Per-Loss Adapters for Gradient Conflict in Physics-Informed Neural Networks

Bum Jun Kim[1,*] **Gnankan Landry Regis N'guessan**[2,3,4]

[1] The University of Tokyo, Japan
[2] Axiom Research Group
[3] Department of Applied Mathematics and Computational Science, NM-AIST, Tanzania
[4] African Institute for Mathematical Sciences (AIMS), Research and Innovation Centre, Rwanda

[*] Corresponding author: bumjun.kim@weblab.t.u-tokyo.ac.jp

Abstract

Physics-informed neural networks (PINNs) train a single neural approximation by minimizing multiple physics- and data-derived losses, but the gradients of these losses often interfere and can stall optimization. Existing remedies typically treat this pathology either through scalar loss balancing or full-parameter-space gradient surgery, leaving it unclear which intervention is most appropriate. We show that PINN gradient conflict is not a uniform failure mode with one universal remedy. Instead, we identify distinct PINN gradient-conflict regimes, each associated with a different intervention class. Persistent directional conflict may require separate loss-indexed parameter subspaces, magnitude imbalance often favors scalar reweighting, and low or transient conflict may require no extra mitigation. To select between scalar reweighting and a lightweight architectural intervention, we propose a diagnostic-first framework. It profiles a 1000-step unmodified PINN run and, when intervention is warranted, uses one low-rank adapter per loss to create explicit loss-indexed parameter subspaces attached to a shared PINN trunk, providing each loss with a direct gradient pathway. Across more than 60 PDE configurations, including forward, inverse, multi-physics, parameter-varying, and high-dimensional problems up to 50D, persistent directional conflict dominates standard forward $K = 3$ benchmarks and a natural $K = 4$ thermoelastic system, where adapters combined with reweighting yield significant improvements. In contrast, $K = 3$ inverse problems and natural $K = 5$ and $K = 6$ multi-physics systems are largely magnitude-dominated and often favor reweighting alone, while full-parameter-space gradient surgery can fail on heterogeneous parameter spaces. A regime-transition theorem and a blockwise neural tangent kernel analysis explain why gains from these separate adapter parameter subspaces arise selectively in persistent-conflict regimes.

1 Introduction

Physics-informed neural networks (PINNs) [34, 15] embed physical laws directly into neural network training by minimizing a composite loss:

$$\mathcal{L}(\theta) = \sum_{k=1}^{K} \mathcal{L}_k(\theta), \tag{1}$$

where $\mathcal{L}_k$ includes partial differential equation (PDE) residuals $\mathcal{L}_{\mathrm{PDE}}(\theta)$, boundary-condition (BC) terms $\mathcal{L}_{\mathrm{BC}}(\theta)$, initial-condition (IC) terms $\mathcal{L}_{\mathrm{IC}}(\theta)$, and data-fidelity terms for inverse problems. Despite their elegance, PINNs suffer from well-documented training difficulties [41, 43, 19, 20, 3, 44]. The gradients $g_k = \nabla_\theta \mathcal{L}_k$ frequently point in conflicting directions, causing destructive interference that impedes convergence; see Figure 1.

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

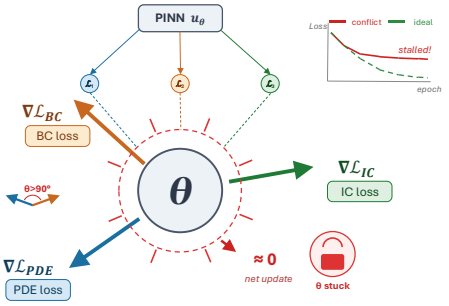

Figure 1: Gradient conflict in PINNs. A PINN $u_{\boldsymbol{\theta}}$ is trained with multiple loss terms, whose gradients $\nabla_{\boldsymbol{\theta}}\mathcal{L}_k$ may point in opposing directions, which causes the training to stall.

Two common approaches to gradient conflicts in PINNs are scalar loss balancing and full-parameter-space gradient surgery. The first category includes loss reweighting methods such as learning rate annealing, gradient normalization (GradNorm), SoftAdapt, and Fast Adaptive Multitask Optimization (FAMO) [41, 2, 6, 24], which adjust the scalar weights λ_k in $\sum_k \lambda_k \mathcal{L}_k$ to balance loss magnitudes but do not modify gradient directions. The second category includes gradient-surgery methods such as projected conflicting gradient (PCGrad), conflict-averse gradient descent (CAGrad), Gradient Vaccine, RotoGrad, and Nash multi-task learning (NashMTL) [49, 23, 46, 14, 32], which project, rotate, or otherwise transform conflicting gradients to remove negative interference but operate on the full parameter space, are computationally expensive, and can fail catastrophically in heterogeneous PINN settings, especially inverse problems that combine network weights with learnable physical constants. Tables 4 and 36 isolate this failure mode through inverse-problem results and grouped-surgery mitigation. Curvature-aware optimizers such as SOAP also improve update alignment [45]. These approaches and recent PINN-specific analyses [10] motivate a regime-level rule for when architectural separation is useful alongside loss balancing and optimizer conditioning.

Rather than advocating a single method, our work argues that PINN gradient conflict should be treated as a regime-dependent optimization pathology. Different PDE-loss systems may require different classes of interventions, even when all exhibit multi-loss interference. We characterize the full landscape of gradient conflict in PINNs across directional conflict, magnitude imbalance, and their K-dependent interaction, providing a principled framework for regime-dependent method selection. Here, a conflict regime refers to the dominant obstruction observed in a short profile, namely persistent directional misalignment, gradient-norm imbalance, low conflict, or transient conflict. This regime determines whether scalar reweighting, architectural separation, or no additional intervention is appropriate.

The central distinction is that PINN losses are not independent tasks in the usual multi-task learning (MTL) sense, because the PDE residual, BC, IC, and data terms all constrain the same physical field $u(x)$. Their gradient geometry is therefore tied to the PDE operator and the physics-loss decomposition, with all losses acting through a shared physical output instead of distinct task heads. This distinction motivates the guiding question of our study:

> *When can PINN gradient conflicts be resolved through scalar reweighting, and when do PDE-induced persistent directional conflicts require explicit separate parameter subspaces?*

No prior work provides a theory-guided answer to determine when each algorithm class is appropriate, validated across different operator families, dimensions, and architectures. We fill this gap with a diagnostic framework that determines whether a conflict warrants intervention. This framework is supported by regime-transition theory and a kernel-level interpretation of per-loss adapters, and it adopts per-loss low-rank adapters [8] as lightweight architectural probes that explicitly instantiate separate loss-indexed parameter subspaces with a 15% parameter overhead. Their gains precisely track the predicted conflict regime.

Our contributions are as follows. First, we show that PINN gradient conflict forms regimes tied to loss count and the problem class. Standard forward $K = 3$ PDEs and natural $K = 4$ thermoelasticity are dominated by directional conflict, whereas $K = 3$ inverse problems and natural $K = 5$ and $K = 6$ multi-physics problems are mostly magnitude-dominated. Second, we introduce the loss-decomposition-tied, shared-output adapter construction with fixed UAM and conflict-conditioned CAM/LCAM mixers. The generic low-rank primitive and FAMO/GradNorm are inherited; uniform averaging is the deliberately simple default mixer. Third, we explain the regime structure with Theorem 1, operator guidance, and a blockwise neural tangent kernel (NTK) view, and validate a 1000-step diagnostic across 60+ PDE configurations, alternative backbones, and optimizer and budget controls.

2 Method

2.1 Problem Setup

Consider a PDE-constrained problem defined on a domain $\Omega \subset \mathbb{R}^d$ with boundary $\partial\Omega$, written at the residual level as

$$\mathcal{R}_{\mathrm{PDE}}[u](x) := \mathcal{F}[u](x) - f(x) = 0 \quad \text{for } x \in \Omega,$$
$$\mathcal{R}_m[u](x) = 0 \quad \text{for } x \in \Gamma_m, \quad m = 1, \dots, M.$$

Here $\mathcal{F}$ is the differential operator, f is a forcing or source term, and the residuals $\mathcal{R}_m$ on the sets Γ_m encode BC, IC, interface, or observation constraints. A PINN approximates the unknown physical field u, the state variable or physical quantity constrained by the governing equations, using a neural network u_θ trained by minimizing the composite objective in Eq. 1. In the PDE setting, its components typically include $\mathcal{L}_{\mathrm{PDE}}(\theta)$, $\mathcal{L}_{\mathrm{BC}}(\theta)$, $\mathcal{L}_{\mathrm{IC}}(\theta)$, ...

