PHYSICAL RESPONSE ATTACK: EXPOSING HIDDEN RESPONSE DISCREPANCIES IN NEURAL OPERATORS

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural operators have emerged as efficient surrogates for numerical PDE solvers. However, a reliable neural operator should not only achieve low prediction error, but also preserve the response characteristics dictated by the underlying physical equations. Existing robustness evaluations are largely model-centric and focus on how perturbations affect the model output, while overlooking whether the learned operator responds consistently with the true physical system. To address this gap, we propose **Physical Response Attack (PRA)**, a fully black-box framework that exploits the physical response structure of the governing PDE. Using only forward queries to a reference numerical solver, PRA identifies physically informative perturbation directions and constructs a compact search subspace, within which it searches for perturbations that maximize the response discrepancy between the neural operator and the solver. PRA requires no access to model parameters, gradients, or internal architecture; it does not require a differentiable solver and performs no backpropagation through either system. We evaluate PRA on the Poisson equation, Darcy flow, and two-dimensional Navier–Stokes equations across four representative neural operator architectures. Experimental results show that PRA effectively reveals failure directions tied to the underlying physical structure; in particular, for time-dependent systems, PRA can induce pronounced deviations in predicted dynamical trajectories. These results demonstrate that the reliability of neural operators should not be determined by prediction accuracy alone; they must also faithfully reproduce the local response and dynamical behavior dictated by the governing equations.

1 INTRODUCTION

Neural operators have emerged as powerful data-driven surrogates for partial differential equations (PDEs), enabling fast prediction across varying initial conditions, forcing terms, and physical parameters. Architectures such as Fourier Neural Operators (FNOs) and Transolver have demonstrated strong performance across a broad range of scientific computing problems (Li et al., 2021; Wu et al., 2024). However, their increasing use in scientific computing raises important questions about their reliability, motivating recent studies on the adversarial robustness of neural operators.

Conventional adversarial attacks do not directly apply to neural operators, as they typically assume the ground truth remains unchanged under small perturbations. In contrast, perturbing a neural operator's input defines a new physical problem, requiring the reference solution to be recomputed. Consequently, attack success should be determined by the discrepancy between the perturbed model prediction and its corresponding reference solution, rather than the original ground truth.

Recent studies have begun to incorporate numerical solvers into adversarial robustness evaluation, using numerical solutions as physical references (Adesoji & Chen, 2022). However, using the solver solely for evaluation does not fully exploit its physical response during perturbation optimization. To address this limitation, solver-integrated attacks directly incorporate the numerical solver into the optimization process to maximize the model–solver discrepancy (Sun, 2026). Nevertheless, such gradient-based approaches can incur substantial computational costs and require differentiable solvers. Black-box attacks have also been explored in neural-operator settings (Roy et al., 2026), but searching for adversarial perturbations in high-dimensional PDE input spaces can require extensive model queries, limiting their computational efficiency.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

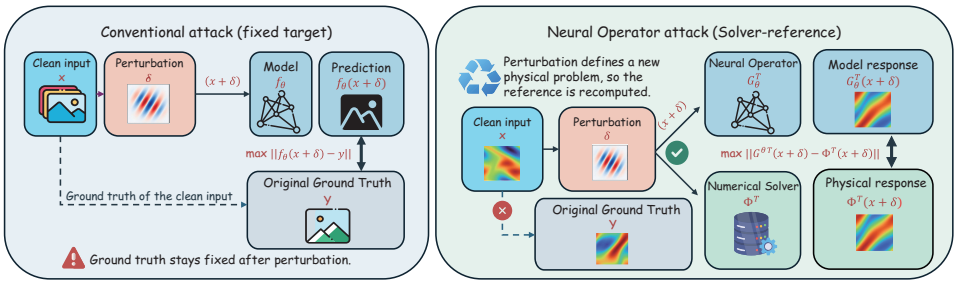


Figure 1: Comparison between conventional adversarial attacks and neural-operator attacks. Conventional attacks typically retain the original ground-truth target after perturbation, whereas perturbing a neural-operator input defines a new physical problem whose reference solution must be recomputed using a numerical solver.

This raises a key question: can we develop a black-box attack that avoids gradient access while substantially reducing query complexity and improving attack efficiency? The inherent physical priors of PDE systems provide a promising avenue for addressing this challenge, as their physical response structure can be exploited to guide adversarial search toward informative perturbation directions.

Based on this principle, we propose **Physical Response Attack (PRA)**, a fully black-box adversarial framework for neural operators. PRA exploits the response structure of the governing PDE to identify a compact set of physically informative perturbation directions and then searches within this subspace for perturbations that maximize the response discrepancy between the neural operator and the reference solver. By restricting the attack to a physics-informed search space, PRA avoids blind exploration of the full high-dimensional input domain and requires only forward queries to both systems. It does not require access to model parameters, gradients, internal representations, or differentiable numerical solvers.

Our main contributions are as follows.

- Formulate neural-operator robustness in a fully black-box, response-based setting, where we evaluate the learned operator against the reference solver on the same perturbed input.
- Propose **Physical Response Attack (PRA)**, a black-box, physics-guided framework that uses only forward queries to construct a solver-informed search subspace and identify vulnerable response directions without requiring gradients or a differentiable solver.
- We evaluate PRA across multiple PDE systems, neural-operator architectures, and physical regimes, showing that it can expose substantial model–solver response discrepancies and trajectory-level failures not fully captured by scalar prediction error alone.

2 Background and Problem Formulation

2.1 Related Work

Traditional Adversarial Attacks. Classical adversarial attacks are typically formulated with respect to a fixed ground-truth target. In classification, methods such as FGSM and PGD perturb the input while assuming that the underlying class label remains unchanged, optimizing the perturbation to increase the loss relative to this fixed label (Goodfellow et al., 2015; Madry et al., 2019). Other attacks differ mainly in how they construct perturbations: C&W reformulates the constrained optimization problem (Carlini & Wagner, 2017), while DeepFool searches for nearby decision boundaries (Moosavi-Dezfooli et al., 2016). Black-box methods, including Boundary Attack, HopSkipJump, and Square Attack, further enable adversarial search without direct gradient access (Brendel et al., 2018; Chen et al., 2020; Andriushchenko et al., 2020).

