlinearity changes the discriminator dynamics qualitatively. Formally, if a finite stationary point exists, it satisfies the equilibrium condition in Proposition 5. In the separable case, however, the limiting relation $\sigma(f_\infty) = \rho_\theta$ may correspond to saturation of the logits rather than convergence to a finite discriminator.

This saturation behavior directly affects the generator-side weighting. Although the logit-level discriminator flow is driven by the classification residual $\rho_\theta - \sigma(f_t)$, the probability-level input derivative satisfies

$$\partial_r D_t(r) = D_t(r)(1 - D_t(r))\, \partial_r f_t(r).$$

Hence the residual-space derivative of the discriminator output is suppressed in saturated regions where $D_t(r) \approx 0$ or $D_t(r) \approx 1$. Compared with IPM and LSGAN, vanilla GAN also introduces an additional confidence factor $D_t(r_i)$ into the sample-wise weighting, so the discriminator acts through both its residual-space score gradient $\partial_r f_t(r_i)$ and its current classification confidence. In this sense, the essential difference is not only the nominal adversarial loss, but also the way the sigmoid nonlinearity reshapes the discriminator-induced residual weighting.

Remark 9. *The purpose of the present appendix is not to claim a new NTK theory of GAN discriminators in itself. Rather, we use the existing discriminator-side NTK framework as an analysis tool and reinterpret it in residual space so that it becomes directly applicable to adversarial PINNs. What is specific to the present setting is that the "generated samples" are residual values, the "real samples" correspond to the zero residual state, and the resulting discriminator flow can therefore be linked explicitly to the generator-side residual weighting and residual-energy dynamics developed in the main text.*

B.4 Generator residual dynamics

We next derive the generator-side dynamics in residual space. Unlike the discriminator, whose evolution is naturally described in function space, the generator affects training through the evolution of the residual values themselves. It is therefore convenient to rewrite the parameter-space gradient flow of the generator as a closed dynamics for the residual vector. This leads directly to the kernel-preconditioned residual evolution used throughout the main text.

Proposition 6 (Generator residual dynamics in residual space)**.** *Let the generator gradient flow be*

$$\dot{\theta}(t) = -\frac{\partial \mathcal{L}_G(\theta(t))}{\partial \theta(t)}. \tag{118}$$

For each collocation point x_j, define the residual

$$r_j(t) := \mathcal{R}(x_j; \theta(t)), \qquad j = 1, \dots, N_r. \tag{119}$$

Let

$$\mathbf{r}(t) := [r_1(t), \dots, r_{N_r}(t)]^\top \in \mathbb{R}^{N_r} \tag{120}$$

be the residual vector, and define the generator residual NTK matrix

$$(K_{rr}^G)_{ij}(t) := \left\langle \frac{\partial \mathcal{R}(x_i; \theta(t))}{\partial \theta(t)}, \frac{\partial \mathcal{R}(x_j; \theta(t))}{\partial \theta(t)} \right\rangle. \tag{121}$$

Further define the diagonal matrix

$$\Gamma(t) := \mathrm{Diag}\big(\Gamma_{11}(t), \dots, \Gamma_{N_r N_r}(t)\big), \tag{122}$$

with entries

$$\Gamma_{ii}(t) = -\frac{1}{N_r} R'\Big(f(r_i(t); \phi(t))\Big)\, \frac{\partial f(r_i(t); \phi(t))}{\partial r_i(t)}. \tag{123}$$

Then the generator-induced residual dynamics satisfy

$$\dot{r}_j(t) = \sum_{i=1}^{N_r} (K_{rr}^G)_{ji}(t)\, \Gamma_{ii}(t), \qquad j = 1, \dots, N_r, \tag{124}$$

or equivalently, in vector form,

$$\dot{\mathbf{r}}(t) = K_{rr}^{G}(t)\,\Gamma(t)\,\mathbf{1}, \tag{125}$$

where $\mathbf{1} \in \mathbb{R}^{N_r}$ denotes the all-one vector.

Equivalently, if one introduces the weighting vector

$$\boldsymbol{\gamma}(t) := \Gamma(t)\mathbf{1} = [\gamma_1(t), \ldots, \gamma_{N_r}(t)]^{\top}, \tag{126}$$

with

$$\gamma_i(t) = -\frac{1}{N_r} R'\Big(f(r_i(t);\phi(t))\Big)\,\frac{\partial f(r_i(t);\phi(t))}{\partial r_i(t)}, \tag{127}$$

then the residual dynamics admit the compact form

$$\dot{\mathbf{r}}(t) = K_{rr}^{G}(t)\,\boldsymbol{\gamma}(t). \tag{128}$$

Proof. Starting from the generator gradient flow,

$$\dot{\theta}(t) = -\frac{\partial \mathcal{L}_G(\theta(t))}{\partial \theta(t)}, \tag{129}$$

and substituting (56), we obtain

$$\dot{\theta}(t) = -\frac{1}{N_r}\sum_{i=1}^{N_r} R'\Big(f(\mathcal{R}(x_i;\theta(t));\phi(t))\Big)\,\frac{\partial f(\mathcal{R}(x_i;\theta(t));\phi(t))}{\partial \mathcal{R}(x_i;\theta(t))}\,\frac{\partial \mathcal{R}(x_i;\theta(t))}{\partial \theta(t)}. \tag{130}$$

For each collocation point x_j, apply the chain rule:

$$\begin{aligned}
\frac{d}{dt}\mathcal{R}(x_j;\theta(t)) &= \left\langle \frac{\partial \mathcal{R}(x_j;\theta(t))}{\partial \theta(t)}, \dot{\theta}(t) \right\rangle \\
&= -\left\langle \frac{\partial \mathcal{R}(x_j;\theta(t))}{\partial \theta(t)}, \frac{1}{N_r}\sum_{i=1}^{N_r} R'\Big(f(\mathcal{R}(x_i;\theta(t));\phi(t))\Big)\,\frac{\partial f(\mathcal{R}(x_i;\theta(t));\phi(t))}{\partial \mathcal{R}(x_i;\theta(t))}\,\frac{\partial \mathcal{R}(x_i;\theta(t))}{\partial \theta(t)} \right\rangle \\
&= -\frac{1}{N_r}\sum_{i=1}^{N_r} R'\Big(f(\mathcal{R}(x_i;\theta(t));\phi(t))\Big)\,\frac{\partial f(\mathcal{R}(x_i;\theta(t));\phi(t))}{\partial \mathcal{R}(x_i;\theta(t))}\left\langle \frac{\partial \mathcal{R}(x_j;\theta(t))}{\partial \theta(t)}, \frac{\partial \mathcal{R}(x_i;\theta(t))}{\partial \theta(t)} \right\rangle.
\end{aligned} \tag{131}$$

Since $r_i(t) = \mathcal{R}(x_i;\theta(t))$, this becomes

$$\dot{r}_j(t) = -\frac{1}{N_r}\sum_{i=1}^{N_r} R'\Big(f(r_i(t);\phi(t))\Big)\,\frac{\partial f(r_i(t);\phi(t))}{\partial r_i(t)}\left\langle \frac{\partial \mathcal{R}(x_j;\theta(t))}{\partial \theta(t)}, \frac{\partial \mathcal{R}(x_i;\theta(t))}{\partial \theta(t)} \right\rangle. \tag{132}$$

Defining the generator residual NTK matrix $K_{rr}^{G}(t)$ and the diagonal weighting matrix $\Gamma(t)$ by (123), we obtain the componentwise representation

$$\dot{r}_j(t) = \sum_{i=1}^{N_r} (K_{rr}^{G})_{ji}(t)\,\Gamma_{ii}(t), \tag{133}$$

which yields the vector form

$$\dot{\mathbf{r}}(t) = K_{rr}^{G}(t)\,\Gamma(t)\,\mathbf{1}. \tag{134}$$

Finally, introducing $\boldsymbol{\gamma}(t) = \Gamma(t)\mathbf{1}$ gives

$$\dot{\mathbf{r}}(t) = K_{rr}^{G}(t)\,\boldsymbol{\gamma}(t). \tag{135}$$

This proves (125) and (128). □

Interpretation. Proposition 6 shows that the generator does not evolve in residual space by an ordinary gradient descent with a scalar loss coefficient. Instead, its residual dynamics are preconditioned by the generator residual NTK $K_{rr}^{G}(t)$ and modulated samplewise by the discriminator-induced weighting $\boldsymbol{\gamma}(t)$.

This decomposition is central to the viewpoint of the main text. The matrix $K_{rr}^{G}(t)$ encodes the geometry of residual propagation through the generator, while $\boldsymbol{\gamma}(t)$ contains the discriminator-side signal through the factor

$$\gamma_i = -\frac{1}{N_r} R'\Big(f(r_i(t);\phi(t))\Big) \frac{\partial f(r_i(t);\phi(t))}{\partial r_i(t)}.$$

Hence, different adversarial objectives do not simply change a scalar divergence value; rather, they reshape the residual dynamics through different sample-dependent weighting mechanisms. In this sense, adversarial PINNs training should be understood as a discriminator-modulated, kernel-preconditioned residual descent process.

B.4.1 Specialization to common adversarial objectives

We finally record the specialization of $\Gamma(t)$ for several standard adversarial objectives.

Vanilla GAN. If

$$R(f) = \log\big(1 - \sigma(f)\big), \tag{136}$$

then

$$R'(f) = -\sigma(f). \tag{137}$$

Hence, by (127),

$$[\Gamma(t)]_{ii} = \frac{1}{N_r} \sigma\big(f(r_i(t);\phi(t))\big) \frac{\partial f(r_i(t);\phi(t))}{\partial r_i(t)}. \tag{138}$$

Least-Squares GAN (LSGAN). If

$$R(f) = \frac{1}{2}(f-1)^2, \tag{139}$$

then

$$R'(f) = f - 1. \tag{140}$$

Hence

$$[\Gamma(t)]_{ii} = -\frac{1}{N_r}\big(f(r_i(t);\phi(t)) - 1\big) \frac{\partial f(r_i(t);\phi(t))}{\partial r_i(t)}. \tag{141}$$

Integral Probability Metric GAN (IPM-GAN). If

$$R(f) = -f, \tag{142}$$

then

$$R'(f) = -1. \tag{143}$$

Therefore,

$$[\Gamma(t)]_{ii} = \frac{1}{N_r} \frac{\partial f(r_i(t);\phi(t))}{\partial r_i(t)}. \tag{144}$$

Equations above show that, once the discriminator state is fixed, different adversarial objectives enter the generator residual dynamics through different sample-wise weighting rules, while the structural form of the residual dynamics remains the same.

B.4.2 Interpretation in our framework

The discriminator-side NTK analysis can be summarized in the language of this paper as follows. First, for a fixed generator, discriminator training against $\hat{\mu}_\theta^r$ and δ_0 produces a finite-time, architecture-dependent residual-space function, rather than an arbitrary empirical optimum. Second, the derivative $\partial_r f_t(r)$ entering the generator dynamics is therefore well-defined and inherits the smoothing and inductive bias of the discriminator NTK. Third, in the IPM case, this derivative is induced by the NTK-MMD witness function comparing $\hat{\mu}_\theta^r$ and δ_0; in the LSGAN case, it is induced by a kernel-smoothed regression solution; and in the vanilla GAN case, it is induced by a kernel-smoothed nonlinear classification flow shaped by the sigmoid nonlinearity.

From our perspective, the essential role of the discriminator in adversarial PINNs is therefore not to realize an abstract optimal divergence critic, but to generate, after finite-time NTK-constrained training, a differentiable residual-space weighting field whose derivative directly modulates the generator residual dynamics. This is precisely the mechanism captured by our framework.

B.5 The effect of the PDE operator on the NTK spectrum

Let u_θ denote the neural network in which the prescribed boundary conditions have already been incorporated. Consider the linear PDE $\mathcal{T}u_\theta(\mathbf{x}) = f(\mathbf{x})$, where $\mathcal{T}$ is an mth-order linear differential operator, and define the residual $r(\mathbf{x};\theta) = \mathcal{T}u_\theta(\mathbf{x}) - f(\mathbf{x})$. The neural-network NTK is $K_{uu}^G(\mathbf{x},\mathbf{x}') = \langle \nabla_\theta u_\theta(\mathbf{x}), \nabla_\theta u_\theta(\mathbf{x}') \rangle$.

Under sufficient smoothness such that the parameter and spatial derivatives commute, the operator-informed NTK relation gives

$$K_{rr}^G(\mathbf{x},\mathbf{x}') = \mathcal{T}_{\mathbf{x}}\mathcal{T}_{\mathbf{x}'} K_{uu}^G(\mathbf{x},\mathbf{x}'), \tag{145}$$

as established in [21]. Thus, the PDE operator not only defines the residual, but also transforms the neural-network parameter features and reshapes the kernel governing the residual dynamics.

Let $\{\lambda_j\}$ and $\{\mu_j\}$ denote the eigenvalues of the integral operators induced by K_{uu}^G and K_{rr}^G, respectively. For a symmetric operator satisfying $\|v\|_{L^2} \le C_{\mathcal{T}}\|\mathcal{T}v\|_{L^2}$, the spectral comparison in [1] yields

$$\lambda_j \le C_{\mathcal{T}}^2 \mu_j. \tag{146}$$

The effect of the differential order is particularly transparent when $\mathcal{T}$ and the neural-network NTK share an eigenbasis. If $\mathcal{T}\varphi_j = \tau_j\varphi_j$ and $\mathcal{I}_u\varphi_j = \lambda_j\varphi_j$, then

$$\mu_j = |\tau_j|^2 \lambda_j. \tag{147}$$

For an mth-order elliptic operator, $|\tau_{\mathbf{k}}| \asymp |\mathbf{k}|^m$ at high frequencies, giving

$$\mu_{\mathbf{k}} \asymp |\mathbf{k}|^{2m} \lambda_{\mathbf{k}}. \tag{148}$$

Hence, the operator order appears twice because $\mathcal{T}$ acts on both arguments of the NTK. For example, the second-order Poisson operator produces the multiplier $|\mathbf{k}|^4$. When the operator and the NTK do not share an eigenbasis, the PDE operator may also mix the original neural-network modes. This discussion is limited to known linear operators; for general nonlinear partial differential equations, the residual NTK may also depend on other conditions. The influence of the boundary on the NTK spectrum will be left for future work.

Remark 10 (Extension to GAN-Based Regression)**.** *The essential difference between GAN-based regression problems and our analysis lies in whether the target distribution is available.*

For a GAN-based regression problem, after redefining the residual as the prediction error, the resulting dynamics share a similar algebraic form with our framework. However, a critical distinction lies in the target distribution. In regression, the true data distribution $p(\mathbf{y})$ is unknown; the training objective is instead defined with respect to the empirical distribution $\hat{p}(\mathbf{y}) = \frac{1}{N}\sum_{i=1}^N \delta_{\mathbf{y}_i}$. Consequently, the

empirical distribution varies with the number of training samples, and it remains unclear whether the gap between the true and empirical distributions may compromise the validity of the first-order indicator derived from our analysis. In contrast, in our PINN setting, the target distribution is always the fixed Dirac measure δ_0, regardless of the number of collocation points, because this zero-residual state is dictated by the physical laws encoded in the PDE. At present, our work is conducted within the PINNs framework. Although our analysis suggests that the method may be extendable to other models, this direction requires further theoretical and experimental validation, which falls beyond the scope of this paper. We therefore leave it as future work.

C Analysis of the residual-energy decay

C.1 Residual-energy law and constant-kernel approximation

We now turn from the residual dynamics themselves to the evolution of the residual energy. This provides a more compact scalar description of whether adversarial training is effectively driving the residual toward zero. Starting from the generator-side residual dynamics derived in the previous appendix, we show that the energy dissipation rate is determined by the interaction between the generator residual NTK and the discriminator-induced weighting vector. This identity will serve as the basis for the later decay estimates, failure-mode analysis, and spectral interpretations.

Theorem 3 (Residual-energy law under adversarial residual dynamics)**.** *Let*

$$E(t) := \frac{1}{2}\|\mathbf{r}(t)\|^2 \tag{149}$$

be the residual energy, where the residual vector evolves according to

$$\dot{\mathbf{r}}(t) = K_{rr}^G(t)\boldsymbol{\gamma}(t), \tag{150}$$

with generator residual NTK $K_{rr}^G(t) \in \mathbb{R}^{N_r \times N_r}$ and discriminator-induced weighting vector

$$\boldsymbol{\gamma}(t) = [\gamma_1(t), \ldots, \gamma_{N_r}(t)]^\top. \tag{151}$$

Then the residual energy satisfies

$$\dot{E}(t) = \mathbf{r}(t)^\top K_{rr}^G(t)\boldsymbol{\gamma}(t). \tag{152}$$

In particular, the sign and magnitude of $\dot{E}(t)$ are determined by how the discriminator-induced weighting $\boldsymbol{\gamma}(t)$ aligns with the residual vector through the kernel geometry imposed by $K_{rr}^G(t)$.

Proof. By definition,

$$E(t) = \frac{1}{2}\|\mathbf{r}(t)\|^2 = \frac{1}{2}\,\mathbf{r}(t)^\top \mathbf{r}(t). \tag{153}$$

Differentiating with respect to time yields

$$\dot{E}(t) = \mathbf{r}(t)^\top \dot{\mathbf{r}}(t). \tag{154}$$

Substituting the residual dynamics (150), we obtain

$$\dot{E}(t) = \mathbf{r}(t)^\top K_{rr}^G(t)\boldsymbol{\gamma}(t), \tag{155}$$

which proves (152). □

Interpretation. Theorem 3 shows that the energy decay in adversarially trained PINNs is not controlled by the generator kernel alone, nor by the discriminator weighting alone, but by their interaction. The matrix $K_{rr}^G(t)$ determines how a local discriminator signal propagates across residual samples, while $\boldsymbol{\gamma}(t)$ determines whether that propagated signal is aligned with residual descent. Therefore, the key quantity is not simply the size of the discriminator gradient, but the signed bilinear form

$$\mathbf{r}(t)^\top K_{rr}^G(t)\boldsymbol{\gamma}(t),$$

which acts as the instantaneous energy-dissipation rate.

To obtain explicit convergence rates and more refined decay mechanisms, it is convenient to further work under the constant-NTK approximation

$$K_{rr}^G(t) \approx K_{rr}^G. \tag{156}$$

Under this approximation, the energy law admits a clearer spectral interpretation, which allows us to distinguish exponential decay, slower algebraic decay, and several representative failure modes.

Remark 11 (On the constant-NTK approximation)**.** *The approximation* (156) *is best understood as a mechanism-level simplification rather than a universally exact property. Under Assumptions 2–6, for some relatively simple or weakly nonlinear problems—especially linear PDEs and certain regular regimes—the generator NTK is known or empirically observed to remain nearly constant during training[5]. By contrast, for strongly nonlinear, multiscale, or more complicated equations, the NTK may evolve nontrivially [22, 23], so the constant-kernel assumption need not hold exactly. Nevertheless, even in such cases, the approximation remains useful for revealing the dominant spectral mechanisms behind residual-energy decay and failure modes.*

C.2 Spectral factorization of the aligned dynamics

We first consider the favorable case in which the discriminator-induced weighting is linearly aligned with the negative residual direction. Instead of treating $K_{rr}^{G}A$ as a symmetric operator, we use a spectral factorization that converts the effective operator into the symmetric positive semidefinite matrix $K_{rr}^{G,1/2}AK_{rr}^{G,1/2}$.

Assume throughout this subsection that the generator residual NTK K_{rr}^{G} is symmetric positive definite on the active residual subspace, and that the discriminator-induced weighting takes the linear alignment form

$$\boldsymbol{\gamma}(t) = -A\mathbf{r}(t), \tag{157}$$

where $A \in \mathbb{R}^{N_r \times N_r}$ is symmetric positive semidefinite. Then the residual dynamics become

$$\dot{\mathbf{r}}(t) = -K_{rr}^{G} A\, \mathbf{r}(t). \tag{158}$$

Theorem 4 (Spectral factorization and convergence rate under linear alignment)**.** *Let*

$$K := K_{rr}^{G}, \qquad M := KA, \qquad H := K^{1/2}AK^{1/2}. \tag{159}$$

Assume that K is symmetric positive definite and that A is symmetric positive semidefinite. Then:

1. *M is similar to H, namely*

$$K^{-1/2}MK^{1/2} = K^{-1/2}(KA)K^{1/2} = K^{1/2}AK^{1/2} = H. \tag{160}$$

   Hence M and H have the same eigenvalues.
2. *Since H is symmetric positive semidefinite, all eigenvalues of M are real and nonnegative. Moreover, if*

$$\lambda_1(A) \geq \lambda_2(A) \geq \cdots \geq \lambda_{N_r}(A) \geq 0 \tag{161}$$

   denote the eigenvalues of A, then for each $j = 1, \ldots, N_r$,

$$\lambda_{\min}(K)\,\lambda_j(A) \;\leq\; \lambda_j(M) = \lambda_j(H) \;\leq\; \lambda_{\max}(K)\,\lambda_j(A). \tag{162}$$

3. *Introducing the transformed variable*

$$\mathbf{z}(t) := K^{-1/2}\mathbf{r}(t), \tag{163}$$

   the dynamics (158) *become*

$$\dot{\mathbf{z}}(t) = -H\mathbf{z}(t). \tag{164}$$

   Therefore,

$$\|\mathbf{z}(t)\|^2 \leq e^{-2\lambda_{\min}(H)t}\|\mathbf{z}(0)\|^2. \tag{165}$$

   Since

$$\lambda_{\min}(K)\|\mathbf{z}\|^2 \leq \|\mathbf{r}\|^2 \leq \lambda_{\max}(K)\|\mathbf{z}\|^2, \tag{166}$$

   the residual energy satisfies

$$E(t) \leq \kappa(K)\,E(0)\,e^{-2\lambda_{\min}(H)t}, \qquad \kappa(K) := \frac{\lambda_{\max}(K)}{\lambda_{\min}(K)}. \tag{167}$$

   In particular, using (162)*,*

$$E(t) \leq \kappa(K)\,E(0)\,e^{-2\lambda_{\min}(K)\lambda_{\min}(A)t}. \tag{168}$$

Proof. The similarity relation follows directly from

$$K^{-1/2}(KA)K^{1/2} = K^{1/2}AK^{1/2}.$$

Since $H = K^{1/2}AK^{1/2}$ is symmetric positive semidefinite, its spectrum is real and nonnegative, and therefore the same is true for M. The eigenvalue bound (162) is the standard congruence bound for symmetric positive semidefinite matrices. Now define $\mathbf{z} = K^{-1/2}\mathbf{r}$. Then

$$\dot{\mathbf{z}} = K^{-1/2}\dot{\mathbf{r}} = -K^{-1/2}KA\mathbf{r} = -K^{1/2}AK^{1/2}\mathbf{z} = -H\mathbf{z},$$

which gives (164). Since $H \succeq 0$,

$$\frac{d}{dt}\frac{1}{2}\|\mathbf{z}(t)\|^2 = -\mathbf{z}(t)^\top H\mathbf{z}(t) \le -\lambda_{\min}(H)\|\mathbf{z}(t)\|^2.$$

Applying Grönwall's inequality yields

$$\|\mathbf{z}(t)\|^2 \le e^{-2\lambda_{\min}(H)t}\|\mathbf{z}(0)\|^2.$$

Finally, using the norm equivalence induced by K,

$$\lambda_{\min}(K)\|\mathbf{z}\|^2 \le \|\mathbf{r}\|^2 \le \lambda_{\max}(K)\|\mathbf{z}\|^2,$$

we obtain (167). The lower bound

$$\lambda_{\min}(H) \ge \lambda_{\min}(K)\lambda_{\min}(A)$$

then gives (168). □

Interpretation. Theorem 4 shows that, under linear alignment, the effective decay rate is controlled not by the generator residual NTK K^G_{rr} alone, but by the coupled operator $K^G_{rr}A$, or equivalently by the symmetric matrix $K^{G,1/2}_{rr}AK^{G,1/2}_{rr}$. Therefore, adversarially induced weighting may accelerate or slow down convergence through its interaction with the spectrum of the generator kernel. In particular, small eigenvalues of A may lead to slow-decaying directions even when K^G_{rr} itself is well-conditioned, while large eigenvalues of A may selectively amplify useful descent directions.