We formalize the resulting multi-loss training pathology as follows.

Definition 1 (Gradient conflict)**.** *Two loss terms $\mathcal{L}_i, \mathcal{L}_j$ exhibit gradient conflict at parameters θ if their nonzero gradients have a negative cosine similarity:*

$$\cos(g_i, g_j) = \frac{g_i^\top g_j}{\|g_i\|\|g_j\|} < 0, \tag{2}$$

where $g_k = \nabla_\theta \mathcal{L}_k(\theta)$. We further define the regularized, magnitude-aware conflict score used in profiling and reporting as:

$$C_{ij} = \left[\frac{g_i^\top g_j}{(\|g_i\| + \varepsilon_g)(\|g_j\| + \varepsilon_g)}\right]_- \cdot \left(1 + \left|\log \frac{\|g_i\| + \varepsilon_g}{\|g_j\| + \varepsilon_g}\right|\right), \tag{3}$$

where $[\cdot]_- = \max(0, -\cdot)$ retains only the negative cosine similarity, and $\varepsilon_g > 0$ is a fixed numerical floor. The magnitude ratio term accounts for gradient norm imbalance, which is a critical factor in PINNs because PDE residual gradients can be orders of magnitude larger than boundary gradients [41].

In all experiments, we use $\varepsilon_g = 10^{-12}$ only to avoid division by zero or $\log 0$ in degenerate cases. When both gradient norms are strictly positive and much larger than ε_g, Eq. 3 reduces to the usual cosine and log-ratio expression.

Definition 1 is used as a diagnostic tool for identifying a pathology observed during PINN training. Remark 3 and Appendix F explain how PDE residuals, BC errors, operator coupling, and network-Jacobian correlations can induce negative alignment between loss gradients. Empirical evidence in the appendix connects persistent conflict to structured errors, adapter specialization, thermoelastic energy mismatch, and unresolved directional conflict under reweighting. For K loss terms, the number of pairwise conflict terms grows as $\binom{K}{2}$, creating $\binom{K}{2}$ potential interference sources that must be managed.

2.2 Profiling Gradient Conflict for Regime Selection

To diagnose gradient conflicts before intervention, we propose a profiling stage for the unmodified PINN. We run a short Vanilla optimization window of $T_{\mathrm{prof}} = 1000$ steps using the planned optimizer and learning rate (LR), and compute raw, unweighted loss-specific gradients $g_k^{(t)} := \nabla_\theta \mathcal{L}_k(\theta_t)$ before optimizer transformation. The profile is repeated when these settings change. For each step, we summarize directional conflict by the fraction and average strength of negative cosine pairs,

$$f_{\mathrm{neg}}^{(t)} := \frac{|\{(i,j) : i < j, \cos(g_i^{(t)}, g_j^{(t)}) < 0\}|}{\binom{K}{2}}, \qquad D^{(t)} := f_{\mathrm{neg}}^{(t)} |\bar{c}_-^{(t)}|,$$

where $\bar{c}_-^{(t)}$ is the mean cosine over the negative pairs and is set to 0 when no negative pair exists. We summarize magnitude imbalance by the coefficient of variation of gradient norms,

$$M^{(t)} := \frac{\mathrm{std}_k(\|g_k^{(t)}\|)}{K^{-1}\sum_k \|g_k^{(t)}\| + \varepsilon_g}.$$

The profiled summaries are

$$\widehat{f}_{\text{neg}} := \frac{1}{T_{\text{prof}}} \sum_t f_{\text{neg}}^{(t)}, \quad \widehat{D} := \frac{1}{T_{\text{prof}}} \sum_t D^{(t)}, \quad \widehat{M} := \frac{1}{T_{\text{prof}}} \sum_t M^{(t)}, \quad \widehat{R} := \frac{\widehat{D}}{\widehat{M} + \varepsilon_R},$$

where $\varepsilon_R = 10^{-8}$. The ratio $\widehat{R}$ is a practical surrogate for the theory-level directional-to-magnitude ratio $\mathcal{U}_K$, not a plug-in estimator: $\widehat{M}$ uses a coefficient-of-variation proxy, whereas Theorem 1 analyzes a max-to-mean imbalance. For deployment, we also measure persistence by comparing the first and last thirds of the profile,

$$P := \frac{\bar{f}_{\text{neg}}^{\text{late}}}{\max(\bar{f}_{\text{neg}}^{\text{early}}, \varepsilon_P)},$$

where $\varepsilon_P = 10^{-8}$; we also compute the least-squares slope against normalized time $(t-1)/(T_{\text{prof}} - 1)$. Low $\widehat{f}_{\text{neg}}$ or rapidly decaying conflict indicates that scalar reweighting is usually sufficient, whereas persistent conflict motivates the adapter intervention described below. Appendix C.2 gives the zero-gradient convention and full profile definitions, and Appendix Algorithm 2 gives executable pseudocode. These profiled diagnostics determine whether scalar reweighting is sufficient or whether explicit loss-indexed architectural separation is needed.

The architectural intervention used in persistent-conflict regimes has two core components. The first is a shared feature trunk, meaning the common network body that computes the hidden representation used by all losses. The second is a set of per-loss low-rank adapters, whose parameters form explicit loss-indexed subspaces and whose outputs are mixed back into the shared representation. The default fixed mixer is Uniform Adapter Mixing (UAM), which uses equal adapter weights. For $K \geq 4$, these components can be augmented with Conflict-aware Adapter Mixing (CAM) variants, including Layerwise Conflict-aware Adapter Mixing (LCAM). The CAM and LCAM definitions apply to any K; our $K = 3$ stress-test ablations use them only to assess whether dynamic mixing is worthwhile in the low-K regime. See Appendix C for the shared trunk architecture, orthogonality regularization, conflict-aware mixing variant, and the full training algorithm in Algorithm 1; the main text focuses on the shared-output adapter construction and the default method choice.

2.3 Per-Loss Low-Rank Adapters

Why per-loss adapters for PINNs. In PINNs, all losses constrain the same physical field $u(x)$ rather than representing independent tasks with separate outputs. This shared-output structure means that the shared trunk naturally captures the common representation, while adapters provide targeted adjustments to the gradient flow. When the PDE residual gradient g_{PDE} conflicts with the BC gradient g_{BC}, the standard shared-parameter approach forces a destructive compromise. Per-loss adapters instead introduce structured low-rank parameter blocks whose Jacobian contributions are encouraged to occupy distinct adapter subspaces through the orthogonality loss in Eq. 10 and are interpreted via the blockwise NTK decomposition in Appendix D.2; because these blocks are mixed into the hidden state, as shown in Eq. 6, they can decorrelate the Jacobians of different losses while maintaining a coherent shared output through the trunk. Thus, the proposed adapters explicitly create separate loss-indexed parameter subspaces within a single-output PINN, rather than replacing the shared physical field with separate task heads. Figure 2 summarizes this block-level pipeline.