Adversarial Attacks on Neural Operators. Conventional adversarial attacks typically assume a fixed ground-truth target. In contrast, perturbing a neural operator's input defines a new physical problem whose ground-truth solution can be recomputed with a numerical solver. Early work investigated the adversarial robustness of Fourier Neural Operators (FNOs) under norm-bounded input

perturbations, using numerical solutions to evaluate prediction errors (Adesoji & Chen, 2022). Although the numerical solver served as a reference, its physical response was not explicitly used to guide adversarial perturbation optimization. More recent studies have integrated numerical solvers directly into adversarial optimization. Solver-integrated attacks jointly consider the responses of the neural operator and the numerical solver under the same perturbed input, optimizing perturbations to maximize their prediction discrepancy (Sun, 2026). Meanwhile, researchers have also explored black-box and derivative-free attacks in neural-operator settings, including using gradient-free differential evolution to identify sparse adversarial perturbations in neural-operator-based digital twins (Roy et al., 2026).

2.2 Problem Formulation

As illustrated in Fig. 1, conventional adversarial attacks typically assume a fixed ground-truth target, whereas perturbing the input of a neural operator defines a new physical problem whose reference solution must be recomputed.

Let $\Phi^T : \mathcal{X} \to \mathcal{Y}$ denote the reference PDE solver over a finite horizon T, and let $G_\theta^T : \mathcal{X} \to \mathcal{Y}$ denote the aligned neural-operator rollout. For a nominal initial condition $x_0 \in \mathcal{X}$, the clean model-solver discrepancy is $e_0 = G_\theta^T(x_0) - \Phi^T(x_0)$.

To distinguish physical sensitivity from model prediction errors, we define the finite-amplitude physical and model responses as

$$\begin{aligned} R^T_{\text{phys}}(\delta) &= \Phi^T(x_0 + \delta) - \Phi^T(x_0), \\ R^T_{\text{model}}(\delta) &= G^T_\theta(x_0 + \delta) - G^T_\theta(x_0). \end{aligned} \tag{1}$$

Their difference, $\Delta R_T(\delta) = R^T_{\text{model}}(\delta) - R^T_{\text{phys}}(\delta)$, measures how differently the learned operator and the reference solver respond to the same perturbation.

The perturbed model–solver discrepancy admits the exact decomposition

$$\begin{aligned} &G^T_\theta(x_0 + \delta) - \Phi^T(x_0 + \delta) \\ &\quad = e_0 + \Delta R_T(\delta). \end{aligned} \tag{2}$$

Accordingly, under a perturbation budget ϵ, the solver-referenced adversarial objective is

$$\delta^\star = \arg \max_{\|\delta\|_2 \le \epsilon} \|e_0 + \Delta R_T(\delta)\|_2 . \tag{3}$$

This formulation separates the existing clean discrepancy from the perturbation-induced response mismatch. An effective attack must increase the model-solver discrepancy rather than merely amplify the physical or model response.

2.3 Limitations of existing work

Limitations of Gradient-Based Solver Integration : Although solver-integrated attacks provide an input-dependent physical reference, computing gradients through the numerical solver can require substantial computational resources, particularly for time-dependent PDEs involving many internal integration steps. For example, the Navier-Stokes solver used in the original FNO data-generation pipeline employs an internal time step of $\Delta t = 10^{-4}$, requiring 10^4 numerical integration steps to advance the solution by one unit of physical time (Li et al., 2021). Direct backpropagation through such a solver involves differentiating through the entire sequence of numerical updates, resulting in substantial memory requirements. Moreover, many established scientific solvers are not differentiable or are implemented outside automatic-differentiation frameworks, making direct backpropagation through the solver impractical (Arisaka & Li, 2024).

Challenges of Black-Box Search. Black-box attacks avoid backpropagation through the numerical solver by relying only on forward queries. However, searching for adversarial perturbations in high-dimensional PDE input spaces can require extensive model and solver evaluations.

Existing gradient-free attacks have explored sparse perturbation strategies to reduce the search dimension (Roy et al., 2026). Nevertheless, optimizing changes in model predictions alone does not account for the PDE's corresponding physical response. Moreover, extending black-box optimization to general high-dimensional input fields remains computationally challenging.

PDE systems provide physical priors that can be exploited through forward solver evaluations to identify informative perturbation directions and construct a compact, physics-informed search space, within which perturbations that induce model-solver response discrepancies can be identified.

2.4 MOTIVATION

Physical Response Structure: PDE systems exhibit nonuniform responses to input perturbations, as their governing dynamics determine how different components propagate, dissipate, amplify, and interact over time. Consequently, perturbations of equal magnitude can induce substantially different physical responses. Consider the two-dimensional incompressible Navier-Stokes equation in vorticity form:

$$\frac{\partial \omega}{\partial t} + \mathbf{u} \cdot \nabla \omega = \nu \Delta \omega + f, \tag{4}$$

where ω denotes vorticity, $\mathbf{u}$ is the velocity field, ν is the viscosity, and f is the external forcing. In Fourier space, viscous dissipation acts on a mode with wavenumber k as $-\nu|k|^2\widehat{\omega}_k$. In the absence of nonlinear coupling, the mode decays as

$$\widehat{\omega}_k(t) \propto e^{-\nu|k|^2 t}\widehat{\omega}_k(0). \tag{5}$$

Thus, higher-frequency components experience stronger viscous dissipation, while other components may persist or interact with nonlinear dynamics over longer horizons.

This nonuniform response structure motivates constructing a compact perturbation subspace from dynamically informative directions identified through reference solver evaluations. Searching within their span reduces the number of optimization variables while focusing on physically meaningful perturbations.

Moreover, the effective perturbation response depends on the numerical solver configuration. For example, different de-aliasing settings in pseudo-spectral Navier-Stokes solvers retain different portions of the Fourier spectrum, potentially altering the evolution of the same perturbation. We therefore extract dynamically informative directions directly from the actual reference solver, rather than prescribing a fixed perturbation basis, to construct a compact, solver-informed subspace for adversarial optimization.

Lyapunov-Informed Subspace. Finite-time Lyapunov analysis characterizes how perturbations evolve along a reference trajectory (Benettin et al., 1980). Let Φ^τ denote the evolution operator over a finite interval τ. For a reference state x and a sufficiently small perturbation direction v, its local evolution is given by $v(\tau) = D\Phi^\tau(x)v$, with the corresponding finite-time Lyapunov growth rate

$$\lambda_\tau(v) = \frac{1}{\tau} \log \frac{\|D\Phi^\tau(x)v\|_2}{\|v\|_2}. \tag{6}$$

Positive and negative values indicate finite-time amplification and contraction, respectively. By characterizing these responses, Lyapunov analysis identifies dynamically informative directions for constructing a compact search space. Importantly, physical amplification does not necessarily imply adversarial effectiveness, as both amplified and contracting directions may exhibit substantial model-solver response discrepancies. The solver-derived subspace therefore determines **where to search**, while subsequent response-mismatch optimization determines **what to optimize**.