Remark 12 (Connection to standard PINN-NTK dynamics)**.** *The standard non-adversarial PINN-NTK law is recovered as the special case*

$$\boldsymbol{\gamma}(t) = -\mathbf{r}(t),$$

that is, $A = I$. In this case the aligned dynamics reduce to

$$\dot{\mathbf{r}}(t) = -K^G_{rr}\mathbf{r}(t),$$

and the residual energy satisfies

$$E(t) \le E(0)e^{-2\lambda_{\min}(K^G_{rr})t}.$$

Thus ordinary PINN training corresponds to the reference case in which the residual is fed back without adversarial distortion.

C.3 Modal view and slower decay under weaker alignment

We now turn to the complementary situation where the residual energy still decreases, but the convergence is no longer fast. In this regime, the discriminator-induced weighting remains descent-oriented, yet its alignment with the residual is weaker than in the strongly aligned case considered above. The resulting decay is therefore slower, and its mechanism is best understood in the eigenbasis of the generator residual NTK.

Theorem 5 (Modal decay and slower convergence under weaker alignment)**.** *Assume the constant-NTK approximation*

$$K^G_{rr}(t) \approx K^G_{rr}, \tag{169}$$

and let

$$K^G_{rr} = U\Lambda U^\top, \qquad \Lambda = \mathrm{Diag}(\lambda_1, \dots, \lambda_{N_r}), \qquad \lambda_j \ge 0, \tag{170}$$

be the eigendecomposition of the generator residual NTK. Define

$$\widetilde{\mathbf{r}}(t) := U^\top\mathbf{r}(t), \qquad \widetilde{\boldsymbol{\gamma}}(t) := U^\top\boldsymbol{\gamma}(t). \tag{171}$$

Then:

1. *The residual dynamics in modal coordinates are*

$$\dot{\widetilde{\mathbf{r}}}(t) = \Lambda\, \widetilde{\boldsymbol{\gamma}}(t), \tag{172}$$

that is,

$$\dot{\widetilde{r}}_j(t) = \lambda_j\, \widetilde{\gamma}_j(t), \qquad j = 1, \dots, N_r. \tag{173}$$

2. *If the discriminator-induced weighting is modewise aligned in the sense that*

$$\widetilde{\gamma}_j(t) = -a_j(t)\widetilde{r}_j(t), \qquad a_j(t) \geq a_* > 0, \tag{174}$$

then each mode satisfies

$$\dot{\widetilde{r}}_j(t) = -\lambda_j a_j(t)\widetilde{r}_j(t), \tag{175}$$

and therefore

$$|\widetilde{r}_j(t)| \leq |\widetilde{r}_j(0)|e^{-\lambda_j a_* t}. \tag{176}$$

Consequently, the residual energy satisfies

$$E(t) \leq \frac{1}{2}\sum_{j=1}^{N_r} \widetilde{r}_j(0)^2 e^{-2\lambda_j a_* t}. \tag{177}$$

3. *More generally, if the feedback remains descent-oriented but no longer scales linearly with the residual, and one only has*

$$\mathbf{r}(t)^\top K_{rr}^G \boldsymbol{\gamma}(t) \leq -c\|\mathbf{r}(t)\|^{2\alpha}, \qquad \alpha > 1, \tag{178}$$

then the residual energy obeys the algebraic decay estimate

$$E(t) \leq C(1+t)^{-1/(\alpha-1)} \tag{179}$$

for some constant $C > 0$.

Proof. Since

$$\dot{\mathbf{r}}(t) = K_{rr}^G \boldsymbol{\gamma}(t), \tag{180}$$

left-multiplying by $U^\top$ and using $K_{rr}^G = U\Lambda U^\top$ gives

$$\dot{\widetilde{\mathbf{r}}}(t) = U^\top \dot{\mathbf{r}}(t) = U^\top K_{rr}^G \boldsymbol{\gamma}(t) = U^\top U \Lambda U^\top \boldsymbol{\gamma}(t) = \Lambda\, \widetilde{\boldsymbol{\gamma}}(t), \tag{181}$$

which proves (172).

Now assume the modewise alignment condition (174). Then each mode satisfies

$$\dot{\widetilde{r}}_j(t) = \lambda_j \widetilde{\gamma}_j(t) = -\lambda_j a_j(t)\widetilde{r}_j(t). \tag{182}$$

Since $a_j(t) \geq a_* > 0$, Grönwall's inequality yields

$$|\widetilde{r}_j(t)| \leq |\widetilde{r}_j(0)|e^{-\lambda_j a_* t}, \tag{183}$$

which proves (176). Summing over all modes gives

$$E(t) = \frac{1}{2}\|\widetilde{\mathbf{r}}(t)\|^2 = \frac{1}{2}\sum_{j=1}^{N_r} \widetilde{r}_j(t)^2 \leq \frac{1}{2}\sum_{j=1}^{N_r} \widetilde{r}_j(0)^2 e^{-2\lambda_j a_* t}, \tag{184}$$

which proves (177).

For the weaker descent-oriented case, starting from (178) and using $\|\mathbf{r}(t)\|^2 = 2E(t)$, we obtain

$$\dot{E}(t) = \mathbf{r}(t)^\top K_{rr}^G \boldsymbol{\gamma}(t) \leq -c(2E(t))^\alpha. \tag{185}$$

Integrating the associated scalar differential inequality yields

$$E(t) \leq C(1+t)^{-1/(\alpha-1)} \tag{186}$$

for some constant $C > 0$, which proves (179). □

Interpretation. Theorem 5 makes clear that residual decay is inherently spectral. Under strong modewise alignment, each mode decays at a rate controlled by $\lambda_j a_j(t)$, so modes associated with larger effective eigenvalues decay first. As a result, long-time training is dominated by the slower unresolved modes. This explains why spectral imbalance can be harmful: once the fast modes have already become small, maintaining a negative overall energy rate requires the discriminator-induced weighting to remain aligned with the slow modes that still dominate the residual energy.

The last part of the proposition shows that if the discriminator feedback remains descent-oriented but is no longer linearly aligned with the residual, then exponential decay is typically lost, and only algebraic decay can be guaranteed. This provides a natural explanation for the long optimization tails often observed in adversarial PINNs training: even when training does not fail, insufficiently strong alignment with the unresolved modes can substantially slow convergence.

C.4 Failure modes: ascent and stagnation

The previous three regimes describe successful but quantitatively different decay. The opposite cases are equally important.

Proposition 7 (Ascent, stagnation, and unstable oscillation regimes)**.** *Under the residual-energy law* (152)*:*

1. *If*
$$\mathbf{r}(t)^\top K_{rr}^G(t)\boldsymbol{\gamma}(t) \geq 0 \tag{187}$$
on some time interval, then $E(t)$ fails to decrease monotonically on that interval. In the strictly positive case, the residual energy increases.
2. *If*
$$\|\boldsymbol{\gamma}(t)\| \to 0 \tag{188}$$
while $\|K_{rr}^G(t)\|$ remains bounded, then
$$\|\dot{\mathbf{r}}(t)\| \leq \|K_{rr}^G(t)\|\,\|\boldsymbol{\gamma}(t)\| \to 0. \tag{189}$$
Thus the first-order residual dynamics stall. If this happens before $\mathbf{r}(t)$ becomes small, training enters a plateau regime.
3. *Let*
$$s(t) := \mathbf{r}(t)^\top K_{rr}^G(t)\boldsymbol{\gamma}(t). \tag{190}$$
If $s(t)$ changes sign repeatedly on a time interval, namely if there exist sequences
$$t_1 < t_2 < t_3 < \cdots \tag{191}$$
such that
$$s(t_{2m-1}) > 0, \qquad s(t_{2m}) < 0, \tag{192}$$
for all admissible m, then the residual-energy slope alternates between ascent-oriented and descent-oriented phases. In this case, $E(t)$ does not exhibit a stable monotone decay trend, but instead undergoes oscillatory evolution. Consequently, training becomes dynamically unstable in the sense that the discriminator-induced update direction repeatedly switches between alignment and misalignment with residual descent.

Remark 13. *Proposition 7 identifies three representative failure modes for adversarially trained PINNs. First, if the discriminator-induced weighting is misaligned with the residual after modulation by the generator residual NTK, so that $\mathbf{r}(t)^\top K_{rr}^G(t)\boldsymbol{\gamma}(t) > 0$, then the residual energy increases rather than decreases. Second, if the discriminator-induced gradient vanishes, namely $\|\boldsymbol{\gamma}(t)\| \to 0$, then the first-order residual dynamics stall; this is especially likely to occur when the discriminator becomes overly strong and produces near-vanishing generator-side gradients, as can happen in GAN-type or LSGAN-type training. Third, even when the discriminator gradient does not vanish, it may fail to remain consistently descent-oriented, so that the induced weighting alternates between favorable and unfavorable directions. In this case, the first-order indicator changes sign repeatedly, leading to oscillatory and unstable training behavior. These three phenomena are also clearly illustrated in the right panel of Fig. 2 in the main text.*

C.5 Discrete-time interpretation for the alternating algorithm

The same logic carries over to the practical alternating algorithm. Recall the first-order discrete expansion

$$E^{m+1} - E^m = \eta_G S^m + \mathcal{O}(\eta_G^2), \qquad S^m := (\mathbf{r}^m)^\top K_{rr}^{G,m} \boldsymbol{\gamma}^m. \tag{193}$$

Thus S^m is the discrete analogue of the continuous energy-dissipation rate.

Contractive regime. If

$$S^m \le -cE^m \tag{194}$$

uniformly and η_G is sufficiently small, then

$$E^{m+1} \le (1 - c\eta_G)E^m, \tag{195}$$

which gives exponential decay in discrete time.

Slow-decay regime. More generally, if S^m scales like a higher power of E^m, then one expects a slower algebraic-type decay, consistent with the continuous-time estimate (179).

Failure regime. If S^m becomes positive or nearly vanishes, then the discrete dynamics exhibit ascent or stagnation, respectively.

Implicit spectral balancing effect of the acceptance rule. Although the proposed alternating algorithm does not explicitly enforce mode-by-mode spectral balancing, the score-based acceptance rule can still provide an indirect benefit from the spectral viewpoint. Indeed, in the eigenbasis of the generator residual NTK,

$$S^m = (\mathbf{r}^m)^\top K_{rr}^{G,m} \boldsymbol{\gamma}^m = \sum_{j=1}^{N_r} \lambda_j^m \, \widetilde{r}_j^m \, \widetilde{\gamma}_j^m. \tag{196}$$

Thus, accepting a discriminator update only when $S^m < 0$ tends to favor discriminator states whose induced weighting yields a negative net contribution across the unresolved dominant modes. In particular, when fast modes have already decayed and their amplitudes $\widetilde{r}_j^m$ become small, maintaining $S^m < 0$ increasingly requires useful correction on the slow modes that still dominate the residual energy. Consequently, the acceptance rule may suppress discriminator states that overemphasize already-resolved modes while underweighting unresolved ones.

From this perspective, the proposed method does not explicitly balance the spectrum, but it may exert an *implicit spectral regularization* effect on the alternating dynamics by filtering out discriminator states that produce poor net modal descent.

Overall, the continuous energy analysis provides a direct interpretation of the practical score used in the alternating algorithm. The convergence rate of the residual energy in adversarial PINNs is governed not by the generator residual NTK alone, but by the interaction between the generator-side kernel geometry, the discriminator-induced weighting, and the evolving residual itself.

D Different Dynamics Induced by Residual Input and Squared-Residual Input

In most existing adversarial PINNs formulations, the discriminator takes the residual itself as input,

$$r_i(\theta) := \mathcal{R}(x_i; \theta), \qquad i = 1, \dots, N_r.$$

By contrast, standard PINNs training is fundamentally driven by the squared residual, since the canonical physics loss is proportional to

$$\frac{1}{N_r} \sum_{i=1}^{N_r} r_i(\theta)^2.$$

This raises a natural question: what changes if the discriminator input is replaced by r_i^2? As we show below, this is not a minor reparameterization. It changes the generator dynamics from an *externally forced preconditioned residual descent*

$$\dot{\mathbf{r}} = K_{rr}^G \Gamma \mathbf{1}$$

into a *state-dependent multiplicative feedback law*

$$\dot{\mathbf{r}} = K_{rr}^{G} \widetilde{\Gamma} \mathbf{r}.$$

Consequently, both the residual evolution law and the energy dissipation mechanism become fundamentally different.

D.1 Generator dynamics with squared-residual input

We now consider the alternative design in which the discriminator no longer takes the residual r_i itself as input, but instead takes the squared residual r_i^2. This seemingly simple modification changes the structure of the generator dynamics in an essential way. In the residual-input case, the discriminator induces an additive forcing term on the residual dynamics. By contrast, under squared-residual input, the chain rule introduces an additional factor of r_i, so that the discriminator enters multiplicatively through the current residual state. The following proposition makes this distinction precise.

Proposition 8 (Generator dynamics with squared-residual discriminator input)**.** *Let*

$$s_i(\theta) := r_i(\theta)^2, \qquad i = 1, \dots, N_r, \tag{197}$$

and define

$$\mathbf{s}(\theta) := [s_1(\theta), \dots, s_{N_r}(\theta)]^\top. \tag{198}$$

Consider the generator loss

$$\mathcal{L}_G^{(X^2)}(\theta, \phi) = \frac{1}{N_r} \sum_{i=1}^{N_r} R\big(f(s_i(\theta); \phi)\big). \tag{199}$$

Then:

1. *The generator gradient is*

$$\nabla_\theta \mathcal{L}_G^{(X^2)} = \frac{2}{N_r} \sum_{i=1}^{N_r} r_i \, R'\big(f_i^{(X^2)}\big) \, \partial_s f_i^{(X^2)} \, \nabla_\theta r_i, \tag{200}$$

   where

$$f_i^{(X^2)} := f(s_i; \phi), \qquad \partial_s f_i^{(X^2)} := \partial_s f(s_i; \phi).$$

2. *The generator gradient flow becomes*

$$\dot{\theta} = -\frac{2}{N_r} \sum_{i=1}^{N_r} r_i \, R'\big(f_i^{(X^2)}\big) \, \partial_s f_i^{(X^2)} \, \nabla_\theta r_i. \tag{201}$$

3. *Let J_r denote the residual Jacobian and $K_{rr}^G(\theta) = J_r J_r^\top$ the generator residual NTK. Then the residual dynamics satisfy*

$$\dot{\mathbf{r}} = -\, K_{rr}^G(\theta) \, \boldsymbol{\gamma}^{(X^2)}, \tag{202}$$

   where

$$\gamma_i^{(X^2)} = \frac{2}{N_r} \, r_i \, R'\big(f(r_i^2; \phi)\big) \, \partial_s f(r_i^2; \phi). \tag{203}$$

4. *Defining*

$$\widetilde{\gamma}_i := -\frac{2}{N_r} R'\big(f(r_i^2; \phi)\big) \, \partial_s f(r_i^2; \phi), \tag{204}$$

   and

$$\widetilde{\Gamma} = \mathrm{diag}(\widetilde{\gamma}_1, \dots, \widetilde{\gamma}_{N_r}), \tag{205}$$

   one has

$$\boldsymbol{\gamma}^{(X^2)} = -\widetilde{\Gamma} \mathbf{r}, \tag{206}$$

   and therefore

$$\dot{\mathbf{r}} = K_{rr}^G(\theta) \, \widetilde{\Gamma} \, \mathbf{r}. \tag{207}$$

5. *The corresponding residual-energy law is*

$$\dot{E} = \mathbf{r}^\top K_{rr}^G(\theta) \, \widetilde{\Gamma} \, \mathbf{r}. \tag{208}$$

Proof. Since $s_i = r_i^2$, we have

$$\nabla_\theta s_i = 2r_i \, \nabla_\theta r_i.$$

Therefore,

$$\begin{aligned}\nabla_\theta \mathcal{L}_G^{(X^2)} &= \frac{1}{N_r} \sum_{i=1}^{N_r} R'\big(f(s_i;\phi)\big)\, \partial_s f(s_i;\phi)\, \nabla_\theta s_i \\ &= \frac{2}{N_r} \sum_{i=1}^{N_r} r_i \, R'\big(f_i^{(X^2)}\big)\, \partial_s f_i^{(X^2)}\, \nabla_\theta r_i, \end{aligned} \tag{209}$$

which proves the first statement. The gradient-flow equation (201) follows immediately.

Next, multiplying by the residual Jacobian J_r, we obtain

$$\dot{\mathbf{r}} = J_r \dot{\theta} = -\, K_{rr}^G(\theta)\, \boldsymbol{\gamma}^{(X^2)}, \tag{210}$$

where

$$\gamma_i^{(X^2)} = \frac{2}{N_r}\, r_i \, R'\big(f(r_i^2;\phi)\big)\, \partial_s f(r_i^2;\phi). \tag{211}$$

Defining

$$\widetilde{\gamma}_i = -\frac{2}{N_r} R'\big(f(r_i^2;\phi)\big)\, \partial_s f(r_i^2;\phi),$$

we have

$$\gamma_i^{(X^2)} = -\widetilde{\gamma}_i \, r_i, \qquad \boldsymbol{\gamma}^{(X^2)} = -\widetilde{\Gamma}\mathbf{r},$$

and hence

$$\dot{\mathbf{r}} = K_{rr}^G(\theta)\, \widetilde{\Gamma}\, \mathbf{r}, \tag{212}$$

which proves (207). Finally,

$$\dot{E} = \mathbf{r}^\top \dot{\mathbf{r}} = \mathbf{r}^\top K_{rr}^G(\theta)\, \widetilde{\Gamma}\, \mathbf{r}, \tag{213}$$

which proves (208). □

Interpretation. Proposition 8 shows that changing the discriminator input from r to r^2 does not merely rescale the adversarial signal: it changes the structure of the residual dynamics. In the residual-input case, the discriminator acts through an additive forcing term of the form

$$\dot{\mathbf{r}} = K_{rr}^G(\theta)\Gamma\mathbf{1},$$

where the weighting is independent of the current residual amplitudes. By contrast, under squared-residual input, the dynamics take the multiplicative form

$$\dot{\mathbf{r}} = K_{rr}^G(\theta)\widetilde{\Gamma}\mathbf{r},$$

so that the discriminator enters through the current residual state itself.

This difference is fundamental. The squared-residual input removes sign information and makes the discriminator sensitive mainly to residual magnitude, while the additional factor $\mathbf{r}$ turns the dynamics into a genuine multiplicative feedback system. As a result, both the residual evolution law and the energy law become quadratic in the residual state. This is why squared-residual adversarial training is structurally different from the standard residual-input formulation, and why its later spectral analysis takes a substantially different form.

D.2 Spectral interpretation

Under the constant-kernel approximation

$$K_{rr}^G(\theta) \approx K_{rr}^G =: K, \tag{214}$$

we analyze the residual-input and squared-residual-input dynamics in the eigenbasis of the generator residual NTK. The main purpose is to clarify how the two input choices differ at the level of modal evolution, and why the squared-residual-input case leads to a genuinely coupled multiplicative system.

Let

$$K = U\Lambda U^\top, \qquad \Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_{N_r}), \qquad \lambda_j \geq 0, \tag{215}$$

where $U = [u_1, \dots, u_{N_r}]$ is orthonormal, and expand

$$\mathbf{r} = \sum_{j=1}^{N_r} c_j u_j. \tag{216}$$

Proposition 9 (Modal dynamics under residual and squared-residual inputs)**.** *Under the above notation:*

1. *If the discriminator input is the residual itself, so that*
$$\dot{\mathbf{r}} = K\Gamma\mathbf{1}, \tag{217}$$
then the modal coefficients satisfy
$$\dot{c}_j = u_j^\top \dot{\mathbf{r}} = \lambda_j \, u_j^\top \Gamma\mathbf{1}. \tag{218}$$
2. *If the discriminator input is the squared residual, so that*
$$\dot{\mathbf{r}} = K\widetilde{\Gamma}\mathbf{r}, \tag{219}$$
then the modal coefficients satisfy
$$\dot{c}_j = u_j^\top \dot{\mathbf{r}} = \lambda_j \sum_{k=1}^{N_r} c_k \, u_j^\top \widetilde{\Gamma} u_k. \tag{220}$$

Proof. For the residual-input dynamics,

$$\dot{\mathbf{r}} = K\Gamma\mathbf{1},$$

we compute

$$\dot{c}_j = u_j^\top \dot{\mathbf{r}} = u_j^\top K\Gamma\mathbf{1} = \lambda_j \, u_j^\top \Gamma\mathbf{1}, \tag{221}$$

which proves (218).

For the squared-residual-input dynamics,

$$\dot{\mathbf{r}} = K\widetilde{\Gamma}\mathbf{r},$$

we similarly obtain

$$\begin{aligned} \dot{c}_j = u_j^\top \dot{\mathbf{r}} &= u_j^\top K\widetilde{\Gamma}\mathbf{r} = \lambda_j \, u_j^\top \widetilde{\Gamma}\mathbf{r} \\ &= \lambda_j \sum_{k=1}^{N_r} c_k \, u_j^\top \widetilde{\Gamma} u_k, \end{aligned} \tag{222}$$

which proves (220). □

Interpretation. Proposition 9 reveals the structural difference between the two input choices. In the residual-input case, each mode is driven by an external discriminator-induced forcing term $u_j^\top \Gamma\mathbf{1}$. In the squared-residual-input case, by contrast, the evolution of mode j depends on all modal amplitudes $\{c_k\}$, so the residual spectrum is fed back into its own dynamics. This is the first indication that squared-residual input leads to a genuinely multiplicative feedback system.

To understand this multiplicative structure more systematically, it is useful to introduce the coupled operator $K\widetilde{\Gamma}$ and its symmetric counterpart. The same spectral factorization used for the linearly aligned residual-input dynamics applies here, with A replaced by $\widetilde{\Gamma}$. The crucial difference is the sign convention: in the residual-input aligned case, decay corresponds to $A \succeq 0$ in $\dot{\mathbf{r}} = -KA\mathbf{r}$, whereas in the squared-residual-input case, decay corresponds to $\widetilde{\Gamma} \preceq 0$ in $\dot{\mathbf{r}} = K\widetilde{\Gamma}\mathbf{r}$.

Proposition 10 (Spectral factorization of the squared-residual-input dynamics)**.** *Define*

$$M := K\widetilde{\Gamma}, \qquad H := K^{1/2}\widetilde{\Gamma}K^{1/2}. \tag{223}$$

Assume that K is symmetric positive definite on the active residual subspace, and that $\widetilde{\Gamma} \in \mathbb{R}^{N_r \times N_r}$ is symmetric. Then:

1. *M and H are similar. M and H have the same eigenvalues.*
2. *Since H is symmetric, all eigenvalues of M are real.*
3. *Since H is congruent to $\widetilde{\Gamma}$, Sylvester's law of inertia implies that H and $\widetilde{\Gamma}$ have the same inertia, and therefore so does M.*
4. *If $\widetilde{\Gamma} \preceq 0$, then all eigenvalues of M are nonpositive; if $\widetilde{\Gamma} \succeq 0$, then all eigenvalues of M are nonnegative; if $\widetilde{\Gamma}$ is indefinite, then M has both positive and negative eigenvalues.*

Proof. The similarity relation follows directly from

$$K^{-1/2} M K^{1/2} = K^{-1/2}(K\widetilde{\Gamma})K^{1/2} = K^{1/2}\widetilde{\Gamma}K^{1/2} = H. \tag{224}$$

Hence M and H have the same eigenvalues. Since H is symmetric whenever $\widetilde{\Gamma}$ is symmetric, its spectrum is real, and therefore so is the spectrum of M. Moreover, H is congruent to $\widetilde{\Gamma}$, so Sylvester's law of inertia implies that they have the same inertia; similarity then transfers the same inertia to M. The sign-definite and indefinite cases follow immediately. □

Interpretation. Proposition 10 shows that the squared-residual-input dynamics are governed not by K or $\widetilde{\Gamma}$ separately, but by their coupled spectrum. The symmetric matrix $H = K^{1/2}\widetilde{\Gamma}K^{1/2}$ provides the most transparent representation of this coupling, and its sign structure determines whether the active modes are contractive, expansive, or mixed.

This also gives a direct way to compare the spectrum of the coupled operator with those of K and $\widetilde{\Gamma}$.