The core innovation is attaching K low-rank adapters to the shared trunk, with one for each loss term. At each residual block ℓ, loss k has an associated adapter:

$$A_k^{(\ell)}(h) = B_k^{(\ell)} \cdot \phi(D_k^{(\ell)} h), \tag{4}$$

where $D_k^{(\ell)} \in \mathbb{R}^{r \times d_h}$ is the down-projection, $B_k^{(\ell)} \in \mathbb{R}^{d_h \times r}$ is the up-projection, ϕ is the adapter nonlinearity, and $r = 16$ is the adapter rank. Let $h_{\text{trunk}}^{(\ell)}$ denote the output of the ℓth shared residual block when applied to the current hidden state $h^{(\ell-1)}$. Each adapter is assigned a latent mixing score $\rho_k^{(\ell)} \in [0, 1]$ for loss k at layer ℓ. The normalized rule below depends only on the relative values of these scores and is well-defined whenever at least one adapter is active. In fixed UAM, we use uniform adapter weights $\pi_k^{(\ell)} = 1/K$; equivalently, any common constant score $\rho_k^{(\ell)} \equiv c < 1$ gives

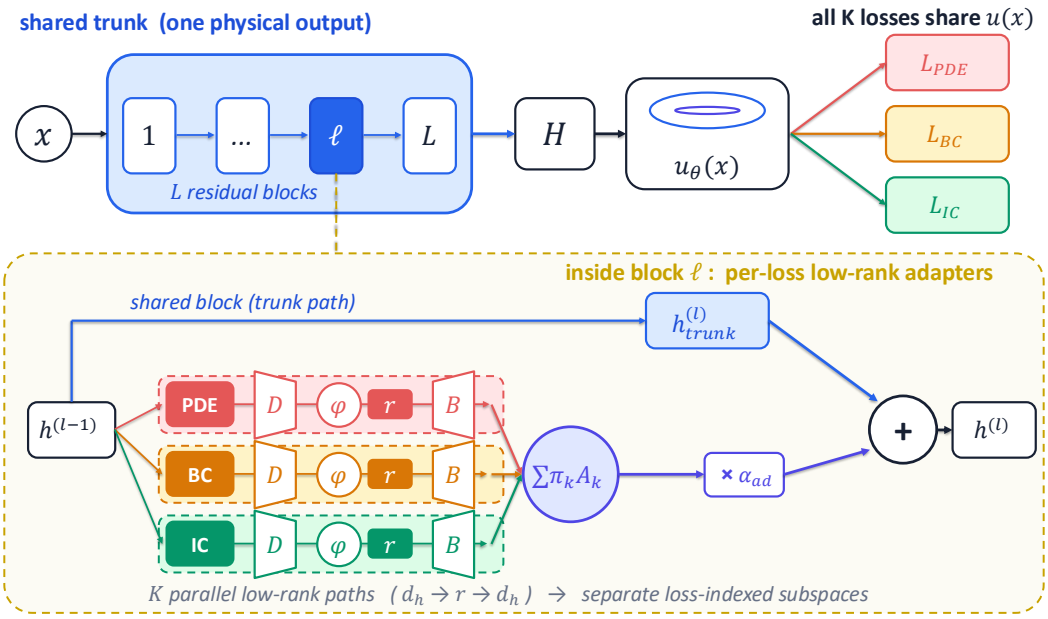

Figure 2: Block-level pipeline for per-loss low-rank adapters in a shared-output PINN. At residual block ℓ, the shared trunk produces $h^{(\ell)}_{\text{trunk}}$, while each loss-indexed adapter maps the previous hidden state through a rank-r bottleneck formed by a down-projection followed by an up-projection. The mixer forms the weighted adapter correction $\sum_k \pi_k^{(\ell)} A_k^{(\ell)}$ and adds it to the shared trunk state, as shown in Eq. 6. A single readout H then produces the physical field $u_\theta(x)$, from which all PDE, BC, IC, or data losses are evaluated; the adapters therefore separate gradient subspaces without creating separate physical outputs.

the same weights, and our implementation stores $c = 0.5$. We then combine the adapters through the normalized mixing weights

$$\pi_k^{(\ell)} := \frac{1 - \rho_k^{(\ell)}}{\sum_{j=1}^{K} (1 - \rho_j^{(\ell)})} \tag{5}$$

These weights satisfy $\sum_{k=1}^{K} \pi_k^{(\ell)} = 1$, and they define the actual shared hidden-state update

$$h^{(\ell)} = h^{(\ell)}_{\text{trunk}} + \alpha_{\text{ad}} \sum_{k=1}^{K} \pi_k^{(\ell)} A_k^{(\ell)}(h^{(\ell-1)}), \tag{6}$$

where α_{ad} controls the overall strength of the adapter correction added to the shared trunk representation. The final output is the single physical field $u_\theta(x) = H(h^{(L)})$, where H is the final linear readout. Each loss $\mathcal{L}_k$ is evaluated on quantities derived from this same shared output, such as boundary values, PDE residuals, or IC mismatches. The loss-indexed mixer therefore modulates only how strongly each adapter subspace perturbs the common representation; it does not create separate physical outputs.

2.4 Method variants and recommended defaults

For compact reporting, GradNorm (GN) denotes the existing GradNorm method, while CAM refers to our global mixing variant, and LCAM to its layer-wise variant. FAMO+UAM combines uniform adapter mixing with loss-value-based reweighting and is our default adapter configuration, with a training-time cost of about $1.3\times$ Vanilla. Strict time budgets favor adapter-free FAMO (Section 4.4). GN+UAM is an alternative for solving spectral-bias-dominated PDEs such as Helmholtz. Alternative adapter architectures and mixing variants are summarized in Appendix Tables 40 and 41.

2.5 Complementarity

Adapters alleviate destructive gradient interference through structured low-rank parameter blocks while preserving one physical field. Every loss still depends on the shared output and adapter bank; loss-indexed blocks do not imply independent losses. Positive scalar weights balance magnitudes, UAM changes the learned Jacobian, and SOAP conditions the full-network update. These interventions can compose: matched gate controls improve geometric-mean error by $1.31\times$ under AdamW and $1.25\times$ on top of SOAP, with FAMO fixed (Appendix J.5).

3 Theoretical Analysis

We organize the theory around the directional-to-magnitude ratio

$$\mathcal{U}_K := \frac{D_K}{M_{\text{eff}}(K)},$$

where D_K denotes pairwise negative directional conflict and $M_{\text{eff}}(K)$ denotes update-level magnitude imbalance in the model analyzed below. The theorem below gives the formal transition statement, while Appendix C.2 gives the profiling definitions used for the empirical diagnostic quantities. See Appendix D for additional supporting theory.

The empirical observation that the balance between directional conflict and magnitude imbalance shifts with the number of loss terms is formalized here through a fixed-model transition theorem for the ratio $\mathcal{U}_K$. Appendix D.1 gives the supporting conditional concentration result for D_K, interpretation remarks, and the expectation-level finite-range corollary.

Theorem 1 (Regime transition in a fixed spherical model)**.** *Fix a subspace $V \subseteq \mathbb{R}^p$ with $\dim(V) = d_{\text{eff}}$. Assume $g_k = m_k u_k$, where $u_1, \dots, u_K$ are independent and uniformly distributed on $V \cap \mathbb{S}^{p-1}$. In orthonormal coordinates on V, these are independent and uniformly distributed on $\mathbb{S}^{d_{\text{eff}}-1}$. The magnitudes $m_1, \dots, m_K$ are positive i.i.d. random variables, independent of $u_1, \dots, u_K$, and satisfy*

$$\mu_m := \mathbb{E}[m_1] \in (0, \infty), \qquad \mu_d := \frac{\Gamma(d_{\text{eff}}/2)}{2\sqrt{\pi}\Gamma((d_{\text{eff}}+1)/2)}.$$

Write

$$A_K := \max_{1 \le k \le K} m_k, \qquad \bar{m} := \frac{1}{K}\sum_{k=1}^{K} m_k,$$

$$M_{\text{eff}}(K) := \frac{A_K}{\bar{m}}, \qquad \mathcal{U}_K := \frac{D_K}{M_{\text{eff}}(K)}.$$

Then, we have

$$\mathbb{E}[\mathcal{U}_K] = \mu_d \mathbb{E}\left[\frac{\bar{m}}{A_K}\right]. \tag{7}$$

Moreover, the following rate statements hold.

