CONFLICT-AWARE PHYSICS-INFORMED NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Due to the mesh-free property and the capability to directly embed physical laws into deep learning models, Physics-Informed Neural Networks (PINNs) have become a promising tool for solving partial differential equations (PDEs). Although these methods have made significant progress, gradients associated with different loss terms can differ substantially in both magnitude and direction, leading to gradient conflicts and difficulties in stable and efficient optimization. While existing balancing methods can mitigate such conflicts to some extent, they typically rely on predefined loss reweighting or gradient manipulation rules that do not explicitly adapt to the evolving conflict state during training. In this paper, we propose **C**onflict-**A**ware **P**hysics-**I**nformed **N**eural **N**etworks (CA-PINN), which introduces an Adaptive Conflict Modulation mechanism to dynamically adjust the strength of conflict mitigation during training. Specifically, we characterize the conflict state from two complementary perspectives: directional disagreement and gradient-norm imbalance. Adaptive Conflict Modulation then uses the estimated conflict state to dynamically regulate the bounded correction range, thereby balancing conflict mitigation with optimization progress. Extensive experiments on five PDE benchmarks demonstrate that CA-PINN achieves competitive and consistent predictive accuracy across different physical systems and loss decompositions.

1 INTRODUCTION

Partial differential equations (PDEs) provide fundamental mathematical tools for describing complex physical phenomena in science and engineering. Since analytical solutions are often unavailable in practical scenarios, developing accurate and efficient PDE solvers has become a long-standing research problem. Traditional numerical methods, such as finite difference, finite element, and finite volume methods Gunzburger et al. (2014); Lynch & Rice (1978); Eymard et al. (2000), have achieved remarkable success, but they usually suffer from heavy computational costs when handling high-dimensional or complex systems. Benefiting from the expressive power of deep neural networks, Physics-Informed Neural Networks (PINNs) Raissi et al. (2019); Karniadakis et al. (2021); Lu et al. (2021); Cuomo et al. (2022); Beck et al. (2020) have recently emerged as a promising paradigm for PDE solving. By embedding governing equations together with initial and boundary conditions into the training objective, PINNs provide a flexible and mesh-free framework that integrates physical priors with data-driven learning.

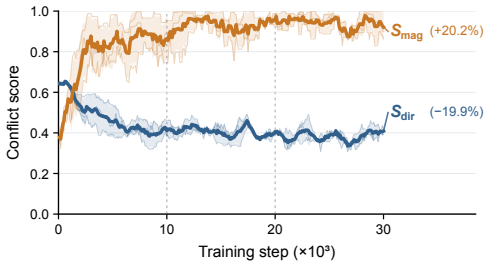


Figure 1: Evolution of directional disagreement and gradient-norm imbalance during training on the Burgers equation.

In recent years, a large number of PINN methods have been proposed to improve training stability and predictive accuracy Wang et al. (2021; 2022); Krishnapriyan et al. (2021). Broadly speaking, existing studies have improved PINNs from multiple perspectives, including adaptive loss weighting Wang et al. (2021); McClenny & Braga-Neto (2023); Xiang et al. (2022); Van Der Meer et al. (2022);

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.

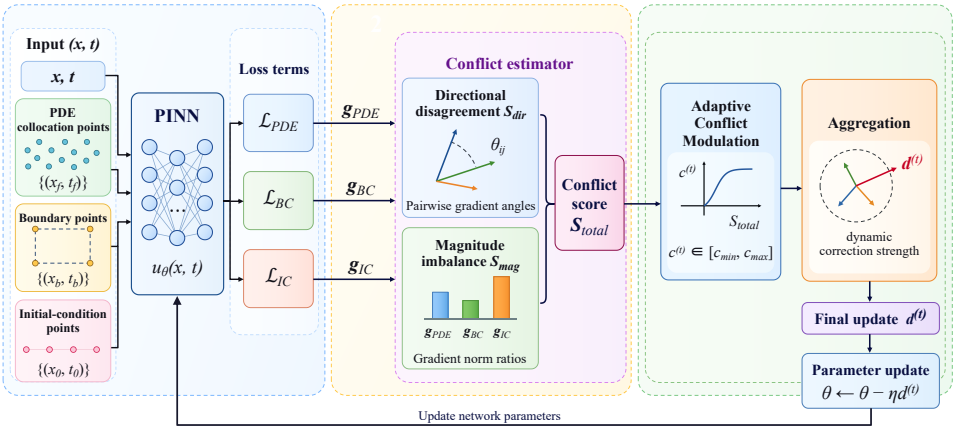


Figure 2: The structure of CA-PINN: the network maps inputs (x, t) to u_θ; gradients from L_{PDE}, L_{BC} and L_{IC} are used to estimate the evolving conflict state, which drives Adaptive Conflict Modulation to regulate the strength of the bounded conflict-aware update.

Bischof & Kraus (2025), constraint relaxation Son et al. (2023), residual-gradient enhancement Yu et al. (2022), network architecture design Jagtap et al. (2020); Jagtap & Karniadakis (2020); Cho et al. (2023), and gradient-based optimization strategies Yu et al. (2020); Hwang & Lim (2024); Liu et al. (2024). Although these methods have achieved encouraging progress, PINN training still suffers from a critical optimization bottleneck Bajaj et al. (2023). Since the PDE residual, boundary conditions, and initial conditions correspond to heterogeneous physical objectives, their gradients often differ substantially in both magnitude and direction Bischof & Kraus (2025); Xiang et al. (2022); Wang et al. (2020). As a result, directly optimizing the aggregated loss may lead the network to make progress on some objectives while sacrificing others, eventually producing inaccurate or even physically inconsistent solutions. Therefore, the gradient conflict among different loss terms has become one key challenge for high-precision PINN training.

Gradient conflicts in PINNs arise from both gradient-norm imbalance and directional disagreement. Existing methods address these issues through adaptive loss weighting Chen et al. (2018) or by modifying the update direction to reduce destructive interference Yu et al. (2020); Liu et al. (2021b;a; 2024); Hwang & Lim (2024). Wang et al. (2026) further analyze normalized gradient alignment from a second-order optimization perspective and motivate quasi-second-order preconditioning. While these approaches can mitigate gradient conflicts, the strength of their intervention is generally governed by predefined weighting or update rules, rather than explicitly responding to how the conflict state evolves during training. For example, as shown in Figure 1, directional disagreement decreases as training progresses on Burgers, whereas gradient-norm imbalance increases. This divergent evolution indicates that the optimization state cannot be adequately characterized by a single conflict signal. It therefore motivates a state-aware strategy that jointly characterizes both properties and adapts the strength of conflict mitigation throughout training.

Based on this observation, we propose **Conflict-Aware Physics-Informed Neural Networks (CA-PINN)**. As shown in Figure 2, CA-PINN couples conflict-state estimation with Adaptive Conflict Modulation. At each iteration, the current optimization state is characterized from two complementary aspects: pairwise directional disagreement and gradient-norm imbalance. These quantities are combined into a conflict score, which is mapped to a bounded coefficient and smoothed over successive iterations. The resulting coefficient regulates the allowable deviation from the average gradient and therefore controls the strength of conflict-aware correction. In this way, CA-PINN adapts the intervention strength to the evolving PINN loss geometry instead of prescribing a fixed correction throughout training. The major contributions of this paper are as follows:

- We reveal that gradient conflict in PINN training is jointly manifested in directional disagreement and gradient-norm imbalance, and that these two aspects can evolve differently throughout optimization.

- Based on this observation, we propose CA-PINN, a state-aware conflict-mitigation framework that characterizes the evolving conflict state from both directional and magnitude perspectives. We further introduce Adaptive Conflict Modulation to map the estimated state to a bounded and temporally smoothed coefficient, thereby dynamically regulating the strength of the conflict-aware update during training.
- We establish the descent property of the resulting bounded time-varying update under the direct-gradient setting, and evaluate CA-PINN on five PDE benchmarks with different loss decompositions. The results, together with optimization diagnostics and ablation studies, demonstrate consistent predictive performance and verify the complementary roles of directional disagreement and gradient-norm imbalance.

2 Related work

2.1 Physics-informed neural networks

Physics-informed neural networks (PINNs) approximate the PDE solution $u(\mathbf{x}, t)$ by a neural network $u_\theta(\mathbf{x}, t)$. Consider a partial differential equation in the following general form:

$$\begin{cases} u_t = \mathcal{N}(u), & x \in \Omega, t \in \Gamma; \\ u(x,0) = u_0(x), & x \in \Omega; \\ u(x,t) = g(x,t), & x \in \partial\Omega, t \in \Gamma. \end{cases} \tag{1}$$

Here, $u(x,t)$ denotes the unknown solution of the PDE, and $\mathcal{N}(u)$ represents a differential operator. The spatial variable x belongs to the physical domain $\Omega \subset \mathbb{R}^d$, and $t \in [0, T]$ denotes time. The function $u_0(x)$ specifies the initial condition at $t = 0$, while $g(x,t)$ prescribes the boundary condition on the boundary $\partial\Omega$ over the time interval Γ.

PINNs enforce these governing equations and constraints through a composite loss function that consists of three components: a PDE residual term evaluated at interior collocation points, an initial condition term, and a boundary condition term. Specifically, the loss functions are defined as:

$$L_{\text{PDE}} = \frac{1}{N_r} \sum_{i=1}^{N_r} \left(\frac{\partial u_\theta}{\partial t}(x_i^r, t_i^r) - \mathcal{N}\big(u_\theta(x_i^r, t_i^r)\big) \right)^2, \tag{2}$$

$$L_{\text{IC}} = \frac{1}{N_{\text{IC}}} \sum_{i=1}^{N_{\text{IC}}} \left(u_\theta(x_i^{\text{IC}}, 0) - u_0(x_i^{\text{IC}}) \right)^2, \tag{3}$$

$$L_{\text{BC}} = \frac{1}{N_{\text{BC}}} \sum_{i=1}^{N_{\text{BC}}} \left(u_\theta(x_i^{\text{BC}}, t_i^{\text{BC}}) - g(x_i^{\text{BC}}, t_i^{\text{BC}}) \right)^2. \tag{4}$$

where $\{(x_i^r, t_i^r)\}_{i=1}^{N_r}$ are collocation points for the PDE residual, $\{x_i^{\text{IC}}\}_{i=1}^{N_{\text{IC}}}$ are points for the initial condition, and $\{(x_i^{\text{BC}}, t_i^{\text{BC}})\}_{i=1}^{N_{\text{BC}}}$ are boundary points. The total loss is then given by a weighted combination:

$$L(\theta) = \lambda_{\text{PDE}} L_{\text{PDE}} + \lambda_{\text{IC}} L_{\text{IC}} + \lambda_{\text{BC}} L_{\text{BC}} + \lambda_{\text{data}} L_{\text{data}}. \tag{5}$$

where $\lambda_{\text{PDE}}, \lambda_{\text{IC}}, \lambda_{\text{BC}}$ are weighting coefficients. Existing studies improve PINNs from four main perspectives: loss weighting strategies, gradient-based optimization algorithms, network architectures, and sampling strategies.