Proposition 11 (Spectral bounds and energy behavior)**.** *Under the assumptions of Proposition 10:*

1. *If $\widetilde{\Gamma} \succeq 0$, then the extreme eigenvalues of M satisfy*

$$\lambda_{\min}(M) = \lambda_{\min}(H) \geq \lambda_{\min}(K)\lambda_{\min}(\widetilde{\Gamma}), \qquad \lambda_{\max}(M) = \lambda_{\max}(H) \leq \lambda_{\max}(K)\lambda_{\max}(\widetilde{\Gamma}). \tag{225}$$

2. *Introduce*

$$\mathbf{z} := K^{-1/2}\mathbf{r}. \tag{226}$$

Then the squared-residual-input dynamics become

$$\dot{\mathbf{z}} = H\mathbf{z}, \tag{227}$$

and the residual energy satisfies

$$\lambda_{\min}(K)\|\mathbf{z}\|^2 \leq \|\mathbf{r}\|^2 \leq \lambda_{\max}(K)\|\mathbf{z}\|^2. \tag{228}$$

3. *If $\widetilde{\Gamma} \preceq 0$, then $H \preceq 0$. In the strictly negative definite case $\widetilde{\Gamma} \prec 0$,*

$$E(t) \leq \kappa(K)\, E(0)\, e^{2\lambda_{\max}(H)t} \leq \kappa(K)\, E(0)\, e^{-2\lambda_{\min}(K)|\lambda_{\max}(\widetilde{\Gamma})|t}, \tag{229}$$

where $\lambda_{\max}(H) < 0$ and

$$\kappa(K) := \frac{\lambda_{\max}(K)}{\lambda_{\min}(K)}. \tag{230}$$

4. *If $\widetilde{\Gamma} \succeq 0$, then the transformed energy $\frac{1}{2}\|\mathbf{z}(t)\|^2$ is nondecreasing. In the strictly positive definite case $\widetilde{\Gamma} \succ 0$, the transformed dynamics are expansive, and the original residual energy admits the lower bound*

$$E(t) \geq \kappa(K)^{-1} E(0)\, e^{2\lambda_{\min}(H)t} \geq \kappa(K)^{-1} E(0)\, e^{2\lambda_{\min}(K)\lambda_{\min}(\widetilde{\Gamma})t}. \tag{231}$$

5. *If $\widetilde{\Gamma}$ is indefinite, then some modes decay while others grow, and the residual energy is not guaranteed to be monotone.*

Proof. For the extreme eigenvalue bounds, note that

$$H = K^{1/2}\widetilde{\Gamma}K^{1/2}$$

is symmetric. If $\widetilde{\Gamma} \succeq 0$, then for any $x \neq 0$,

$$x^\top H x = (K^{1/2}x)^\top \widetilde{\Gamma}(K^{1/2}x) \geq \lambda_{\min}(\widetilde{\Gamma})\, \|K^{1/2}x\|^2 \geq \lambda_{\min}(K)\lambda_{\min}(\widetilde{\Gamma})\, \|x\|^2. \tag{232}$$

Taking the minimum over unit vectors x yields

$$\lambda_{\min}(H) \geq \lambda_{\min}(K)\lambda_{\min}(\widetilde{\Gamma}).$$

Similarly,

$$x^\top H x \leq \lambda_{\max}(\widetilde{\Gamma})\, \|K^{1/2}x\|^2 \leq \lambda_{\max}(K)\lambda_{\max}(\widetilde{\Gamma})\, \|x\|^2, \tag{233}$$

and taking the maximum over unit vectors gives

$$\lambda_{\max}(H) \le \lambda_{\max}(K)\lambda_{\max}(\widetilde{\Gamma}).$$

Since M and H are similar by Proposition 10, they have the same eigenvalues, which proves (225).

Next, setting $\mathbf{z} = K^{-1/2}\mathbf{r}$, we obtain

$$\dot{\mathbf{z}} = K^{-1/2}\dot{\mathbf{r}} = K^{-1/2}K\widetilde{\Gamma}\mathbf{r} = K^{1/2}\widetilde{\Gamma}K^{1/2}\mathbf{z} = H\mathbf{z},$$

which proves (227). Since $\mathbf{r} = K^{1/2}\mathbf{z}$, the norm equivalence (228) follows immediately.

If $\widetilde{\Gamma} \preceq 0$, then $H \preceq 0$. In the strictly negative definite case, $\lambda_{\max}(H) < 0$, and

$$\frac{d}{dt}\frac{1}{2}\|\mathbf{z}(t)\|^2 = \mathbf{z}(t)^\top H\mathbf{z}(t) \le \lambda_{\max}(H)\|\mathbf{z}(t)\|^2. \tag{234}$$

Grönwall's inequality yields

$$\|\mathbf{z}(t)\|^2 \le e^{2\lambda_{\max}(H)t}\|\mathbf{z}(0)\|^2.$$

Combining this with (228) gives

$$E(t) \le \kappa(K)\, E(0)\, e^{2\lambda_{\max}(H)t}.$$

Moreover, since $\widetilde{\Gamma} \prec 0$, we have

$$\lambda_{\max}(H) \le \lambda_{\min}(K)\lambda_{\max}(\widetilde{\Gamma}) = -\lambda_{\min}(K)|\lambda_{\max}(\widetilde{\Gamma})|,$$

which proves (229).

If $\widetilde{\Gamma} \succeq 0$, then $H \succeq 0$, and therefore

$$\frac{d}{dt}\frac{1}{2}\|\mathbf{z}(t)\|^2 = \mathbf{z}(t)^\top H\mathbf{z}(t) \ge 0, \tag{235}$$

so the energy is nondecreasing. In the strictly positive definite case $\widetilde{\Gamma} \succ 0$, one has $\lambda_{\min}(H) > 0$, and

$$\frac{d}{dt}\frac{1}{2}\|\mathbf{z}(t)\|^2 \ge \lambda_{\min}(H)\|\mathbf{z}(t)\|^2. \tag{236}$$

Applying Grönwall's inequality gives

$$\|\mathbf{z}(t)\|^2 \ge e^{2\lambda_{\min}(H)t}\|\mathbf{z}(0)\|^2,$$

which, together with (228), yields

$$E(t) \ge \kappa(K)^{-1}E(0)e^{2\lambda_{\min}(H)t}.$$

Using

$$\lambda_{\min}(H) \ge \lambda_{\min}(K)\lambda_{\min}(\widetilde{\Gamma}),$$

we obtain (231).

Finally, if $\widetilde{\Gamma}$ is indefinite, then by Proposition 10, H has both positive and negative eigenvalues. Hence the linear system

$$\dot{\mathbf{z}} = H\mathbf{z}$$

contains both contractive and expansive modes, so some components decay while others grow, and monotonicity of the residual energy is no longer guaranteed. □

Interpretation. The three propositions above make the spectral distinction between the two input choices precise. With residual input, the discriminator acts as an external forcing projected onto each kernel mode. With squared-residual input, the residual state is fed back into the dynamics through the coupled operator $K\widetilde{\Gamma}$. As a result, the squared-residual-input case forms a genuine multiplicative feedback system.

The energy behavior is therefore controlled by the sign structure of the coupled spectrum. If $\widetilde{\Gamma} \preceq 0$, the active modes are non-expansive; if $\widetilde{\Gamma} \prec 0$, the residual energy decays exponentially; if $\widetilde{\Gamma} \succeq 0$, the active modes become expansive and the energy grows; and if $\widetilde{\Gamma}$ is indefinite, the system contains both damping and amplification, which naturally leads to oscillatory or unstable behavior. This is why replacing the discriminator input r by r^2 should be viewed not as a cosmetic modification, but as a structural change in the adversarial PINNs dynamics.

D.3 Interpretation and implications

The analysis above shows that replacing the discriminator input r by r^2 changes the adversarial PINNs dynamics in a structural rather than merely parametric manner. In the residual-input case,

$$\dot{\mathbf{r}} = K_{rr}^{G}\Gamma\mathbf{1},$$

so the discriminator acts as an external sample-wise forcing term. In the squared-residual-input case,

$$\dot{\mathbf{r}} = K_{rr}^{G}\widetilde{\Gamma}\mathbf{r},$$

so the discriminator enters multiplicatively through the current residual state. At the same time, the map $r \mapsto r^2$ removes sign information, making the discriminator sensitive only to residual magnitude rather than to its sign.

This modification has two opposite consequences. On the positive side, the extra factor r_i naturally amplifies the contribution of large-residual samples, which makes the adversarial signal more consistent with the squared-residual structure of standard PINNs training and may help concentrate optimization on poorly fitted regions. On the negative side, the same mechanism automatically weakens the update as $r_i \to 0$, which can slow late-stage refinement, and it may over-emphasize a small subset of large-residual modes, thereby worsening spectral imbalance.

Correspondingly, the energy law changes from the linear form

$$\dot{E} = \mathbf{r}^{\top}K_{rr}^{G}\Gamma\mathbf{1}$$

to the quadratic form

$$\dot{E} = \mathbf{r}^{\top}K_{rr}^{G}\widetilde{\Gamma}\mathbf{r}.$$

Thus, changing the discriminator input from r to r^2 should not be viewed as a minor implementation choice: it changes the continuous-time training law, the NTK-level spectral coupling, and the mechanism by which the discriminator influences convergence.

E Soft-Constrained PINNs under Adversarial Training

In the hard-constrained setting considered in the main text, the solution ansatz is designed so that boundary and initial conditions are satisfied identically, and adversarial training is applied only to the interior PDE residual. In the soft-constrained setting, by contrast, the solution network does not enforce these constraints exactly. As a result, the PDE residual, the boundary-condition mismatch, and the initial-condition mismatch must all be controlled during training. This naturally leads to a multi-discriminator adversarial formulation, in which a single generator u_θ is coupled to three discriminators f_1, f_2, f_3, each acting on a different violation channel.

In this appendix, we derive the resulting coupled dynamics in a unified way and show that, unlike the hard-constrained case, soft-constrained adversarial PINNs training is governed by a block-structured kernel dynamics with cross-coupling among the interior, boundary, and initial-condition channels.

E.1 Problem setup and violation channels

Consider a PDE of the form

$$\mathcal{N}[u](x) = a(x), \qquad x \in \Omega, \tag{237}$$

with boundary condition

$$\mathcal{G}_B[u](x) = g_B(x), \qquad x \in \partial\Omega, \tag{238}$$

and initial condition

$$\mathcal{G}_0[u](x) = g_0(x), \qquad x \in \Omega_0, \tag{239}$$

where Ω_0 denotes the initial manifold (for example, $t = 0$ in time-dependent problems).

Let u_θ be the generator (solution network). In the soft-constrained setting, we define three types of violations:

Interior residual. Given interior collocation points $\{x_i^r\}_{i=1}^{N_r} \subset \Omega$, define

$$r_i(\theta) := \mathcal{R}(x_i^r;\theta) := \mathcal{N}[u_\theta](x_i^r) - a(x_i^r), \qquad i = 1, \dots, N_r. \tag{240}$$

Collect them into

$$\mathbf{r}(\theta) = [r_1(\theta), \dots, r_{N_r}(\theta)]^{\top} \in \mathbb{R}^{N_r}. \tag{241}$$

Boundary-condition mismatch. Given boundary collocation points $\{x_j^b\}_{j=1}^{N_b} \subset \partial\Omega$, define

$$b_j(\theta) := \mathcal{B}(x_j^b;\theta) := \mathcal{G}_B[u_\theta](x_j^b) - g_B(x_j^b), \qquad j = 1,\dots,N_b. \tag{242}$$

Collect them into

$$\mathbf{b}(\theta) = [b_1(\theta),\dots,b_{N_b}(\theta)]^\top \in \mathbb{R}^{N_b}. \tag{243}$$

Initial-condition mismatch. Given initial-condition points $\{x_k^0\}_{k=1}^{N_0} \subset \Omega_0$, define

$$c_k(\theta) := \mathcal{I}(x_k^0;\theta) := \mathcal{G}_0[u_\theta](x_k^0) - g_0(x_k^0), \qquad k = 1,\dots,N_0. \tag{244}$$

Collect them into

$$\mathbf{c}(\theta) = [c_1(\theta),\dots,c_{N_0}(\theta)]^\top \in \mathbb{R}^{N_0}. \tag{245}$$

Thus, instead of a single residual vector, soft-constrained PINNs involve three generator-dependent violation vectors:

$$\mathbf{r}, \qquad \mathbf{b}, \qquad \mathbf{c}.$$

E.2 Adversarial formulation with three discriminators

As in the main text, the target for each violation channel is the zero state. Therefore, each channel induces a generated empirical distribution and a real distribution δ_0.

For the interior residual channel,

$$\hat{\mu}_\theta^r = \frac{1}{N_r}\sum_{i=1}^{N_r} \delta_{r_i(\theta)}, \qquad \mu_{\text{real}}^r = \delta_0. \tag{246}$$

For the boundary channel,

$$\hat{\mu}_\theta^b = \frac{1}{N_b}\sum_{j=1}^{N_b} \delta_{b_j(\theta)}, \qquad \mu_{\text{real}}^b = \delta_0. \tag{247}$$

For the initial channel,

$$\hat{\mu}_\theta^0 = \frac{1}{N_0}\sum_{k=1}^{N_0} \delta_{c_k(\theta)}, \qquad \mu_{\text{real}}^0 = \delta_0. \tag{248}$$

We introduce three discriminators with separate parameters:

$$f_1(\cdot;\phi_1), \qquad f_2(\cdot;\phi_2), \qquad f_3(\cdot;\phi_3),$$

where

- f_1 acts on interior residual samples r_i,
- f_2 acts on boundary mismatch samples b_j,
- f_3 acts on initial mismatch samples c_k.

Using the unified GAN notation P, Q, R from the main text, the discriminator objectives are

$$\max_{\phi_1} \mathcal{L}_{D_1}(\theta,\phi_1) = \frac{1}{N_r}\sum_{i=1}^{N_r} P\big(f_1(0;\phi_1)\big) + \frac{1}{N_r}\sum_{i=1}^{N_r} Q\big(f_1(r_i(\theta);\phi_1)\big), \tag{249}$$

$$\max_{\phi_2} \mathcal{L}_{D_2}(\theta,\phi_2) = \frac{1}{N_b}\sum_{j=1}^{N_b} P\big(f_2(0;\phi_2)\big) + \frac{1}{N_b}\sum_{j=1}^{N_b} Q\big(f_2(b_j(\theta);\phi_2)\big), \tag{250}$$

$$\max_{\phi_3} \mathcal{L}_{D_3}(\theta,\phi_3) = \frac{1}{N_0}\sum_{k=1}^{N_0} P\big(f_3(0;\phi_3)\big) + \frac{1}{N_0}\sum_{k=1}^{N_0} Q\big(f_3(c_k(\theta);\phi_3)\big). \tag{251}$$

Correspondingly, the generator minimizes the sum of three adversarial objectives:

$$\mathcal{L}_G(\theta,\phi_1,\phi_2,\phi_3) = \lambda_r \mathcal{L}_G^{(r)} + \lambda_b \mathcal{L}_G^{(b)} + \lambda_0 \mathcal{L}_G^{(0)}, \tag{252}$$

where

$$\mathcal{L}_G^{(r)} = \frac{1}{N_r}\sum_{i=1}^{N_r} R\big(f_1(r_i(\theta);\phi_1)\big), \tag{253}$$

$$\mathcal{L}_G^{(b)} = \frac{1}{N_b}\sum_{j=1}^{N_b} R\big(f_2(b_j(\theta);\phi_2)\big), \tag{254}$$

$$\mathcal{L}_G^{(0)} = \frac{1}{N_0}\sum_{k=1}^{N_0} R\big(f_3(c_k(\theta);\phi_3)\big). \tag{255}$$

Here $\lambda_r, \lambda_b, \lambda_0 > 0$ are channel weights balancing the three soft constraints.

E.3 Generator gradient flow and channel-wise adversarial weights

We now derive the continuous-time generator dynamics with the discriminator parameters ϕ_1, ϕ_2, ϕ_3 frozen.

For notational convenience, define

$$f_{1,i} := f_1(r_i;\phi_1), \qquad \partial_r f_{1,i} := \partial_r f_1(r_i;\phi_1), \tag{256}$$

$$f_{2,j} := f_2(b_j;\phi_2), \qquad \partial_b f_{2,j} := \partial_b f_2(b_j;\phi_2), \tag{257}$$

$$f_{3,k} := f_3(c_k;\phi_3), \qquad \partial_c f_{3,k} := \partial_c f_3(c_k;\phi_3). \tag{258}$$

Differentiating (252) with respect to θ yields

$$\nabla_\theta \mathcal{L}_G = \frac{\lambda_r}{N_r}\sum_{i=1}^{N_r} R'(f_{1,i})\,\partial_r f_{1,i}\,\nabla_\theta r_i + \frac{\lambda_b}{N_b}\sum_{j=1}^{N_b} R'(f_{2,j})\,\partial_b f_{2,j}\,\nabla_\theta b_j + \frac{\lambda_0}{N_0}\sum_{k=1}^{N_0} R'(f_{3,k})\,\partial_c f_{3,k}\,\nabla_\theta c_k. \tag{259}$$

Under generator gradient flow,

$$\dot\theta = -\nabla_\theta \mathcal{L}_G, \tag{260}$$

we define the three discriminator-induced sample-weight vectors

$$\gamma_i^{(r)} := -\frac{\lambda_r}{N_r} R'(f_{1,i})\,\partial_r f_{1,i}, \qquad i = 1,\dots,N_r, \tag{261}$$

$$\gamma_j^{(b)} := -\frac{\lambda_b}{N_b} R'(f_{2,j})\,\partial_b f_{2,j}, \qquad j = 1,\dots,N_b, \tag{262}$$

$$\gamma_k^{(0)} := -\frac{\lambda_0}{N_0} R'(f_{3,k})\,\partial_c f_{3,k}, \qquad k = 1,\dots,N_0. \tag{263}$$

Let

$$\boldsymbol{\gamma}^{(r)} = [\gamma_1^{(r)},\dots,\gamma_{N_r}^{(r)}]^\top, \qquad \boldsymbol{\gamma}^{(b)} = [\gamma_1^{(b)},\dots,\gamma_{N_b}^{(b)}]^\top, \qquad \boldsymbol{\gamma}^{(0)} = [\gamma_1^{(0)},\dots,\gamma_{N_0}^{(0)}]^\top.$$

Then (259) can be rewritten compactly as

$$\dot\theta = J_r^\top \boldsymbol{\gamma}^{(r)} + J_b^\top \boldsymbol{\gamma}^{(b)} + J_0^\top \boldsymbol{\gamma}^{(0)}, \tag{264}$$

where the three Jacobians are

$$J_r(\theta) := \begin{bmatrix} (\nabla_\theta r_1)^\top \\ \vdots \\ (\nabla_\theta r_{N_r})^\top \end{bmatrix} \in \mathbb{R}^{N_r \times p}, \tag{265}$$

$$J_b(\theta) := \begin{bmatrix} (\nabla_\theta b_1)^\top \\ \vdots \\ (\nabla_\theta b_{N_b})^\top \end{bmatrix} \in \mathbb{R}^{N_b \times p}, \tag{266}$$

$$J_0(\theta) := \begin{bmatrix} (\nabla_\theta c_1)^\top \\ \vdots \\ (\nabla_\theta c_{N_0})^\top \end{bmatrix} \in \mathbb{R}^{N_0 \times p}, \tag{267}$$

and $p = \dim(\theta)$.

Equation (264) already reveals the key difference from the hard-constrained case: the generator update is now the sum of three adversarially weighted Jacobian-transpose terms, one for each constraint channel.

E.4 Coupled violation dynamics and block kernel structure

We now derive the generator-induced dynamics for the three violation channels in the soft-constrained setting. Unlike the hard-constrained case, where the state is a single residual vector, the soft-constrained formulation involves three coupled channels: the interior residual, the boundary violation, and the initial-condition violation. Since all three channels depend on the same generator parameters, their dynamics are not independent. Instead, the resulting evolution is governed by a block kernel operator with both self-kernel and cross-kernel interactions.

Proposition 12 (Coupled soft-constrained violation dynamics)**.** *Let* $\mathbf{r}$*,* $\mathbf{b}$*, and* $\mathbf{c}$ *denote the interior, boundary, and initial-condition violation vectors, respectively, and let the generator flow be given by* (264)*. Then the induced channel-wise dynamics satisfy*

$$\dot{\mathbf{r}} = J_r\dot{\theta} = J_r J_r^\top \boldsymbol{\gamma}^{(r)} + J_r J_b^\top \boldsymbol{\gamma}^{(b)} + J_r J_0^\top \boldsymbol{\gamma}^{(0)}, \tag{268}$$

$$\dot{\mathbf{b}} = J_b\dot{\theta} = J_b J_r^\top \boldsymbol{\gamma}^{(r)} + J_b J_b^\top \boldsymbol{\gamma}^{(b)} + J_b J_0^\top \boldsymbol{\gamma}^{(0)}, \tag{269}$$

$$\dot{\mathbf{c}} = J_0\dot{\theta} = J_0 J_r^\top \boldsymbol{\gamma}^{(r)} + J_0 J_b^\top \boldsymbol{\gamma}^{(b)} + J_0 J_0^\top \boldsymbol{\gamma}^{(0)}. \tag{270}$$

Define the self-kernel blocks

$$K_{rr}^G := J_r J_r^\top, \qquad K_{bb}^G := J_b J_b^\top, \qquad K_{00}^G := J_0 J_0^\top, \tag{271}$$

and the cross-kernel blocks

$$K_{rb}^G := J_r J_b^\top, \qquad K_{r0}^G := J_r J_0^\top, \qquad K_{b0}^G := J_b J_0^\top, \tag{272}$$

with

$$K_{br}^G = (K_{rb}^G)^\top, \qquad K_{0r}^G = (K_{r0}^G)^\top, \qquad K_{0b}^G = (K_{b0}^G)^\top.$$

Then the channel-wise dynamics become

$$\dot{\mathbf{r}} = K_{rr}^G \boldsymbol{\gamma}^{(r)} + K_{rb}^G \boldsymbol{\gamma}^{(b)} + K_{r0}^G \boldsymbol{\gamma}^{(0)}, \tag{273}$$

$$\dot{\mathbf{b}} = K_{br}^G \boldsymbol{\gamma}^{(r)} + K_{bb}^G \boldsymbol{\gamma}^{(b)} + K_{b0}^G \boldsymbol{\gamma}^{(0)}, \tag{274}$$

$$\dot{\mathbf{c}} = K_{0r}^G \boldsymbol{\gamma}^{(r)} + K_{0b}^G \boldsymbol{\gamma}^{(b)} + K_{00}^G \boldsymbol{\gamma}^{(0)}. \tag{275}$$

Further define the stacked violation vector

$$\mathbf{z} := \begin{bmatrix} \mathbf{r} \\ \mathbf{b} \\ \mathbf{c} \end{bmatrix} \in \mathbb{R}^{N_r + N_b + N_0}, \tag{276}$$

the stacked adversarial-weight vector

$$\boldsymbol{\gamma} := \begin{bmatrix} \boldsymbol{\gamma}^{(r)} \\ \boldsymbol{\gamma}^{(b)} \\ \boldsymbol{\gamma}^{(0)} \end{bmatrix}, \tag{277}$$

and the block kernel matrix

$$K_{\mathrm{soft}}^G := \begin{bmatrix} K_{rr}^G & K_{rb}^G & K_{r0}^G \\ K_{br}^G & K_{bb}^G & K_{b0}^G \\ K_{0r}^G & K_{0b}^G & K_{00}^G \end{bmatrix}. \tag{278}$$

Then the coupled soft-constrained dynamics admit the compact form

$$\dot{\mathbf{z}} = K_{\mathrm{soft}}^G \, \boldsymbol{\gamma}. \tag{279}$$

Proof. By differentiating the three violation vectors $\mathbf{r}$, $\mathbf{b}$, and $\mathbf{c}$ along the generator flow (264), we obtain

$$\dot{\mathbf{r}} = J_r\dot{\theta} = J_r J_r^\top \boldsymbol{\gamma}^{(r)} + J_r J_b^\top \boldsymbol{\gamma}^{(b)} + J_r J_0^\top \boldsymbol{\gamma}^{(0)}, \tag{280}$$

$$\dot{\mathbf{b}} = J_b\dot{\theta} = J_b J_r^\top \boldsymbol{\gamma}^{(r)} + J_b J_b^\top \boldsymbol{\gamma}^{(b)} + J_b J_0^\top \boldsymbol{\gamma}^{(0)}, \tag{281}$$

$$\dot{\mathbf{c}} = J_0\dot{\theta} = J_0 J_r^\top \boldsymbol{\gamma}^{(r)} + J_0 J_b^\top \boldsymbol{\gamma}^{(b)} + J_0 J_0^\top \boldsymbol{\gamma}^{(0)}. \tag{282}$$

This proves (268)–(270).