Motivated by this principle, we propose **Physical Response Attack (PRA)**, a fully black-box, physics-informed framework that constructs a compact perturbation subspace from solver-derived physical response information and searches within it to maximize the model-solver discrepancy under the same perturbed input.

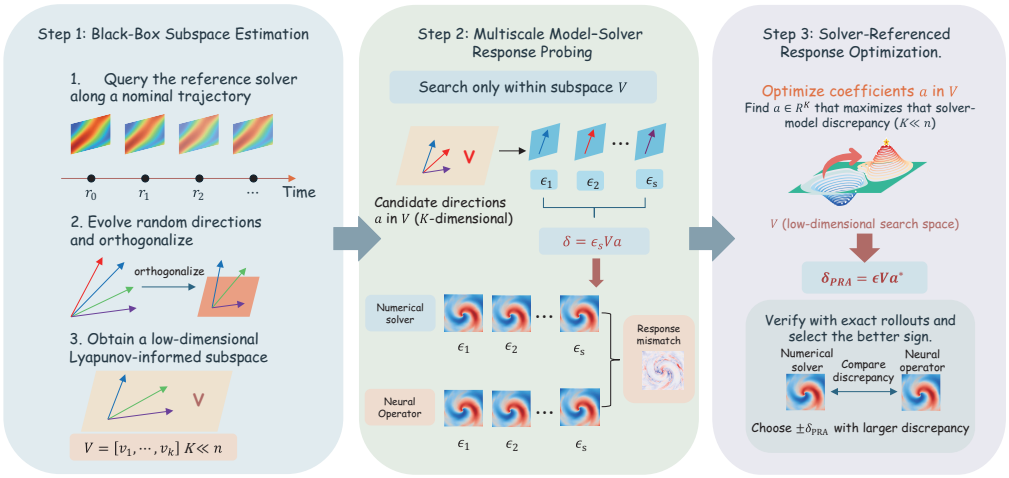


Figure 2: **Overview of PRA.** (1) We estimate a low-dimensional Lyapunov-informed subspace V ($K \ll n$) by evolving and orthogonalizing random directions along a nominal trajectory of the black-box solver. (2) Perturbations are restricted to $\delta = \epsilon V \alpha$, and the responses of the numerical solver and neural operator are compared. (3) We optimize $\alpha \in \mathbb{R}^K$ to maximize their discrepancy and, after verification with exact rollouts, select the sign of δ_{PRA} yielding the larger discrepancy.

3 PRA Framework

As illustrated in Fig. 2, PRA proceeds in three stages: (1) estimating a dynamics-informed search subspace through black-box solver queries, (2) comparing model and solver responses along this subspace at multiple perturbation scales, and (3) optimizing the perturbation within the resulting low-dimensional space and validating it with exact nonlinear rollouts. The appendix 1 provides detailed pseudocode for the proposed method.

Step 1: Black-Box Lyapunov Subspace Estimation. Since the reference solver is treated as a black box, PRA approximates the finite-time evolution of perturbation directions using only forward solver queries. We initialize K random directions and orthonormalize them to obtain $Q^{(0)} = [q_1^{(0)}, \ldots, q_K^{(0)}]$. Dividing the propagation horizon into intervals of length τ, we evolve the reference trajectory as $r_k = \Phi^\tau(r_{k-1})$ and approximate the propagation of each perturbation direction using finite differences:

$$z_i^{(k)} = \frac{\Phi^\tau(r_{k-1} + h q_i^{(k-1)}) - r_k}{h}. \tag{7}$$

The propagated directions are collected into $Z^{(k)} = [z_1^{(k)}, \ldots, z_K^{(k)}]$ and reorthogonalized through QR decomposition:

$$Z^{(k)} = Q^{(k)} R^{(k)}. \tag{8}$$

The resulting $Q^{(k)}$ serves as the basis for the next propagation interval. Repeating this procedure over the finite horizon yields the Lyapunov-informed basis $V = [v_1, \ldots, v_K]$, which defines the dynamics-informed search subspace used by PRA.

Step 2: Multiscale Model–Solver Response Probing The Lyapunov-informed basis identifies physically informative directions but does not determine their model–solver response discrepancies. PRA therefore compares the responses of both systems to the same perturbations at multiple finite amplitudes $\{\epsilon_s\}_{s=1}^S$. For each basis direction v_i, we construct

$$\begin{aligned} M_{\mathrm{phys}}^{(s)}[:, i] &= \frac{\Phi^T(x_0 + \epsilon_s v_i) - \Phi^T(x_0)}{\epsilon_s}, \\ M_{\mathrm{model}}^{(s)}[:, i] &= \frac{G_\theta^T(x_0 + \epsilon_s v_i) - G_\theta^T(x_0)}{\epsilon_s}. \end{aligned} \tag{9}$$

Their difference, $B_s = M^{(s)}_{\text{model}} - M^{(s)}_{\text{phys}}$, captures the finite-amplitude response mismatch. For a combined perturbation $\delta = \epsilon_s V a$, with $\|a\|_2 = 1$, we approximate its response mismatch using

$$\Delta R_T(\epsilon_s V a) \approx \epsilon_s B_s a. \tag{10}$$

This finite-amplitude superposition approximation assumes approximately additive directional responses within the subspace. Since nonlinear interactions may invalidate this assumption, the approximation serves only as a search surrogate, it serves only as a search surrogate, and the resulting perturbations are subsequently evaluated using exact nonlinear rollouts.

Step 3: Solver-Referenced Response Optimization. Using Eq. 10, PRA optimizes a shared coefficient vector across all perturbation scales:

$$a^\star = \arg\max_{\|a\|_2=1} \frac{1}{S}\sum_{s=1}^{S} \|e_0 + \epsilon_s B_s a\|_2^2 . \tag{11}$$

This objective favors response mismatches that enlarge, rather than cancel, the existing model–solver discrepancy. Expanding Eq. 11 and removing terms independent of a yields

$$\begin{aligned} &\max_{\|a\|_2=1} \quad a^\top H a + 2b^\top a, \\ &H = \frac{1}{S}\sum_{s=1}^{S} \epsilon_s^2 B_s^\top B_s, \qquad b = \frac{1}{S}\sum_{s=1}^{S} \epsilon_s B_s^\top e_0. \end{aligned} \tag{12}$$

Since $K \ll n$, we solve this problem in the low-dimensional coefficient space using multi-start projected gradient ascent on the unit sphere. For $J(a) = a^\top H a + 2b^\top a$, the gradient $g = 2(Ha + b)$ is projected onto the tangent space as $g_{\tan} = g - a(a^\top g)$, followed by normalization and backtracking line search. The resulting perturbation direction and its realization at the target budget ϵ are

$$d_{\text{PRA}} = V a^\star, \qquad \delta_{\text{PRA}} = \epsilon\, d_{\text{PRA}}. \tag{13}$$

The multiscale amplitudes $\{\epsilon_s\}$ guide direction selection, while ϵ specifies the target evaluation budget. Finally, we evaluate both signs $\pm d_{\text{PRA}}$ using exact nonlinear rollouts and retain the one yielding the larger solver-referenced discrepancy, thereby validating the finite-amplitude surrogate.