Exponential-tail magnitudes. *Suppose there exist constants $\nu > 0$, $0 < a \le b < \infty$, and $t_0 > 0$ such that for all $t \ge t_0$,*

$$\exp\left(-\frac{bt}{\nu}\right) \le \Pr(m_1 > t) \le \exp\left(-\frac{at}{\nu}\right).$$

Then

$$\mathbb{E}[\mathcal{U}_K] = \Theta(d_{\text{eff}}^{-1/2}/\log K).$$

Pareto-tail magnitudes. *Suppose*

$$\Pr(m_1 > t) = \left(\frac{t_0}{t}\right)^{\alpha}, \qquad t \ge t_0,$$

for some $\alpha > 1$. Then

$$\mathbb{E}[\mathcal{U}_K] = \Theta(d_{\text{eff}}^{-1/2} K^{-1/\alpha}).$$

Table 1: 10K-epoch benchmark I. Relative L_2 error on Burgers, Helmholtz, and Allen–Cahn. We report the mean $\pm$ std with $d_h = 128$ and 5 seeds unless otherwise noted. Best values are bold and second-best values are underlined.

Method	Burgers	Helmholtz	Allen–Cahn
Baselines			
Vanilla	$(8.4 \pm 2.4) \times 10^{-3}$	$(2.8 \pm 1.1) \times 10^{-2}$	$(1.60 \pm 0.33) \times 10^{-4}$
Loss Weighting			
FAMO	$\underline{(4.7 \pm 0.14) \times 10^{-3}}$	$(1.4 \pm 0.63) \times 10^{-3}$	$(2.17 \pm 0.2) \times 10^{-4}$
GradNorm	$(1.3 \pm 1.9) \times 10^{-2}$	$\underline{(6.40 \pm 6.4) \times 10^{-4}}$	$(5.7 \pm 0.83) \times 10^{-3}$
Causal	$(7.5 \pm 3) \times 10^{-3}$ [3]	$(5.3 \pm 1.1) \times 10^{-2}$ [3]	$(1.71 \pm 0.096) \times 10^{-4}$ [3]
SelfAdapt	$(1.1 \pm 0.8) \times 10^{-2}$ [3]	$(1.2 \pm 0.5) \times 10^{-2}$ [3]	$(1.70 \pm 0.39) \times 10^{-4}$ [3]
Uncertainty	$(1.1 \pm 0.7) \times 10^{-2}$	$(1.8 \pm 1.7) \times 10^{-2}$ [4]	$\underline{(1.31 \pm 0.27) \times 10^{-4}}$
Gradient Surgery			
ConFIG	$(9.6 \pm 2.7) \times 10^{-2}$	$\mathbf{(2.49 \pm 0.77) \times 10^{-4}}$ [3]	1.4×10^{-2} [1]
PCGrad	$(4.8 \pm 0.11) \times 10^{-3}$	$(9.06 \pm 2.8) \times 10^{-4}$ [4]	2.25×10^{-4} [1]
CAGrad	$(1.5 \pm 1) \times 10^{-2}$	$(3.3 \pm 1.5) \times 10^{-2}$	1.96×10^{-4} [1]
NashMTL	$(6.5 \pm 6.7) \times 10^{-2}$	$(1.65 \pm 1.11) \times 10^{-1}$	$(3.13 \pm 0.43) \times 10^{-4}$
AlignedMTL	$(5.1 \pm 0.34) \times 10^{-3}$	$(1.1 \pm 0.33) \times 10^{-3}$ [4]	1.81×10^{-4} [1]
Architecture (Ours)			
UAM	$(4.9 \pm 0.075) \times 10^{-3}$	$(1.5 \pm 0.1) \times 10^{-2}$	$\mathbf{(1.19 \pm 0.21) \times 10^{-4}}$
FAMO+UAM	$(4.8 \pm 0.082) \times 10^{-3}$	$(1.1 \pm 0.46) \times 10^{-3}$	$(1.84 \pm 0.28) \times 10^{-4}$
GN+UAM	$\mathbf{(4.6 \pm 0.12) \times 10^{-3}}$	$(1.0 \pm 0.44) \times 10^{-3}$	$(1.6 \pm 0.21) \times 10^{-3}$
FAMO+LCAM	$(4.8 \pm 0.17) \times 10^{-3}$	$(1.3 \pm 0.41) \times 10^{-3}$	$(2.59 \pm 0.39) \times 10^{-4}$
GN+LCAM	$\mathbf{(4.6 \pm 0.12) \times 10^{-3}}$	$(7.81 \pm 2.8) \times 10^{-4}$	$(2.3 \pm 0.45) \times 10^{-3}$

Superscript $[n]$ denotes seed count when <5.

See Appendix D.1 for supporting statements and Appendix E for the full proofs. Empirically, we use the short-run profile described in Section 2.2 and the ratio $\widehat{R} := \widehat{D}/(\widehat{M} + \varepsilon_R)$ as a practical surrogate for $\mathcal{U}_K$. Appendix C.2 gives the zero-gradient convention and additional reporting details for the profiled summaries.

4 Experiments

We evaluate on 60+ PDE configurations spanning forward, inverse, multi-physics, parameter-varying, and high-dimensional settings. See Appendix G and Appendix I for detailed benchmark definitions, the natural K-scaling setup, compared methods, evaluation protocol, computational-cost analysis, and supplementary diagnostics. The main text focuses on the core empirical findings and the resulting deployment heuristic. The forward-benchmark tables below refer to the conflict-free gradient method (ConFIG) and aligned multi-task learning (AlignedMTL) by their standard short names, alongside the previously defined PCGrad, CAGrad, NashMTL, UAM, GN, and LCAM labels.

4.1 Main Results on Forward Benchmarks

Tables 1 and 2 present the full result matrices for five main PDE benchmarks at 10K epochs. Table 1 groups Burgers, Helmholtz, and Allen–Cahn, while Table 2 groups convection-diffusion (Conv-Diff), Klein–Gordon, and the average rank.

Architectural and loss-weighting combinations are strongest. On 10K-epoch Burgers, GN+UAM and FAMO+UAM are statistically tied at about 4.6×10^{-3}. This Burgers tie corresponds to a $1.8\times$ improvement over Vanilla, with a 95% confidence interval (CI) of $[1, 2]$ and a Cohen's $d = 2.1$. On Helmholtz, GN+UAM with 1.0×10^{-3} achieves a $28\times$ improvement over Vanilla, with a CI $[18, 46]$ and $d = 3.4$. FAMO+UAM achieves $25\times$, with a CI $[16, 39]$ and the same effect size $d = 3.4$. The gap persists across width and three-dimensional (3D) evaluations in Table 26. All architecture-based methods complete the five main forward benchmarks, whereas CAGrad and NashMTL diverge on inverse problems. ConFIG wins the original Helmholtz comparison at 2.49×10^{-4} but reaches 0.81

Table 2: 10K-epoch benchmark II. Relative L_2 error on Conv-Diff and Klein–Gordon together with the average rank. We report mean $\pm$ std with $d_h = 128$ and 5 seeds, unless otherwise noted.

Method	Conv-Diff	Klein–Gordon	Avg. Rank
Baselines			
Vanilla	$(3.55 \pm 0.74) \times 10^{-4}$	$(4.6 \pm 1) \times 10^{-2}$	7.8
Loss Weighting			
FAMO	$(3.95 \pm 1.4) \times 10^{-4}$	$(2.5 \pm 0.39) \times 10^{-3}$	6.4
GradNorm	$(1.8 \pm 0.46) \times 10^{-3}$	$(1.13 \pm 0.57) \times 10^{-1}$	11.4
Causal	$(2.98 \pm 0.39) \times 10^{-4}$ [3]	$(9.80 \pm 0.13) \times 10^{-1}$ [3]	9.6
SelfAdapt	$(4.12 \pm 0.91) \times 10^{-4}$ [3]	$(3.8 \pm 0.34) \times 10^{-2}$ [3]	8.1
Uncertainty	$(3.66 \pm 0.73) \times 10^{-4}$	$(4.89 \pm 4.87) \times 10^{-1}$	9.3
Gradient Surgery			
ConFIG	$(8.12 \pm 4.44) \times 10^{-1}$	$(2.39 \pm 3) \times 10^{-1}$	12.6
PCGrad	$(3.92 \pm 1.7) \times 10^{-4}$	$(9.0 \pm 3.7) \times 10^{-3}$	6.0
CAGrad	$(4.63 \pm 1.5) \times 10^{-4}$	$(4.7 \pm 1) \times 10^{-2}$	11.2
NashMTL	$(5.41 \pm 1.1) \times 10^{-4}$	$(3.6 \pm 1.1) \times 10^{-2}$	12.0
AlignedMTL	$(4.16 \pm 1.4) \times 10^{-4}$	$(1.14 \pm 2.1) \times 10^{-1}$	8.7
Architecture (Ours)			
UAM	$\mathbf{(2.64 \pm 0.32) \times 10^{-4}}$	$(4.3 \pm 0.7) \times 10^{-2}$	5.4
FAMO+UAM	$(2.70 \pm 0.21) \times 10^{-4}$	$\mathbf{(1.6 \pm 0.38) \times 10^{-3}}$	**4.4**
GN+UAM	$(1.0 \pm 0.19) \times 10^{-3}$	$(4.8 \pm 1.1) \times 10^{-2}$	8.5
FAMO+LCAM	$(3.07 \pm 0.56) \times 10^{-4}$	$\mathbf{(1.6 \pm 0.11) \times 10^{-3}}$	5.9
GN+LCAM	$(1.4 \pm 0.41) \times 10^{-3}$	$(6.2 \pm 1.7) \times 10^{-2}$	8.8