Introducing the self-kernel and cross-kernel blocks by

$$K_{rr}^G := J_r J_r^\top, \quad K_{bb}^G := J_b J_b^\top, \quad K_{00}^G := J_0 J_0^\top,$$

and

$$K_{rb}^G := J_r J_b^\top, \quad K_{r0}^G := J_r J_0^\top, \quad K_{b0}^G := J_b J_0^\top,$$

with the transpose relations

$$K_{br}^G = (K_{rb}^G)^\top, \qquad K_{0r}^G = (K_{r0}^G)^\top, \qquad K_{0b}^G = (K_{b0}^G)^\top,$$

we immediately obtain (273)–(275).

Finally, stacking the three channels into the single vector

$$\mathbf{z} = \begin{bmatrix} \mathbf{r} \\ \mathbf{b} \\ \mathbf{c} \end{bmatrix}, \qquad \boldsymbol{\gamma} = \begin{bmatrix} \boldsymbol{\gamma}^{(r)} \\ \boldsymbol{\gamma}^{(b)} \\ \boldsymbol{\gamma}^{(0)} \end{bmatrix},$$

and defining the block kernel matrix K_{soft}^G as in (278), the three coupled equations are exactly equivalent to the compact block form

$$\dot{\mathbf{z}} = K_{\mathrm{soft}}^G \boldsymbol{\gamma},$$

which proves (279). □

Interpretation. Proposition 12 is the soft-constrained analogue of the hard-constrained adversarial PINNs dynamics in the main text. The key difference is that the state is no longer a single residual vector, but a stacked multi-channel violation vector. Accordingly, the generator-side kernel is no longer a single NTK matrix, but a block operator containing both self-channel and cross-channel interactions.

The cross-kernel blocks K_{rb}^G, K_{r0}^G, and K_{b0}^G are especially important. They show that even if a discriminator acts only on one channel, the resulting generator update will generally propagate to the other channels through the shared generator parameters. Thus, soft-constrained adversarial PINNs training is intrinsically coupled: interior, boundary, and initial-condition corrections cannot be analyzed independently, but must be treated as a single block dynamical system.

E.5 Diagonal-weight representation

To make the connection with the main text more explicit, define diagonal matrices

$$\Gamma_r = \mathrm{diag}(\gamma_1^{(r)}, \dots, \gamma_{N_r}^{(r)}), \qquad \Gamma_b = \mathrm{diag}(\gamma_1^{(b)}, \dots, \gamma_{N_b}^{(b)}), \qquad \Gamma_0 = \mathrm{diag}(\gamma_1^{(0)}, \dots, \gamma_{N_0}^{(0)}). \tag{283}$$

Let $\mathbf{1}_r, \mathbf{1}_b, \mathbf{1}_0$ denote the all-one vectors of dimensions N_r, N_b, N_0, respectively. Then

$$\boldsymbol{\gamma}^{(r)} = \Gamma_r \mathbf{1}_r, \qquad \boldsymbol{\gamma}^{(b)} = \Gamma_b \mathbf{1}_b, \qquad \boldsymbol{\gamma}^{(0)} = \Gamma_0 \mathbf{1}_0.$$

Define

$$\Gamma_{\mathrm{soft}} := \mathrm{diag}(\Gamma_r, \Gamma_b, \Gamma_0), \qquad \mathbf{1}_{\mathrm{soft}} := \begin{bmatrix} \mathbf{1}_r \\ \mathbf{1}_b \\ \mathbf{1}_0 \end{bmatrix}. \tag{284}$$

Then (279) can be rewritten as

$$\dot{\mathbf{z}} = K_{\mathrm{soft}}^G \, \Gamma_{\mathrm{soft}} \, \mathbf{1}_{\mathrm{soft}}. \tag{285}$$

This form makes clear that soft-constrained adversarial training is still a discriminator-induced sample-dependent preconditioned dynamics, but now at the level of a coupled three-channel system rather than a single residual channel.

E.6 Energy dynamics

A natural total violation energy is

$$E_{\mathrm{soft}}(t) := \frac{1}{2}\|\mathbf{r}(t)\|^2 + \frac{1}{2}\|\mathbf{b}(t)\|^2 + \frac{1}{2}\|\mathbf{c}(t)\|^2 = \frac{1}{2}\|\mathbf{z}(t)\|^2. \tag{286}$$

Differentiating (286) and using (279) gives

$$\dot{E}_{\mathrm{soft}} = \mathbf{z}^\top \dot{\mathbf{z}} = \mathbf{z}^\top K^G_{\mathrm{soft}}\, \boldsymbol{\gamma} = \mathbf{z}^\top K^G_{\mathrm{soft}}\, \Gamma_{\mathrm{soft}}\, \mathbf{1}_{\mathrm{soft}}. \tag{287}$$

Equivalently, in expanded form,

$$\begin{aligned}\dot{E}_{\mathrm{soft}} = {} & \mathbf{r}^\top \Big(K^G_{rr}\boldsymbol{\gamma}^{(r)} + K^G_{rb}\boldsymbol{\gamma}^{(b)} + K^G_{r0}\boldsymbol{\gamma}^{(0)} \Big) \\ & + \mathbf{b}^\top \Big(K^G_{br}\boldsymbol{\gamma}^{(r)} + K^G_{bb}\boldsymbol{\gamma}^{(b)} + K^G_{b0}\boldsymbol{\gamma}^{(0)} \Big) \\ & + \mathbf{c}^\top \Big(K^G_{0r}\boldsymbol{\gamma}^{(r)} + K^G_{0b}\boldsymbol{\gamma}^{(b)} + K^G_{00}\boldsymbol{\gamma}^{(0)} \Big).\end{aligned} \tag{288}$$

Therefore, unlike in the hard-constrained case, the decrease of total violation energy is jointly determined by

1. the self-kernel blocks $K^G_{rr}, K^G_{bb}, K^G_{00}$,
2. the cross-kernel blocks $K^G_{rb}, K^G_{r0}, K^G_{b0}$,
3. the three discriminator-induced sample-weight vectors $\boldsymbol{\gamma}^{(r)}, \boldsymbol{\gamma}^{(b)}, \boldsymbol{\gamma}^{(0)}$.

This means that improving one channel adversarially may either help or hurt the others, depending on the sign and structure of the corresponding cross-kernel couplings.

E.7 Constant-kernel regime and NTK interpretation

In the infinite-width or lazy-training regime, it is natural to approximate the block kernel matrix by a constant operator,

$$K^G_{\mathrm{soft}}(\theta) \approx K^G_{\mathrm{soft}}. \tag{289}$$

The soft-constrained adversarial dynamics then reduce to

$$\dot{\mathbf{z}} = K^G_{\mathrm{soft}}\, \Gamma_{\mathrm{soft}}\, \mathbf{1}_{\mathrm{soft}}, \tag{290}$$

where

$$\mathbf{z} = \begin{bmatrix} \mathbf{r} \\ \mathbf{b} \\ \mathbf{c} \end{bmatrix}, \qquad \Gamma_{\mathrm{soft}} = \mathrm{diag}(\Gamma_r, \Gamma_b, \Gamma_0).$$

Equation (290) shows that, unlike the hard-constrained case, the generator is no longer governed by a single residual-regression dynamics, but by a coupled block-NTK system in which the three discriminators jointly modulate the evolution of the interior residual, boundary mismatch, and initial-condition mismatch.

The essential new feature is the presence of cross-channel coupling. Although f_1, f_2, f_3 act on the three channels separately and induce channel-wise diagonal weightings $\Gamma_r, \Gamma_b, \Gamma_0$, these weightings are all propagated through the same shared generator kernel K^G_{soft}. As a consequence, an update encouraged by one discriminator generally affects not only its own channel, but also the other two through the off-diagonal kernel blocks. Therefore, the success of soft-constrained adversarial training depends not only on whether each discriminator provides a useful weighting within its own channel, but also on whether the three channels remain balanced under the shared generator dynamics.

A convenient coarse diagnostic for this balance can be obtained by examining the diagonal block interactions between the generator kernel and the discriminator-induced weighting matrices. More specifically, define

$$T_r := \mathrm{Tr}\big(K^G_{rr}\Gamma_r\big), \qquad T_b := \mathrm{Tr}\big(K^G_{bb}\Gamma_b\big), \qquad T_0 := \mathrm{Tr}\big(K^G_{00}\Gamma_0\big). \tag{291}$$

Each of these quantities summarizes, at an aggregated level, how strongly the generator-side sensitivity and the discriminator-induced sample weighting interact within the corresponding channel. In this

sense, T_r, T_b, T_0 can be viewed as coarse channel-wise driving strengths for the residual, boundary, and initial-condition modules, respectively.

To compare the three channels on a common scale, we further introduce the normalized balance indicators

$$\rho_r := \frac{T_r}{T_r + T_b + T_0}, \qquad \rho_b := \frac{T_b}{T_r + T_b + T_0}, \qquad \rho_0 := \frac{T_0}{T_r + T_b + T_0}, \tag{292}$$

whenever the denominator is nonzero. By construction,

$$\rho_r + \rho_b + \rho_0 = 1. \tag{293}$$

These ratios provide a simple first-order measure of how the total effective adversarial drive is distributed across the three channels. If

$$\rho_r \approx \rho_b \approx \rho_0 \approx \frac{1}{3},$$

then the residual, boundary, and initial-condition modules are being driven at roughly comparable levels. If one ratio is significantly larger than the others, then the corresponding channel is dominating the generator update; if one ratio is much smaller, then that channel is comparatively under-driven.

To quantify this deviation from balanced evolution, one may further define a scalar imbalance score

$$\mathcal{B}_{\mathrm{imb}} := \left(\rho_r - \frac{1}{3}\right)^2 + \left(\rho_b - \frac{1}{3}\right)^2 + \left(\rho_0 - \frac{1}{3}\right)^2. \tag{294}$$

A smaller value of $\mathcal{B}_{\mathrm{imb}}$ indicates that the three channels are being updated in a more balanced manner, whereas a larger value indicates that the training dynamics are biased toward only one or two components. In particular, a persistently large $\mathcal{B}_{\mathrm{imb}}$ suggests that the soft-constrained adversarial training may be over-emphasizing one class of violations while neglecting the others.

Of course, these trace-based quantities do not capture the full off-diagonal coupling structure of K^G_{soft}. Nevertheless, they provide a useful and easily interpretable first-order diagnostic: they isolate the interaction between the generator-side diagonal kernel blocks and the discriminator-side weighting matrices, and thereby allow one to infer whether the residual, boundary, and initial-condition modules are evolving in a balanced way before analyzing the more detailed cross-channel effects.

E.8 Discriminator-side functional dynamics

For completeness, we also write the discriminator-side gradient-flow dynamics. Since the three discriminators are independent given the current generator state, each evolves on its own violation channel.

For $\ell \in \{1, 2, 3\}$, let $f_\ell(\cdot; \phi_\ell)$ denote the corresponding discriminator and let k^D_ℓ be its NTK kernel. Let the fake samples be

$$\xi^{(1)}_i := r_i, \qquad \xi^{(2)}_j := b_j, \qquad \xi^{(3)}_k := c_k.$$

Then, under discriminator gradient ascent,

$$\dot{\phi}_\ell = \nabla_{\phi_\ell} \mathcal{L}_{D_\ell}, \tag{295}$$

the induced functional dynamics satisfy, for any input z,

$$\frac{d}{dt} f_\ell(z; \phi_\ell(t)) = \frac{1}{N_\ell} \sum_{m=1}^{N_\ell} k^D_\ell\big(z, 0\big)\, P'\big(f_\ell(0; \phi_\ell(t))\big) + \frac{1}{N_\ell} \sum_{m=1}^{N_\ell} k^D_\ell\big(z, \xi^{(\ell)}_m\big)\, Q'\big(f_\ell(\xi^{(\ell)}_m; \phi_\ell(t))\big), \tag{296}$$

where $N_1 = N_r$, $N_2 = N_b$, and $N_3 = N_0$. Thus each discriminator learns a channel-specific weighting function shaped by its own NTK and its own empirical violation distribution.

F Existing Adaptive and Weak-Adversarial PINNs Methods as Constrained-Discriminator Special Cases

In this appendix, we show that several existing PINNs variants, including SA-PINN, LA-PINN, and weak-adversarial methods, can be embedded into a unified adversarial PINNs framework by restricting

the discriminator to a prescribed functional subspace. Under this viewpoint, these methods do not require a fundamentally different generator-side training law; rather, they differ in the admissible class of discriminator functions and therefore in the induced sample-weighting or mode-weighting structure acting on the generator dynamics. However, in contrast to the label convention adopted in the main text—where the real target 0 is assigned label 1 and the generated data is assigned label 0—the following methods use a different normalization. Specifically, the discriminator is constrained to satisfy $f(0) = y_{\mathrm{real}} := 0$ rather than 1.

F.1 General constrained-discriminator formulation

Let $x \in \mathcal{X}$ denote the generated discrepancy variable. Depending on the method, x may represent a pointwise residual, a boundary or initial mismatch, or a weak-form discrepancy. The real target is always the zero-violation state, so the real distribution is concentrated at $x = 0$.

Consider the generator-side adversarial objective

$$\mathcal{L}_G(\theta, \phi) = \frac{1}{N} \sum_{i=1}^{N} R\big(f(x_i(\theta); \phi)\big), \tag{297}$$

where $f(\cdot; \phi)$ is the discriminator (or critic), R is the generator-side adversarial potential, and $\{x_i(\theta)\}_{i=1}^N$ are the generated discrepancies induced by the solution network u_θ.

We now restrict f to a structured family

$$\mathcal{F}_{\mathrm{constr}} \subset \mathcal{F},$$

and assume that every admissible discriminator satisfies the anchoring condition

$$f(0; \phi) = y_{\mathrm{real}} = 0, \tag{298}$$

where y_{real} is the prescribed real label or real-score baseline. In other words, the discriminator is forced to match the real target exactly at the zero-violation state, and only its response on generated discrepancies $x \neq 0$ is trainable.

Let

$$\mathbf{x}(\theta) = [x_1(\theta), \ldots, x_N(\theta)]^\top$$

and define

$$J_x(\theta) := \begin{bmatrix} (\nabla_\theta x_1)^\top \\ \vdots \\ (\nabla_\theta x_N)^\top \end{bmatrix}, \qquad K_{xx}^G(\theta) := J_x J_x^\top.$$

If the generator is trained by gradient flow,

$$\dot{\theta} = -\nabla_\theta \mathcal{L}_G,$$

then the induced discrepancy dynamics are given by

$$\dot{\mathbf{x}} = K_{xx}^G(\theta)\, \boldsymbol{\gamma}, \qquad \gamma_i = -\frac{1}{N} R'\big(f(x_i; \phi)\big)\, \partial_x f(x_i; \phi). \tag{299}$$

Therefore, once the discriminator class is restricted, the generator-side dynamics are completely determined by the admissible form of $f(x; \phi)$, or equivalently by the induced weighting law γ_i.

F.2 SA-PINN as a monotone masked discriminator

In SA-PINN, the discriminator acts on pointwise violation quantities. More precisely, the input variable x denotes a generic pointwise discrepancy, which may represent any one of the three soft-constrained PINNs channels (240), (242) and (244):

$$x \in \{\, r_i,\ b_j,\ c_k \,\},$$

where r_i is the PDE residual at an interior collocation point, b_j is the boundary-condition mismatch at a boundary point, and c_k is the initial-condition mismatch at an initial point.

From the viewpoint of the present paper, SA-PINN can be interpreted within the IPM-type (36) adversarial framework, namely the same general class as WGAN-type objectives, but with the discriminator restricted to a highly structured function family. Instead of optimizing over an unconstrained critic class, SA-PINN introduces trainable nonnegative pointwise weights $m(\lambda_i)$, where the mask m is nonnegative, differentiable, and strictly increasing, and the mask parameters are updated by ascent while the solution network is updated by descent. This is equivalent to constraining the discriminator to the family

$$f_{\mathrm{SA}}(x;\lambda) = -\frac{1}{2}m(\lambda)x^2, \qquad m(\lambda) \geq 0, \qquad m'(\lambda) > 0. \tag{300}$$

Thus the discriminator is restricted to a one-dimensional monotone family acting on squared discrepancies, rather than being free over an arbitrary nonlinear hypothesis class.

Moreover,

$$\partial_x f_{\mathrm{SA}}(x;\lambda) = -m(\lambda)x,$$

and, for the IPM/WGAN-type generator choice $Q(t) = R(t) = -t$, the induced generator-side weighting reduces to

$$\gamma_i^{\mathrm{SA}} = -\frac{1}{N}m(\lambda_i)\, x_i. \tag{301}$$

Hence SA-PINN can be interpreted as a constrained-discriminator adversarial PINNs under an IPM-type objective, whose critic is limited to a monotone mask family on x^2, consistent with its original trainable-mask construction [19].

F.3 LA-PINN as a linear attentional discriminator

In LA-PINN, the discriminator also acts on pointwise violation quantities. As in the SA-PINN case, the input variable x denotes a generic pointwise discrepancy, which may represent any one of the three soft-constrained PINNs channels:

$$x \in \{\, r_i,\ b_j,\ c_k \,\}.$$

From the viewpoint of the present paper, LA-PINN can also be interpreted within an IPM-type adversarial framework, namely the same general class as WGAN-type objectives, but with the discriminator restricted to a structured linear attentional family. LA-PINN feeds pointwise squared errors into independent loss-attentional networks (LANs) and explicitly emphasizes that the attentional transformation is linear, without nonlinear activation in the LAN weighting stage. This leads naturally to the constrained discriminator family

$$f_{\mathrm{LA}}(x;\xi) = -\frac{1}{2}\,\mathrm{LAN}_\xi(x^2), \qquad \mathrm{LAN}_\xi(0) = 0, \tag{302}$$

where LAN_ξ is a linear map parameterized by ξ. The condition $\mathrm{LAN}_\xi(0) = 0$ ensures that

$$f_{\mathrm{LA}}(0;\xi) = 0,$$

which is consistent with the zero-discrepancy target in the IPM setting.

If LAN_ξ acts pointwise, for instance,

$$\mathrm{LAN}_\xi(x_i^2) = a_i(\xi)x_i^2 + b_i(\xi),$$

then the constraint $\mathrm{LAN}_\xi(0) = 0$ implies $b_i(\xi) = 0$, and hence

$$f_{\mathrm{LA}}(x_i;\xi) = -\frac{1}{2}a_i(\xi)x_i^2.$$

In our notation, this form is exactly equivalent to Eq. (17) in [9]. Therefore,

$$\partial_x f_{\mathrm{LA}}(x_i;\xi) = -a_i(\xi)x_i,$$

and, for the IPM/WGAN-type choice $Q(t) = R(t) = -t$, the induced weighting becomes

$$\gamma_i^{\mathrm{LA}} = -\frac{1}{N}a_i(\xi)\, x_i. \tag{303}$$

Thus LA-PINN may be viewed as a constrained-discriminator adversarial PINNs under an IPM-type objective, with a more flexible linear attentional family than SA-PINN, but still far more structured than a generic MLP critic. This interpretation matches the original LAN design, where the pointwise squared errors are the inputs and the weighting transformation is purely linear [9].

F.4 Weak-adversarial methods as projection-type discriminators

Weak-adversarial methods such as WAN [20] do not act on pointwise strong-form residuals directly. Instead, they optimize a saddle-point problem induced by the weak formulation, where the adversarial network parameterizes a test function and seeks the most challenging weak direction for the current solution network. From the viewpoint of the present paper, this can be interpreted within the same IPM-type adversarial framework, namely the same general class as WGAN-type objectives with

$$Q(t) = R(t) = -t,$$

but with the discriminator restricted to a normalized projection-type family in weak space:

$$f_{\mathrm{W}}(x;\eta) = -\frac{1}{2}\log\left(\frac{|\langle x, \varphi_\eta\rangle|^2}{\|\varphi_\eta\|_2^2}\right), \tag{304}$$

where φ_η is the adversarial test function. Here x denotes the discrepancy object in weak space, such as the weak residual functional induced by the current generator. By contrast, the boundary and initial conditions are not modified through the discriminator, but are still enforced through standard L^2-type errors.

Unlike SA-PINN or LA-PINN, the discriminator is now restricted not to pointwise masks on x^2, but to normalized rank-one projection-type forms in function space. The adversarial variable η searches for the most challenging direction φ_η, while the generator attempts to suppress that weak projection.

Formally, writing

$$g(x,\eta) := \frac{|\langle x, \varphi_\eta\rangle|^2}{\|\varphi_\eta\|_2^2},$$

we have

$$\partial_x f_{\mathrm{W}}(x;\eta) = -\frac{1}{2}\frac{1}{g(x,\eta)}\,\partial_x g(x,\eta). \tag{305}$$

Since

$$\partial_x g(x,\eta) = \frac{2\langle x, \varphi_\eta\rangle}{\|\varphi_\eta\|_2^2}\,\varphi_\eta,$$

it follows that

$$\partial_x f_{\mathrm{W}}(x;\eta) = -\frac{\varphi_\eta}{\langle x, \varphi_\eta\rangle}. \tag{306}$$

Thus, under the IPM/WGAN-type choice $Q(t) = R(t) = -t$, the induced generator update is still directed by the most adversarial weak test direction, but now with an additional inverse-projection normalization coming from the logarithmic form. The induced weighting becomes

$$\gamma_i^{\mathrm{W}} = -\frac{1}{N}\,\frac{\varphi_\eta}{\langle x, \varphi_\eta\rangle}. \tag{307}$$

Therefore, weak-adversarial PINNs can be interpreted as constrained-discriminator adversarial PINNs under an IPM-type objective, whose critic is limited to normalized weak-space projection forms, consistent with the original WAN saddle-point formulation based on induced operator norms [20].

F.5 Unified special-case theorem

Theorem 2 (SA-PINN, LA-PINN, and weak-adversarial methods as constrained-discriminator special cases). Each of the following methods can be embedded into the general generator-side dynamics

$$\dot{\mathbf{x}} = K_{xx}^G(\theta)\,\boldsymbol{\gamma}$$

by restricting the discriminator to a prescribed structured family:

$$\text{SA-PINN:} \qquad f(x) = -\frac{1}{2}m(\lambda)x^2, \qquad m(\lambda) \geq 0,\ \ m'(\lambda) > 0, \tag{308}$$

$$\text{LA-PINN:} \qquad f(x) = -\frac{1}{2}\mathrm{LAN}_\xi(x^2), \qquad \mathrm{LAN}_\xi \text{ linear}, \tag{309}$$

$$\text{Weak-adversarial PINN:} \qquad f(x) = -\frac{1}{2}\log\left(\frac{|\langle x, \varphi_\eta\rangle|^2}{\|\varphi_\eta\|_2^2}\right), \qquad \varphi_\eta \text{ is the test function.} \tag{310}$$

In all three cases, the discriminator is anchored at the zero-violation state,

$$f(0) = 0,$$

and is only allowed to vary on generated violations inside a prescribed structured subspace. Consequently, these methods differ primarily in the admissible structure of the discriminator-induced weighting $\boldsymbol{\gamma}$, rather than in the generator-side kernel form itself.

Proof. For each of the three constructions above, the discriminator family is explicitly restricted to a structured class and satisfies the zero-state anchoring condition $f(0) = 0$. Moreover, in all cases the generator-side weighting can be written in the unified form

$$\gamma_i = -\frac{1}{N} R'\big(f(x_i)\big)\, \partial_x f(x_i),$$

or its weak-space analogue in the weak-adversarial case. Therefore, by Proposition 1, each method admits the same generator-side representation

$$\dot{\mathbf{x}} = K_{xx}^{G}(\theta)\, \boldsymbol{\gamma}.$$

Hence they are all realizations of a broader adversarial PINNs framework with constrained discriminator classes. □

Theorem 2 has an immediate consequence for the NTK-based analysis developed in the main text.