4 Experiments

4.1 Experimental Setup

PDEs, data, and neural surrogates. We evaluate PRA on three representative PDE systems: Navier–Stokes (NS), Darcy flow, and Poisson equations, covering both time-dependent and static settings. Each dataset contains 1,000 training samples and 200 test samples. For NS and Darcy, we follow the data-generation settings of the FNO benchmark. We consider NS at $\nu \in 10^{-3}, 10^{-4}$ on a 64×64 periodic grid, where the neural surrogate takes a 10-frame input window and autoregressively predicts future states. We treat Darcy and Poisson as static input-to-output mappings, and we solve the Poisson equation using a finite-difference reference solver. We evaluate attacks on four neural-operator architectures: FNO (Li et al., 2021), UNO (Rahman et al., 2023), Transolver (Wu et al., 2024), and UPT (Alkin et al., 2025). For all PDEs, the reference solver is independent of the attacked neural operator and is evaluated separately on both clean and perturbed inputs.

Evaluation protocol. For an initial field u_0 with n spatial degrees of freedom, we use

$$\|\delta\|_2 = c\,\text{std}(u_0)\sqrt{n}, \qquad c \in \{0.25, 0.5, 1.0\}. \tag{14}$$

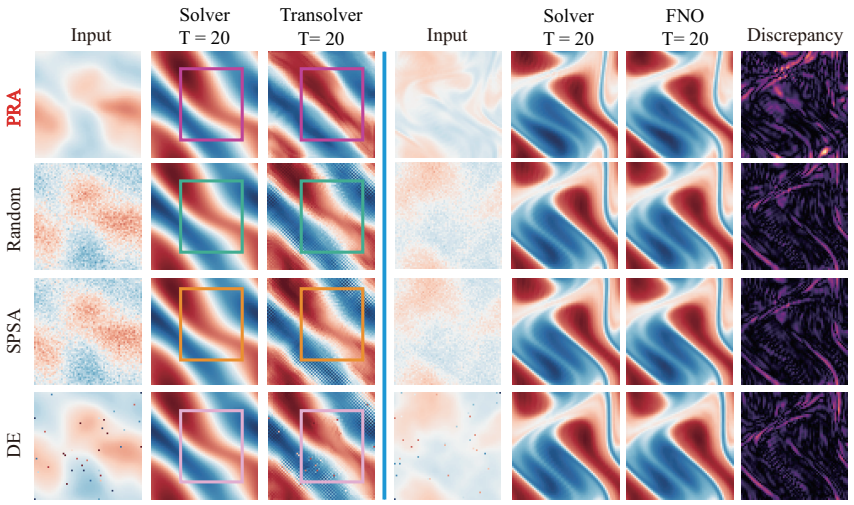
For NS, the perturbed initial condition is first propagated by the reference solver to reconstruct the 10-frame model input window, after which the neural operator is rolled out autoregressively. We evaluate rollout $T \in \{16, 18, 20\}$. For static PDEs, the perturbed field is passed directly to both operators. Attack effectiveness is measured by the amplification of the solver-referenced relative error,

$$A_T = \frac{E_{\text{adv}}(T)}{E_{\text{clean}}(T)}, \qquad E(T) = \frac{\|\widehat{u}_T - u_T\|_2}{\|u_T\|_2}. \tag{15}$$

Random perturbations with the same norm are used as the primary baseline.

Table 1: Attack amplification of PRA, Random Attack, SPSA, and DE (Roy et al., 2026) on Navier-Stokes across different neural operators and viscosity regimes.

Model	Attack	ϵ/σ	$\nu = 10^{-3}$			$\nu = 10^{-4}$		
			$T = 16$	$T = 18$	$T = 20$	$T = 16$	$T = 18$	$T = 20$
Transolver	PRA	50%	2.22×	2.26×	2.51×	**2.51×**	**2.78×**	**2.88×**
		100%	4.78×	**4.26×**	**3.96×**	**3.79×**	**3.62×**	**4.27×**
	Random Attack	50%	2.29×	2.37×	2.50×	1.64×	1.72×	1.45×
		100%	2.96×	2.60×	2.43×	2.00×	2.26×	1.98×
	SPSA	50%	–	–	2.63×	–	–	1.65×
		100%	–	–	2.55×	–	–	1.69×
	DE	50%	**4.07×**	**3.07×**	**2.82×**	1.63×	1.91×	1.58×
		100%	**4.94×**	3.40×	2.89×	1.83×	2.25×	1.86×
FNO	PRA	50%	**6.70×**	**8.37×**	**8.74×**	**1.12×**	**1.12×**	**1.43×**
		100%	**26.44×**	**29.59×**	**23.85×**	**1.52×**	**1.39×**	**1.92×**
	Random Attack	50%	1.15×	0.95×	0.94×	1.06×	1.06×	1.06×
		100%	1.61×	0.98×	0.92×	1.13×	1.15×	1.17×
	SPSA	50%	–	–	1.23×	–	–	1.02×
		100%	–	–	1.82×	–	–	1.16×
	DE	50%	1.23×	1.05×	0.96×	1.07×	1.05×	1.09×
		100%	1.75×	1.19×	1.00×	1.19×	1.12×	1.17×
UNO	PRA	50%	**3.98×**	**6.59×**	**8.29×**	**1.53×**	**1.88×**	**1.35×**
		100%	**19.29×**	**25.55×**	**24.22×**	**2.48×**	**2.23×**	**1.71×**
	Random Attack	50%	1.12×	0.95×	0.82×	1.04×	1.05×	0.94×
		100%	1.54×	0.98×	0.77×	1.11×	1.15×	0.91×
	SPSA	50%	–	–	1.11×	–	–	1.01×
		100%	–	–	1.82×	–	–	0.99×
	DE	50%	1.28×	1.18×	1.83×	1.01×	0.97×	0.78×
		100%	2.04×	1.70×	4.30×	1.11×	0.93×	0.71×
UPT	PRA	50%	**5.90×**	**6.86×**	**6.54×**	**1.52×**	**1.41×**	**1.36×**
		100%	**13.43×**	**14.43×**	**13.08×**	**2.21×**	**1.85×**	**1.40×**
	Random Attack	50%	1.05×	0.98×	1.04×	0.85×	0.77×	0.81×
		100%	1.20×	1.09×	1.09×	0.85×	0.58×	0.62×
	SPSA	50%	–	–	1.85×	–	–	1.04×
		100%	–	–	3.30×	–	–	1.08×
	DE	50%	1.03×	1.00×	1.09×	1.15×	1.04×	1.10×
		100%	1.77×	1.77×	1.76×	1.36×	1.08×	1.17×



(a) NS Equation (1e-3) on Transolver (b) NS Equation (1e-4) on FNO

Figure 3: Qualitative comparison of PRA and baseline attacks on Navier-Stokes. (a) Transolver at $\nu = 10^{-3}$; (b) FNO at $\nu = 10^{-4}$. PRA induces substantially larger structural changes in the model-solver response, whereas the baseline perturbations are dominated by finer-scale artifacts.