Table 3: Optimizer substitution at 10K updates: terminal relative L_2, mean $\pm$ sample std over three seeds. SOAP uses the PINN settings of Wang et al. [45]. HF uses $\omega = 20$ and KG-Heavy uses $m = 2$.

PDE	SOAP	SOAP+FAMO	SOAP+FAMO+UAM
Helmholtz	$(4.11 \pm 2.03) \times 10^{-3}$	$(2.69 \pm 0.68) \times 10^{-4}$	$\mathbf{(2.04 \pm 0.42) \times 10^{-4}}$
Klein–Gordon	$(7.58 \pm 4.62) \times 10^{-3}$	$(1.32 \pm 2.26) \times 10^{-1}$	$\mathbf{(6.49 \pm 2.95) \times 10^{-4}}$
Allen–Cahn	$\mathbf{(3.62 \pm 0.30) \times 10^{-5}}$	$(4.04 \pm 0.18) \times 10^{-5}$	$(4.26 \pm 0.31) \times 10^{-5}$
Helmholtz-HF	$(8.39 \pm 2.37) \times 10^{-1}$	$(6.55 \pm 5.64) \times 10^{-1}$	$\mathbf{(1.27 \pm 0.50) \times 10^{-3}}$
KG-Heavy	$(5.61 \pm 0.15) \times 10^{-3}$	$(1.11 \pm 0.18) \times 10^{-3}$	$\mathbf{(6.87 \pm 1.09) \times 10^{-4}}$

on Conv-Diff. A separate eight-seed, LR-tuned Helmholtz control yields a near tie under AdamW and 36.7% lower mean error for FAMO+UAM under the same SOAP optimizer (Appendix N.3.1). No single method dominates, and the right method depends on the PDE conflict profile. Both causal training and self-adaptive weights further confirm this PDE dependence. Causal training matches FAMO+UAM on Burgers but catastrophically fails on Klein–Gordon, reaching 0.98 at 10K and 0.92 at 20K, whereas FAMO+UAM reaches 1.6×10^{-3}. Self-adaptive weights also show high variance across PDEs. Additional PDE checks in Appendix M.1.6 confirm these findings. On Klein–Gordon, FAMO+UAM achieves a $29\times$ gain, with a CI $[24, 36]$ and $d = 6.1$. On Darcy flow, the gain is $2.0\times$, with a CI $[1, 3]$ and $d = 2.3$. Appendix Figure 4 provides a qualitative view of the mechanism underlying regime dependence by visualizing the learned adapter contributions in a persistent-conflict case and a low-conflict control.

Stronger optimization. Table 3 extends the regime evidence to SOAP: the complete configuration improves on SOAP in all 12 paired runs on the four persistent-conflict PDEs, while SOAP wins the Allen–Cahn control. At each seed's SOAP-10K time budget, gains remain $16.4\times$, $9.14\times$, $208\times$, and $6.48\times$ on Helmholtz, Klein–Gordon, Helmholtz-HF, and KG-Heavy. Appendix J separates these complete-configuration comparisons from matched adapter-gate controls.

Table 4: Inverse problem results at 10K epochs with $d_h = 128$ and 3 to 5 seeds. Full-parameter-space gradient surgery methods diverged on these benchmarks and are omitted here. †GradNorm entries use a simplified loss-ratio variant rather than the reference gradient-norm algorithm used in Tables 1 and 2.

Method	Inverse Burgers ($K = 3$)	Inverse Heat ($K = 4$)	Inverse Poisson ($K = 3$)
Vanilla	$(8.0 \pm 0.7) \times 10^{-2}$	$(2.2 \pm 3.8) \times 10^{-2}$	$(2.71 \pm 0.65) \times 10^{-1}$
FAMO	$(6.5 \pm 0.4) \times 10^{-2}$	$(1.1 \pm 2.3) \times 10^{-2}$	$\mathbf{(2.3 \pm 1.6) \times 10^{-3}}$
GradNorm†	$\mathbf{(5.9 \pm 0.2) \times 10^{-2}}$	$(1.1 \pm 0.1) \times 10^{-3}$	$(1.0 \pm 1.5) \times 10^{-2}$
FAMO+UAM	$(7.2 \pm 0.4) \times 10^{-2}$	$(2.8 \pm 2.6) \times 10^{-4}$	$(2.8 \pm 0.9) \times 10^{-3}$
GN+UAM†	$(6.6 \pm 0.4) \times 10^{-2}$	$(8.7 \pm 0.9) \times 10^{-4}$	$(3.4 \pm 0.9) \times 10^{-3}$
FAMO+LCAM	$(7.3 \pm 0.2) \times 10^{-2}$	$\mathbf{(2.6 \pm 0.74) \times 10^{-4}}$	$(3.2 \pm 0.7) \times 10^{-3}$

4.2 Inverse Problems and Robustness

Full-parameter-space gradient surgery methods, namely, PCGrad, CAGrad, NashMTL, and ConFIG, produce invalid values within the first 50 to 200 epochs during inverse-problem runs due to the $10^4\times$ gradient norm mismatch between network weights and scalar physical constants. Table 36 in the appendix gives the details. At 10K epochs in Table 4, a K-dependent pattern emerges. For $K = 3$ inverse problems, FAMO and GradNorm without adapters outperform adapter combinations by 10% to 25%. On the $K = 4$ inverse heat problem, adapter variants substantially improve stability compared with Vanilla, with FAMO+LCAM numerically best and FAMO+UAM close behind, while Vanilla fails catastrophically on one of three seeds. Parameter-grouped surgery variants, such as PCGrad-Grouped and CAGrad-Grouped, resolve divergence but remain far worse than the direct baselines shown in Table 4, which confirms that the failure is structural. Gradient surgery corrupts the low-dimensional physical-parameter learning signal.

4.3 Multi-Physics and K-Scaling

The natural scaling studies clarify the regime picture. In natural $K = 4$ thermoelasticity, FAMO+UAM improves FAMO from 1.04×10^{-1} to 7.95×10^{-2} (Table 21), demonstrating complementarity for an intrinsic physics-loss decomposition. Appendix Figure 10 shows that this improvement is physically structured. The adapter variant mainly improves the coupled displacement branch and the associated elastic-energy exchange rather than merely reducing a pixel-wise error metric.

Natural higher-K validation. Natural multi-physics systems with $K = 5$ reactive transport and $K = 6$ magnetohydrodynamics (MHD), as discussed in Appendix M.5, fall into the empirically magnitude-dominated regime. FAMO alone outperforms FAMO+UAM, consistent with the low-profile proxy $\widehat{R}$. Beyond the main suite, FAMO+UAM improves FAMO by $2.47\times$ on Helmholtz-3D, $2.59\times$ on Poisson-2D, $1.74\times$ on KG-Heavy, and $5.73\times$ on Poisson-Multiscale (Appendix M.1).