Corollary 3. The unified discriminator-induced weighting framework is not limited to standard GAN or IPM critics, but also applies to SA-PINN, LA-PINN, and weak-adversarial PINNs once their discriminators are interpreted as belonging to constrained subspaces.

Interpretation. Under this viewpoint,

- SA-PINN corresponds to a monotone mask-induced weighting on pointwise squared violations;
- LA-PINN corresponds to a linear attentional weighting on pointwise squared violations;
- weak-adversarial methods correspond to a projection-induced weighting in weak space.

Therefore, these methods can be analyzed within the same generator-side NTK framework, even though their discriminator classes are much more restricted than those of generic adversarial neural networks.

More importantly, this viewpoint clarifies that the essential distinction among these methods is not whether they are formally written as GANs, but rather the functional subspace in which the discriminator is allowed to live. Once this subspace is fixed, the induced weighting law $\boldsymbol{\gamma}$ is fixed accordingly, and the generator dynamics follow the same kernel-driven mechanism as in the general adversarial PINNs analysis.

G A broader design space for adversarial PINNs training / Future work

The analysis developed in this paper suggests that adversarial PINNs training should be viewed as a broad design space rather than as a small collection of isolated named algorithms. Two largely orthogonal design dimensions emerge naturally: the *discriminator input* and the *discriminator objective or functional class*. Existing methods only occupy a limited subset of this space.

An additional important point is that several methods that appear distinct from standard adversarial PINNs can in fact be understood within the same general framework. More precisely, many such methods do not leave the GAN/IPM paradigm altogether; instead, they can be interpreted as standard GAN-type or IPM-type adversarial schemes in which the discriminator is restricted to a structured function space. From this viewpoint, the essential distinction among methods is often not whether they are "adversarial" in a completely different sense, but rather what discriminator input is used and in what functional class the discriminator is allowed to live.

G.1 Dimension I: residual input versus squared-residual input

The first design choice concerns the quantity fed into the discriminator. Two natural options are the residual itself,

$$x = r,$$

and the squared residual,

$$x = r^2.$$

When the discriminator takes r as input, the sign information of the violation is preserved. The discriminator can therefore distinguish positive and negative residuals and induce a direction-sensitive correction signal. In the NTK residual-dynamics formulation of the main text, this leads to an additive forcing law

$$\dot{\mathbf{r}} = K_{rr}^{G} \Gamma \mathbf{1}.$$

When the discriminator instead takes r^2 as input, the sign information is removed and only the magnitude of the violation is retained. In this case, the discriminator naturally emphasizes large residuals, and the generator dynamics become multiplicatively coupled:

$$\dot{\mathbf{r}} = K_{rr}^{G} \widetilde{\Gamma} \mathbf{r}.$$

Thus the distinction between r and r^2 is structural: the former yields a sign-sensitive additive mechanism, while the latter yields a sign-free magnitude-gated multiplicative mechanism.

G.2 Dimension II: discriminator objective and functional class

The second design choice concerns the discriminator itself. At a broad level, the methods considered in this work may be grouped into the following categories.

Classification-type discriminators (GAN). These discriminators are trained through a classification-style adversarial objective and produce a signal tied primarily to separation of the zero target from the generated residual samples.

Regression-type discriminators (LSGAN). These discriminators learn a smooth regression from residual samples to target labels. Compared with classification-type objectives, they generate a smoother and more local correction signal, which can be particularly suitable when the training goal is to drive every residual point toward zero.

Witness-function discriminators (IPM/WGAN). These discriminators learn a witness function measuring the discrepancy between the empirical generated distribution and the real target distribution. Under the NTK view, the generator is driven by the input derivative of this witness field, so these methods emphasize global distributional discrepancy rather than direct pointwise fitting.

Regularized witness discriminators (WGAN-GP). These methods retain the witness-function interpretation of IPM/WGAN while additionally regularizing the discriminator, for instance through gradient penalties, in order to improve smoothness and stability.

Monotone constrained discriminators (SA-PINN-type). Here the discriminator is restricted to a trainable monotone mask acting on pointwise discrepancy magnitudes. From the present viewpoint, SA-PINN is naturally interpreted as a standard adversarial framework with the discriminator restricted to a monotone mask family.

Linear constrained discriminators (LA-PINN-type). In this class, the discriminator is restricted to a linear attentional transform of pointwise squared errors. Thus LA-PINN may be understood as an adversarial scheme with a more expressive but still highly structured discriminator class.

Weak-space discriminators (WAN-type). Weak adversarial methods do not act directly on strong-form pointwise residuals. Instead, they parameterize an adversarial test function in weak space and maximize the normalized weak-form discrepancy. From the present viewpoint, this again fits naturally into the same broader adversarial framework, but with the discriminator restricted to a weak-space projection class.

Table 3: Combinations of discriminator input and basic adversarial objective in adversarial PINNs. A check mark indicates that a representative method already exists in the literature or has a clear correspondence, while △ highlights combinations that we believe are particularly promising.

Discriminator input	**GAN**	**LSGAN**	**IPM-GAN**
Residual r	✓[6]	✓[This paper]	✓[20]
Squared residual r^2	×	△	✓[7, 9, 19]

Table 4: Adversarial objectives and discriminator function classes. The first column corresponds to the unconstrained setting, while the other columns correspond to additional structural restrictions on the discriminator.

Adversarial framework	**Unconstrained**	**Spectra Normalization**	**Gradient-Penalty(GP)**	**Monotone-Mask**	**Linear-Transform**	**Weak-Space**
GAN	×	✓[6]	×	×	×	×
LSGAN	✓[This paper]	×	×	×	×	△
IPM-GAN	×	×	✓[7]	✓[19]	✓[9]	✓[20]

G.3 Why SA-PINN, LA-PINN, and WAN each have distinctive advantages

This taxonomy helps explain why SA-PINN, LA-PINN, and WAN can all be effective, but for different reasons.

SA-PINN. The strength of SA-PINN lies in its simplicity and robustness. By assigning a trainable nonnegative weight to each point and updating these weights by ascent, it automatically increases the emphasis on points whose losses remain large. This makes it particularly effective at highlighting stiff or hard-to-fit regions.

LA-PINN. LA-PINN inherits the pointwise weighting idea but replaces the simple monotone mask with a more expressive linear attentional map. This gives the discriminator more flexibility in redistributing emphasis across points while preserving the stability and interpretability of a structured weighting rule.

WAN. The advantage of WAN is of a different nature. Rather than refining pointwise weighting, WAN changes the space in which adversarial training is performed. The discriminator becomes a weak test function, and the adversarial interaction takes place over the weak formulation of the PDE. This is particularly attractive when the weak form is more natural or more stable than the strong form.

G.4 A two-dimensional taxonomy of adversarial PINNs training strategies

The discussion in Appendix G.1–G.3 is summarized here by the two-dimensional taxonomy in Tables 3 and 4. The first table emphasizes the role of the discriminator input, while the second highlights the role of the discriminator objective or structured function class. Here, a check mark indicates that a representative method already exists in the literature or has a clear correspondence, while △ highlights combinations that we believe are particularly promising.

Several observations follow immediately. First, the current literature only explores a limited subset of the full design space. Second, many methods that appear quite different can still be interpreted within a unified adversarial framework once the discriminator is viewed as living in a structured function space. Third, the missing entries are themselves informative: they indicate combinations that are meaningful from the present theory but have not yet been systematically explored.

In particular, we believe that the two highlighted cases, namely r^2+LSGAN and LSGAN with weak-space constraints, are especially promising and may potentially yield stronger performance.

G.5 Summary

In summary, adversarial PINNs training is best understood as a broad design space. The choice between r and r^2 determines whether the discriminator preserves sign information or focuses on magnitude, and whether the induced generator dynamics are additive or multiplicative. The choice of discriminator objective or function class determines whether the discriminator behaves like a classifier, a regression map, a witness function, or a constrained weighting operator. Existing methods

such as SA-PINN, LA-PINN, and WAN should therefore be viewed not as isolated constructions, but as representative points in a much larger structured family of adversarial PINNs training strategies.

H Experiments

H.1 Experimental setup details

This appendix provides the detailed experimental setup corresponding to the results reported in Section 5.

H.1.1 Compared methods

We compare the following adversarial PINNs methods:

- **DEQGAN**,
- **GAN**,
- **LSGAN**,
- **WGAN-GP**,
- **GAN-RB**,
- **LSGAN-RB**,
- **WGAN-GP-RB**.

Here the suffix "RB" denotes the proposed rollback strategy. For the rollback-enhanced variants, rollback selection is performed within at most 20 generator updates and 20 discriminator updates in each outer iteration.

Although the preceding theoretical discussion shows that a broader family of existing methods can also be interpreted as special cases of adversarial training under constrained discriminator classes, in the experiments we deliberately focus on the most basic adversarial objectives, namely GAN, LSGAN, and WGAN-GP. The reason is that these methods represent the canonical adversarial design choices and therefore provide the cleanest setting for evaluating the proposed framework. By starting from these standard baselines, we can more clearly separate the effect of the adversarial objective itself from the effect of the proposed update-control mechanism, without introducing additional architectural or formulation-specific factors.

DEQGAN is included for a different reason. In fact, our study was initially motivated by the distinctive phenomena observed in DEQGAN under different generator–discriminator update ratios. These observations led us to identify a gap between the empirical behavior of adversarial PINNs training and the classical divergence-based explanation, and further motivated the unified residual-dynamics framework developed in this paper. In this sense, DEQGAN serves not only as a strong existing baseline, but also as the starting point from which we first discovered the phenomenon, constructed the explanatory framework, and then generalized the resulting analysis to a broader class of adversarial PINNs methods.

The purpose of including both the original and rollback-enhanced variants is therefore twofold: first, to evaluate the adversarial objectives themselves under a unified setting; and second, to isolate the contribution of the proposed rollback mechanism from the choice of adversarial objective.

H.1.2 Benchmark equations

To evaluate the robustness of the proposed framework across different PDE types, we consider the following benchmark problems:

- **Poisson equation**,
- **Laplace equation**,
- **Viscous Burgers equation**,
- **Reaction–Diffusion equation**,
- **Klein–Gordon equation**.

These problems cover a diverse range of settings, including linear and nonlinear equations, stationary and time-dependent systems, as well as equations with different propagation, diffusion, and stiffness

Table 5: Hyperparameter settings for DEQGAN on PDE benchmarks.

Hyperparameter	**Poisson**	**Laplace**	**Viscous–Burgers**	**Reaction–Diffusion**	**Klein–Gordon**
Num. iterations	3000	3000	3300	4500	4000
Num. grid points	32×32	32×32	32×32	32×32	32×32
G units/layer	50	50	50	50	50
G num. layers	4	4	3	3	3
D units/layer	30	50	20	20	20
D num. layers	2	2	5	5	5
Activations	tanh	tanh	tanh	tanh	tanh
G learning rate	0.019	0.012	0.012	0.007	0.012
D learning rate	0.021	0.088	0.005	0.009	0.005
G β_1 (Adam)	0.139	0.295	0.185	0.185	0.185
G β_2 (Adam)	0.369	0.358	0.594	0.594	0.594
D β_1 (Adam)	0.745	0.575	0.093	0.093	0.093
D β_2 (Adam)	0.759	0.133	0.184	0.184	0.184
Exponential LR decay (γ)	0.957	0.953	0.954	0.954	0.954
Decay step size	9	10	25	10	10

characteristics. This diversity is intended to test whether the conclusions drawn from the proposed residual-dynamics analysis generalize beyond a single PDE family.

H.1.3 Network architectures

For fairness, the generator architecture is kept fixed across all methods and uses residual links. The discriminator architecture depends on the adversarial objective.

For **DEQGAN**, **GAN**, and **GAN-RB**, we use the standard discriminator configuration of GAN-style adversarial training:

- residual links,
- spectral normalization,
- a non-regressive output with a final sigmoid activation.

For **LSGAN**, **LSGAN-RB**, **WGAN-GP**, and **WGAN-GP-RB**, the discriminator is treated as a critic:

- no residual links,
- no spectral normalization,
- no final sigmoid activation.

All hidden layers use $\tanh$ activation.

H.1.4 Optimization settings

A central practical distinction in our study is that, except for DEQGAN, all methods are trained under the same simple default optimization setting:

$$\mathrm{lr}_G = \mathrm{lr}_D = 10^{-3}, \tag{311}$$

with Adam optimizer and default hyperparameters

$$(\beta_1, \beta_2) = (0.9, 0.999). \tag{312}$$

By contrast, **DEQGAN** follows its original tuning-dependent protocol (Table 5), which relies on:

- two-timescale generator/discriminator learning rates,
- tuned Adam hyperparameters,
- Ray Tune-based hyperparameter search.

This distinction is particularly important because one of the central goals of our work is to demonstrate that favorable alternating training dynamics can be effectively recovered through the proposed rollback principle, while avoiding the substantial hyperparameter tuning burden and implementation cost typically required by DEQGAN.

Table 6: Running time (**hours**) for different PDE benchmarks and compared methods. All experiments were conducted on a laptop workstation equipped with an NVIDIA GeForce RTX 4070 Laptop GPU and an Intel Core i7-14700HX CPU.

PDE	DEQGAN	GAN	LSGAN	WGAN-GP	GAN-RB	LSGAN-RB	WGAN-GP-RB
Poisson	0.24	0.33	0.28	0.29	3.17	3.02	2.98
Laplace	0.28	0.27	0.27	0.29	3.33	3.27	3.24
Burgers	0.46	0.41	0.42	0.37	4.62	4.48	4.47
Reaction	0.33	0.32	0.29	3.86	4.11	4.09	4.07
Klein	0.45	0.43	0.44	0.36	4.98	4.82	4.69

H.1.5 Training protocol

For each outer training iteration, the generator and discriminator are updated alternately. The rollback-enhanced variants additionally evaluate candidate intermediate states and retain only those updates that improve the corresponding first-order criterion or residual-energy criterion, as described in the main text.

Unless otherwise specified, all methods are trained under the same sampling and optimization pipeline on each benchmark problem. This design ensures that performance differences are primarily attributable to the adversarial objective and the rollback mechanism, rather than to unrelated implementation choices. All experiments are implemented in PyTorch and conducted on a laptop workstation equipped with an NVIDIA GeForce RTX 4070 Laptop GPU and an Intel Core i7-14700HX CPU. The detailed hardware configuration and running-time (hours) statistics are summarized in Table 6.

H.1.6 Sampling Strategy and Training–Validation Relationship

Different PDEs adopt slightly different sampling strategies according to the spatial size of their computational domains. For each problem, however, the sampling strategy is predetermined before training and is independent of the final validation error. To quantitatively characterize the spatial relationship between the training and validation points, we first normalize the coordinates of all PDEs to $[0,1]^2$. For each validation point $\mathbf{x}_i^{\mathrm{val}}$, we compute its two-dimensional Euclidean distance to the nearest training point:

$$d_i = \min_j \left\| \mathbf{x}_i^{\mathrm{val}} - \mathbf{x}_j^{\mathrm{tr}} \right\|_2 .$$

The mean nearest-neighbor distance is then defined as

$$\bar{d} = \frac{1}{N_{\mathrm{val}}} \sum_{i=1}^{N_{\mathrm{val}}} d_i .$$

To account for differences in validation-grid resolution, we further normalize this quantity by the diagonal spacing h_{diag} between neighboring validation-grid points:

$$\rho = \frac{\bar{d}}{h_{\mathrm{diag}}} \times 100\%.$$

Therefore, the second column of Table 7 reports the mean distance from each validation point to its nearest training point, while the third column reports this distance relative to one diagonal spacing of the validation grid.

For sampling strategies involving random perturbations, we perform ten independent random realizations and report the averaged distance statistics.

Overall, the training and validation sets are spatially close but are generally not identical. The mean nearest-neighbor distance is approximately 21%–32% of one diagonal validation-grid spacing, indicating that the training samples provide a dense spatial coverage of the computational domain. At the same time, most validation points do not exactly coincide with the training points, so the reported validation error is evaluated predominantly at distinct spatial locations within the same domain.

Table 7: Spatial relationship between training and validation samples after normalizing all PDE coordinates to $[0,1]^2$.

PDE	Mean nearest-neighbor distance	Relative grid spacing
Poisson	0.00971	21.3%
Laplace	0.01129	31.9%
Burgers	0.00483	21.9%
Reaction	0.01259	32.2%
Klein	0.01206	32.1%

H.1.7 Evaluation metrics

We report both **training error** (L2 error) and **validation error** (L2 error) as the primary accuracy metrics. To connect the experiments with the theory developed in the main text, we additionally track:

- the residual energy
$$E = \frac{1}{2}\|\mathbf{r}\|^2, \tag{313}$$
- the first-order indicator
$$S = \mathbf{r}^\top K_{rr}^G \boldsymbol{\gamma}, \tag{314}$$
- and, for rollback-based methods, the selected discriminator and generator rollback depths.

These quantities serve as useful diagnostics and provide a direct empirical bridge between adversarial PINN training and the NTK-based residual-dynamics analysis developed in this paper.

H.1.8 Relation to the main text

The main text reports the core quantitative comparisons and the principal mechanistic observations. Detailed training curves, per-equation dynamics, rollback-step statistics, and additional ablation results are deferred to the appendix in order to keep the main paper focused on the central theoretical and algorithmic message.

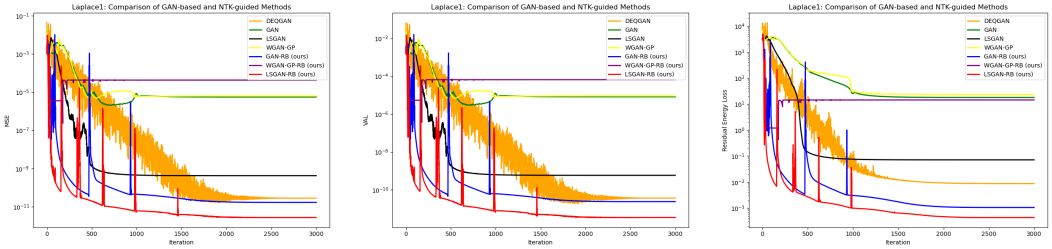
We note that the mechanistic analysis of DEQGAN under different $G : D$ ratios has already been presented systematically in Section 5.2 of the main text. There, based on the first-order indicator and the NTK-based failure-mode analysis, we explained the phenomena identified in Section 3, clarified the apparent mismatch between empirical observations and the classical theoretical picture, reconstructed the training-dynamics viewpoint, and supported the resulting interpretation with numerical evidence. Since these components already form a coherent narrative in the main paper, we do not repeat them again in the appendix.

H.2 Benchmark equations Comparison

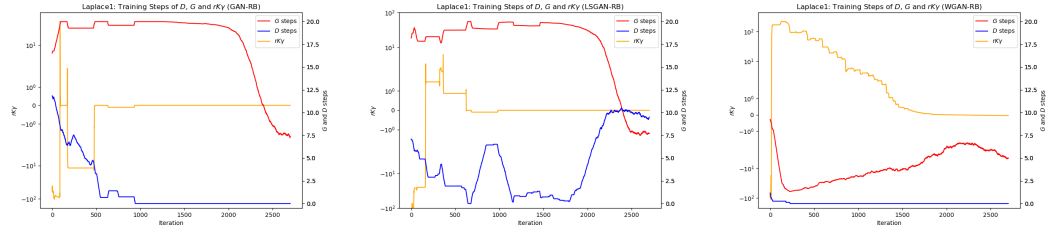
H.2.1 Laplace

Laplace equation: We first consider a two-dimensional Laplace equation posed on the unit square domain
$$\Omega = (0,1) \times (0,1),$$
with Dirichlet boundary conditions. The governing equation is
$$\begin{cases} \dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial y^2} = 0, & (x,y) \in \Omega, \\ u(0,y) = 0, & y \in [0,1], \\ u(1,y) = 0, & y \in [0,1], \\ u(x,0) = 0, & x \in [0,1], \\ u(x,1) = \dfrac{1}{2\cosh(\pi)} \sin(\pi x)\big(e^{\pi} - e^{-\pi}\big)\big), & x \in [0,1]. \end{cases}$$

The Laplace equation is one of the most fundamental elliptic partial differential equations and commonly arises in steady-state heat conduction, electrostatic potential analysis, incompressible fluid flow, and other equilibrium problems.



(a) Training MSE (b) Validation MSE (c) Residual energy



(d) GAN-RB (e) LSGAN-RB (f) WGAN-GP-RB

Figure 3: Laplace equation: The first row reports the training MSE, validation MSE, and residual energy for all compared methods. The second row shows the rollback-related dynamics for GAN-RB, LSGAN-RB, and WGAN-GP-RB, respectively, including the selected generator/discriminator inner-step behavior and the discriminator-induced quantity $r^\top K_{rr}\gamma$.

In the present example, the problem is defined on a bounded rectangular domain with mixed homogeneous and non-homogeneous Dirichlet boundary conditions, making it a standard benchmark for evaluating the approximation quality of physics-informed methods.

For this problem, an analytical solution is available:

$$u(x,y) = \frac{1}{2\cosh(\pi)} \sin(\pi x)\left(e^{\pi y} - e^{-\pi y}\right),$$

Hence, the numerical prediction can be directly compared with the exact solution in both solution error and PDE residual.

Comparison among different methods: Fig. 3(a)–(c) reports the training error, validation error, and residual energy dynamics of different adversarial PINNs variants on the Laplace equation. Overall, the rollback-based methods significantly improve the corresponding original adversarial methods. In particular, LSGAN-RB and GAN-RB achieve the best performance, with both the residual energy and prediction errors reduced by several orders of magnitude compared with their non-rollback counterparts. This demonstrates that the proposed rollback strategy can effectively stabilize and accelerate adversarial PINNs training by selecting more favorable generator–discriminator update states.

An interesting observation is that WGAN-GP does not perform well in this experiment, although it is commonly believed to alleviate the gradient-vanishing issue in standard GAN training. Even after applying the rollback strategy, WGAN-GP-RB still fails to achieve comparable accuracy. This suggests that avoiding discriminator saturation alone is not sufficient for adversarial PINNs training. From the perspective of our framework, the key factor is whether the discriminator-induced weighting γ provides an informative and well-aligned residual descent direction under the generator NTK. In this case, the WGAN-type gradient field appears to be less effective for the residual dynamics of the Laplace equation.

We further analyze the three rollback variants in Fig. 3(d)–(f), which show the evolution of the first-order indicator, the selected generator update steps, and the selected discriminator update steps during training. For better visualization, all three quantities are smoothed by a moving average over 300 iterations. In the following experiments, we adopt the same analysis strategy. The main quantity of interest is the number of selected generator update steps. According to the rollback criterion, the

generator is allowed to continue updating only when the current discriminator-induced direction leads to a decrease in the residual energy. Therefore, a larger selected generator update ratio indicates that the discriminator feedback remains useful over a longer generator trajectory and can continuously drive the residual energy downward.

Fig. 3(d) and (e) correspond to GAN-RB and LSGAN-RB, respectively. We observe that both methods maintain a high generator update ratio for most of the training process, indicating that their discriminator-induced gradient fields provide effective descent directions for the generator. This is consistent with their strong performance in Fig. 3(a)–(c). In contrast, Fig. 3(f) shows that WGAN-GP-RB keeps the generator update steps at a very low level throughout training. This means that the rollback criterion frequently rejects generator updates, implying that the WGAN-GP discriminator fails to provide a sufficiently useful residual descent direction in this problem.

The behavior of the first-order indicator further supports this interpretation. Under our plotting convention, a negative value corresponds to a favorable energy-decreasing direction. For GAN-RB and LSGAN-RB, the first-order indicator is largely suppressed in the negative region during training, which explains why their generator updates can be repeatedly accepted. By contrast, for WGAN-GP-RB, the indicator rapidly moves toward the positive region at the early stage and then becomes ineffective, indicating poor alignment between the discriminator-induced weighting and the generator NTK descent direction. This provides direct empirical evidence for our theoretical claim that the success of adversarial PINNs training depends not only on the adversarial loss itself, but more fundamentally on the quality of the induced residual-space weighting and its alignment with the generator residual dynamics.