Black-box baselines. In addition to Random Attack, we compare PRA with two query-based black-box attacks: SPSA (Uesato et al., 2018), which performs zeroth-order optimization in the full input space, and Differential Evolution (DE), following the sparse gradient-free attack of Roy et al. (2026). All methods are evaluated under the same solver-referenced protocol and perturbation budgets for each PDE.

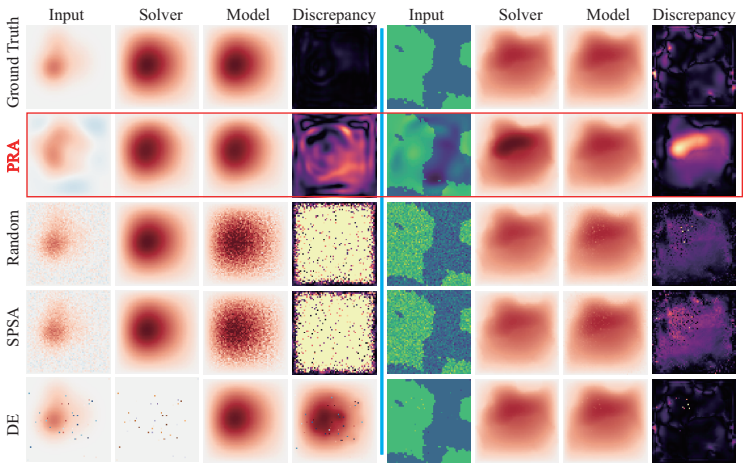


Figure 4: Qualitative comparison of the static PDEs. (a) FNO on Poisson; (b) Transolver on Darcy. PRA and the baseline attacks produce qualitatively different model-solver discrepancies under the same perturbation budgets.

4.2 EXPERIMENTAL RESULTS

4.2.1 NS EQUATION

As shown in Table 1, PRA achieves the strongest amplification in 44 of the 48 model-budget-horizon settings. Its advantage is particularly pronounced for FNO, UNO, and UPT at $\nu = 10^{-3}$, where the prediction errors under the baseline attacks generally remain close to the clean prediction error while PRA produces substantially larger model–solver discrepancies.

Transolver at $\nu = 10^{-3}$ is the main exception, where DE achieves comparable or larger scalar amplification in several settings. However, Fig. 3(a) shows a clear qualitative difference: the baseline attacks mainly introduce high-frequency artifacts while largely preserving the large-scale flow structure, whereas PRA produces a coherent deviation of the predicted trajectory from the reference dynamics. Thus, large scalar amplification does not necessarily correspond to a structural model-solver mismatch.

At $\nu = 10^{-4}$, amplification decreases substantially for FNO, UNO, and UPT. This behavior is consistent with the weaker viscous dissipation in this regime, where perturbations can persist and interact across finer spatial scales. Nevertheless, PRA remains the strongest attack across all models, budgets, and horizons in this regime. As shown in Fig. 3, PRA also induces more pronounced structural changes in the predicted response than the baseline attacks, whose discrepancies are dominated by finer-scale perturbations. Overall, these results indicate that PRA's effectiveness depends on the underlying physical regime, while the solver-informed subspace remains effective at identifying model-solver response discrepancies across different viscosities.

PRA's predominantly low-frequency subspace yields smooth perturbations, limiting the effectiveness of standard defenses such as low-pass, Gaussian, and median filtering; detailed results appear in Appendix A.2.

4.2.2 POISSON AND DARCY EQUATIONS

As shown in Table 2, PRA achieves the strongest amplification in 11 of the 12 Darcy settings, with particularly large gains on FNO and Transolver.

On Poisson, the difference in amplification between PRA and the baseline attacks is closely related to the operator's smoothing property. For $-\Delta u = f$, high-frequency components of the forcing are strongly attenuated by the solver. Consequently, high-frequency perturbations can leave the reference solution nearly unchanged while still causing large prediction errors in the neural operator. Random Attack, SPSA, and DE can therefore achieve large amplification by exploiting such noise-

Table 2: Attack amplification of PRA, Random Attack, SPSA, and DE on Darcy flow and the Poisson equation under different perturbation budgets. E_{clean} denotes the clean prediction error, and the remaining values report the amplification ratio relative to the clean error; bold marks the strongest attack for each model and budget. Darcy budgets are defined in log-permeability space as $\|\delta\|_2 \le c\sigma\sqrt{n}$, where $\sigma = \text{std}(\log a_0)$ and $n = 4096$. †Following Roy et al. (2026), Darcy DE perturbs $k = 41$ entries by at most $\pm\sigma$, giving $\|\delta\|_2 \le \sqrt{k}\sigma \approx 6.4\sigma$, below even the 25% budget of 16σ. Hence, the global L_2 budget is inactive for DE, and the same result holds for all three budgets.

PDE	Model	E_{clean}	PRA			Random Attack			SPSA			DE		
			25%	50%	100%	25%	50%	100%	25%	50%	100%	25%	50%	100%
Darcy	UNO	4.41×10^{-2}	**2.81**×	**5.77**×	10.36×	2.61×	4.86×	**10.65**×	2.56×	4.76×	10.12×	1.48×†	1.48×†	1.48×†
	FNO	3.38×10^{-2}	**2.33**×	**11.01**×	**144.44**×	1.22×	2.76×	31.43×	1.25×	2.97×	37.32×	1.17×†	1.17×†	1.17×†
	Transolver	2.07×10^{-2}	**3.70**×	**7.85**×	**17.77**×	1.44×	3.79×	11.24×	1.38×	3.81×	12.48×	1.07×†	1.07×†	1.07×†
	UPT	2.20×10^{-2}	**5.63**×	**9.59**×	**17.15**×	2.29×	6.16×	15.73×	2.16×	6.29×	16.70×	0.92×†	0.92×†	0.92×†
Poisson	UNO	1.42×10^{-3}	2.74×	10.02×	37.72×	50.02×	99.60×	198.56×	50.99×	102.15×	205.90×	**59.76**×	**125.11**×	**256.31**×
	FNO	1.05×10^{-3}	2.70×	8.23×	26.20×	94.74×	188.96×	378.83×	97.11×	194.90×	396.21×	**134.98**×	**265.82**×	**510.91**×
	Transolver	2.05×10^{-3}	49.31×	65.73×	74.88×	38.41×	49.16×	55.13×	39.14×	51.31×	56.53×	**60.12**×	**89.09**×	**78.33**×
	UPT	2.23×10^{-2}	5.60×	9.04×	16.69×	4.24×	9.35×	16.35×	5.94×	12.27×	17.19×	**7.08**×	**14.88**×	**19.70**×

Table 3: Total queries per attack to construct one perturbation. PRA's counts include constructing its search subspace.