4.4 Method Selection and Practical Guidelines

The results above reveal a clear regime structure that motivates a practical, profiling-based deployment heuristic. In practice, budget-constrained settings favor FAMO. Forward problems with $K = 3$ often benefit from FAMO+UAM. Inverse problems with $K = 3$ favor FAMO or GradNorm without adapters, while inverse problems with $K = 4$ commonly benefit from FAMO+UAM. The natural higher-K multi-physics systems in our suite are better handled by FAMO alone. The resulting deployment rule is diagnostic-first. Run the 1000-step Vanilla profile from Section 2.2; if the profiled Vanilla error is already below 10^{-3} or $\widehat{f}_{\text{neg}} < 0.05$, use FAMO alone. If the conflict persists with $P > 0.8$, use FAMO+UAM; if it decays with $P < 0.5$ and slope < -0.02, use FAMO alone. For ambiguous profiles, report a borderline decision; a strict budget favors FAMO, while three short profiles can be aggregated when greater confidence is needed. The ratio $\widehat{R}$ is reported as a compact direction-to-magnitude summary, while P distinguishes persistent conflict from transient negative pairs. Appendix Algorithm 2 gives the full pseudocode; the computational-cost analysis and the qualitative profiling-to-outcome bridge are provided in Appendix I.

Diagnostic robustness and time budgets. With thresholds fixed, 100 profiles spanning five PDEs, five seeds, three AdamW LRs, and nominal-LR SOAP select UAM in all 40 Helmholtz/Klein–Gordon cases. Boundary cases vary with the training configuration (Appendix J.3). Under matched Vanilla-10K time budgets, FAMO is best on Burgers/Helmholtz, and FAMO+UAM is best on Klein–Gordon; LCAM's additional cost is not recovered (Appendix J.2). Thus strict budgets favor FAMO, while persistent-conflict gains can justify UAM's modest overhead.

Baseline fairness. The original benchmark tables share the same trunk, outer AdamW optimizer, scheduler, and nominal 10K budget, with PDE-specific learning rates stated in Appendix H. Most entries use 5 seeds. Any results with fewer seeds are explicitly marked in Tables 1 and 2. NashMTL is additionally swept at a learning rate of 3×10^{-4}. ConFIG is swept over $\{10^{-4}, 5 \times 10^{-4}, 10^{-3}, 2 \times 10^{-3}, 5 \times 10^{-3}\}$, as reported in Appendix N.3. GradNorm follows the reference algorithm, with gradient norms computed with respect to the last shared layer. Implementation details appear in Appendix H. The source code is provided in the supplementary material.

Domain decomposition and inference. Extended physics-informed neural networks (XPINNs) partition the input space [12], whereas per-loss adapters partition the gradient space. These are orthogonal interventions, as Tables 42 and 43 show. Adapters increase the model parameter count by 15% to 41% in the configurations reported in Tables 8 and 42. See Appendix I for training-curve, NTK-weighting, and rank-ablation diagnostics.

5 Conclusion

We characterized gradient conflict in PINNs and showed that its structure is regime-dependent rather than uniformly requiring a single mitigation strategy. Across 60+ PDE configurations, directional conflict dominates the standard forward $K = 3$ benchmarks and the natural $K = 4$ thermoelastic system, making loss-indexed adapters combined with reweighting the most effective choice. In contrast, $K = 3$ inverse problems and natural $K = 5$ and $K = 6$ multi-physics systems are largely magnitude-dominated, often favoring reweighting alone, and expose the unreliability of full-parameter-space gradient surgery on heterogeneous parameter spaces. A 1000-step profiling rule turns this observation into a practical method-selection heuristic, while the regime-transition theorem and the blockwise NTK view explain why adapter gains appear only in the appropriate conflict regime. Overall, our results support a diagnostic-first workflow for selecting PINN training interventions.

Acknowledgements

B.J.K. was supported by JSPS KAKENHI Grant Number JP26K21295. This research used resources of the Argonne Leadership Computing Facility under the ALCF project RobustViT and resources of the Oak Ridge Leadership Computing Facility under OLCF Project ID CSC704 through Director's Discretionary allocation awards. The authors declare no competing interests.

References

[1] Rafael Bischof and Michael A. Kraus. Multi-Objective Loss Balancing for Physics-Informed Deep Learning. *CoRR*, abs/2110.09813, 2021.

[2] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. GradNorm: Gradient Normalization for Adaptive Loss Balancing in Deep Multitask Networks. In *ICML*, pages 793–802, 2018.

[3] Arka Daw, Jie Bu, Sifan Wang, Paris Perdikaris, and Anuj Karpatne. Mitigating Propagation Failures in Physics-informed Neural Networks using Retain-Resample-Release (R3) Sampling. In *ICML*, pages 7264–7302, 2023.

[4] Chris Fifty, Ehsan Amid, Zhe Zhao, Tianhe Yu, Rohan Anil, and Chelsea Finn. Efficiently Identifying Task Groupings for Multi-Task Learning. In *NeurIPS*, pages 27503–27516, 2021.

[5] Zhongkai Hao, Jiachen Yao, Chang Su, Hang Su, Ziao Wang, Fanzhi Lu, Zeyu Xia, Yichi Zhang, Songming Liu, Lu Lu, and Jun Zhu. PINNacle: A Comprehensive Benchmark of Physics-Informed Neural Networks for Solving PDEs. In *NeurIPS*, 2024.

[6] A. Ali Heydari, Craig A. Thompson, and Asif Mehmood. SoftAdapt: Techniques for Adaptive Loss Weighting of Neural Networks with Multi-Part Loss Functions. *CoRR*, abs/1912.12355, 2019.

[7] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-Efficient Transfer Learning for NLP. In *ICML*, pages 2790–2799, 2019.

[8] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. In *ICLR*, 2022.

[9] Zheyuan Hu, Ameya D. Jagtap, George Em Karniadakis, and Kenji Kawaguchi. Augmented Physics-Informed Neural Networks (APINNs): A gating network-based soft domain decomposition methodology. *Eng. Appl. Artif. Intell.*, 126:107183, 2023.

[10] Youngsik Hwang and Dong-Young Lim. Dual Cone Gradient Descent for Training Physics-Informed Neural Networks. In *NeurIPS*, 2024.

[11] Arthur Jacot, Clément Hongler, and Franck Gabriel. Neural Tangent Kernel: Convergence and Generalization in Neural Networks. In *NeurIPS*, pages 8580–8589, 2018.

[12] Ameya D. Jagtap and George E. Karniadakis. Extended Physics-informed Neural Networks (XPINNs): A Generalized Space-Time Domain Decomposition based Deep Learning Framework for Nonlinear Partial Differential Equations. In *AAAI Spring Symposium: MLPS*, 2021.

[13] 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.

[14] Adrián Javaloy and Isabel Valera. RotoGrad: Gradient Homogenization in Multitask Learning. In *ICLR*, 2022.

[15] 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.

[16] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics. In *CVPR*, pages 7482–7491, 2018.

[17] Ehsan Kharazmi, Zhongqiang Zhang, and George Em Karniadakis. Variational Physics-Informed Neural Networks For Solving Partial Differential Equations. *CoRR*, abs/1912.00873, 2019.

[18] Ehsan Kharazmi, Zhongqiang Zhang, and George Em Karniadakis. hp-VPINNs: Variational Physics-Informed Neural Networks With Domain Decomposition. *CoRR*, abs/2003.05385, 2020.

[19] Aditi S. Krishnapriyan, Amir Gholami, Shandian Zhe, Robert M. Kirby, and Michael W. Mahoney. Characterizing possible failure modes in physics-informed neural networks. In *NeurIPS*, pages 26548–26560, 2021.

[20] Raphael Leiteritz and Dirk Pflüger. How to Avoid Trivial Solutions in Physics-Informed Neural Networks. *CoRR*, abs/2112.05620, 2021.

[21] Zongyi Li, Nikola Borislavov Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew M. Stuart, and Anima Anandkumar. Fourier Neural Operator for Parametric Partial Differential Equations. In *ICLR*, 2021.

[22] Xi Lin, Hui-Ling Zhen, Zhenhua Li, Qingfu Zhang, and Sam Kwong. Pareto Multi-Task Learning. In *NeurIPS*, pages 12037–12047, 2019.

[23] Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-Averse Gradient Descent for Multi-task learning. In *NeurIPS*, pages 18878–18890, 2021.