Despite these advancements, many existing methods do not explicitly adapt the strength of gradient-conflict intervention to the evolving optimization state of PINNs. Traditional weighting schemes and gradient-based approaches often rely on static or heuristic adjustments that are agnostic to the changing interplay between PDE residuals and physical constraints. This lack of adaptability can limit their ability to respond when the direction and scale of loss-specific gradients change during training.

2.2 Multi-task learning

Multi-task learning (MTL) Ruder (2017) aims to enhance generalization and learning efficiency by training a single model to address multiple related tasks simultaneously. A fundamental challenge in

MTL is gradient conflict Yu et al. (2020), which occurs when gradients originating from different task objectives point in opposing directions. Such interference results in parameter updates that benefit one task at the expense of others, a phenomenon known as negative transfer. This conflict often leads to suboptimal performance, potentially underperforming compared to independently trained single-task models.

Training PINNs can be viewed as a multi-objective optimization problem in which the PDE, boundary, and initial losses are optimized together Wang et al. (2021); Bischof & Kraus (2025). Their gradients may disagree in direction and scale Liu et al. (2024); Li et al. (2023); Vemuri & Denzler (2023). Existing approaches include gradient projection Yu et al. (2020), impartial aggregation Liu et al. (2021b), CAGrad Liu et al. (2021a), sign-consistency masking Chen et al. (2020), bargaining-based aggregation Navon et al. (2022), and adaptive task balancing Liu et al. (2023).

Gradient-based optimization methods mitigate interference among objectives through mechanisms such as gradient projection, impartial aggregation, sign-consistency constraints, bargaining-based updates, and bounded multi-objective optimization Yu et al. (2020); Liu et al. (2021b;a); Chen et al. (2020); Navon et al. (2022). These approaches suggest that gradient conflict can be handled not only by reweighting individual objectives, but also by regulating the geometry of the shared update. Meanwhile, Wang et al. (2026) study gradient alignment in PINNs from a second-order optimization perspective. Inspired by the general principle of controlling the admissible update region, our work focuses on a different question: how to estimate the evolving conflict state of physical objectives and use it to adapt the intervention strength throughout PINN training.

3 METHOD

3.1 OVERVIEW

Gradient conflicts among heterogeneous physical objectives evolve throughout PINN training. In particular, directional disagreement and gradient-norm imbalance may follow different trajectories, making a fixed conflict-mitigation strength difficult to adapt to the changing optimization state. To address this issue, we propose Conflict-Aware Physics-Informed Neural Networks (CA-PINN), a state-aware optimization framework that dynamically adjusts the strength of conflict mitigation according to the evolving gradient geometry.

CA-PINN consists of three components. First, a conflict-state estimator characterizes the current optimization state from directional disagreement and gradient-norm imbalance. Second, Adaptive Conflict Modulation maps the estimated state to a bounded and temporally smoothed coefficient $c^{(t)}$. Third, this coefficient regulates a bounded conflict-aware update around the average gradient, balancing shared descent with conflict correction.

The following sections first introduce the bounded conflict-aware update, then describe conflict-state estimation and Adaptive Conflict Modulation, and finally analyze the descent property of the resulting time-varying update.

3.2 CONFLICT-AWARE CONSTRAINED UPDATE

We begin by revisiting the optimization structure of PINNs. In general, PINN training can be formulated as a multi-objective optimization problem over M physical objectives, where each loss term corresponds to a distinct constraint, such as the PDE residual, boundary condition, or initial condition:

$$\boldsymbol{\mathcal{L}}(\theta) = \left[\mathcal{L}_1(\theta), \ldots, \mathcal{L}_M(\theta)\right]^\top. \tag{6}$$

For each objective, we denote its gradient by $\mathbf{g}_i = \nabla_\theta \mathcal{L}_i(\theta)$.

A central difficulty is that these physical objectives are not always cooperative. Their gradients can disagree in direction and differ substantially in magnitude, so directly averaging or statically combining them may cause destructive interference. Existing gradient-based approaches address this issue through loss reweighting, gradient projection, or constrained update rules. These strategies suggest that conflict mitigation can be viewed as controlling how much the update is allowed to deviate from the common descent tendency of the physical objectives.

Inspired by bounded-neighborhood formulations for multi-objective gradient optimization Liu et al. (2021a), we formulate a conflict-aware update that promotes balanced descent while restricting excessive deviation from the average gradient:

$$\max_{\mathbf{d}} \min_{i\in\{1,\dots,M\}} \mathbf{g}_i^\top \mathbf{d} \quad \text{s.t.} \quad \|\mathbf{d}-\mathbf{g}_0\| \le c\|\mathbf{g}_0\|, \tag{7}$$

where $\mathbf{g}_0 = \frac{1}{M}\sum_{i=1}^{M}\mathbf{g}_i$ denotes the average gradient.

The bounded neighborhood serves as a controllable interface between shared descent and conflict correction. A smaller c keeps the update close to the average gradient, whereas a larger c allows greater deviation when stronger intervention is needed. Rather than prescribing this correction strength throughout training, CA-PINN determines it from the current gradient-conflict state, as described in Sections 3.3.3 and 3.4.

For efficient computation, the constrained problem can be expressed through the following dual formulation:

$$\min_{\mathbf{w}\in\Delta^M} F(\mathbf{w}) = \mathbf{g}_\mathbf{w}^\top \mathbf{g}_0 + c\|\mathbf{g}_0\|\|\mathbf{g}_\mathbf{w}\|, \tag{8}$$

where $\mathbf{g}_\mathbf{w} = \sum_{i=1}^{M} w_i \mathbf{g}_i$ is a convex combination of the objective gradients and Δ^M denotes the probability simplex. The first term evaluates support from the average descent direction, whereas the second controls the allowable conflict correction within the bounded neighborhood. Minimizing this objective yields the optimal dual weights $\mathbf{w}^*$. Given a modulation coefficient c, the resulting conflict-aware update direction is constructed as

$$\mathbf{d} = \mathbf{g}_0 + c\|\mathbf{g}_0\|\frac{\mathbf{g}_{\mathbf{w}^*}}{\|\mathbf{g}_{\mathbf{w}^*}\|+\epsilon}, \tag{9}$$

where $\epsilon > 0$ is a small constant for numerical stability. In CA-PINN, c is instantiated by the adaptive coefficient $c^{(t)}$ introduced in Section 3.4. For $M = 2$, we write $\mathbf{w} = [\gamma, 1-\gamma]^\top$ with $\gamma \in [0,1]$, reducing the dual optimization to a one-dimensional search. For $M > 2$, the dual variables are optimized iteratively.

3.3 Conflict-aware score estimation

The conflict state is designed to reflect two properties of the current physical-loss gradients. Directional disagreement describes whether the objectives compete geometrically, while norm imbalance describes whether one objective dominates the update by scale. These two effects can evolve differently during optimization; for example, the gradients may become more aligned while their norms become increasingly imbalanced. We therefore model them separately and combine them to determine the strength of conflict-aware correction.

3.3.1 Directional conflict score

We first measure directional conflict, since disagreement in gradient direction is the most direct source of destructive interference among objectives. If two loss terms produce gradients that point toward substantially different directions, improving one objective may immediately hinder another. This phenomenon is especially common in PINNs, where PDE residuals and constraint losses often induce heterogeneous optimization signals.

To quantify this effect, we compute the pairwise cosine similarity matrix $\mathbf{C} \in \mathbb{R}^{M\times M}$ over normalized gradients. For each loss i, we define

$$\hat{\mathbf{g}}_i = \frac{\mathbf{g}_i}{\|\mathbf{g}_i\|+\epsilon}, \tag{10}$$

and compute

$$C_{ij} = \hat{\mathbf{g}}_i^\top \hat{\mathbf{g}}_j. \tag{11}$$

Based on this matrix, we define the directional conflict score as

$$S_{\text{dir}} = \frac{1}{M(M-1)}\sum_{i\neq j}\frac{\arccos(C_{ij})}{\pi}. \tag{12}$$

A smaller S_{dir} indicates that the gradients are largely aligned, whereas a larger value implies stronger directional disagreement and, consequently, more severe optimization conflict.

3.3.2 MAGNITUDE CONFLICT SCORE

Directional agreement alone does not guarantee balanced optimization. Even when gradients point in roughly similar directions, a large disparity in their magnitudes may still cause the optimization process to be dominated by a single objective. In PINNs, this issue frequently arises because different physical losses naturally operate on different scales.

To capture this imbalance, we measure the spread of gradient norms as

$$r_{\mathrm{mag}} = \frac{\max_i \|\mathbf{g}_i\|}{\min_i \|\mathbf{g}_i\| + \epsilon}, \tag{13}$$

and convert it into a normalized magnitude conflict score:

$$S_{\mathrm{mag}} = \min\left(1, \max\left(0, \frac{r_{\mathrm{mag}} - 1}{\kappa}\right)\right), \tag{14}$$

where κ is a sensitivity threshold and ϵ is a small constant for numerical stability. In this way, larger values of S_{mag} correspond to stronger magnitude imbalance and a higher risk that the optimizer will disproportionately follow the largest-gradient objective.

3.3.3 TOTAL CONFLICT SCORE

The two conflict types above are complementary: directional conflict reflects whether objectives compete geometrically, while magnitude conflict reflects whether they compete in scale. Since both factors can degrade PINNs optimization, we combine them into a unified conflict indicator:

$$S_{\mathrm{total}}^{(t)} = \lambda_{\mathrm{dir}} S_{\mathrm{dir}}^{(t)} + (1 - \lambda_{\mathrm{dir}}) S_{\mathrm{mag}}^{(t)}. \tag{15}$$

In our experiments, we set $\lambda_{\mathrm{dir}} = 0.8$, which places slightly greater emphasis on directional disagreement while still accounting for scale imbalance.

3.4 ADAPTIVE CONFLICT MODULATION

The combined score is used to update the modulation coefficient $c^{(t)}$. A larger score permits a larger deviation from the average gradient, whereas a smaller score keeps the update closer to the average direction. We map the score to the target modulation coefficient as

$$c_{\mathrm{target}}^{(t)} = c_{\min} + (c_{\max} - c_{\min}) \cdot S_{\mathrm{total}}^{(t)}. \tag{16}$$

A larger c enlarges the search radius $R = c\|\mathbf{g}_0\|$, allowing the optimizer to move further away from the average gradient and search for an update direction that better reconciles the competing physical objectives.

By contrast, when the measured conflict is weak, excessive deviation from $\mathbf{g}_0$ is unnecessary and may even slow convergence. In this case, reducing c shrinks the feasible region toward the average descent direction, so that the optimizer behaves more like standard gradient descent and enjoys faster progress. To avoid abrupt oscillations caused by transient conflict fluctuations, we further smooth the target value using an exponential moving average:

$$c^{(t)} = \alpha_{\mathrm{EMA}} c_{\mathrm{target}}^{(t)} + (1 - \alpha_{\mathrm{EMA}}) c^{(t-1)}. \tag{17}$$

This temporal smoothing stabilizes the modulation process and makes the transition between low-conflict and high-conflict regimes more gradual.