	Navier-Stokes			Poisson / Darcy		
Attack	**Solver calls**	**Solver frames**	**Model passes**	**Poisson solves**	**Darcy solves**	**Model passes**
PRA	497	3,068	720	72	153	72
SPSA (Uesato et al., 2018)	60	1,140	600	60	60	60
DE (Roy et al., 2026)	6,465	58,185	64,650	1	1	6,464

sensitive directions, whereas PRA searches within solver-informed response directions and focuses on discrepancies in the physical response.

This distinction is visible in Fig. 4. On Poisson, Random Attack and SPSA introduce substantial high-frequency noise into the model prediction and produce large pointwise errors. However, the underlying large-scale response remains largely consistent with the solver, and the resulting discrepancy is dominated by spatially scattered noise rather than a coherent structural deviation. In contrast, PRA produces a more organized model-solver discrepancy. A similar behavior is observed on Darcy, where PRA induces a response mismatch that is not captured by the baseline attacks. These results show that large scalar amplification does not necessarily imply a larger structural discrepancy between the neural operator and the reference solver.

4.2.3 Query cost

Table 3 reports the query cost per perturbation, including subspace construction. PRA builds its subspace using only forward solver evaluations, enabling reuse across neural surrogates for the same physical input. The compact search substantially reduces model queries relative to DE, especially on NS. Costs count simulated solver frames and autoregressive model steps for NS, and individual solver or model calls for static PDEs.

5 Conclusion

In this work, we proposed Physical Response Attack (PRA), a fully black-box, physics-informed framework for evaluating the adversarial robustness of neural operators. Using only forward queries to a reference numerical solver, PRA constructs a compact perturbation subspace from the governing PDE's physical response structure and searches within it for perturbations that expose discrepancies between the learned operator and the underlying physics, requiring neither model gradients nor a differentiable solver. Experiments on Poisson, Darcy, and Navier–Stokes equations show that PRA is effective across architectures and physical regimes, exposing model–solver discrepancies of up to $24.22\times$ the clean prediction error on Navier–Stokes at $T = 20$. These results highlight the importance of evaluating neural operators against the physical response of the perturbed problem, rather than relying solely on model sensitivity or prediction error.

REPRODUCIBILITY STATEMENT

The accompanying code includes all training, attack, and data-generation scripts, together with the experimental configurations needed to reproduce the reported results.

AI USE STATEMENT

The authors proposed the research idea. LLMs were used as assistive tools as described below; they are not authors, and the authors take full responsibility for all content in the paper and the supplementary material.

Idea organization and validation. LLMs helped the authors organize the idea and carry out its preliminary validation.

Literature search. LLMs helped search for related literature. The authors selected the references and are responsible for how they are cited.

Code. The reference PDE solvers were provided by the authors; an LLM-based coding assistant (Anthropic Claude, through Claude Code) organized and refactored them. The same assistant implemented the attack, the baselines, and the evaluation scripts (including the scripts that export raw result panels), and tested them, for example, by checking the solvers against the recorded solutions of each dataset and by re-running experiments to confirm that the reported numbers are reproducible. The authors specified the experiments and are responsible for all reported results.

Writing and tables. The paper was written by the authors. LLMs were used to polish the text (grammar, wording, and clarity) and to help construct some of the tables.

Review. LLMs were used to review the draft and the experiments and to raise comments. The authors decided whether to adopt each comment.

Figures. All figures were designed, composed, and finalized independently by the authors.

REFERENCES

Abolaji D. Adesoji and Pin-Yu Chen. Evaluating the adversarial robustness for fourier neural operators, 2022. URL https://arxiv.org/abs/2204.04259.

Benedikt Alkin, Andreas Fürst, Simon Schmid, Lukas Gruber, Markus Holzleitner, and Johannes Brandstetter. Universal physics transformers: A framework for efficiently scaling neural operators, 2025. URL https://arxiv.org/abs/2402.12365.

Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search, 2020. URL https://arxiv.org/abs/1912.00049.

Sohei Arisaka and Qianxiao Li. Accelerating legacy numerical solvers by non-intrusive gradient-based meta-solving, 2024. URL https://arxiv.org/abs/2405.02952.

Giancarlo Benettin, Luigi Galgani, Antonio Giorgilli, and Jean-Marie Strelcyn. Lyapunov characteristic exponents for smooth dynamical systems and for hamiltonian systems; a method for computing all of them. part 1: Theory. *Meccanica*, 15(1):9–20, 1980. doi: 10.1007/BF02128236.

Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models, 2018. URL https://arxiv.org/abs/1712.04248.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks, 2017. URL https://arxiv.org/abs/1608.04644.

Jianbo Chen, Michael I. Jordan, and Martin J. Wainwright. Hopskipjumpattack: A query-efficient decision-based attack, 2020. URL https://arxiv.org/abs/1904.02144.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2015. URL https://arxiv.org/abs/1412.6572.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations, 2021. URL https://arxiv.org/abs/2010.08895.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks, 2019. URL https://arxiv.org/abs/1706.06083.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks, 2016. URL https://arxiv.org/abs/1511.04599.

Md Ashiqur Rahman, Zachary E. Ross, and Kamyar Azizzadenesheli. U-no: U-shaped neural operators, 2023. URL https://arxiv.org/abs/2204.11127.

Samrendra Roy, Kazuma Kobayashi, Souvik Chakraborty, Rizwan-uddin, and Syed Bahauddin Alam. Adversarial vulnerabilities in neural operator digital twins: Gradient-free attacks on nuclear thermal-hydraulic surrogates, 2026. URL https://arxiv.org/abs/2603.22525.

Yifei Sun. Solver-integrated adversarial attacking and training of neural operators, 2026. URL https://arxiv.org/abs/2510.18989.

Jonathan Uesato, Brendan O'Donoghue, Aaron van den Oord, and Pushmeet Kohli. Adversarial risk and the dangers of evaluating against weak attacks, 2018. URL https://arxiv.org/abs/1802.05666.