[24] Bo Liu, Yihao Feng, Peter Stone, and Qiang Liu. FAMO: Fast Adaptive Multitask Optimization. In *NeurIPS*, 2023.

[25] Qiang Liu, Mengyu Chu, and Nils Thuerey. ConFIG: Towards Conflict-free Training of Physics Informed Neural Networks. In *ICLR*, 2025.

[26] Shikun Liu, Edward Johns, and Andrew J. Davison. End-To-End Multi-Task Learning With Attention. In *CVPR*, pages 1871–1880, 2019.

[27] Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nat. Mach. Intell.*, 3(3):218–229, 2021.

[28] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H. Chi. Modeling Task Relationships in Multi-task Learning with Multi-gate Mixture-of-Experts. In *KDD*, pages 1930–1939, 2018.

[29] Levi D. McClenny and Ulisses M. Braga-Neto. Self-adaptive physics-informed neural networks. *J. Comput. Phys.*, 474:111722, 2023.

[30] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-Stitch Networks for Multi-task Learning. In *CVPR*, pages 3994–4003, 2016.

[31] Ben Moseley, Andrew Markham, and Tarje Nissen-Meyer. Finite basis physics-informed neural networks (FBPINNs): a scalable domain decomposition approach for solving differential equations. *Adv. Comput. Math.*, 49(4):62, 2023.

[32] Aviv Navon, Aviv Shamsian, Idan Achituve, Haggai Maron, Kenji Kawaguchi, Gal Chechik, and Ethan Fetaya. Multi-Task Learning as a Bargaining Game. In *ICML*, pages 16428–16446, 2022.

[33] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred A. Hamprecht, Yoshua Bengio, and Aaron C. Courville. On the Spectral Bias of Neural Networks. In *ICML*, pages 5301–5310, 2019.

[34] 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. *J. Comput. Phys.*, 378:686–707, 2019.

[35] Ozan Sener and Vladlen Koltun. Multi-Task Learning as Multi-Objective Optimization. In *NeurIPS*, pages 525–536, 2018.

[36] Dmitry Senushkin, Nikolay Patakin, Arseny Kuznetsov, and Anton Konushin. Independent Component Alignment for Multi-Task Learning. In *CVPR*, pages 20083–20093, 2023.

[37] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. In *ICLR*, 2017.

[38] Trevor Standley, Amir Zamir, Dawn Chen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. Which Tasks Should Be Learned Together in Multi-task Learning? In *ICML*, pages 9120–9132, 2020.

[39] N. Sukumar and Ankit Srivastava. Exact imposition of boundary conditions with distance functions in physics-informed deep neural networks. *CoRR*, abs/2104.08426, 2021.

[40] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains. In *NeurIPS*, 2020.

[41] Sifan Wang, Yujun Teng, and Paris Perdikaris. Understanding and Mitigating Gradient Flow Pathologies in Physics-Informed Neural Networks. *SIAM J. Sci. Comput.*, 43(5):A3055–A3081, 2021.

[42] Sifan Wang, Shyam Sankaran, and Paris Perdikaris. Respecting causality is all you need for training physics-informed neural networks. *CoRR*, abs/2203.07404, 2022.

[43] Sifan Wang, Xinling Yu, and Paris Perdikaris. When and why PINNs fail to train: A neural tangent kernel perspective. *J. Comput. Phys.*, 449:110768, 2022.

[44] Sifan Wang, Shyam Sankaran, Hanwen Wang, and Paris Perdikaris. An Expert's Guide to Training Physics-informed Neural Networks. *CoRR*, abs/2308.08468, 2023.

[45] Sifan Wang, Ananyae Kumar Bhartari, Bowen Li, and Paris Perdikaris. Gradient Alignment in Physics-informed Neural Networks: A Second-Order Optimization Perspective. *CoRR*, abs/2502.00604, 2025.

[46] Zirui Wang, Yulia Tsvetkov, Orhan Firat, and Yuan Cao. Gradient Vaccine: Investigating and Improving Multi-task Optimization in Massively Multilingual Models. In *ICLR*, 2021.

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

[48] Jeremy Yu, Lu Lu, Xuhui Meng, and George Em Karniadakis. Gradient-enhanced physics-informed neural networks for forward and inverse PDE problems. *CoRR*, abs/2111.02801, 2021.

[49] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient Surgery for Multi-Task Learning. In *NeurIPS*, 2020.

[50] Leo Zhiyuan Zhao, Xueying Ding, and B. Aditya Prakash. PINNsFormer: A Transformer-Based Framework For Physics-Informed Neural Networks. In *ICLR*, 2024.

Appendix Table of Contents

A List of Notation **16**

B Related Work **16**

C Supplementary Method Details **17**

C.1 Shared Feature Trunk 17

C.2 Conflict Metrics and Profiled Summaries 18

C.3 Additional Adapter Details 18

C.4 Conflict-Aware Adapter Mixing 19

C.5 Training Algorithm 19

D Supplementary Theory Details **19**

D.1 Supporting Directional-to-Magnitude Results 20

D.2 Blockwise NTK View for Per-Loss Adapters 21

D.3 Capacity Heuristics and Per-Layer Mixing 21

D.4 PDE Structure and Gradient Conflict 21

E Proofs for the Regime-Transition Results **22**

F Justification of the PDE-Structure Remarks **26**

G Experimental Setup **28**

G.1 PDE Benchmarks 28

G.2 K-Scaling Experiments 28

G.3 Methods Compared 29

G.4 Evaluation and Reproducibility Protocol 29

H Architecture and Hyperparameter Details **29**

H.1 Default Hyperparameters 29

H.2 PDE-Specific Settings 30

H.3 Experiment-wise Dataset, Task, and Budget Summary 30

H.4 Baseline Implementation Details 30

H.5 Model Sizes 31

H.6 Natural $K = 4$ Thermoelastic System 31

H.7 Parameter-Grouped Gradient Surgery 32

H.8 ConFIG Baseline 32

I Supplementary Cost, Scalability, and Diagnostic Analyses **32**

I.1 Regime-Aware Method Selection 32

I.2 Profiling-to-Outcome Bridge 33

I.3 Adapter Specialization Visualization 34

I.4 Computational Cost and Scalability . . . 34
I.5 Training Dynamics Visualization . . . 35
I.6 NTK Weighting Comparison . . . 36
I.7 Adaptive Rank Selection Validation . . . 37
I.8 Scalability and Ablation Tables . . . 37

J Optimizer, Budget, and Diagnostic Validation 38
J.1 Optimizer Substitution Protocol . . . 38
J.2 Accuracy under Equal Training Time . . . 39
J.3 Seed, Learning-Rate, and Optimizer Sensitivity of Profiling . . . 40
J.4 Matched Adapter Gates and Conflict Trajectories . . . 41
J.5 Raw-Gradient and Optimizer-Transformed Alignment . . . 43

K Supplementary Benchmark Tables 44

L Statistical Comparison 44

M Extended Validation and Robustness 46
M.1 Extended Forward Validation . . . 46
M.1.1 3D Validation . . . 46
M.1.2 Additional Forward Benchmarks . . . 47
M.1.3 Reference-Consistent KdV Evaluation . . . 47
M.1.4 Criterion Expansion Validation . . . 48
M.1.5 Helmholtz Scaling Across Width, Dimension, and Training Budget . . . 48
M.1.6 Additional PDE Generalization Checks . . . 49
M.2 Diagnostic Transfer to Standard MTL Benchmarks . . . 49
M.3 Mitigating Catastrophic Weight Divergence with Bounded FAMO . . . 50
M.4 Architecture Generality . . . 50
M.5 Natural K-Scaling and Multi-Physics Diagnostics . . . 51
M.6 Extended Seed Coverage . . . 51

N Additional Baseline and Ablation Analyses 52
N.1 Missing Baseline Comparisons . . . 52
N.2 Gradient Surgery Mitigation Details . . . 53
N.3 ConFIG Sensitivity Analysis . . . 53
N.3.1 Targeted Eight-Seed Helmholtz Comparison . . . 53
N.4 Implicit Regularization Ablation . . . 55
N.5 Adapter Variant Checks . . . 56
N.6 Domain Decomposition Comparison . . . 56