H.2.2 Poisson

Poisson equation We next consider a two-dimensional Poisson equation posed on the unit square domain

$$\Omega = (0,1) \times (0,1),$$

with homogeneous Dirichlet boundary conditions. The governing equation is given by

$$\begin{cases} \dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial y^2} = 2x(y-1)\big(y - 2x + xy + 2\big)e^{x-y}, & (x,y) \in \Omega, \\ u(x,y) = 0, & (x,y) \in \partial\Omega. \end{cases}$$

The Poisson equation is one of the most classical elliptic partial differential equations and arises widely in electrostatics, steady-state heat conduction, gravitational potential modeling, and incompressible flow analysis. In the present example, the problem is defined on a bounded rectangular domain with a prescribed source term and zero boundary values, which makes it a standard benchmark for testing the approximation accuracy and training stability of PINNs-type methods.

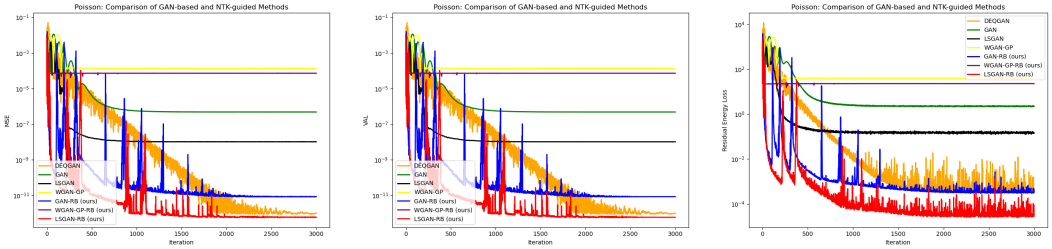
For this problem, an analytical solution is available:

$$u(x,y) = x(1-x)y(1-y)e^{x-y}.$$

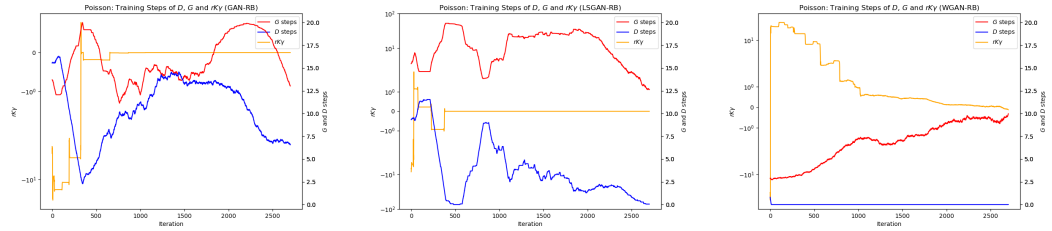
Therefore, both the solution error and the PDE residual can be evaluated directly, making this example particularly suitable for quantitative Comparison among different methods: .

Comparison among different methods: For the Poisson equation, Fig. 4(a)–(c) shows that LSGAN-RB achieves the best convergence behavior among all compared methods. It attains lower training error, validation error, and residual energy than the original DEQGAN baseline, indicating that the proposed rollback strategy can further improve adversarial PINNs training beyond the existing GAN-based formulation. Here, WGAN-GP still performs poorly, and the application of rollback does not lead to a clear improvement. This again suggests that the non-saturating property of WGAN-type objectives alone is insufficient to guarantee effective residual descent.

Fig. 4(d)–(f) further explain this behavior from the perspective of the rollback dynamics. GAN-RB and LSGAN-RB maintain relatively high selected generator update steps during most of the training process, which means that the discriminator-induced directions can support a longer sequence of energy-decreasing generator updates. In contrast, WGAN-GP-RB keeps the generator update steps at a much lower level, implying that many candidate generator updates are rejected by the rollback criterion.



(a) Training MSE (b) Validation MSE (c) Residual energy



(d) GAN-RB (e) LSGAN-RB (f) WGAN-GP-RB

Figure 4: Poisson equation: The first row reports the training MSE, validation MSE, and residual energy for all compared methods. The second row shows the rollback-related dynamics for GAN-RB, LSGAN-RB, and WGAN-GP-RB, respectively, including the selected generator/discriminator inner-step behavior and the discriminator-induced quantity $r^\top K_{rr}\gamma$.

The first-order indicator shows a consistent trend. For GAN-RB and LSGAN-RB, the indicator remains at a lower level during training, suggesting better alignment between the discriminator-induced residual weighting and the generator NTK dynamics. By contrast, the WGAN-GP-RB indicator becomes large already in the early stage, indicating an unfavorable or poorly aligned descent direction. Since the qualitative interpretation is consistent with the Laplace experiment, we omit repeated details here. Overall, the Poisson results further confirm that the effectiveness of adversarial PINNs training depends on the quality of the induced residual-space weighting rather than merely on the nominal GAN objective.

H.2.3 Reaction-Difussion

Reaction-Difussion equation We consider a one-dimensional time-dependent parabolic equation on the space–time domain

$$\Omega = (0,1) \times (0,5),$$

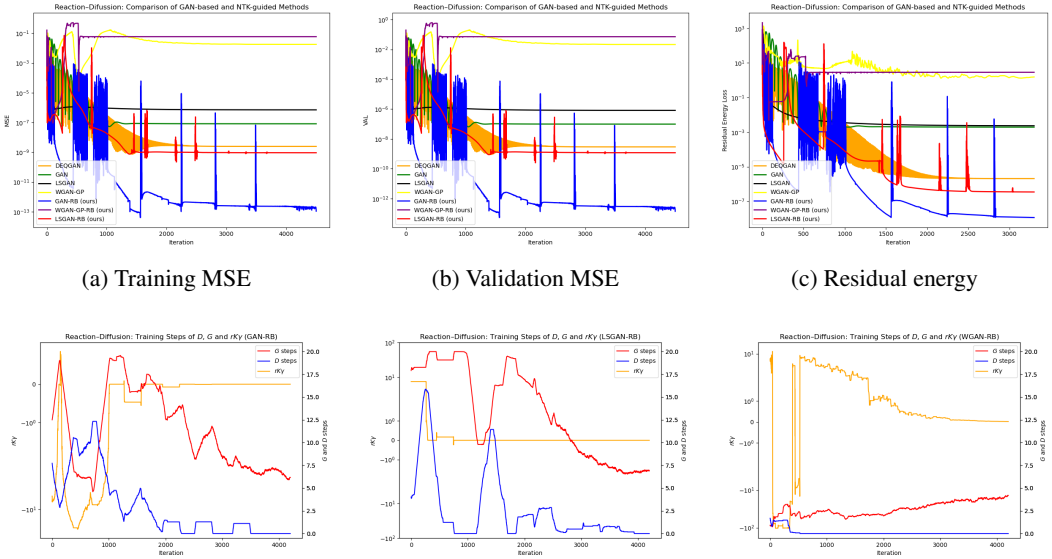
subject to homogeneous Dirichlet boundary conditions and a zero initial condition. The governing equation is

$$\begin{cases} u_t - u_{xx} - 10u = \dfrac{2}{5}t - \left(2t - \dfrac{1}{5}\right) x(1-x), & (x,t) \in \Omega, \\ u(0,t) = 0, & t \in [0,5], \\ u(1,t) = 0, & t \in [0,5], \\ u(x,0) = 0, & x \in [0,1]. \end{cases}$$

This type of parabolic equation is commonly used to describe diffusion and heat-transfer processes, and more generally appears in time-dependent transport-reaction problems. For the present example, an analytical solution is available:

$$u(x,t) = \frac{t}{5} x(1-x).$$

Hence, the predicted solution can be directly compared with the exact solution, which makes this problem convenient for evaluating both approximation accuracy and training behavior.



(a) Training MSE (b) Validation MSE (c) Residual energy

(d) GAN-RB (e) LSGAN-RB (f) WGAN-GP-RB

Figure 5: Reaction-Difussion equation: The first row reports the training MSE, validation MSE, and residual energy for all compared methods. The second row shows the rollback-related dynamics for GAN-RB, LSGAN-RB, and WGAN-GP-RB, respectively, including the selected generator/discriminator inner-step behavior and the discriminator-induced quantity $r^\top K_{rr}\gamma$.

Comparison among different methods: For the reaction–diffusion equation, Fig. 5(a)–(c) shows that GAN-RB achieves the best accuracy among all compared methods. Compared with the other baselines, GAN-RB improves the training accuracy by approximately four orders of magnitude, demonstrating that the rollback strategy can be particularly effective for nonlinear or reaction-dominated residual dynamics.

Fig. 5(d)–(f) further illustrate the underlying update behavior. Both GAN-RB and LSGAN-RB maintain relatively high selected generator update steps during training, indicating that their discriminator-induced directions can still support repeated energy-decreasing generator updates. However, compared with GAN-RB, the first-order indicator of LSGAN-RB remains relatively large at the early stage. Since this indicator is an aggregated quantity, this suggests that the discriminator feedback of LSGAN-RB is less well aligned with the global residual-energy descent direction, although it can still produce decreasing residual energy. This explains why LSGAN-RB converges, but not as rapidly or accurately as GAN-RB.

For WGAN-GP-RB, the rollback strategy brings a slight improvement over the original WGAN-GP. As shown in Fig. 5(f), the WGAN-GP discriminator provides a useful descent direction at the very beginning of training. Nevertheless, this favorable behavior quickly disappears, and the selected generator updates become less effective afterward. As a result, the overall training performance remains unsatisfactory. These results again indicate that the success of the rollback mechanism depends on whether the discriminator-induced weighting can provide a persistently informative residual descent direction.

H.2.4 Viscous-Burgers

Viscous Burgers equation We consider a one-dimensional viscous Burgers equation on the space–time domain

$$\Omega = (-5, 5) \times (0, 2.5),$$

with viscosity coefficient $\nu = 0.001$. The governing equation is

$$\begin{cases} u_t + u\, u_x - \nu u_{xx} = 0, & (x,t) \in \Omega, \\ u(x,0) = \dfrac{1}{\cosh(x)}, & x \in [-5,5]. \end{cases}$$

In the implementation, the boundary values at $x = \pm 5$ are fixed to $1/\cosh(5)$, which is numerically very close to zero.

The viscous Burgers equation is a classical nonlinear convection–diffusion model and is widely used as a benchmark for studying nonlinear transport, diffusion, and sharp-gradient dynamics. Owing to the competition between nonlinear advection and viscosity, it provides a representative test case for assessing the stability and accuracy of physics-informed methods.

For this problem, the reference solution is not available in closed form. Instead, a high-accuracy numerical solution computed by an FFT-based solver is used for comparison.

Comparison among different methods: For the Burgers equation, Fig. 6(a)–(c) shows that all methods exhibit strongly oscillatory training dynamics. This behavior is reasonable because the Burgers equation contains nonlinear convection, and its residual landscape is more sensitive to local sharp structures and dynamically evolving error modes. As a result, the adversarial feedback may change rapidly during training, leading to more pronounced fluctuations than in the linear elliptic problems.

Despite these oscillations, GAN-RB achieves the best overall performance. This indicates that, for nonlinear convection-dominated residual dynamics, the sharper discriminator-induced weighting of GAN, together with the rollback selection mechanism, can provide more effective residual descent directions than the smoother LSGAN-type weighting.

Fig. 6(d)–(f) further support this interpretation. Among the three rollback variants, GAN-RB maintains the largest selected generator update steps, followed by LSGAN-RB, while WGAN-GP-RB keeps the lowest generator update level. This ordering is consistent with their final accuracy. The first-order indicator of GAN-RB is also the most favorable and remains better suppressed during training, suggesting that its discriminator-induced weighting is better aligned with the generator residual dynamics.

For LSGAN-RB, the first-order indicator is negative at the beginning, indicating an initially useful descent direction. However, it gradually moves toward the positive region and remains there, which weakens the effectiveness of subsequent generator updates and leads to an inferior final result compared with GAN-RB. WGAN-GP-RB performs the worst: its first-order indicator is poorly suppressed and the selected generator update steps remain low, implying that the WGAN-GP-RB discriminator fails to provide a persistently useful residual descent direction for this nonlinear problem.

H.2.5 Klein-Gordon

Klein-Gordon equation We further consider a one-dimensional Klein–Gordon-type equation on the space–time domain

$$\Omega = (0,1) \times (0,2),$$

with homogeneous Dirichlet boundary conditions. The governing equation is

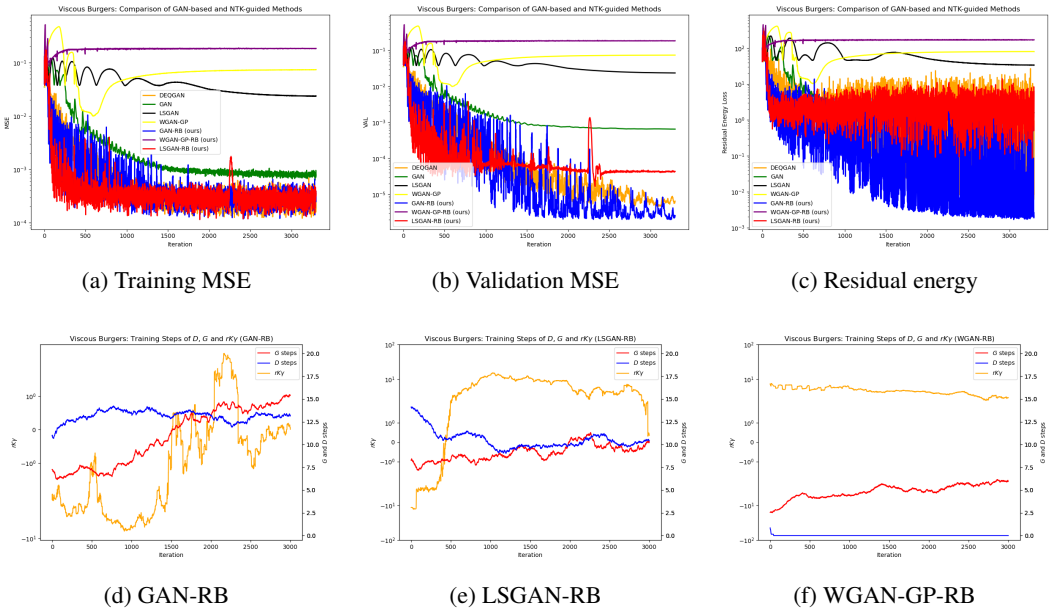
$$\begin{cases} u_{tt} - c^2 u_{xx} - 4u = 0.8 \sin(\pi x)\cos(\pi t), & (x,t) \in \Omega, \\ u(0,t) = 0, \qquad t \in [0,2], \\ u(1,t) = 0, \qquad t \in [0,2], \\ u(x,0) = -0.2\sin(\pi x), \qquad x \in [0,1], \\ u_t(x,0) = 0, \qquad x \in [0,1]. \end{cases}$$

The Klein–Gordon equation is a classical hyperbolic partial differential equation and is widely used to describe wave propagation with restoring or reaction effects. It also serves as a representative benchmark for testing whether learning-based solvers can accurately capture second-order temporal dynamics together with spatial oscillations.

For the default choice $c = 1$, the problem admits the analytical solution

$$u(x,t) = -0.2\sin(\pi x)\cos(\pi t).$$

Hence, the predicted solution can be directly compared with the exact solution in both solution error and PDE residual.



(a) Training MSE (b) Validation MSE (c) Residual energy

(d) GAN-RB (e) LSGAN-RB (f) WGAN-GP-RB

Figure 6: Viscous Burgers equation: The first row reports the training MSE, validation MSE, and residual energy for all compared methods. The second row shows the rollback-related dynamics for GAN-RB, LSGAN-RB, and WGAN-GP-RB, respectively, including the selected generator/discriminator inner-step behavior and the discriminator-induced quantity $r^\top K_{rr}\gamma$.

Comparison among different methods: For the Klein–Gordon equation, Fig. 7(a)–(c) shows that GAN-RB achieves the best accuracy among all methods. Its final error is approximately four orders of magnitude lower than the other baselines, demonstrating that GAN-RB is particularly effective for this oscillatory and wave-like problem.

From Fig. 7(d)–(f), we observe that both GAN-RB and LSGAN-RB maintain relatively high selected generator update steps during training. However, GAN-RB obtains a much larger improvement after applying the rollback strategy. This indicates that, in the Klein–Gordon problem, the GAN-induced residual weighting is better aligned with the generator residual dynamics than the smoother LSGAN-type weighting.

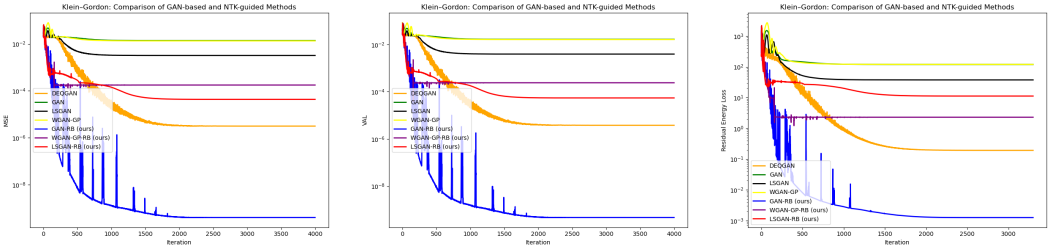
The first-order indicator provides further evidence. For GAN-RB, the indicator is globally well suppressed in the favorable region throughout training, which is a strong indication that the discriminator feedback persistently provides useful descent directions. In contrast, LSGAN-RB also produces a favorable indicator at the beginning, but it gradually increases and eventually stays near a small positive value around zero. This means that the discriminator-induced direction becomes less effective in later training stages, explaining why LSGAN-RB is inferior to GAN-RB.

For WGAN-GP-RB, the rollback strategy also brings a clear improvement, reducing the error by about two orders of magnitude compared with the original WGAN-GP. Different from the previous examples, Fig. 7(f) shows that the selected generator update steps of WGAN-GP-RB are significantly higher in this experiment. This suggests that the WGAN-GP discriminator can provide more useful descent directions for the Klein–Gordon residual dynamics than in the Laplace, Poisson, or reaction–diffusion cases. Nevertheless, its first-order indicator is still less stable and less favorable than that of GAN-RB, so its final performance remains worse.

H.3 Ablation experiment

H.3.1 Laplace equation (Under different boundary conditions)

Here, we conduct the first ablation experiment to examine the robustness of the proposed method under changes in boundary conditions. The motivation for this experiment is that, even when the governing PDE remains unchanged, the solution can be highly sensitive to the prescribed boundary



(a) Training MSE (b) Validation MSE (c) Residual energy



(d) GAN-RB (e) LSGAN-RB (f) WGAN-GP-RB

Figure 7: Klein-Gordon equation: The first row reports the training MSE, validation MSE, and residual energy for all compared methods. The second row shows the rollback-related dynamics for GAN-RB, LSGAN-RB, and WGAN-GP-RB, respectively, including the selected generator/discriminator inner-step behavior and the discriminator-induced quantity $r^\top K_{rr}\gamma$.

conditions. Therefore, it is important to verify whether the proposed algorithm can still maintain stable and accurate performance when the boundary conditions are modified.

Specifically, we keep the differential operator and the computational domain fixed, and only change the boundary condition setting. This allows us to isolate the effect of boundary-induced changes in the solution structure, while avoiding interference from changes in the PDE type itself. If the proposed method still achieves strong performance under this setting, it provides evidence that the improvement is not limited to one specific boundary configuration, but reflects a more robust residual optimization mechanism.

Laplace with different boundary condition: We also consider a two-dimensional Laplace equation on the unit square domain

$$\Omega = (0,1) \times (0,1),$$

subject to Dirichlet boundary conditions:

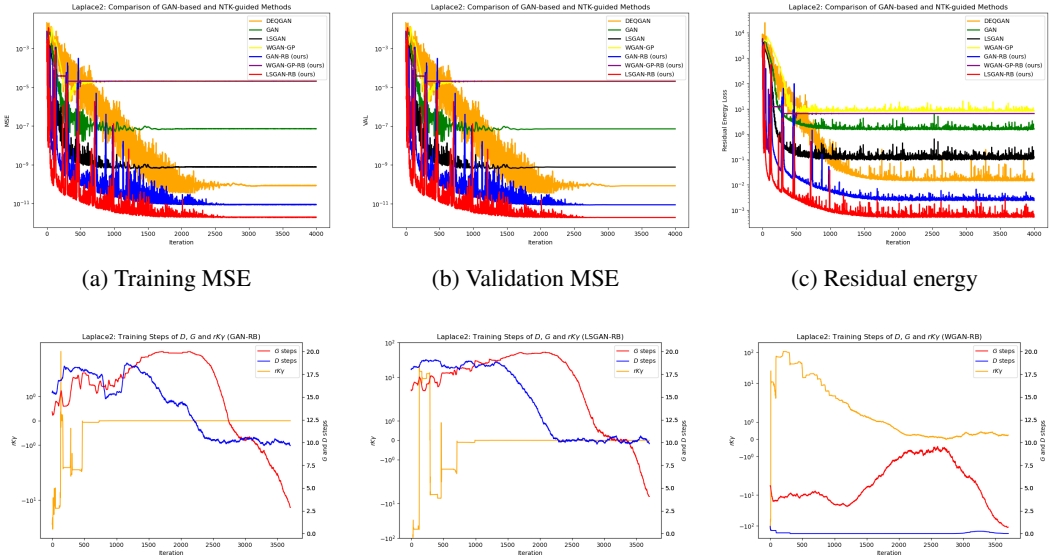
$$\begin{cases} u_{xx} + u_{yy} = 0, & (x,y) \in \Omega, \\ u(0,y) = 0, & y \in [0,1], \\ u(1,y) = 0, & y \in [0,1], \\ u(x,0) = \sin(\pi x), & x \in [0,1], \\ u(x,1) = \dfrac{1}{\cosh(\pi)} \sin(\pi x) + \dfrac{1}{2} \tanh(\pi) \cos\left(\pi x + \dfrac{\pi}{2}\right), & x \in [0,1]. \end{cases}$$

Compared with simpler homogeneous-boundary examples, this test case involves nontrivial non-homogeneous boundary data and is therefore useful for evaluating the capability of a method to handle more complex boundary structures.

For this problem, an analytical solution is available:

$$u(x,y) = \frac{1}{2\cosh(\pi)} \sin(\pi x)\left(e^{\pi(y-1)} + e^{\pi(1-y)}\right) + \frac{1}{4\cosh(\pi)} \cos\left(\pi x + \frac{\pi}{2}\right)\left(e^{\pi y} - e^{-\pi y}\right).$$

Hence, the numerical prediction can be directly compared with the exact solution.



(a) Training MSE (b) Validation MSE (c) Residual energy

(d) GAN-RB (e) LSGAN-RB (f) WGAN-GP-RB

Figure 8: Ablation study on the same pde with different boundary condition: The first row reports the training MSE, validation MSE, and residual energy for all compared methods. The second row shows the rollback-related dynamics for GAN-RB, LSGAN-RB, and WGAN-GP-RB, respectively, including the selected generator/discriminator inner-step behavior and the discriminator-induced quantity $r^\top K_{rr}\gamma$.

Comparison among different methods: For the second Laplace setting, Fig. 8 evaluates the same Laplace equation under different boundary conditions. As shown in Fig. 8(a)–(c), both GAN-RB and LSGAN-RB achieve strong convergence performance. Compared with the other methods, they improve the final accuracy by about two orders of magnitude. This result indicates that the proposed rollback strategy remains effective even when the boundary conditions are changed, suggesting good robustness with respect to boundary-induced variations in the solution structure.

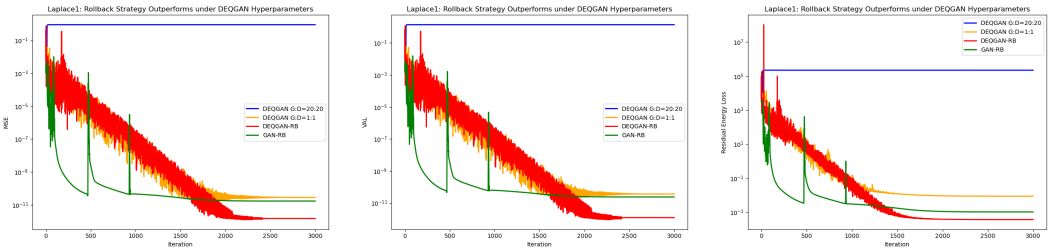
Fig. 8(d) and (e) further show that GAN-RB and LSGAN-RB maintain a large number of selected generator update steps during training. At the same time, the discriminator updates help keep the first-order indicator mostly below zero, meaning that the discriminator-induced directions remain favorable for residual-energy reduction. This explains why both methods can achieve stable and accurate convergence in this modified boundary setting.