Mean-gradient cancellation analysis. The bounded update radius is defined relative to the mean-gradient norm as $R^{(t)} = c^{(t)}\|\mathbf{g}_0^{(t)}\|$. To examine the regime in which individual gradients partially cancel, we define

$$\rho_t = \frac{\|\mathbf{g}_0^{(t)}\|}{M^{-1}\sum_i \|\mathbf{g}_i^{(t)}\| + \epsilon}, \qquad \widetilde{R}_t = c^{(t)}\rho_t. \tag{18}$$

A small ρ_t indicates stronger cancellation among the loss-specific gradients. The behavior of $c^{(t)}$ and the normalized radius in this regime is examined empirically in Appendix C.

For computational efficiency, we use different solvers depending on the number of objectives. When $M = 2$, $\mathbf{w} = [\gamma, 1-\gamma]^\top$ and the optimal $\gamma \in [0,1]$ is found by a vectorized one-dimensional grid search. For $M > 2$, we solve the dual problem by gradient descent and initialize $\mathbf{w}$ with the previous-step solution $\mathbf{w}^{(t-1)}$. This warm-start strategy reduces the inner-loop optimization cost in practice.

3.5 Descent property under bounded time-varying modulation

The bounded-neighborhood formulation naturally extends to a time-varying modulation coefficient. Let $\mathcal{L}_0(\theta) = M^{-1}\sum_i \mathcal{L}_i(\theta)$ and $\mathbf{g}_0^{(t)} = \nabla\mathcal{L}_0(\theta_t)$. Substituting $c = c^{(t)}$ into Eq. 9, the ϵ-stabilized correction has norm no larger than $c^{(t)}\|\mathbf{g}_0^{(t)}\|$. Therefore, the update direction satisfies

$$\|\mathbf{d}^{(t)} - \mathbf{g}_0^{(t)}\| \le c^{(t)}\|\mathbf{g}_0^{(t)}\|. \tag{19}$$

If $c^{(t)} \le c_{\max} < 1$ for every iteration, then

$$(\mathbf{g}_0^{(t)})^\top \mathbf{d}^{(t)} \ge (1-c_{\max})\|\mathbf{g}_0^{(t)}\|^2, \qquad \|\mathbf{d}^{(t)}\| \le (1+c_{\max})\|\mathbf{g}_0^{(t)}\|. \tag{20}$$

For a deterministic, L-smooth, lower-bounded $\mathcal{L}_0$ and the idealized direct update $\theta_{t+1} = \theta_t - \eta\mathbf{d}^{(t)}$, the descent lemma yields

$$\mathcal{L}_0(\theta_{t+1}) \le \mathcal{L}_0(\theta_t) - \left[\eta(1-c_{\max}) - \frac{L\eta^2}{2}(1+c_{\max})^2\right]\|\mathbf{g}_0^{(t)}\|^2. \tag{21}$$

Thus, if

$$0 < \eta < \frac{2(1-c_{\max})}{L(1+c_{\max})^2}, \tag{22}$$

then $\sum_t \|\mathbf{g}_0^{(t)}\|^2 < \infty$ and hence $\|\mathbf{g}_0^{(t)}\| \to 0$. Thus, the direct-gradient descent argument permits any bounded history-dependent sequence $c^{(t)}$, including our EMA controller.

4 Experiments

4.1 Experimental setup

4.1.1 PDE benchmarks

We evaluate CA-PINN on five widely used PDE benchmarks with different gradient-conflict characteristics: the 1D Burgers equation, the Schrödinger equation, the Heat equation, the Kovasznay flow, and the 3D Beltrami flow. These benchmarks cover shock waves, high-frequency oscillations, diffusion dynamics, steady-state vortex structures, and complex three-dimensional flow patterns.

Beyond the diversity of physical dynamics, these benchmarks also involve different constraint structures, making it necessary to examine CA-PINN under different loss decompositions. We therefore consider two loss configurations. In the two-term setting, the boundary and initial losses are merged as $\mathcal{L}_{\mathcal{BI}} = \mathcal{L}_{\mathcal{B}} + \mathcal{L}_{\mathcal{I}}$, which highlights the conflict between the PDE residual $\mathcal{L}_{\mathcal{N}}$ and data constraints. In the three-term setting, the PDE residual, boundary condition, and initial condition are optimized as three independent objectives. The Kovasznay flow is steady-state and has no initial condition, so it only uses the two-term configuration. Detailed network architectures, training schedules, sampling sizes, evaluation protocols, optimizer settings, random seeds, and CA-PINN hyperparameters are provided in Appendix B.

4.1.2 Baselines

We compare CA-PINN with Adam Kingma & Ba (2014), PCGrad Yu et al. (2020), IMTL-G Liu et al. (2021b), MinMax Liu & Wang (2021), ReLoBRaLo Bischof & Kraus (2025), ConFIG Liu et al. (2024), and M-ConFIG Liu et al. (2024). An additional fixed-coefficient control comparison is provided in Appendix C.

Table 1: Mean squared error (MSE) on the five PDE benchmarks. The scale factor is reported in the "Scale" column, and each value should be multiplied by the corresponding factor. Symbols ① and ② denote the two-term and three-term loss configurations, respectively.

Equation	Scale	Loss	Adam	PCGrad	IMTL-G	MinMax	ReLoBRaLo	ConFIG	M-ConFIG	CA-PINN
			ICLR'15	NeurIPS'20	ICLR'21	NN'21	CMAME'25	ICLR'25	ICLR'25	Ours
Schrödinger	$\times 10^{-4}$	①	3.383	1.344	5.118	42.364	3.017	0.710	4.292	**0.562**
		②	3.383	22.478	172.497	42.311	3.826	1.338	15.217	**0.660**
Burgers	$\times 10^{-4}$	①	1.484	1.344	1.339	1.889	1.419	1.301	1.282	**1.253**
		②	1.484	1.279	22.478	1.582	1.402	1.322	1.287	**1.275**
Beltrami	$\times 10^{-4}$	①	1.112	1.041	0.622	1.851	1.181	0.617	0.795	**0.559**
		②	1.112	0.756	1.032	2.215	0.949	0.571	0.910	**0.508**
Heat	$\times 10^{-7}$	①	1.673	5.854	1.263	5.589	1.939	1.507	1.873	**0.720**
		②	1.673	2.824	1.873	5.691	**0.564**	1.919	2.554	0.632
Kovasznay	$\times 10^{-7}$	①	1.044	1.036	**0.086**	7.743	0.987	0.424	**0.086**	0.117

4.1.3 Evaluation metrics

The prediction accuracy is quantitatively evaluated by the Mean Squared Error (MSE) between the predicted solution u_{pred} and the reference solution u_{ref}. Specifically, MSE measures the average squared discrepancy over all N evaluation points in the spatiotemporal domain:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left(u_{\text{pred}}(\mathbf{x}_i, t_i) - u_{\text{ref}}(\mathbf{x}_i, t_i)\right)^2. \tag{23}$$

For vector-valued or complex-valued solutions, the squared errors are first computed component-wise and then averaged over all components.

4.2 Main results

Table 1 reports the MSE results on the five PDE benchmarks. CA-PINN gives the lowest error on both Schrödinger settings, both Burgers settings, both Beltrami settings, and the two-term Heat setting. ReLoBRaLo is better on the three-term Heat setting, while IMTL-G and M-ConFIG obtain lower error on Kovasznay. Overall, CA-PINN remains competitive across the different PDEs and loss decompositions. Supplementary Relative L_2 results are reported in Appendix I.

Empirical convergence behavior. To further examine the optimization behavior of CA-PINN, we report the training-loss trajectories on the Beltrami flow benchmark under the $[\mathcal{L}_{\mathcal{N}}, \mathcal{L}_{\mathcal{BI}}]$ setting. As shown in Figure 3, CA-PINN exhibits a stable decrease in the training objective throughout optimization and reaches a low final training loss relative to the displayed baselines. More importantly, the trajectory does not show persistent oscillation or late-stage degradation, suggesting that dynamically adjusting the conflict-mitigation strength does not destabilize the underlying optimization process.

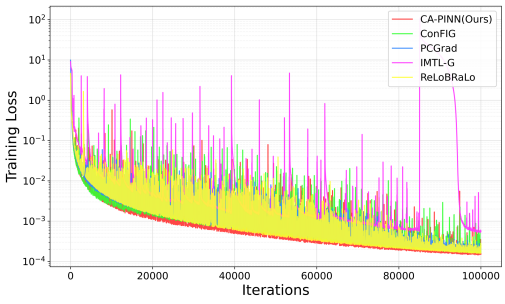


Figure 3: Empirical training-loss trajectories on the Beltrami flow benchmark.

This empirical behavior is consistent with the bounded time-varying update analyzed in Section 3.5. The adaptive coefficient is restricted to a bounded interval, which prevents the conflict-aware correction from deviating arbitrarily far from the average descent direction. The training trajectory therefore provides empirical evidence that CA-PINN can adapt the correction strength while maintaining stable optimization progress. We emphasize that the theoretical result in Section 3.5 concerns an idealized direct-gradient update and should not be interpreted as a convergence guarantee for Adam.

4.3 Optimization analysis

The conflict state does not remain stationary during training, but changes substantially as the different physical objectives are progressively optimized. Table 2 summarizes this evolution on Burgers with three loss terms. The directional conflict score decreases from 0.478 in the first training stage to 0.383 in the final stage, corresponding to a reduction of approximately 19.9%. In contrast, the magnitude conflict score increases from 0.784 to 0.941, an increase of approximately 20.0%. Meanwhile, the validation loss decreases from 1.47×10^{-3} to 1.80×10^{-4}.

Table 2: Evolution of the conflict state on Burgers with three loss terms.

Stage	S_{dir}	S_{mag}	S_{total}	$c^{(t)}$	r_{mag}
1–10k	.478	.784	.539	.518	13.5
10–20k	.408	.928	.512	.503	24.1
20–30k	.383	.941	.495	.499	24.7

This behavior supports the conflict-state characterization in CA-PINN. Directional disagreement decreases while magnitude imbalance increases, so neither signal alone fully reflects the evolving optimization state. As a result, S_{total} decreases only moderately from 0.539 to 0.495, and the modulation coefficient $c^{(t)}$ changes from 0.518 to 0.499, showing that the correction strength responds to their joint evolution.

4.4 Ablation study

Table 3 examines whether directional disagreement and gradient-norm imbalance provide redundant or complementary information for adaptive conflict modulation. CA-PINN-1 retains only magnitude conflict, CA-PINN-2 retains only directional conflict, and CA-PINN-3 uses the complete conflict-state estimator.

Table 3: Ablation on Burgers under two-term (①) and three-term (②) losses. DC and MC denote directional and magnitude conflict.