Haixu Wu, Huakun Luo, Haowen Wang, Jianmin Wang, and Mingsheng Long. Transolver: A fast transformer solver for pdes on general geometries, 2024. URL https://arxiv.org/abs/2402.02366.

A APPENDIX

A.1 SUBSPACE ABLATION

To isolate the contribution of the solver-derived search subspace, we replace the finite-time Lyapunov subspace with a random Gaussian subspace of the same dimension, while keeping all subsequent stages of PRA unchanged, including multiscale response probing, coefficient optimization, perturbation budgets, and exact nonlinear validation.

As shown in Table 4, replacing the solver-derived subspace substantially reduces attack amplification on FNO, UNO, and UPT. At the 100% budget, the random-subspace variant achieves only $1.35\times$, $2.56\times$, and $1.95\times$ amplification, compared with $23.85\times$, $24.22\times$, and $13.08\times$ for PRA, respectively. Moreover, these amplification ratios remain much closer to those of Random Attack than to those of full PRA. This ablation therefore shows that the solver-derived subspace is a key source of PRA's effectiveness, rather than the gain being explained by the subsequent low-dimensional optimization alone. For Transolver, the gap is smaller, consistent with the weaker PRA advantage observed in the main experiments.

Table 4: Subspace ablation on Navier–Stokes at $\nu = 10^{-3}$ and $T = 20$. The solver-derived finite-time Lyapunov subspace in PRA is replaced by a random Gaussian subspace of the same dimension, while all subsequent stages remain unchanged.

Model	**PRA**		**Random Subspace**		**Random Attack**	
	50%	100%	50%	100%	50%	100%
Transolver	**2.51**×	**3.96**×	2.40×	2.43×	2.50×	2.43×
FNO	**8.74**×	**23.85**×	1.20×	1.35×	0.94×	0.92×
UNO	**8.29**×	**24.22**×	1.90×	2.56×	0.82×	0.77×
UPT	**6.54**×	**13.08**×	1.42×	1.95×	1.04×	1.09×

A.2 DEFENSE

Tables 5 and 6 report the defended-error decomposition for representative Navier–Stokes trajectories at $T = 20$. Each cell reports e_{total} (e_{model}, e_{filter}), normalized by the undefended clean error of the corresponding configuration. The *None* column therefore gives the undefended amplification, while the *clean* row characterizes the cost introduced by each defense.

Here, $e_{\text{total}} = \|M(F(w)) - S(w_9)\|$ measures the error against the true continuation, $e_{\text{model}} = \|M(F(w)) - S(F(w_9))\|$ measures the surrogate error relative to the exact continuation of the filtered state, and $e_{\text{filter}} = \|S(F(w_9)) - S(w_9)\|$ measures the physical change introduced by filtering, with $e_{\text{total}} \leq e_{\text{model}} + e_{\text{filter}}$. All attack directions are optimized against the undefended models (non-adaptive).

Table 5: Defense error decomposition on NS at $T = 20$ under the 100% perturbation budget. Each cell reports e_{total} (e_{model}, e_{filter}), normalized by the undefended clean error.

Setting	Input	None	Low-pass K=8	Gauss. σ=1	Gauss. σ=2	Median 3×3
FNO, $\nu = 10^{-3}$	clean	1.00	1.00 (1.00, 0.00)	1.90 (0.93, 1.41)	5.51 (1.42, 5.60)	1.07 (0.99, 0.17)
	PRA	23.85	23.85 (23.85, 0.00)	24.11 (23.38, 1.91)	25.29 (22.01, 7.50)	23.85 (23.80, 0.22)
	Random	0.92	0.89 (0.89, 0.00)	1.77 (0.81, 1.41)	5.30 (1.32, 5.47)	0.97 (0.90, 0.16)
	SPSA	1.83	1.83 (1.83, 0.00)	2.58 (1.76, 1.40)	5.88 (2.02, 5.52)	1.90 (1.82, 0.16)
	DE	1.00	0.89 (0.89, 0.00)	1.78 (0.90, 1.35)	5.20 (1.46, 5.34)	1.08 (1.00, 0.14)
FNO, $\nu = 10^{-4}$	clean	1.00	1.00 (0.97, 0.07)	1.00 (1.11, 1.29)	3.68 (1.59, 4.92)	0.95 (1.02, 0.14)
	PRA	1.92	1.92 (1.92, 0.05)	2.05 (1.87, 1.38)	4.06 (1.87, 4.93)	1.91 (1.90, 0.14)
	Random	1.17	1.16 (1.15, 0.03)	1.01 (1.29, 1.30)	3.50 (1.77, 4.87)	1.11 (1.19, 0.16)
	SPSA	1.16	1.15 (1.15, 0.02)	0.99 (1.21, 1.25)	3.44 (1.67, 4.80)	1.08 (1.15, 0.15)
	DE	1.17	1.20 (1.19, 0.03)	1.15 (1.29, 1.31)	3.62 (1.76, 4.93)	1.17 (1.23, 0.17)
Transolver, $\nu = 10^{-3}$	clean	1.00	1.01 (1.01, 0.00)	0.99 (0.96, 0.08)	1.04 (0.93, 0.31)	1.94 (1.93, 0.01)
	PRA	3.96	3.37 (3.37, 0.00)	3.11 (3.10, 0.09)	2.87 (2.84, 0.36)	4.10 (4.09, 0.01)
	Random	2.43	1.20 (1.20, 0.00)	1.48 (1.45, 0.08)	1.08 (0.96, 0.30)	2.06 (2.06, 0.01)
	SPSA	2.55	1.26 (1.26, 0.00)	1.46 (1.43, 0.08)	1.05 (0.93, 0.30)	2.47 (2.46, 0.01)
	DE	2.89	1.40 (1.40, 0.00)	1.86 (1.83, 0.08)	1.18 (1.06, 0.32)	2.06 (2.06, 0.01)
Transolver, $\nu = 10^{-4}$	clean	1.00	0.94 (0.91, 0.04)	1.00 (1.21, 0.74)	1.92 (2.10, 2.81)	1.68 (1.69, 0.08)
	PRA	4.27	4.61 (4.60, 0.05)	4.61 (4.00, 0.74)	4.81 (2.54, 2.74)	4.74 (4.63, 0.15)
	Random	1.98	1.43 (1.42, 0.03)	1.65 (1.40, 0.74)	2.44 (2.03, 2.83)	1.92 (1.88, 0.08)
	SPSA	1.69	1.27 (1.26, 0.02)	1.31 (1.16, 0.75)	2.08 (1.81, 2.79)	1.58 (1.54, 0.10)
	DE	1.86	0.99 (0.99, 0.03)	1.15 (0.95, 0.77)	2.19 (1.78, 2.87)	1.94 (1.92, 0.10)

Table 6: Defense error decomposition on NS at $T = 20$ under the 50% perturbation budget. Each cell reports e_{total} (e_{model}, e_{filter}), normalized by the undefended clean error.