In contrast, WGAN-GP-RB still performs poorly. As shown in Fig. 8(f), its first-order indicator quickly moves to a large positive value at the early training stage, indicating poor alignment between the WGAN-GP-induced weighting and the generator residual dynamics. Consequently, even with rollback, WGAN-GP-RB cannot provide sufficiently effective descent directions for this problem.

H.3.2 Compatibility with DEQGAN

To further evaluate the compatibility of the proposed rollback strategy, we combine it with DEQGAN and test the resulting method on the Laplace equation introduced in Section H.2.1. The purpose of this experiment is not merely to seek another performance gain, but to clarify an important conceptual point: rollback is not a standalone algorithm disconnected from existing adversarial PINNs methods. Rather, it should be understood as a dynamical modification of the alternating optimization process motivated by the NTK-based residual analysis developed in this paper.

From this viewpoint, rollback does not replace the underlying adversarial objective or discriminator design. Instead, it operates at the level of update selection, using the first-order residual-dynamics criterion to filter candidate alternating steps and retain those more consistent with descent-oriented residual evolution. Therefore, if our interpretation is correct, the rollback principle should be compatible not only with GAN and LSGAN, but also with existing adversarial PINNs frameworks such as DEQGAN.



(a) Training MSE (b) Validation MSE (c) Residual energy



(d) DEQGAN-RB dynamics (e) GAN-RB dynamics (f) Comparison of $r^\top K_{rr}\gamma$

Figure 9: Ablation study on the DEQGAN and RB. The top row reports the training MSE, validation MSE, and residual energy of all compared methods. The bottom row illustrates the rollback dynamics of adversarial variants, including the selected generator/discriminator inner-step evolution and the discriminator-induced quantity $r^\top K_{rr}\gamma$.

To verify this, we apply rollback on top of DEQGAN under the Laplace benchmark. This setting is particularly informative because DEQGAN already relies on carefully tuned hyperparameters and a specific two-timescale training protocol. If rollback can still improve performance in this case, then its benefit cannot be attributed merely to replacing weak baselines or to using simpler optimization settings. Instead, it would support the stronger claim that rollback captures a more general principle of alternating adversarial PINNs training.

The results confirm this expectation: even when combined with DEQGAN, the rollback-enhanced variant achieves improved convergence behavior and better final accuracy on the Laplace problem. This indicates that the proposed rollback mechanism is compatible with existing adversarial PINNs methods and can serve as an additional optimization layer on top of them. In this sense, the contribution of rollback is not to define a completely separate training paradigm, but to provide a principled update-control mechanism that can be integrated into broader adversarial PINNs frameworks.

Comparison among different methods: In Fig. 9, we further combine the rollback strategy with the existing DEQGAN method to demonstrate the flexibility and broad applicability of the proposed strategy. As shown in Fig. 9(a)–(c), the resulting DEQGAN-RB method achieves the best accuracy among all compared methods. This indicates that rollback is not restricted to the standard GAN, LSGAN, or WGAN-GP formulations, but can also be incorporated into existing adversarial PINNs frameworks to further improve their stability and accuracy.

An interesting observation is that the original DEQGAN performs poorly when the generator–discriminator update ratio is fixed to $20:20$. This is reasonable because the original DEQGAN hyperparameters were tuned for a different update setting, such as a $1:1$ update ratio. In contrast, after incorporating rollback, DEQGAN-RB can adaptively select more favorable update states and therefore achieves much better performance under the same update budget. We also include GAN-RB as a comparison. The results show that, when rollback is applied on top of a properly tuned adversarial training method, it can produce more stable training dynamics and higher final accuracy.

This observation suggests that rollback can serve as a general stabilization module for adversarial PINNs training. Across the previous five numerical experiments, adversarial training methods often exhibit strong fluctuations, especially on nonlinear or oscillatory PDEs. Therefore, combining rollback with existing adversarial PINNs methods is a meaningful direction for improving both robustness and training stability.

Fig. 9(d) and (e) compare the rollback dynamics of DEQGAN-RB and GAN-RB. We observe that DEQGAN-RB selects fewer generator update steps than GAN-RB. This does not contradict the previous analysis, because the two methods use different learning rates. DEQGAN-RB adopts a larger learning rate, so each accepted generator update can produce a larger decrease in the residual energy. As a result, fewer accepted generator steps are sufficient to achieve strong convergence. In contrast, GAN-RB uses more generator updates, but its first-order indicator shows much stronger fluctuations. After hyperparameter tuning and integration with the existing DEQGAN framework, the rollback indicator becomes more stable, which explains the superior performance of DEQGAN-RB in this experiment.

In Fig. 9(f), we compare the first-order indicator across four methods. It can be observed that the DEQGAN method achieves its best performance under a 1:1 generator–discriminator update ratio. This is because, when the number of alternating updates per outer iteration is small, the generator and discriminator—after careful hyperparameter tuning—are less likely to become excessively strong or weak relative to each other. However, since the total number of updates is strictly limited under the 1:1 setting, the overall optimization process remains insufficiently explored, and the final performance is therefore suboptimal. When the update budget is increased, e.g., to a 20:20 ratio, a different phenomenon emerges. We observe that the first-order indicator rapidly approaches zero within a very small number of iterations. This suggests that the discriminator becomes effectively over-trained relative to the generator: the discriminator gradients quickly vanish, leading to $\boldsymbol{\gamma} \approx \mathbf{0}$ and thus a near-degenerate training signal for the generator. In other words, simply increasing the update budget of DEQGAN is not sufficient to guarantee improved optimization performance. For the GAN-RB method, although no extensive hyperparameter tuning is performed, it achieves better results than the 1:1 baseline. However, its first-order indicator is noticeably unstable. As shown in Fig. F, there exist several abrupt positive spikes (i.e., indicator values exceeding zero), which correspond to regions where the loss exhibits significant oscillations. This instability reflects inconsistent descent behavior in the residual energy. Finally, we emphasize that the proposed DEQGAN-RB method achieves the most stable first-order indicator under the 20:20 update ratio. This stability directly translates into more consistent energy decay and smoother training dynamics, ultimately leading to higher solution accuracy. These observations suggest that combining existing adaptive alternating training strategies with the proposed first-order rollback mechanism could yield new methods with both improved stability and accuracy, which provides an important direction for future research.

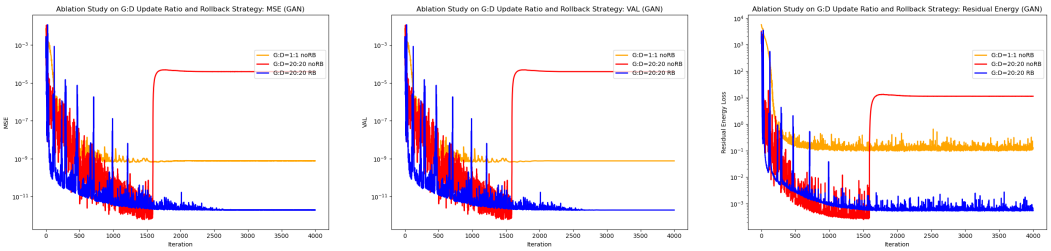
H.3.3 Does rollback help beyond increasing the update budget?

A natural concern about the proposed rollback strategy is that it evaluates multiple candidate updates within each outer iteration, and therefore may appear to benefit simply from using more update steps. To rule out this possibility, we perform a controlled ablation and examine whether the observed performance gain truly comes from rollback itself.

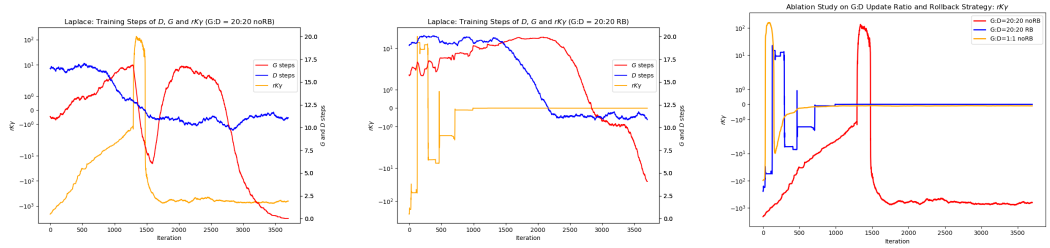
We focus on the two rollback-enhanced methods that perform best overall in our experiments, namely **GAN-RB** and **LSGAN-RB**. For **GAN-RB**, we use the Klein–Gordon equation benchmark in Section H.2.5, where GAN-RB performs best. For **LSGAN-RB**, we use the Laplace equation in Section H.3.1, where this method achieves its strongest performance.

For each method, we compare three training settings: **(i) no rollback with** $G : D = 1:1$**, (ii) no rollback with** $G : D = 20:20$**, and (iii) rollback with** $G : D = 20:20$**.** This design is chosen to answer two separate questions. The comparison between settings (i) and (ii) tests whether simply increasing the number of generator and discriminator updates can improve the solution accuracy. The comparison between settings (ii) and (iii) then isolates the effect of rollback under the same $20:20$ update budget, and asks whether adding the proposed optimization strategy on top of that fixed budget leads to further improvement. Therefore, if the third setting consistently outperforms the second, the gain cannot be attributed merely to more candidate updates, but must come from the rollback mechanism itself.

Laplace. From Fig. 10, several clear observations can be made. First, Fig. 10 (a)–(c) show that among the three compared settings, the fixed $G : D = 20$:20 scheme without rollback performs the worst, while the rollback-enhanced 20:20 scheme achieves the best overall convergence. A particularly interesting phenomenon is that the non-rollback 20:20 method actually exhibits very good optimization behavior before roughly 1700 iterations, but then suddenly collapses, with both the training and validation errors increasing sharply.



(a) Training MSE (b) Validation MSE (c) Residual energy



(d) No rollback, $G : D = 20{:}20$ (e) Rollback, $G : D = 20{:}20$ (f) Comparison of $r^\top K_{rr}\gamma$

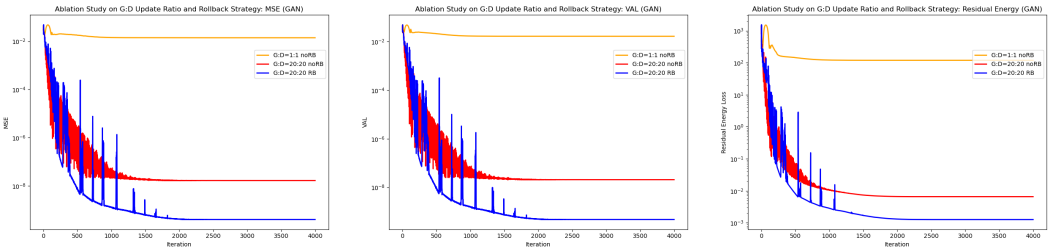
Figure 10: Controlled ablation for LSGAN on the Laplace benchmark. The first row reports the training MSE, validation MSE, and residual energy. The second row compares the convergence behavior without rollback and with rollback under the same update budget $G : D = 20{:}20$, and further shows the corresponding evolution of the first-order quantity $r^\top K_{rr}\gamma$ across the compared settings.

This abrupt failure can be better understood from Fig. 10 (d), where we plot, for the non-rollback 20:20 method, the best discriminator/generator inner-step statistics together with the first-order indicator $r^\top K_{rr}\gamma$. At the beginning of training, this indicator remains stably negative, suggesting that the discriminator-induced signal is well aligned with residual descent. However, around iteration 1500, the indicator undergoes a large jump, and at almost the same time the best generator update depth drops sharply to around 6 steps. Since the generator update budget is fixed to 20 steps, this means that after the useful descent-oriented regime has already been exhausted, the generator continues to update for many additional steps under a deteriorated discriminator signal. As a result, the accumulated error is rapidly amplified, causing the sudden jump in loss and the eventual training failure.

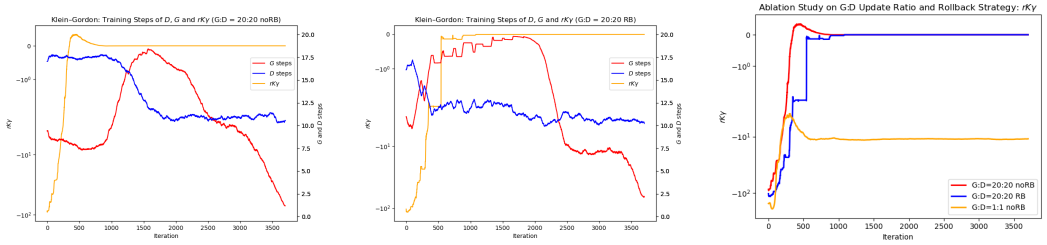
By contrast, Fig. 10 (e) shows that after introducing rollback, the first-order indicator is maintained within a slightly negative but much more stable range, and the effective generator update depth remains consistently high and stable throughout training. At the same time, the discriminator also keeps a relatively high effective update ratio during the early stage, indicating that it is still able to push the first-order indicator toward a favorable regime. In this sense, rollback acts on both sides of the alternating process: it prevents the generator from taking too many harmful steps once the discriminator signal becomes unreliable, and it also prevents the discriminator from continuing unnecessary updates after a sufficiently good state has already been reached.

This observation also helps explain why the non-rollback 20:20 scheme can perform well at the beginning of training. In the early stage, allowing the discriminator to take many updates is indeed beneficial, because it helps drive the first-order indicator into a good negative region and thus yields rapid error reduction. However, as training proceeds, keeping the discriminator fixed at 20 updates becomes excessive. At that point, the discriminator no longer needs such a strong update budget, and continuing to optimize it too aggressively starts to destroy the previously favorable dynamical balance. The rollback mechanism effectively corrects this issue by adapting the usable update depth to the current training state, thereby maintaining stability while preserving accuracy.

Finally, Fig. 10 (f) compares the evolution of $r^\top K_{rr}\gamma$ across the three settings. One can see that the overall indicator of the 1:1 baseline is comparable to that of 20:20 with rollback. However, the important point is that once the generator and discriminator update budgets are both increased, maintaining a favorable first-order indicator becomes much more difficult. This is precisely why the plain 20:20 scheme fails: without rollback, the enlarged update budget makes the dynamics far more



(a) Training MSE (b) Validation MSE (c) Residual energy



(d) No rollback, $G : D = 20{:}20$ (e) Rollback, $G : D = 20{:}20$ (f) Comparison of $r^\top K_{rr}\gamma$

Figure 11: Controlled ablation for GAN on the Klein–Gordon benchmark. The first row reports the training MSE, validation MSE, and residual energy. The second row compares the convergence behavior without rollback and with rollback under the same update budget $G : D = 20{:}20$, and further shows the corresponding evolution of the first-order quantity $r^\top K_{rr}\gamma$ across the compared settings.

vulnerable to abrupt misalignment, which then causes the catastrophic jump observed in training. In this sense, the comparison shows that the benefit of rollback is not merely to improve performance, but to make larger alternating-update budgets practically usable.

Klein–Gordon. From Fig. 11, a different but equally informative pattern can be observed on the Klein–Gordon benchmark. Fig. 11 (a)–(c) show that the $G : D = 1{:}1$ baseline gives the worst final accuracy, the fixed $20{:}20$ scheme without rollback performs better, and the rollback-enhanced $20{:}20$ scheme achieves the best overall result. In this example, increasing the generator and discriminator update budgets does indeed improve the solution accuracy, which is different from the Laplace case above.

At the same time, Fig. 11 (d) shows that for the non-rollback $20{:}20$ scheme, the first-order indicator $r^\top K_{rr}\gamma$ is kept in a relatively stable negative region throughout most of training. This indicates that the alternating dynamics are more stable here than in the previous example. However, the same Fig. 11 also shows that the best generator update depth is only around 10 steps in the early stage. Therefore, when the generator budget is fixed to 20 updates, a substantial fraction of these updates are already beyond the most effective descent-oriented range. In other words, even though the overall indicator remains negative, the generator is still being updated more than necessary.

Fig. 11 (e) shows the effect of adding rollback under the same $20{:}20$ budget. In this case, the generator consistently operates within a higher and more stable effective update range, while the first-order indicator is further improved. This suggests that rollback does not merely preserve stability, but also makes the enlarged update budget more efficient by filtering out less useful updates and retaining those that are better aligned with the current descent direction.

Finally, Fig. 11 (f) compares the evolution of $r^\top K_{rr}\gamma$ across the three settings. One can see that although the $1{:}1$ baseline has the most favorable first-order indicator overall, its small update budget leads to inferior final accuracy. Once the generator and discriminator update numbers are increased, the first-order indicator deteriorates immediately in the plain $20{:}20$ setting. After rollback is introduced, however, the indicator is clearly improved again. This explains why the rollback-enhanced method achieves the best performance: it not only allows the larger alternating-update budget to be used, but also restores a better first-order descent quality under that larger budget.

Conclusion: This ablation supports the interpretation that rollback improves adversarial PINNs training by selecting more descent-consistent alternating updates, rather than by merely enlarging the optimization budget. In other words, the performance improvement is attributable to better update control, not to more iterations.

I Contributions, Limitations, and Future Work

Contributions: Although the present work still has several limitations, we believe that its main contributions are already clear. First, from the perspective of residual dynamics and NTK analysis, we develop a unified framework for adversarially trained PINNs. In contrast to the classical static picture based on the optimal discriminator and divergence minimization, we emphasize that practical training is governed by finite-step alternating dynamics, and we show how different adversarial objectives induce different residual-weighting mechanisms through the discriminator input derivative. Second, based on this framework, we propose a first-order rollback strategy and demonstrate across multiple PDE benchmarks that it can substantially improve GAN-based and LSGAN-based adversarial PINNs without requiring expensive hyperparameter search. Third, we show that this viewpoint is not limited to standard GAN variants: it also provides a unified interpretation of squared-residual input, soft-constrained settings, and existing methods such as SA-PINN, LA-PINN, and WAN, thereby suggesting a broader design space for adversarial PINNs training.

Limitations: Against this background, several limitations remain.

First, the rollback strategy, while effective, can still introduce noticeable oscillations during training. In some sense, this is natural, since rollback explicitly compares and reverts among candidate alternating updates, so the resulting trajectory is not expected to be completely smooth. Nevertheless, such oscillations indicate that the current rollback mechanism still has room for improvement in terms of stability. At the same time, our ablation results (H.3.2) show that when rollback is combined with an already well-tuned model, it can achieve both higher accuracy and more stable dynamics. This suggests that rollback should not be viewed only as a substitute for hyperparameter tuning; it may also serve as an additional control layer on top of a good optimization configuration, further improving training quality.

Second, our interpretation of discriminator behavior across different GAN variants is still built on the NTK regime. In particular, the present theory describes how, in the infinite-width and linearized-training limit, the discriminator influences generator residual descent through its input derivative. Compared with the finite-width and often shallow networks used in practice, this analysis remains an approximation. However, the value of the framework is precisely that it does not aim to reproduce every finite-network phenomenon exactly; rather, it provides a mechanistic picture that explains success and failure modes and, more importantly, guides algorithm design. The numerical results suggest that this infinite-width viewpoint remains informative for finite-width training.

Third, the current theory does not explicitly cover common training techniques such as spectral normalization or gradient penalties. This limitation is not specific to our analysis alone, but reflects a broader boundary of the standard NTK framework itself. In particular, classical NTK theory is typically developed under assumptions such as infinite network width, smooth activations, and a near-linearized training regime. By contrast, mechanisms such as spectral normalization actively modify the network parameterization and training geometry, and therefore are not naturally captured by the usual fixed-kernel NTK approximation. For this reason, our present theory does not yet characterize how such techniques alter the discriminator NTK, the induced gradient field, or the resulting residual dynamics in a strict sense. At the same time, this does not prevent us from analyzing the resulting training behavior empirically through the network output and residual-based indicators. One practical strength of the present framework is that these indicators remain informative even when such regularization mechanisms are used in training. A fuller theoretical integration of spectral normalization, gradient penalties, and related techniques would make the framework both more complete and more realistic.

Fourth, the generator-side analysis relies on the assumption that the generator NTK remains approximately constant. For relatively regular problems such as Poisson and Laplace equations, this approximation appears reasonable and is consistent with both existing studies and our numerical observations [5]. However, for more complex strongly nonlinear, multiscale, stiff, or higher-order

PDEs, the generator NTK may evolve more substantially during training, and the constant-kernel approximation may become less accurate [22, 23]. In this sense, the present theory is better suited to describing early-stage or weakly nonlinear training dynamics than fully global dynamics in strongly nonlinear regimes. Even so, one of the main messages of this paper is that a linearized infinite-width analysis can still provide practically useful guidance for finite-network training control.

Fifth, under finite-width neural network training, the first-order quantity $r^\top K_{rr}^G \gamma$ should be understood as an aggregated and low-cost first-order indicator rather than as a strict per-step descent certificate. It effectively separates successful regimes from representative failure modes and provides a practical basis for rollback, but it remains a low-dimensional global scalar and cannot fully resolve finer interactions among spectral modes, spatial regions, or multiple loss channels. In difficult PDEs, the dominant error may come from a small subset of slow modes, boundary channels, or localized regions, while a single aggregated score may not fully reflect such structural imbalance. At the same time, this indicator is not spectrally blind. Through the decomposition of $r^\top K_{rr}^G \gamma$, it implicitly constrains how discriminator-induced gradients are distributed across modes. In particular, fast modes associated with larger kernel eigenvalues typically decay first, so their residual amplitudes become small early in training. If one still wishes to maintain $r^\top K_{rr}^G \gamma < 0$ afterwards, then the discriminator-induced weighting must become more consistent with the slower unresolved modes that continue to dominate the residual energy. In this sense, the first-order indicator does not explicitly optimize spectral modes one by one, but it implicitly favors a training process that first reduces the fast modes and then increasingly emphasizes the slow ones. It is precisely because this quantity is global, low-cost, and available online that it provides a workable bridge between theory and algorithm.

Sixth, while our experiments already cover a diverse set of PDEs, including Poisson, Laplace, viscous Burgers, Reaction–Diffusion, and Klein–Gordon equations, the overall benchmark range is still limited. We have not yet systematically studied higher-dimensional PDEs, complex geometries, noisy inverse problems, parameter identification tasks, or more extreme stiffness and boundary-condition regimes. Therefore, although the current conclusions already show a certain degree of generality, their broader scope still requires further validation. On the other hand, the existing experiments are sufficient to show that the proposed framework is not tied to a single equation or isolated phenomenon, but has explanatory and algorithmic value across multiple PDE types.

Future Work: These limitations also point naturally to several directions for future work.

First, the rollback strategy itself can be made more stable. For example, one may incorporate rollback as an additional control layer on top of an already well-tuned optimization setting, or design smoother rollback criteria with momentum, confidence thresholds, or multi-step prediction mechanisms to reduce oscillations.

Second, it would be valuable to move beyond the constant-NTK or weakly varying NTK approximation and study more general time-varying kernel dynamics, especially for strongly nonlinear, multiscale, and higher-order PDEs. Such an analysis could lead to a deeper understanding of how generator kernels and discriminator feedback co-evolve across residual modes.

Third, common practical techniques such as spectral normalization, gradient penalties, and residual links should be incorporated into a unified theoretical framework. Understanding how these mechanisms reshape the discriminator NTK, its input-gradient field, and the induced preconditioned residual descent may lead not only to a more faithful theory, but also to better training strategies.

Fourth, the broader design space suggested by this work deserves more systematic exploration. In particular, many combinations among residual input versus squared-residual input, GAN/LSGAN/IPM-type objectives, and constrained discriminator function classes remain unexplored. Such combinations may exhibit better stability, spectral balance, or stiffness-handling properties.

Fifth, future work may develop more refined control criteria beyond the scalar quantity $r^\top K_{rr}^G \gamma$, for instance by using spectral information, channel-wise loss decomposition, or block-structured residual diagnostics for interior, boundary, and initial-condition terms. This could yield a more fine-grained and PDE-aware update-control mechanism. Balancing the computational cost of solving the spectrum and the resulting accuracy during training remains an open problem.