Variant	**Scale**	①	②	**DC**	**MC**
CA-PINN-1	$\times 10^{-4}$	1.467	1.328	$\times$	✓
CA-PINN-2	$\times 10^{-4}$	<u>1.460</u>	<u>1.317</u>	✓	$\times$
CA-PINN-3	$\times 10^{-4}$	**1.253**	**1.275**	✓	✓

Several observations can be made. First, either conflict signal alone already provides a meaningful optimization cue, since the two single-signal variants achieve comparable performance on Burgers. However, neither signal alone captures the complete optimization state. Under the two-term setting, CA-PINN-3 reduces the MSE from 1.460×10^{-4} for the stronger single-signal variant to 1.253×10^{-4}, corresponding to an improvement of approximately 14.2%. Under the three-term setting, the error decreases from 1.317×10^{-4} to 1.275×10^{-4}, giving a further improvement of approximately 3.2%.

The joint use of DC and MC is consistent with the optimization dynamics above. Directional disagreement captures geometric conflict, while magnitude imbalance reflects scale dominance. Since the two signals can evolve differently, using both provides a more complete conflict state for adaptive modulation.

5 Conclusion

In this paper, we study gradient conflicts among heterogeneous physical objectives in PINN training. Our analysis shows that directional disagreement and gradient-norm imbalance capture complementary aspects of the conflict state and can evolve differently throughout optimization. Based on this observation, we propose Conflict-Aware Physics-Informed Neural Networks (CA-PINN), which characterizes the evolving conflict state from both perspectives and uses Adaptive Conflict Modulation to dynamically regulate the strength of the bounded conflict-aware update. We further establish the descent property of the resulting bounded time-varying update under the direct-gradient setting. Across five PDE benchmarks and multiple loss decompositions, CA-PINN maintains competitive accuracy across different physical systems.

Reproducibility statement

The optimization procedure is specified in Section 3 and Algorithm 1. Appendix B reports the network architectures, training schedules, evaluation grids, sampling rules, optimizer settings, random seeds, CA-PINN hyperparameters, the fixed-CAGrad tuning protocol, and compute resources. Appendix C provides the detailed gradient-distribution, sensitivity, cancellation, and fixed-coefficient control analyses.

AI use statement

Generative AI tools were used to assist with language polishing, translation. All AI-assisted suggestions were reviewed by the authors, and the technical claims, equations, experimental values, theoretical arguments, and citations were independently checked. The authors take responsibility for the final content of the paper.

References

Chandrajit Bajaj, Luke McLennan, Timothy Andeen, and Avik Roy. Recipes for when physics fails: recovering robust learning of physics informed neural networks. *Machine learning: science and technology*, 4(1):015013, 2023.

Christian Beck, Martin Hutzenthaler, Arnulf Jentzen, and Benno Kuckuck. An overview on deep learning-based approximation methods for partial differential equations. *arXiv preprint arXiv:2012.12348*, 2020.

Rafael Bischof and Michael A Kraus. Multi-objective loss balancing for physics-informed deep learning. *Computer Methods in Applied Mechanics and Engineering*, 439:117914, 2025.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International conference on machine learning*, pp. 794–803. PMLR, 2018.

Zhao Chen, Jiquan Ngiam, Yanping Huang, Thang Luong, Henrik Kretzschmar, Yuning Chai, and Dragomir Anguelov. Just pick a sign: Optimizing deep multitask models with gradient sign dropout. *Advances in Neural Information Processing Systems*, 33:2039–2050, 2020.

Junwoo Cho, Seungtae Nam, Hyunmo Yang, Seok-Bae Yun, Youngjoon Hong, and Eunbyung Park. Separable physics-informed neural networks. *Advances in Neural Information Processing Systems*, 36:23761–23788, 2023.

Salvatore Cuomo, Vincenzo Schiano Di Cola, Fabio Giampaolo, Gianluigi Rozza, Maziar Raissi, and Francesco Piccialli. Scientific machine learning through physics–informed neural networks: Where we are and what's next. *Journal of Scientific Computing*, 92(3):88, 2022.

Robert Eymard, Thierry Gallouët, and Raphaèle Herbin. Finite volume methods. *Handbook of numerical analysis*, 7:713–1018, 2000.

Max D Gunzburger, Clayton G Webster, and Guannan Zhang. Stochastic finite element methods for partial differential equations with random input data. *Acta Numerica*, 23:521–650, 2014.

Youngsik Hwang and Dong-Young Lim. Dual cone gradient descent for training physics-informed neural networks. *Advances in Neural Information Processing Systems*, 37:98563–98595, 2024.

Ameya D Jagtap and George Em Karniadakis. Extended physics-informed neural networks (xpinns): A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations. *Communications in Computational Physics*, 28(5), 2020.

Ameya D Jagtap, Ehsan Kharazmi, and George Em Karniadakis. Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems. *Computer Methods in Applied Mechanics and Engineering*, 365:113028, 2020.

George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Aditi Krishnapriyan, Amir Gholami, Shandian Zhe, Robert Kirby, and Michael W Mahoney. Characterizing possible failure modes in physics-informed neural networks. *Advances in neural information processing systems*, 34:26548–26560, 2021.

Xiaojian Li, Yuhao Liu, and Zhengxian Liu. Physics-informed neural network based on a new adaptive gradient descent algorithm for solving partial differential equations of flow problems. *Physics of Fluids*, 35(6), 2023.

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 18878–18890, 2021a.

Bo Liu, Yihao Feng, Peter Stone, and Qiang Liu. Famo: Fast adaptive multitask optimization. *Advances in Neural Information Processing Systems*, 36:57226–57243, 2023.

Dehao Liu and Yan Wang. A dual-dimer method for training physics-constrained neural networks with minimax architecture. *Neural Networks*, 136:112–125, 2021.

Liyang Liu, Yi Li, Zhanghui Kuang, Jing-Hao Xue, Yimin Chen, Wenming Yang, Qingmin Liao, and Wayne Zhang. Towards impartial multi-task learning. In *International conference on learning representations*, 2021b.

Qiang Liu, Mengyu Chu, and Nils Thuerey. Config: Towards conflict-free training of physics informed neural networks. *arXiv preprint arXiv:2408.11104*, 2024.

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

Robert E Lynch and John R Rice. High accuracy finite difference approximation to solutions of elliptic partial differential equations. *Proceedings of the National Academy of Sciences*, 75(6): 2541–2544, 1978.

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

Michael D McKay, Richard J Beckman, and William J Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 42(1):55–61, 2000.

Aviv Navon, Aviv Shamsian, Idan Achituve, Haggai Maron, Kenji Kawaguchi, Gal Chechik, and Ethan Fetaya. Multi-task learning as a bargaining game. *arXiv preprint arXiv:2202.01017*, 2022.

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

Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

Hwijae Son, Sung Woong Cho, and Hyung Ju Hwang. Enhanced physics-informed neural networks with augmented lagrangian relaxation method (al-pinns). *Neurocomputing*, 548:126424, 2023.

Remco Van Der Meer, Cornelis W Oosterlee, and Anastasia Borovykh. Optimally weighted loss functions for solving pdes with neural networks. *Journal of Computational and Applied Mathematics*, 405:113887, 2022.

Sai Karthikeya Vemuri and Joachim Denzler. Gradient statistics-based multi-objective optimization in physics-informed neural networks. *Sensors*, 23(21):8665, 2023.

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.

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.

Sifan Wang, Bowen Li, Paris Perdikaris, et al. Gradient alignment in physics-informed neural networks: A second-order optimization perspective. *Advances in Neural Information Processing Systems*, 38:168482–168532, 2026.

Zirui Wang, Yulia Tsvetkov, Orhan Firat, and Yuan Cao. Gradient vaccine: Investigating and improving multi-task optimization in massively multilingual models. *arXiv preprint arXiv:2010.05874*, 2020.

Zixue Xiang, Wei Peng, Xu Liu, and Wen Yao. Self-adaptive loss balanced physics-informed neural networks. *Neurocomputing*, 496:11–34, 2022.

Jeremy Yu, Lu Lu, Xuhui Meng, and George Em Karniadakis. Gradient-enhanced physics-informed neural networks for forward and inverse pde problems. *Computer Methods in Applied Mechanics and Engineering*, 393:114823, 2022.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in neural information processing systems*, 33: 5824–5836, 2020.

APPENDIX

This appendix provides additional algorithmic details, implementation settings, detailed PDE benchmark descriptions, supplementary visualization analysis, gradient-update illustrations, and additional quantitative results to support the main submission.

- In Section A, we present the detailed optimization procedure of CA-PINN.
- In Section B, we report the network architectures, training schedules, evaluation protocols, sampling sizes, random seeds, optimizer settings, CA-PINN hyperparameters, and compute resources used in our experiments.
- In Section C, we provide additional optimization diagnostics, including the fixed-coefficient control comparison, loss-specific gradient-norm evolution, hyperparameter sensitivity, and mean-gradient cancellation.
- In Section D, we provide the detailed descriptions of the PDE benchmarks used in our experiments.
- In Section E, we collect the qualitative solution comparisons, including the 3D Beltrami visualization moved from the main text.
- In Section F, we provide an intuitive illustration of how CA-PINN adaptively regulates the update direction under conflicting loss-specific gradients.
- In Section G, we examine the compatibility of CA-PINN with SOAP and compare CA-PINN+SOAP with the SOAP-based PINN system studied by Wang et al. (2026).
- In Section H, we report additional ablation results on four PDE benchmarks to further verify the contribution of directional conflict and magnitude conflict.
- In Section I, we provide supplementary Relative L_2 error results to evaluate the normalized prediction accuracy of CA-PINN across different PDE benchmarks.
- In Section J, we discuss the limitations of the current work and potential future extensions.

A ALGORITHM AND OPTIMIZATION DETAILS

Algorithm 1 summarizes the overall CA-PINN procedure. At each iteration, the method first quantifies gradient conflicts across multiple loss terms from both directional and magnitude perspectives, and then adaptively updates the modulation coefficient. Using this coefficient, CA-PINN solves the dual optimization problem to obtain aggregation weights and constructs the bounded conflict-aware update direction for the network parameters.