Setting	Input	None	Low-pass K=8	Gauss. σ=1	Gauss. σ=2	Median 3×3
FNO, $\nu = 10^{-3}$	clean	1.00	1.00 (1.00, 0.00)	1.90 (0.93, 1.41)	5.51 (1.42, 5.60)	1.07 (0.99, 0.17)
	PRA	8.74	8.74 (8.74, 0.00)	9.23 (8.27, 1.81)	11.61 (7.11, 6.96)	8.78 (8.69, 0.23)
	Random	0.94	0.93 (0.93, 0.00)	1.83 (0.86, 1.43)	5.40 (1.37, 5.54)	1.00 (0.93, 0.16)
	SPSA	1.23	1.23 (1.23, 0.00)	2.13 (1.16, 1.43)	5.65 (1.56, 5.57)	1.31 (1.22, 0.17)
	DE	0.96	0.94 (0.94, 0.00)	1.85 (0.90, 1.41)	5.38 (1.44, 5.48)	1.05 (0.97, 0.15)
FNO, $\nu = 10^{-4}$	clean	1.00	1.00 (0.97, 0.07)	1.00 (1.11, 1.29)	3.68 (1.59, 4.92)	0.95 (1.02, 0.14)
	PRA	1.43	1.43 (1.44, 0.04)	1.45 (1.48, 1.28)	3.54 (1.89, 4.81)	1.40 (1.44, 0.15)
	Random	1.06	1.05 (1.03, 0.05)	1.00 (1.17, 1.30)	3.61 (1.70, 4.94)	1.00 (1.07, 0.15)
	SPSA	1.02	1.02 (1.00, 0.04)	0.92 (1.13, 1.28)	3.54 (1.64, 4.87)	0.95 (1.03, 0.16)
	DE	1.09	1.10 (1.07, 0.06)	1.10 (1.18, 1.31)	3.71 (1.66, 4.97)	1.05 (1.12, 0.16)
Transolver, $\nu = 10^{-3}$	clean	1.00	1.01 (1.01, 0.00)	0.99 (0.96, 0.08)	1.04 (0.93, 0.31)	1.94 (1.93, 0.01)
	PRA	2.51	1.74 (1.74, 0.00)	1.58 (1.55, 0.09)	1.37 (1.27, 0.34)	2.99 (2.99, 0.01)
	Random	2.50	1.08 (1.08, 0.00)	1.30 (1.27, 0.08)	1.06 (0.93, 0.30)	1.93 (1.93, 0.01)
	SPSA	2.63	1.10 (1.10, 0.00)	1.25 (1.23, 0.08)	1.04 (0.92, 0.30)	2.15 (2.15, 0.01)
	DE	2.82	1.15 (1.15, 0.00)	1.48 (1.45, 0.08)	1.10 (0.98, 0.31)	1.98 (1.98, 0.01)
Transolver, $\nu = 10^{-4}$	clean	1.00	0.94 (0.91, 0.04)	1.00 (1.21, 0.74)	1.92 (2.10, 2.81)	1.68 (1.69, 0.08)
	PRA	2.88	2.93 (2.93, 0.04)	2.91 (2.29, 0.76)	3.72 (1.61, 2.88)	3.70 (3.62, 0.11)
	Random	1.45	1.03 (1.01, 0.04)	1.29 (1.29, 0.73)	2.13 (1.97, 2.83)	1.64 (1.63, 0.08)
	SPSA	1.65	1.02 (0.99, 0.03)	1.12 (1.11, 0.74)	2.05 (1.84, 2.80)	1.66 (1.64, 0.09)
	DE	1.58	1.14 (1.13, 0.04)	1.33 (1.20, 0.76)	2.23 (1.84, 2.85)	1.75 (1.73, 0.08)

A.3 ALGORITHM

Algorithm 1 PRA for one reference condition x_0

Require: solver Φ, operator G_θ, condition x_0, horizon T, subspace size K, FD step h, sub-window τ, passes L, amplitudes $\{\epsilon_s\}_{s=1}^{S}$, budget ϵ

Step 1: Black-Box Lyapunov Subspace Estimation

$Q \leftarrow$ random orthonormal $n \times K$ matrix
for $\ell = 1, \dots, L$ **do**
  $r_0 \leftarrow x_0$
  for $k = 1, \dots, m$ **do**
    $r_k \leftarrow \Phi^\tau(r_{k-1}); \quad z_i \leftarrow \big(\Phi^\tau(r_{k-1} + h\, q_i) - r_k\big)/h, \quad i = 1, \dots, K$
    $Q, R \leftarrow \mathrm{QR}([z_1, \dots, z_K])$
  end for
end for
$V \leftarrow Q$

Step 2: Multiscale Model–Solver Response Comparison

$e_0 \leftarrow G_\theta^T(x_0) - \Phi^T(x_0)$
for $s = 1, \dots, S$ **do**
  $M_{\mathrm{phys}}^{(s)}[:, i] \leftarrow \big(\Phi^T(x_0 + \epsilon_s v_i) - \Phi^T(x_0)\big)/\epsilon_s, \quad i = 1, \dots, K$
  $M_{\mathrm{model}}^{(s)}[:, i] \leftarrow \big(G_\theta^T(x_0 + \epsilon_s v_i) - G_\theta^T(x_0)\big)/\epsilon_s$
  $B_s \leftarrow M_{\mathrm{model}}^{(s)} - M_{\mathrm{phys}}^{(s)}$
end for

Step 3: Solver-Referenced Response Optimization

$H \leftarrow \frac{1}{S} \sum_s \epsilon_s^2 B_s^\top B_s; \quad b \leftarrow \frac{1}{S} \sum_s \epsilon_s B_s^\top e_0$
$a^\star \leftarrow \arg\max_{\|a\|_2 = 1} \; a^\top H a + 2 b^\top a$ ▷ multi-start projected gradient ascent on the unit sphere
$a^\star \leftarrow \arg\max_{\pm a^\star} \rho(\pm a^\star)$ ▷ sign check by exact nonlinear rollouts
return $\delta_{\mathrm{PRA}} = \epsilon\, V a^\star$

with $\rho(a) = E(x_0 + \epsilon V a)/E(x_0)$ and $E(x) = \|G_\theta^T(x) - \Phi^T(x)\|_2 / \|\Phi^T(x)\|_2$