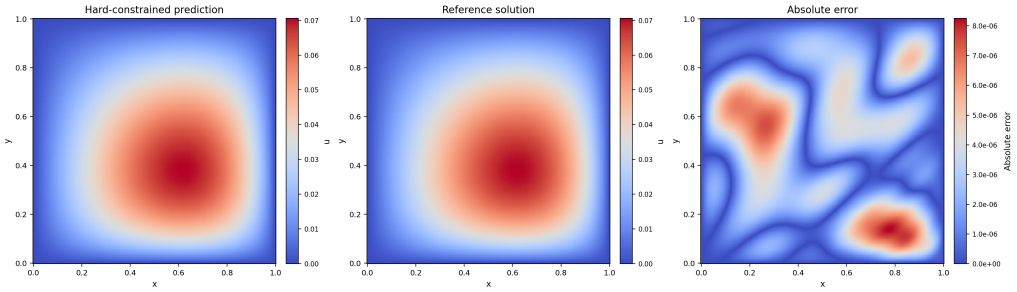
Sixth, the present framework could be extended to broader scientific machine learning settings, including soft-constrained PINNs, inverse problems, optimal control, weak-form methods, and function-space generative models. Since the key idea of this paper is that the discriminator modulates

generator dynamics by inducing a gradient field on residual space, this mechanism is not restricted to standard PINNs and may be relevant in a wider class of problems.

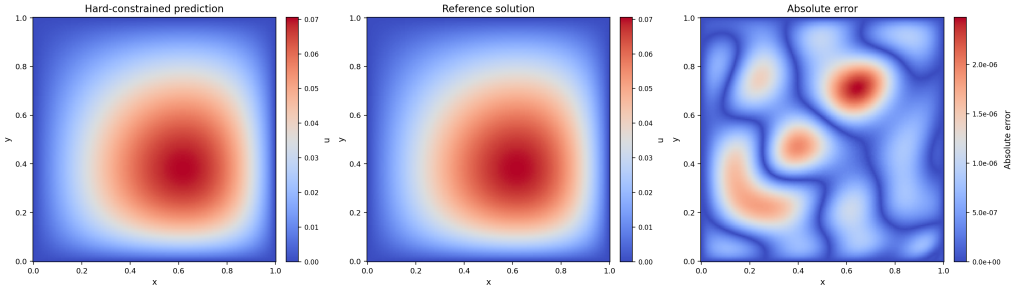
J Two-Dimensional Heatmap Comparison of GAN-Based Solutions

This section presents a comparative visual assessment of the hard-constrained solutions obtained using GAN-RB, LSGAN-RB, and WGAN-RB for the six PDE problems. In each figure, the three rows correspond, from top to bottom, to GAN-RB, LSGAN-RB, and WGAN-RB, respectively. Within each row, the panels from left to right display the hard-constrained prediction, the reference solution, and the pointwise absolute error, defined as $|\hat{u}(\mathbf{x}) - u_{\mathrm{ref}}(\mathbf{x})|$.

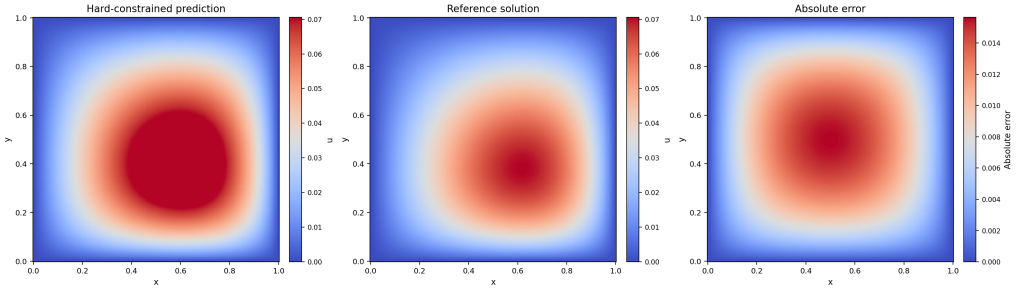
Figure 12 compares the predicted and reference solutions and their absolute errors for the Poisson equation. Figures 13 and 14 present the corresponding results for the two Laplace problems with different boundary conditions. Figures 15, 16, and 17show the space–time solutions and error distributions for the reaction–diffusion, viscous Burgers, and Klein–Gordon equations, respectively.



(a) GAN-RB

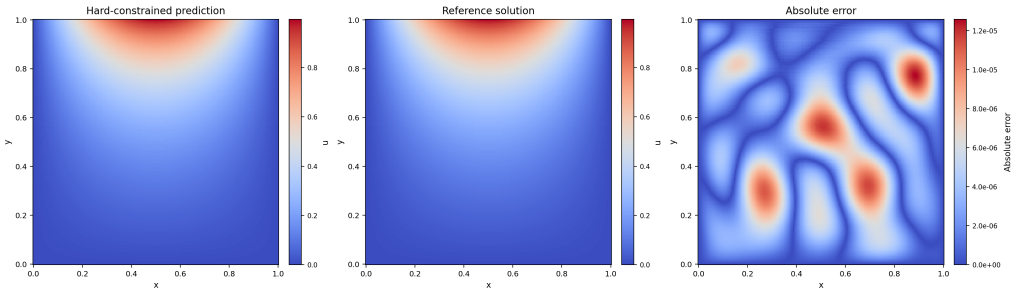


(b) LSGAN-RB

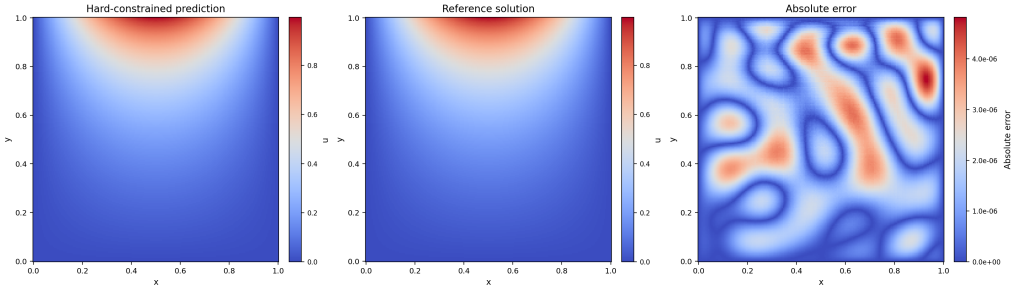


(c) WGAN-RB

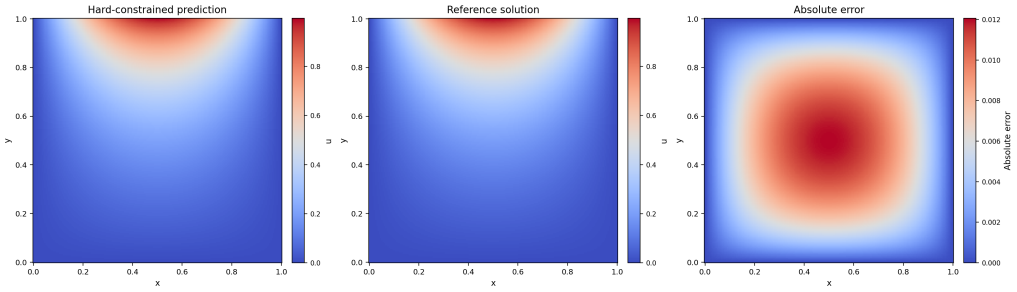
Figure 12: Poisson equation: From top to bottom, the rows present the results obtained using GAN-RB, LSGAN-RB, and WGAN-RB, respectively. Within each row, the panels from left to right show the hard-constrained prediction, the reference solution, and the pointwise absolute error.



(a) GAN-RB

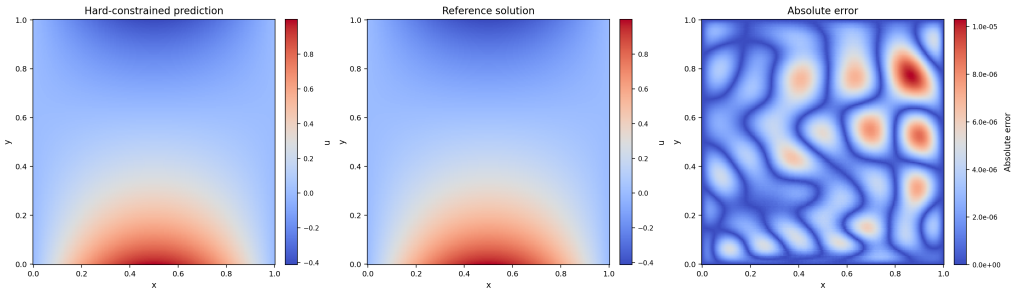


(b) LSGAN-RB

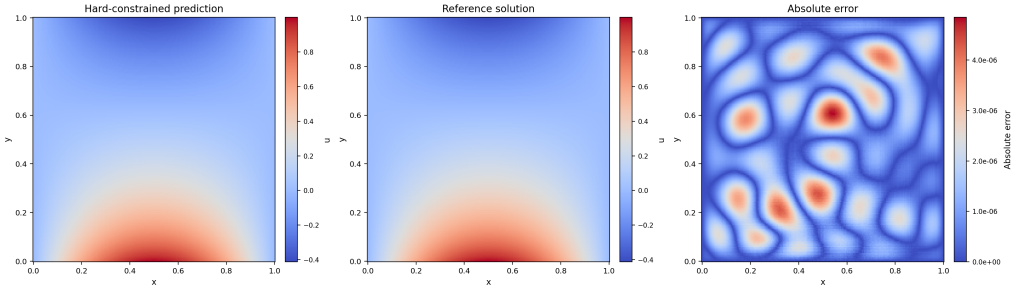


(c) WGAN-RB

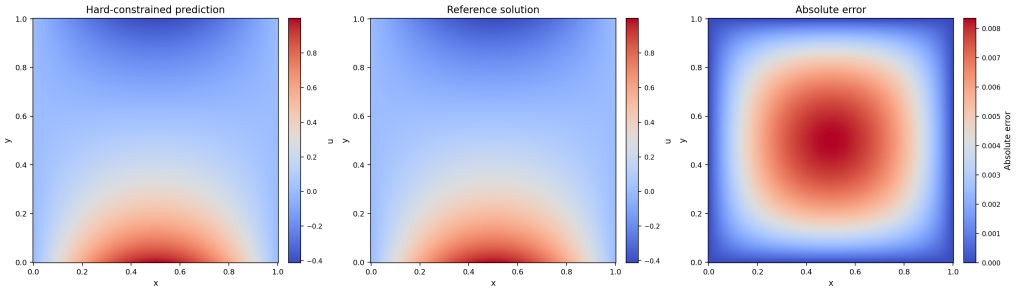
Figure 13: Laplace equation 1: From top to bottom, the rows present the results obtained using GAN-RB, LSGAN-RB, and WGAN-RB, respectively. Within each row, the panels from left to right show the hard-constrained prediction, the reference solution, and the pointwise absolute error.



(a) GAN-RB

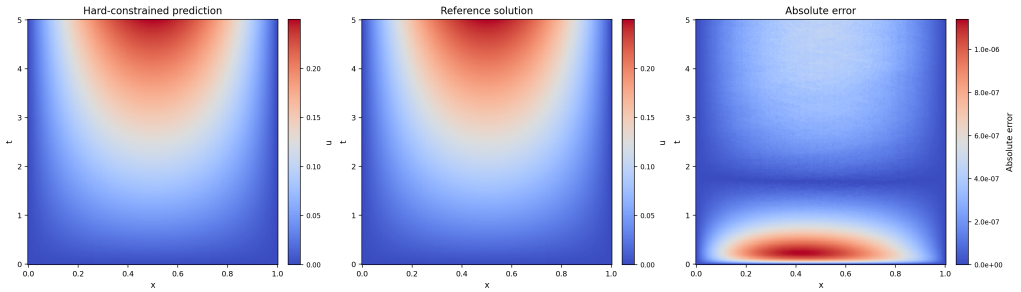


(b) LSGAN-RB

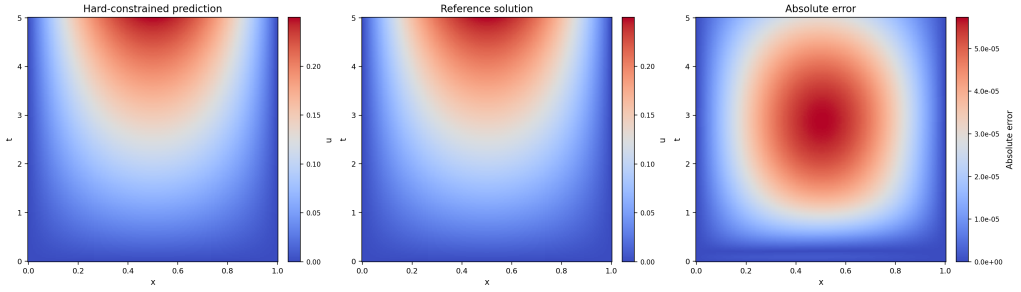


(c) WGAN-RB

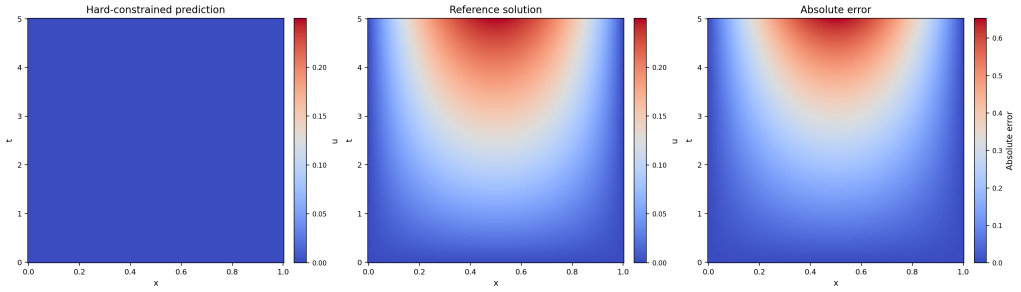
Figure 14: Laplace equation 2: From top to bottom, the rows present the results obtained using GAN-RB, LSGAN-RB, and WGAN-RB, respectively. Within each row, the panels from left to right show the hard-constrained prediction, the reference solution, and the pointwise absolute error.



(a) GAN-RB

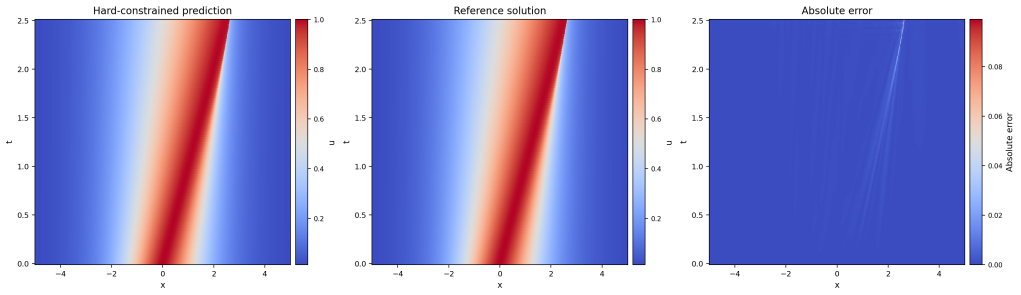


(b) LSGAN-RB

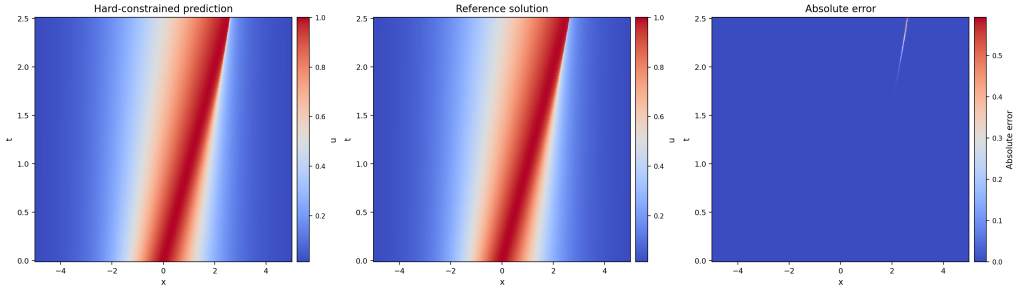


(c) WGAN-RB

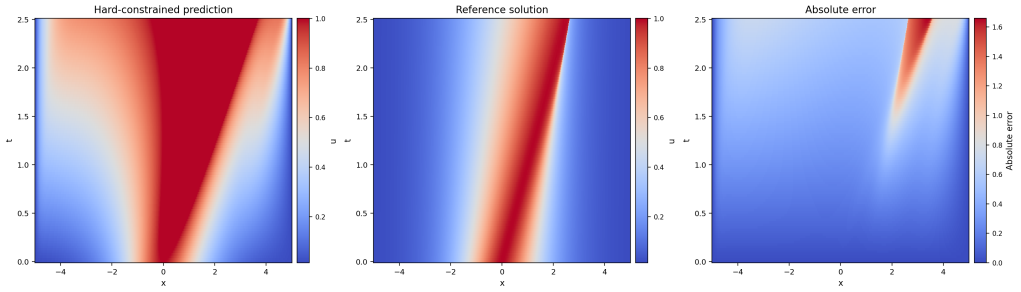
Figure 15: Reaction-Difussion equation: From top to bottom, the rows present the results obtained using GAN-RB, LSGAN-RB, and WGAN-RB, respectively. Within each row, the panels from left to right show the hard-constrained prediction, the reference solution, and the pointwise absolute error.



(a) GAN-RB

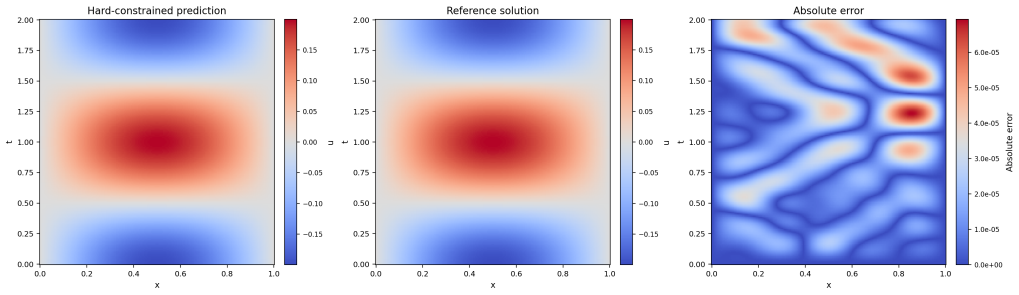


(b) LSGAN-RB

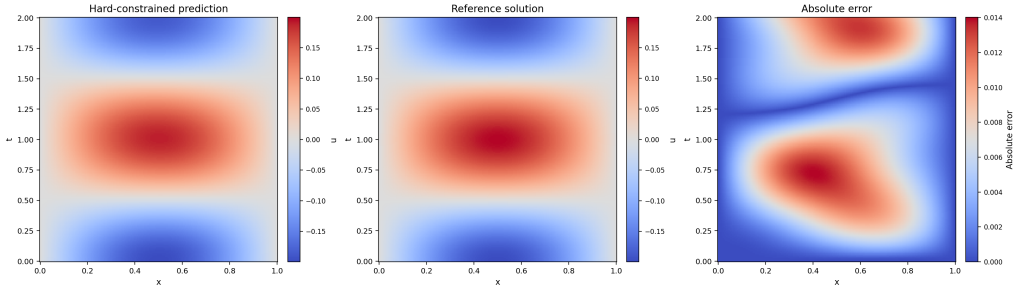


(c) WGAN-RB

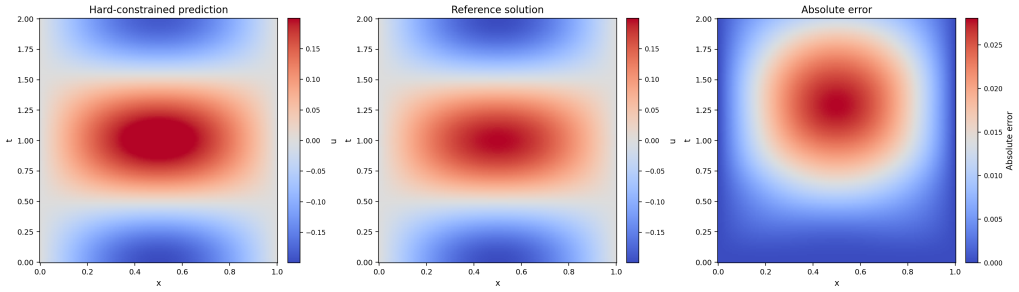
Figure 16: Viscous Burgers equation: From top to bottom, the rows present the results obtained using GAN-RB, LSGAN-RB, and WGAN-RB, respectively. Within each row, the panels from left to right show the hard-constrained prediction, the reference solution, and the pointwise absolute error.



(a) GAN-RB



(b) LSGAN-RB



(c) WGAN-RB

Figure 17: Klein-Gordon equation: From top to bottom, the rows present the results obtained using GAN-RB, LSGAN-RB, and WGAN-RB, respectively. Within each row, the panels from left to right show the hard-constrained prediction, the reference solution, and the pointwise absolute error.

NeurIPS Paper Checklist

1. **Claims**
   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?
   Answer: [Yes]
   Justification: The claims in the abstract and introduction are supported by the theoretical analysis in Sec. 3–4, with full assumptions, proofs and analysis provided in Appendix B and C. Numerical experiments supporting the theoretical analysis are presented in Sec. 5, while more detailed experimental settings, algorithmic comparisons, and ablation studies are provided in the appendix H.
   Guidelines:
   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.
2. **Limitations**
   Question: Does the paper discuss the limitations of the work performed by the authors?
   Answer: [Yes]
   Justification: The limitations of this work are discussed in Sec. 7. More detailed discussions of these limitations, together with potential future research directions motivated by them, are provided in the appendix I.
   Guidelines:
   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.
3. **Theory assumptions and proofs**
   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?
   Answer: [Yes]

Justification: All theoretical results are stated with explicit assumptions introduced in the preliminaries (Sec. 2), which build on the NTK background detailed in Appendix A.4. Complete derivations and proofs for all main results are provided in the appendix B and C, with key intuitions summarized in the main text.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification:All experimental procedures are described in detail in Sec. 5 and Appendix H, including model architectures, hyperparameters, training protocols, update ratios, rollback strategies, and evaluation metrics. These details provide sufficient information to reproduce the main results and conclusions.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?
Answer: [Yes]
Justification: We will provide data, code, and instructions in the final paper upon acceptance.
Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**
Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?
Answer: [Yes]
Justification: All training and evaluation details are specified in Sec. 5 and Appendix H, including data sampling strategies, model architectures, hyperparameters, optimizer settings, update ratios, and rollback configurations. These details are sufficient to understand, interpret, and reproduce the reported results.
Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**
Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?
Answer: [Yes]
Justification: We evaluate the numerical accuracy by comparing the learned solutions with reference solutions generated from analytical solutions or numerical solvers. The resulting relative L_2 errors are reported in Table 2, providing quantitative evidence for the experimental comparisons.
Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**
   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?
   Answer: [Yes]
   Justification: The compute resources used in the experiments are reported in the appendix H, including the hardware configuration, implementation framework, and approximate running time required to reproduce the main experiments.
   Guidelines:
   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).
9. **Code of ethics**
   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?
   Answer: [Yes]
   Justification: The research complies with the NeurIPS Code of Ethics. The work focuses on numerical methods for PDEs and does not involve human subjects, sensitive data, or ethically concerning applications.
   Guidelines:
   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).
10. **Broader impacts**
    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?
    Answer: [No]
    Justification: Our work shares the general ethical considerations common to AI research, and does not present any unique or specific societal impact that warrants separate discussion.
    Guidelines:
    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to

any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All existing assets used in this work, including standard PDE benchmarks, numerical solvers, and software packages, are properly cited where applicable. Their licenses and terms of use are respected, and implementation details are provided in the appendix H.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [No]
Justification: At submission time, this paper does not release new public assets. We plan to provide the code, together with a README containing installation, training, evaluation, and reproduction instructions, in the final version upon acceptance.
Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**
Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?
Answer: [N/A]
Justification: The paper does not involve crowdsourcing nor research with human subjects.
Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**
Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?
Answer: [N/A]
Justification: The paper does not involve crowdsourcing nor research with human subjects.
Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**
Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.
Answer: [N/A]
Justification: LLM was used only for writing, editing, and formatting purposes.
Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.

- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.