Algorithm 1 Conflict-Aware Physics-Informed Neural Networks (CA-PINN)

Require: Loss gradients $\mathcal{G} = \{\mathbf{g}_1, \ldots, \mathbf{g}_M\}$, previous state $c^{(t-1)}$, weight cache $\mathbf{w}_{cache}$

Parameters: Range $[c_{min}, c_{max}]$, EMA factor α_{EMA}, threshold κ, weight λ_{dir}

Average gradient: $\mathbf{g}_0 = \frac{1}{M}\sum_{i=1}^{M} \mathbf{g}_i$

Normalize gradients: $\hat{\mathbf{g}}_i = \mathbf{g}_i / (\|\mathbf{g}_i\| + \epsilon)$

Directional score: $S_{\text{dir}} = \text{mean}_{i \neq j}[\arccos(\hat{\mathbf{g}}_i^\top \hat{\mathbf{g}}_j)/\pi]$

Magnitude score: $S_{\text{mag}} = \min\Big(1, \max\Big(0, \Big(\frac{\max \|\mathbf{g}_i\|}{\min \|\mathbf{g}_i\| + \epsilon} - 1\Big)/\kappa\Big)\Big)$

Total conflict: $S_{\text{total}} = \lambda_{\text{dir}} S_{\text{dir}} + (1 - \lambda_{\text{dir}}) S_{\text{mag}}$

Map to target: $c^{(t)}_{\text{target}} = c_{min} + (c_{max} - c_{min}) \cdot S_{\text{total}}$

Update modulation coefficient: $c^{(t)} = \alpha_{\text{EMA}} c^{(t)}_{\text{target}} + (1 - \alpha_{\text{EMA}}) c^{(t-1)}$

if $M = 2$ **then**

  Solve the one-dimensional dual objective via vectorized grid search for $\mathbf{w}^* = [\gamma, 1-\gamma]^\top$

else

  Solve $\mathbf{w}^* = \arg\min_{\mathbf{w} \in \Delta^M} (\mathbf{g}_{\mathbf{w}}^\top \mathbf{g}_0 + c^{(t)} \|\mathbf{g}_0\| \|\mathbf{g}_{\mathbf{w}}\|)$ with $\mathbf{w}_{cache}$

  Update cache $\mathbf{w}_{cache} \leftarrow \mathbf{w}^*$

end if

Construct final update: $\mathbf{d}^{(t)} = \mathbf{g}_0 + c^{(t)} \|\mathbf{g}_0\| \, \mathbf{g}_{\mathbf{w}^*} / (\|\mathbf{g}_{\mathbf{w}^*}\| + \epsilon)$

return $\mathbf{d}^{(t)}$

B IMPLEMENTATION DETAILS

Experimental settings. For all PDE benchmarks, we use a fully connected MLP with Tanh activation and Xavier initialization. Unless otherwise specified, the network is trained with Adam using an initial learning rate of 10^{-3} and a final learning rate of 10^{-4} under a cosine-constant scheduler. Training samples are drawn by Latin hypercube sampling McKay et al. (2000). For repeated evaluations, the same random seeds $\{52943, 71349, 79774\}$ are used for all candidate methods. Table 1 reports the benchmark results, and Table C.4 reports the comparison between fixed CAGrad and CA-PINN.

Table B.1: Experimental settings of CA-PINN on the PDE benchmarks. Here N_f, N_b, and N_i denote the numbers of collocation, boundary, and initial points, respectively. The loss-split column indicates whether two-term, three-term, or both loss decompositions are evaluated.

Equation	Network	Steps	LR	N_f	N_b	N_i	Loss split
Schrödinger	2-50×5-2	100000	$10^{-3} \to 10^{-4}$	20000	500	500	2/3
Burgers	2-50×5-1	30000	$10^{-3} \to 10^{-4}$	10000	250	250	2/3
Beltrami	4-50×5-4	100000	$10^{-3} \to 10^{-4}$	25000	5000	5000	2/3
Heat	2-50×5-1	30000	$10^{-3} \to 10^{-4}$	50000	1000	1000	2/3
Kovasznay	2-50×5-3	100000	$10^{-3} \to 10^{-4}$	30000	2000	–	2

The network notation d_{in}-50×5-d_{out} denotes an MLP with input dimension d_{in}, five hidden layers of width 50, and output dimension d_{out}. The notation 2/3 in the loss-split column means that both the two-term and three-term loss decompositions are evaluated, while 2 means that only the two-term decomposition is used.

Evaluation protocol. All MSE values are computed on fixed evaluation grids or reference data, rather than on the training collocation points. For Burgers and Heat, we evaluate on uniform grids of $100 \times 256 = 25{,}600$ points. For Schrödinger, we use the fixed NLS reference dataset with a $201 \times 256 = 51{,}456$ space-time grid. For Beltrami, the validation set is a uniform grid generated from the analytic solution with $n_t = 5$ time slices; due to integer rounding in the grid construction, the actual number of evaluation points is $5 \times 7 \times 7 \times 8 = 1{,}960$. For Kovasznay, the validation grid is also generated from the analytic solution, with an actual size of $27 \times 37 = 999$ points.

Table B.2: Evaluation grid and reference solution used for MSE computation.

Equation	Evaluation points	Grid/source	Reference solution
Schrödinger	201×256	fixed dataset	NLS benchmark data
Burgers	100×256	fixed dataset	`simulation_data.npy`
Beltrami	$5 \times 7 \times 7 \times 8$	generated grid	analytic solution
Heat	100×256	fixed grid	$u(x,t) = \sin(\pi x)e^{-t}$
Kovasznay	27×37	generated grid	analytic solution

The Schrödinger reference follows the fixed NLS benchmark data used by Raissi et al. Raissi et al. (2019); the Burgers reference is the precomputed numerical solution stored in `simulation_data.npy`. For Beltrami and Kovasznay, evaluation grids are generated from the corresponding analytic solutions.

Sampling and fairness. Training samples are drawn by Latin hypercube sampling McKay et al. (2000). Since each trainer has one batch per epoch and `update_training_data=True`, the collocation, boundary, and initial points are resampled at every optimization step. All baselines use the same network architecture, random seeds, training steps, sampling rule, and learning-rate scheduler for the same PDE and loss split. The only optimizer exception is M-ConFIG, which follows its method implementation and uses SGD; Adam, PCGrad, IMTL-G, MinMax, ReLoBRaLo, ConFIG, and CA-PINN use Adam.

CA-PINN hyperparameters. CA-PINN uses the bounded conflict-aware update described in Section 3.2 with

$$c_{\text{init}} = 0.5, \quad c_{\min} = 0.3, \quad c_{\max} = 0.7, \quad \alpha_{\text{EMA}} = 0.15,$$

and warm start is enabled for the dual-variable optimization. The conflict score combines directional and magnitude conflicts as

$$S_{\text{total}} = 0.8\, S_{\text{dir}} + 0.2\, S_{\text{mag}},$$

where S_{dir} is computed from pairwise gradient angles and S_{mag} is computed from the ratio of gradient magnitudes. The modulation coefficient c is initialized as $c^{(0)} = c_{\text{init}}$ and is updated by an exponential moving average toward the linearly mapped target value in $[c_{\min}, c_{\max}]$, using α_{EMA} in Eq. 17.

Fixed-CAGrad tuning protocol. For Table C.4, fixed CAGrad is tuned independently for every PDE/loss setting over $c \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$. All candidates use the same initialization procedure, resampling protocol, optimizer schedule, random seeds, and evaluation grid as CA-PINN. The coefficient is selected using the average validation loss over the final 500 epochs, and the same protocol is applied consistently to all fixed-c candidates.

Compute resources. All experiments were run on a single NVIDIA GeForce RTX 3090 GPU with 24GB memory. The machine contains two RTX 3090 GPUs, but each training run used one GPU only. All reported methods fit within the 24GB memory budget.

Table B.3: Logged computational cost over all PDE benchmarks, loss settings, and repeated runs.

Method	Total running (GPU-h)	Mean/run	Range/run	Total training (GPU-h)
Adam	22.40	49.8 min	5.3–183.6 min	17.25
PCGrad	24.68	54.8 min	5.8–238.2 min	23.21
IMTL-G	22.63	50.3 min	6.0–162.0 min	20.91
MinMax	22.03	48.9 min	5.7–165.5 min	16.98
ReLoBRaLo	21.37	47.5 min	5.5–152.4 min	16.78
ConFIG	23.01	51.1 min	6.0–160.9 min	21.67
M-ConFIG	11.03	24.5 min	3.1–111.0 min	10.00
CA-PINN	28.31	62.9 min	6.2–249.3 min	27.13

C Additional optimization diagnostics

C.1 Fixed-coefficient control comparison

To isolate adaptive modulation from the underlying constrained update rule, we compare CA-PINN with a tuned fixed-coefficient CAGrad baseline. We evaluate $c \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ independently for every PDE/loss setting and choose the coefficient with the lowest average validation loss over the final 500 epochs. Fixed CAGrad and CA-PINN use the same initialization, sampling procedure, optimizer schedule, and evaluation grid.

Table C.4: Mean MSE for tuned fixed CAGrad and CA-PINN. Fixed c is selected separately for every PDE/loss setting.

PDE (losses)	**Best c**	**Fixed CAGrad**	**CA-PINN**
Schrödinger (2)	0.5	6.909e−5	5.620e−5
Schrödinger (3)	0.5	9.009e−5	6.600e−5
Burgers (2)	0.9	1.747e−4	1.253e−4
Burgers (3)	0.7	1.735e−4	1.275e−4
Beltrami (2)	0.7	5.346e−5	5.590e−5
Beltrami (3)	0.1	6.635e−5	5.080e−5
Heat (2)	0.9	2.206e−7	7.200e−8
Heat (3)	0.5	4.393e−6	6.320e−8
Kovasznay (2)	0.5	1.033e−7	1.170e−8

The selected fixed coefficient ranges from 0.1 to 0.9, indicating a strong dependence on the PDE and loss decomposition. In contrast, CA-PINN uses the same adaptive configuration throughout the benchmark suite and achieves a lower mean MSE in eight of the nine settings. The only exception is the two-term Beltrami problem, where tuned fixed CAGrad gives a slightly lower mean error. These results show that adaptive conflict modulation can accommodate the problem-dependent variation in the preferred correction strength without selecting a separate coefficient for each PDE.

We further compare the update geometry of CA-PINN with the tuned fixed-CAGrad baseline on Burgers. Besides S_{dir}, S_{mag}, and S_{total}, we measure the worst-objective alignment and the proportion of updates that are descent directions for all physical losses,

$$D_{\mathrm{worst}}^{(t)} = \min_i \frac{(\mathbf{g}_i^{(t)})^\top \mathbf{d}^{(t)}}{\|\mathbf{g}_i^{(t)}\| \, \|\mathbf{d}^{(t)}\| + \epsilon}, \qquad P_+ = \Pr\Big((\mathbf{g}_i^{(t)})^\top \mathbf{d}^{(t)} > 0, \ \forall i\Big). \tag{24}$$

Table C.5: Mean conflict statistics for tuned fixed CAGrad and CA-PINN on Burgers. P_+ is reported in percent.

Quantity	**Fixed $c = 0.7$**	**CA-PINN**	**Change**
S_{dir}	0.4410	0.4231	−4.1%
S_{mag}	0.8894	0.8844	−0.6%
S_{total}	0.5307	0.5153	−2.9%
D_{worst}	0.2041	0.2218	+8.7%
P_+	69.0	70.0	+1.0 pp

CA-PINN gives lower directional and total conflict scores and a higher worst-objective alignment than the tuned fixed-CAGrad baseline. Together with the improvement in P_+, these results show that the adaptive modulation changes the update geometry in the intended direction rather than only changing the final prediction error.

C.2 Gradient-norm evolution

The main text reports the evolution of S_{dir}, S_{mag}, S_{total}, $c^{(t)}$, and r_{mag} on Burgers with three loss terms. Here we provide complementary loss-specific gradient-norm diagnostics to explain why magnitude imbalance can increase even while the overall optimization progresses. At checkpoint t and repeated run s, for $\ell \in \{\mathrm{PDE}, \mathrm{BC}, \mathrm{IC}\}$ we compute

$$G_{\ell,s,t} = \|\nabla_\theta L_\ell(\theta_{s,t})\|_2. \tag{25}$$

Table C.6 reports the phase-wise median gradient norms. All three objectives decrease substantially over training, but the boundary and initial-condition gradients shrink faster than the PDE gradient.

This unequal contraction explains why the relative gradient-norm imbalance can become larger even as all physical objectives are being optimized.

Table C.6: Phase-wise median loss-specific gradient norms on Burgers with three loss terms, aggregated over repeated runs.

Phase	**Median** $\|g_{\mathrm{PDE}}\|$	**Median** $\|g_{\mathrm{BC}}\|$	**Median** $\|g_{\mathrm{IC}}\|$
1–10k	1.194	.108	.142
10–20k	.684	.041	.035
20–30k	.453	.028	.023

To quantify the contraction of the gradient distributions, within each phase we pool checkpoint-level norms across runs and define the 10–90% width

$$W_\ell = Q_{0.90}(G_\ell) - Q_{0.10}(G_\ell). \tag{26}$$

Table C.7: Contraction of gradient-norm distributions from steps 1–10k to 20–30k.

Gradient	**Steps 1–10k**	**Steps 20–30k**	**Contraction**
PDE	1.16×10^{-2}	2.20×10^{-3}	81.0%
BC	1.45×10^{-3}	2.56×10^{-4}	82.3%
IC	1.59×10^{-3}	1.91×10^{-4}	88.0%

All three distributions contract, but BC/IC shrink faster than the PDE gradient. Comparing median losses over the first and last 1,000 steps, the PDE loss decreases from 4.02×10^{-2} to 1.83×10^{-4}, the boundary loss from 3.93×10^{-3} to 1.01×10^{-5}, and the initial loss from 6.97×10^{-2} to 2.12×10^{-5}. Thus, all physical constraints improve rather than one being sacrificed.

C.3 Hyperparameter sensitivity

We use one default across all PDEs:

$$(c_{\mathrm{init}}, [c_{\min}, c_{\max}], \lambda_{\mathrm{dir}}, \kappa, \alpha_{\mathrm{EMA}}) = (0.5, [0.3, 0.7], 0.8, 10, 0.15). \tag{27}$$

Table C.8 reports three-loss Burgers MSEs (all values $\times 10^{-4}$) when varying one choice at a time.

Table C.8: Sensitivity analysis on three-loss Burgers. The middle/default choice is used except for λ_{dir}, where 0.8 is the default.

Parameter	**Values**	**MSE** ($\times 10^{-4}$)
Bounds	[.1, .7]/[.3, .7]/[.3, .9]	<u>1.62</u>/**1.275**/1.74
α_{EMA}	.05/.15/.30	1.75/**1.275**/<u>1.65</u>
κ	5/10/20	<u>1.67</u>/**1.275**/3.19
λ_{dir}	.25/.50/.75/.80	2.29/<u>1.80</u>/6.55/**1.275**
c_{init}	.3/.5/.7	<u>3.36</u>/**1.275**/4.77

The default configuration gives the lowest MSE among the tested settings. The radius bounds and EMA factor show moderate variation around the default, whereas κ and λ_{dir} have a stronger influence on the final error. In particular, the response to λ_{dir} is non-monotonic, which suggests that the relative emphasis on directional and magnitude conflict should be chosen with care.

C.4 Mean-gradient cancellation analysis

We use $\rho_t < 0.1$ in Eq. 18 to identify iterations with strong cancellation among the loss-specific gradients. Such iterations account for 2.44% of the CA-PINN trajectory and 2.21% of the fixed-c

trajectory on Burgers. The mean adaptive coefficient increases from 0.507 over the full trajectory to 0.546 in this regime, showing that the controller applies a stronger relative correction when cancellation is more pronounced. The corresponding normalized radius $\widetilde{R}_t$ is 0.0332 for CA-PINN and 0.0364 for fixed $c = 0.7$.

D Detailed PDE benchmark setup

Burgers equation We consider the one-dimensional viscous Burgers' equation, a canonical benchmark for modeling nonlinear convection-diffusion phenomena. The governing equation is defined as follows:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2}, \quad x \in [-1, 1], \quad t \in [0, 1] \tag{28}$$

where the kinematic viscosity is set to $\nu = 0.01/\pi$. The system is initialized with a sinusoidal profile $u(x, 0) = -\sin(\pi x)$ and is subject to zero Dirichlet boundary conditions $u(-1, t) = u(1, t) = 0$. These settings drive the solution to develop a sharp gradient, or shock wave, at $x = 0$.

Beltrami flow We consider the three-dimensional unsteady incompressible Navier-Stokes equations, which govern the motion of viscous fluids. The equations are defined as follows:

$$\begin{gathered} \nabla \cdot \mathbf{u} = 0, \\ \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\nabla p + \nu \nabla^2 \mathbf{u}, \\ x \in [-1, 1], y \in [-1, 1], z \in [-1, 1], t \in [0, 1], \end{gathered} \tag{29}$$

where $\mathbf{u} = (u, v, w)$ denotes the velocity field, p is the pressure, and ν is the kinematic viscosity. To validate the model, we utilize the exact Beltrami flow solution to prescribe initial and boundary conditions. The analytical solution is given by:

$$\begin{aligned} u(x, y, z, t) &= -a\left[e^{ax}\sin(ay + dz) + e^{az}\cos(ax + dy)\right] e^{-d^2 t} \\ v(x, y, z, t) &= -a\left[e^{ay}\sin(az + dx) + e^{ax}\cos(ay + dz)\right] e^{-d^2 t} \\ w(x, y, z, t) &= -a\left[e^{az}\sin(ax + dy) + e^{ay}\cos(az + dx)\right] e^{-d^2 t} \end{aligned} \tag{30}$$

In our experiments, we set the parameters $a = 1, d = 1$, and $\nu = 1$.

Schrödinger equation We study the one-dimensional non-linear Schrödinger (NLS) equation, a standard model for wave propagation in nonlinear media. The complex-valued solution $h(x, t) = u(x, t) + iv(x, t)$ is governed by the equation:

$$ih_t + 0.5h_{xx} + |h|^2 h = 0, \quad x \in [-5, 5], t \in [0, \pi/2] \tag{31}$$

The system is initialized with a soliton solution $h(x, 0) = 2\,\mathrm{sech}(x)$ and is subject to periodic boundary conditions $h(-5, t) = h(5, t)$ and $h_x(-5, t) = h_x(5, t)$.

Heat equation We consider the one-dimensional heat equation to evaluate the model's performance on pure diffusion processes. The governing equation is defined as follows:

$$\begin{gathered} \frac{\partial u}{\partial t} = \frac{1}{\pi^2}\frac{\partial^2 u}{\partial x^2}, \\ x \in [-1, 1], t \in [0, 1], \end{gathered} \tag{32}$$

The system is initialized with a sinusoidal profile $u(x, 0) = \sin(\pi x)$ and is subject to zero Dirichlet boundary conditions $u(-1, t) = u(1, t) = 0$. These conditions lead to a smooth decay of the solution over time.

Kovasznay flow We consider the two-dimensional steady incompressible Navier-Stokes equations to model the Kovasznay flow, a laminar flow behind a grid often used for code validation. The governing equations are defined as follows:

$$\begin{aligned} \nabla \cdot \mathbf{u} &= 0, \\ (\mathbf{u} \cdot \nabla)\mathbf{u} &= -\nabla p + \nu \nabla^2 \mathbf{u}, \end{aligned} \tag{33}$$

where $\nu = 1/Re$ with the Reynolds number set to $Re = 40$. The system is subject to Dirichlet boundary conditions determined by the exact solution:

$$\begin{aligned} u &= 1 - e^{\lambda x} \cos(2\pi y), \\ v &= \frac{\lambda}{2\pi} e^{\lambda x} \sin(2\pi y), \\ p &= \frac{1}{2}(1 - e^{2\lambda x}), \end{aligned} \tag{34}$$

with $\lambda = \frac{1}{2\nu} - \sqrt{\frac{1}{4\nu^2} + 4\pi^2}$. Since the Kovasznay flow is a steady-state problem independent of time, it inherently lacks initial conditions and thus involves only two loss components. In contrast, the other time-dependent benchmarks involve three loss components, including the PDE residual, boundary conditions, and initial conditions.

E Additional visualization analysis

This section provides supplementary qualitative analyses that complement the quantitative results in Section 4.2. We report the 3D Beltrami comparison in Figure E.1, broader visual comparisons on four representative benchmarks in Figure E.2, and cross-sectional comparisons on the Burgers and Schrödinger equations in Figures E.3 and E.4, respectively.

Figure E.1 shows that the compared methods recover the overall 3D flow structure, while CA-PINN exhibits lower and more localized error in several regions. We treat this visualization as qualitative support and rely on the quantitative comparisons in Tables 1 and C.4 for the main claims.

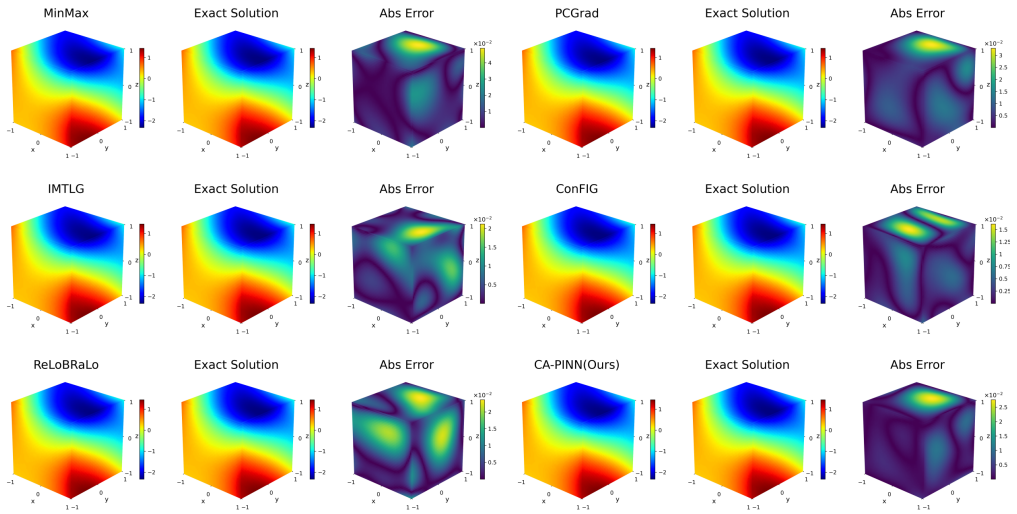
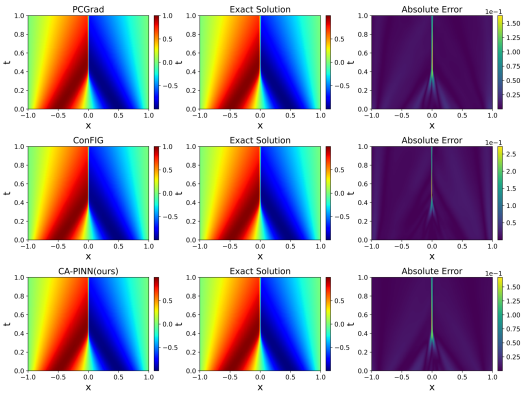
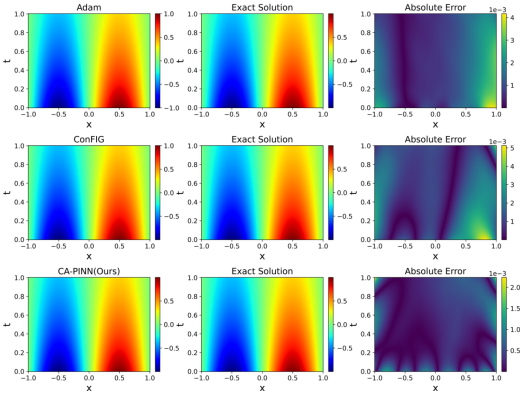


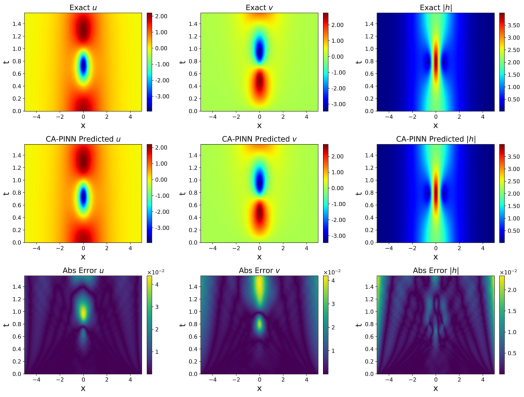
Figure E.1: Qualitative comparison of predicted solutions, reference solutions, and absolute errors on the 3D Beltrami flow benchmark.



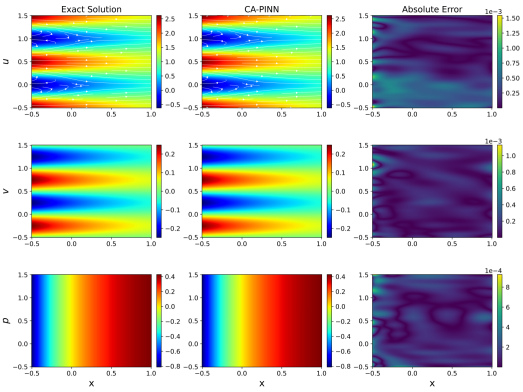
(a) Visualization results on the Burgers equation.



(b) Visualization results of two baselines and CA-PINN on the Heat equation.



(c) Solution and error contours of the NLS equation. Columns represent the real part u, imaginary part v, and magnitude $|h|$.



(d) Visualization of the steady-state Kovasznay flow results. The rows display the exact solutions and the absolute error maps for the horizontal velocity u, vertical velocity v, and pressure p.

Figure E.2: Visual comparison of CA-PINN and baseline methods on four representative benchmarks.

As shown in Figure E.2, CA-PINN demonstrates strong qualitative performance across different physical scenarios. For the Burgers equation, CA-PINN more accurately captures the sharp transition region and yields a more concentrated error distribution near the shock interface. For the Heat equation, CA-PINN recovers a smoother diffusion profile and exhibits lower overall error than the baselines, which is consistent with its strong quantitative performance. For the Schrödinger equation, CA-PINN better preserves the structures of the real part, imaginary part, and magnitude, and the predicted solution remains visually closer to the reference contours. For the Kovasznay flow, although the task involves coupled velocity-pressure fields, CA-PINN still reconstructs the main flow patterns and pressure distribution with localized error regions, showing competitive qualitative accuracy on this steady-state problem.

Figure E.3 further presents the velocity profiles of the Burgers equation at three representative time instants. We observe that the predictions of CA-PINN closely follow the exact solution across all slices. In particular, the method maintains good agreement in both smooth regions and the steep-gradient area. This qualitative comparison shows that CA-PINN closely preserves the shock location and profile shape across the selected time slices.

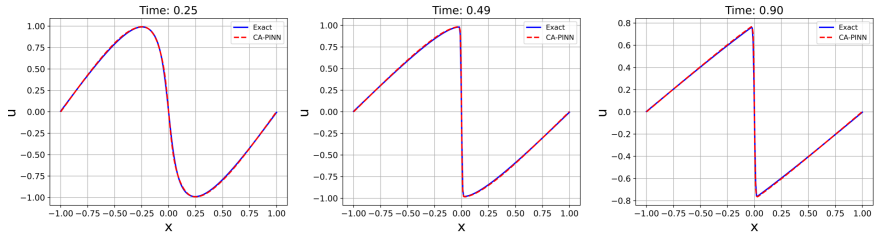


Figure E.3: Velocity profiles. CA-PINN predictions (red dashed lines) versus exact solutions (blue solid lines) at $t = 0.24, 0.49, 0.75$.

Figure E.4 shows the predicted magnitude $|h|$ on the Schrödinger equation at $t = \{0.59, 0.79, 0.98\}$. We can observe that CA-PINN remains highly consistent with the exact solution over the selected time slices, accurately tracking both the peak amplitude and the spatial profile of the wave packet. The close agreement across multiple time slices suggests stable prediction quality for this oscillatory complex-valued solution.

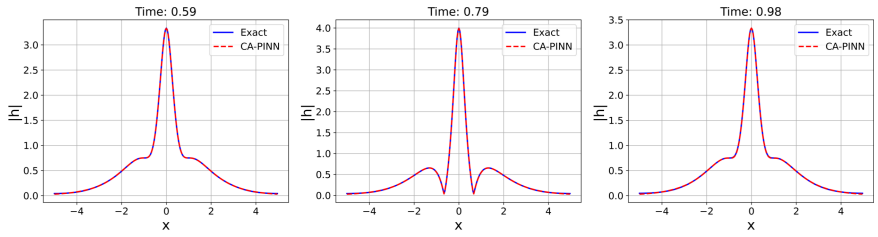


Figure E.4: The predicted magnitude $|h|$ by CA-PINN versus the exact solution at $t = \{0.59, 0.79, 0.98\}$.

Overall, these supplementary visual results provide qualitative support for the quantitative results in Table 1. Across the selected examples, CA-PINN maintains accurate solution profiles for PDEs with shocks, diffusion dynamics, oscillatory structures, and coupled flow fields.

F ADAPTIVE CONFLICT-AWARE GRADIENT UPDATE

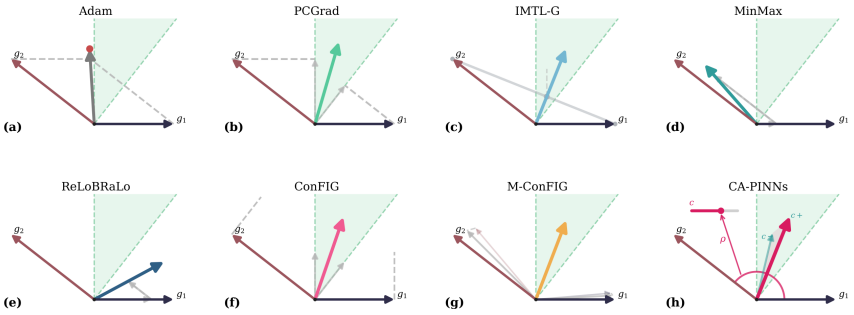


Figure F.5: Schematic comparison of gradient update directions under conflicting loss-specific gradients. The green region denotes the shared descent cone. Existing methods use fixed weighting or projection rules, while CA-PINN maps the estimated conflict level S_{total} to the modulation coefficient c, adaptively adjusting the feasible update region according to the current optimization state.

Figure F.5 provides an intuitive illustration of how CA-PINN differs from representative gradient-based optimization methods. When loss-specific gradients are inconsistent, directly averaging them may produce an update that is biased toward the dominant objective or even conflicts with some physical constraints. Projection-based methods, such as PCGrad and IMTL-G, reduce part of this

interference by modifying the update direction, while ConFIG and M-ConFIG further encourage conflict-free updates within a shared descent region. However, these methods still follow relatively fixed update rules and do not explicitly adjust the strength of conflict mitigation according to the current training state.

In contrast, CA-PINN introduces an adaptive conflict-aware mechanism. Specifically, it estimates the conflict level S_{total} from directional disagreement and magnitude imbalance, and maps this score to the modulation coefficient c. When the gradients exhibit strong conflict, a larger c allows the update direction to deviate more from the average gradient, providing more freedom to seek a shared descent direction. When the gradients are relatively compatible, a smaller c keeps the update closer to the average descent direction, avoiding unnecessary correction. Therefore, Figure F.5 visually demonstrates the core motivation of CA-PINN: conflict mitigation should not be static, but should be adaptively regulated according to the evolving optimization state of PINN training.

G Comparison and complementarity with SOAP-based optimization

Table G.9: Comparison of CA-PINN, CA-PINN+SOAP, and our reproduction of the SOAP-based PINN system in Wang et al. (2026). Bold and underlined values indicate the best and second-best results in each row, respectively. "Ref." denotes the reproduced SOAP-based reference system, which reports one value per PDE and is repeated across the corresponding loss-split rows for comparison.

PDE	Losses	Metric	CA-PINN	CA-PINN+SOAP	Ref.
Schrödinger	2	MSE	$\underline{5.620 \times 10^{-5}}$	$\mathbf{1.8950 \times 10^{-5}}$	2.79×10^{-2}
	2	Relative L_2	$\underline{1.290 \times 10^{-2}}$	$\mathbf{6.8871 \times 10^{-3}}$	2.640×10^{-1}
	3	MSE	$\underline{6.600 \times 10^{-5}}$	$\mathbf{1.8874 \times 10^{-5}}$	2.79×10^{-2}
	3	Relative L_2	$\underline{1.148 \times 10^{-2}}$	$\mathbf{6.8452 \times 10^{-3}}$	2.640×10^{-1}
Burgers	2	MSE	$\mathbf{1.253 \times 10^{-4}}$	1.6338×10^{-4}	$\underline{1.577 \times 10^{-4}}$
	2	Relative L_2	2.143×10^{-2}	$\underline{2.1017 \times 10^{-2}}$	$\mathbf{2.065 \times 10^{-2}}$
	3	MSE	$\mathbf{1.275 \times 10^{-4}}$	$\underline{1.5703 \times 10^{-4}}$	1.577×10^{-4}
	3	Relative L_2	2.066×10^{-2}	$\mathbf{2.0604 \times 10^{-2}}$	$\underline{2.065 \times 10^{-2}}$
Beltrami	2	MSE	$\underline{5.590 \times 10^{-5}}$	$\mathbf{4.6353 \times 10^{-6}}$	2.07×10^{-3}
	2	Relative L_2	$\underline{4.159 \times 10^{-3}}$	$\mathbf{1.2669 \times 10^{-3}}$	2.771×10^{-2}
	3	MSE	$\underline{5.080 \times 10^{-5}}$	$\mathbf{3.5606 \times 10^{-6}}$	2.07×10^{-3}
	3	Relative L_2	$\underline{4.790 \times 10^{-3}}$	$\mathbf{1.1436 \times 10^{-3}}$	2.771×10^{-2}
Heat	2	MSE	$\underline{7.200 \times 10^{-8}}$	2.9459×10^{-7}	$\mathbf{1.08 \times 10^{-12}}$
	2	Relative L_2	$\underline{9.500 \times 10^{-4}}$	1.1544×10^{-3}	$\mathbf{2.24 \times 10^{-6}}$
	3	MSE	6.320×10^{-8}	$\underline{1.4900 \times 10^{-9}}$	$\mathbf{1.08 \times 10^{-12}}$
	3	Relative L_2	1.652×10^{-3}	$\underline{6.7242 \times 10^{-5}}$	$\mathbf{2.24 \times 10^{-6}}$
Kovasznay	2	MSE	$\underline{1.170 \times 10^{-8}}$	$\mathbf{7.5520 \times 10^{-9}}$	1.23×10^{-6}
	2	Relative L_2	$\underline{2.762 \times 10^{-4}}$	$\mathbf{1.1653 \times 10^{-4}}$	1.501×10^{-3}

Wang et al. (2026) study gradient alignment in PINNs from a second-order optimization perspective and evaluate a SOAP-based training system that combines PirateNet, RFF/RWF, GradNorm, causal weighting, and SOAP. Following their system design, we reproduce this SOAP-based pipeline and evaluate it on our five benchmark settings. Although both approaches target gradient-related optimization difficulties, CA-PINN operates at the gradient-aggregation stage by estimating the current gradient-conflict state and adaptively regulating conflict-aware gradient aggregation. Importantly, the standard CA-PINN configuration uses Adam rather than SOAP. As shown in Table G.9, CA-PINN without SOAP already obtains lower MSE than our reproduced SOAP-based reference on four of the five PDE benchmarks—Schrödinger, Burgers, Beltrami, and Kovasznay—and lower Relative L_2 error on Schrödinger, Beltrami, and Kovasznay. These results show that CA-PINN remains effective without SOAP-based optimization.

We further examine whether the two approaches can be combined. Specifically, we replace Adam with SOAP in CA-PINN while keeping the conflict-state estimator and adaptive modulation unchanged, and

denote the resulting variant as **CA-PINN+SOAP**. We use $(\beta_1, \beta_2) = (0.99, 0.999)$ for Schrödinger, $(0.99, 0.99)$ for Burgers and Beltrami, $(0.95, 0.999)$ for Heat, and $(0.90, 0.999)$ for Kovasznay. Table G.9 reports CA-PINN, CA-PINN+SOAP, and our reproduced SOAP-based reference following Wang et al. (2026). Since the reference is a complete training system rather than a two-/three-loss variant of our formulation, it provides one result for each PDE; the same reference value is repeated across the corresponding loss-split rows only for comparison.

The comparison reveals two complementary observations. First, even without SOAP, CA-PINN already outperforms the reproduced SOAP-based reference in MSE on four of the five PDEs and in Relative L_2 error on three of the five PDEs, indicating that adaptive conflict modulation provides an optimization benefit independently of SOAP. Second, incorporating SOAP can further improve CA-PINN on several benchmarks. CA-PINN+SOAP reduces both MSE and Relative L_2 error relative to CA-PINN on Schrödinger, Beltrami, Kovasznay, and the three-loss Heat setting, while also slightly improving Relative L_2 on three-loss Burgers. The combination therefore further enlarges the advantage over the SOAP-based reference on Schrödinger, Beltrami, and Kovasznay, produces closely matched results on Burgers, while the reproduced SOAP-based reference remains stronger on Heat. These results support the complementarity of the two mechanisms: CA-PINN regulates the aggregation of conflicting physical gradients, whereas SOAP improves the subsequent parameter update through second-order-aware preconditioning.

H Additional ablation study

Table H.10: Additional ablation study on four PDE benchmarks. The scale factor is reported in the "Scale" column, and each value should be multiplied by the corresponding factor. Symbols ① and ② denote the two-term and three-term loss configurations, respectively. DC and MC denote directional conflict and magnitude conflict.

Equation	**Scale**	**Loss**	**Ablation variant**		
			CA-PINN-1 MC only	**CA-PINN-2** DC only	**CA-PINN-3** DC + MC
Schrödinger	$\times 10^{-4}$	①	1.388	**0.520**	<u>0.562</u>
		②	<u>0.829</u>	1.012	**0.660**
Beltrami	$\times 10^{-4}$	①	<u>1.110</u>	1.132	**0.559**
		②	0.823	<u>0.612</u>	**0.508**
Heat	$\times 10^{-7}$	①	<u>0.836</u>	2.612	**0.720**
		②	<u>0.717</u>	0.782	**0.632**
Kovasznay	$\times 10^{-7}$	①	<u>0.357</u>	0.489	**0.117**

To test whether the Burgers ablation trends generalize beyond the main-text example, Table H.10 evaluates the same three variants defined in Table 3 on four additional PDE benchmarks. The complete CA-PINN-3 achieves the lowest error in six of the seven additional settings and the second-lowest error in the remaining two-term Schrödinger setting, showing that the benefit of jointly modeling directional disagreement and magnitude imbalance is not limited to Burgers.

More specifically, the relative importance of the two conflict signals varies across physical systems and loss decompositions. On the two-term Schrödinger problem, directional conflict alone performs best, whereas on the three-term setting the complete model reduces the error from 0.829×10^{-4} to 0.660×10^{-4} compared with the stronger single-signal variant. A similar complementary effect is more pronounced on the flow problems. For Beltrami, CA-PINN-3 reduces the error from 1.110×10^{-4} to 0.559×10^{-4} in the two-term setting and from 0.612×10^{-4} to 0.508×10^{-4} in the three-term setting. On Kovasznay, the error further decreases from 0.357×10^{-7} to 0.117×10^{-7}.

The Heat equation exhibits another useful pattern: magnitude conflict alone is consistently stronger than directional conflict alone, yet incorporating both signals still gives the lowest error in both loss decompositions. These results indicate that neither directional disagreement nor magnitude imbalance

is uniformly dominant across PDEs. Instead, their relative contributions depend on the underlying physical dynamics and objective structure. Jointly characterizing both signals therefore provides a more complete and transferable description of the conflict state, leading to more consistent adaptive modulation across different PDE benchmarks.

I SUPPLEMENTARY RELATIVE L_2 EVALUATION

Table I.11: Relative L_2 error on the five PDE benchmarks. The scale factor is reported in the "Scale" column, and each value should be multiplied by the corresponding factor. Symbols ① and ② denote the two-term and three-term loss configurations, respectively.

Equation	Scale	Loss	Adam	PCGrad	IMTL-G	MinMax	ReLoBRaLo	ConFIG	M-ConFIG	CA-PINN
			ICLR'15	NeurIPS'20	ICLR'21	NN'21	CMAME'25	ICLR'25	ICLR'25	Ours
Schrödinger	$\times 10^{-2}$	①	3.041	2.173	2.948	11.197	3.504	<u>1.608</u>	3.663	**1.290**
		②	3.033	<u>1.702</u>	62.737	10.830	2.729	2.385	6.465	**1.148**
Burgers	$\times 10^{-2}$	①	2.310	3.665	**2.107**	9.022	2.723	5.316	2.246	<u>2.143</u>
		②	2.082	2.176	13.158	2.575	**2.029**	11.040	2.320	<u>2.066</u>
Beltrami	$\times 10^{-3}$	①	6.536	6.347	<u>4.779</u>	8.652	6.718	4.854	5.790	**4.159**
		②	5.940	5.383	5.351	9.185	6.292	**4.712**	6.625	<u>4.790</u>
Heat	$\times 10^{-3}$	①	10.091	12.465	16.019	4.308	**0.846**	1.869	2.236	<u>0.950</u>
		②	10.091	9.782	9.461	9.546	<u>2.056</u>	6.570	17.707	**1.652**
Kovasznay	$\times 10^{-4}$	①	6.411	8.276	4.051	20.098	4.887	**1.822**	3.878	<u>2.762</u>

Besides the MSE results, Relative L_2 error provides a complementary evaluation of the normalized solution accuracy. Therefore, we report the Relative L_2 error on five PDE benchmarks in Table I.11. From these results, we can obtain the following observations. (1) CA-PINN achieves consistently competitive performance across all benchmarks and loss configurations. Specifically, it obtains the best Relative L_2 error on the Schrödinger equation under both two-term and three-term settings, the Beltrami flow under the two-term setting, and the Heat equation under the three-term setting. For the remaining cases, CA-PINN still achieves the second-best performance, showing that the proposed method does not suffer from obvious performance degradation across different physical systems.

(2) Compared with representative gradient-based baselines, CA-PINN maintains competitive performance under more challenging multi-objective settings. For example, on the three-term Schrödinger equation, CA-PINN reduces the Relative L_2 error from 1.702 to 1.148 compared with the strongest competing baseline. Similarly, on the three-term Heat equation, CA-PINN achieves the lowest error among all methods, indicating that Adaptive Conflict Modulation can better handle the interactions among PDE residuals, boundary conditions, and initial conditions.

(3) Although some baselines perform well on specific benchmarks, their performance is less stable across different PDEs. For instance, ReLoBRaLo achieves strong results on the Heat and Burgers equations, while ConFIG performs well on the Kovasznay and Beltrami cases. In contrast, CA-PINN maintains best or second-best results in all settings, which suggests that dynamically regulating the conflict mitigation strength leads to more consistent performance across these benchmarks. Overall, the Relative L_2 results provide complementary evidence that CA-PINN maintains competitive normalized solution accuracy across diverse PDE benchmarks.

J LIMITATIONS

Although CA-PINN performs well on the five PDE benchmarks, several directions remain to be explored. The current experiments focus on standard forward problems with clean references, and broader evaluation on inverse problems, irregular geometries, noisy observations, and larger multiphysics systems would further test its applicability. CA-PINN also requires loss-specific gradients, pairwise conflict scores, and an inner constrained aggregation subproblem. With M objectives and P parameters, gradient storage is $O(MP)$ and pairwise scoring is $O(M^2P)$, which

increases the computational cost when either the network or the number of objectives grows. In addition, the sensitivity results suggest that the controller parameters can be further simplified or selected automatically. These issues provide useful directions for extending CA-PINN to larger-scale scientific learning problems.