O Limitations and Future Work 57

Table 5: List of notation

Symbol	Meaning
$\Omega, \partial\Omega, x, d$	Domain, boundary, point, dimension.
$\mathcal{F}, f, \mathcal{R}_{\mathrm{PDE}}, \mathcal{R}_m, \Gamma_m, u, u_\theta$	PDE operator, source, residuals, constraint sets, and fields.
$\theta, \theta_{\text{trunk}}, \theta_k$	All, trunk, and kth adapter parameters.
$\mathcal{L}, \mathcal{L}_k, \mathcal{L}_{\mathrm{PDE}}, \mathcal{L}_{\mathrm{BC}}, \mathcal{L}_{\mathrm{IC}}, K, \lambda_k$	Losses, loss count, and weights.
ν, κ, α	Physical coefficients in inverse and coupled problems.
$g_k, \cos(g_i, g_j), C_{ij}, \varepsilon_g$	Gradient, cosine, conflict score, floor.
$g_k^{(\ell)}, C_k^{(\ell)}$	Layer gradient and conflict.
$C_k, \bar{r}$	Mean conflict, average norm ratio.
$h^{(\ell)}, H, L, d_h$	Layer feature, readout, depth, width.
$h_{\text{trunk}}^{(\ell)}$	Pre-adapter trunk feature.
$A_k^{(\ell)}(h), D_k^{(\ell)}, B_k^{(\ell)}, r$	Adapter, projections, rank.
$\rho_k^{(\ell)}, \pi_k^{(\ell)}, \alpha_{\text{ad}}$	Mixing score, adapter mixing weight, adapter strength.
$\mathcal{R}_{\text{ortho}}, \lambda_{\text{self}}, \lambda_{\text{cross}}$	Orthogonality regularizer and weights.
$\Phi(x), \omega_f, F, \omega_{\max}$	Fourier features, frequencies, bands, max frequency.
$D_K, M_{\text{eff}}(K), \mathcal{U}_K, K_{\mathrm{E}}^\star$	Conflict, imbalance, regime ratio, endpoint.
$T_{\mathrm{prof}}, \widehat{f}_{\mathrm{neg}}, \widehat{D}, \widehat{M}, \widehat{R}, P$	Profile length, negative-pair fraction, summaries, persistence.
$J_\theta(x), r_i, e_j$	Jacobian, pointwise PDE residual, pointwise BC error.
$\Theta_{ij}, \Theta_{ij}^{\text{shared}}, \Theta_{ij}^{(k)}$	Block NTK, shared and adapter terms.

A List of Notation

We list the symbols used throughout the paper.

B Related Work

Training difficulties and loss reweighting in PINNs. Raissi et al. [34] introduced PINNs, and Karniadakis et al. [15] reviewed the broader physics-informed machine-learning landscape. Subsequent work identified gradient pathologies [41], NTK-driven spectral bias [43, 33], soft-constraint trivial solutions [20], poor propagation [3], and broader failure modes [19]. Recent benchmarks and guides systematize these observations [5, 44]. Loss reweighting methods include learning rate annealing [41], NTK-based weighting [43], self-adaptive weights [29, 47], GradNorm [2], SoftAdapt [6], FAMO [24], uncertainty weighting [16], and causal training [42]. These methods adjust when or how much each loss contributes, but they do not directly resolve directional gradient conflict. Causal training, for example, enforces temporal causality by progressively activating loss terms from earlier to later time points.

Other PINN-side interventions modify the loss structure or update geometry itself. Gradient-enhanced PINNs add derivative-based residual terms [48], exact boundary imposition removes soft BC penalties via distance functions [39], and recent cone-based or gradient-alignment analyses study non-conflicting PINN updates directly [10, 45]. Bischof and Kraus [1] showed that multi-objective loss balancing can match gradient surgery. These methods are complementary to our per-loss adapters. They alter the loss schedule, loss scale, or update geometry, while our adapters adjust how each loss influences parameters through explicit loss-indexed subspaces.

Gradient surgery for MTL. Representative MTL conflict-mitigation and multi-objective methods include the multiple-gradient descent algorithm [35], Pareto MTL [22], PCGrad [49], CAGrad [23], NashMTL [32], Gradient Vaccine [46], RotoGrad [14], ConFIG [25], and AlignedMTL [36]. These methods assume homogeneous shared parameters; however, this assumption makes them prone to failure on PINN problems with heterogeneous parameter types. In our setting, this heterogeneity combines network weights with learnable physical constants and produces a $10^4\times$ norm mismatch, as we systematically demonstrate in Section 4. Indeed, ConFIG's unit-vector normalization discards gradient magnitude information critical for PDE loss landscapes where gradient magnitude ratios can exceed 10^3 in our evaluation suites. AlignedMTL's singular value decomposition (SVD)-based alignment also incurs $O(p^2)$ cost per step.

Architecture-level approaches and adapters. Domain decomposition methods such as conservative PINNs [13], XPINNs [12], finite basis PINNs [31], and augmented PINNs [9] partition the input space into subdomains, each handled by a local PINN with interface constraints. Variational formulations such as variational PINNs [17] and their hp-variants [18] instead modify the weak-form residual and can be combined with domain decomposition. These methods address spatial complexity, weak-form structure, or discontinuities, but they do not resolve gradient conflict between different loss terms within each subdomain or formulation. Our per-loss adapters are orthogonal to this line of work. They augment the shared trunk with loss-indexed low-rank parameter blocks and can be integrated with domain decomposition by placing such adapters inside each XPINN subdomain when a problem exhibits both spatial and loss-term complexity.

Beyond the PINN setting, broader MTL studies architecture-level soft sharing, where representations, routing weights, or parameter subsets are partially shared across tasks rather than tied through a single monolithic trunk. Representative examples include Cross-Stitch Networks [30], task-attention mechanisms [26], and Multi-gate mixture-of-experts (MoE) [28], in addition to generic sparse MoE [37] and task grouping [4, 38]. Task-specific adapters were popularized in transfer learning [7], and Low-Rank Adaptation (LoRA) introduced rank-controlled adapters for parameter-efficient large language model fine-tuning [8]. We use this low-rank parameterization not for task adaptation, but to instantiate loss-indexed perturbation subspaces inside a single shared-output PINN. Our loss-indexed adapter mixing is derived from gradient conflict rather than input features, which makes the method complementary to both domain decomposition and MoE approaches.

NTK analysis and spectral diagnostics in deep learning. The NTK [11] provides a spectral characterization of neural network training dynamics. Wang et al. [43] applied NTK analysis to PINNs, revealing how eigenvalue disparity between PDE and BC kernels causes spectral bias and contributes to training difficulty. Our work complements this line by using a blockwise NTK perspective to interpret how per-loss architectural modifications can reduce destructive interference between losses.

Neural operators. Neural operators such as the Deep Operator Network [27] and the Fourier Neural Operator [21] learn function-to-function solution maps, typically from paired solution data, although physics-informed and hybrid variants can also include residual losses. This setting is complementary to ours. We study data-scarce PINN training in which a single physical field is optimized using PDE, BC, IC, and optional data losses, making inter-loss gradient geometry the central bottleneck.

Positioning our contribution. Prior work provides strong individual tools for PINN loss balancing, gradient surgery, domain decomposition, adapters, and NTK analysis, but it does not, to our knowledge, characterize the conflict regimes that determine when scalar reweighting, architectural separation, or no additional intervention is appropriate. Our novelty is therefore not the adapter primitive itself, but the diagnostic-first regime characterization linking conflict structure to method choice. We support this characterization with regime-transition theory, operator-structure arguments, a blockwise NTK interpretation of per-loss parameter subspaces, and validation across 60+ PDE configurations, MTL benchmarks, NTK-weighting baselines [43], and architecture-transfer checks spanning multilayer perceptron (MLP), Modified MLP, and Transformer backbones.

C Supplementary Method Details

C.1 Shared Feature Trunk

All methods share a common feature trunk that maps inputs $x \in \mathbb{R}^d$ to hidden features, consisting of Fourier feature encoding [40] followed by residual blocks:

$$\Phi(x) = [\sin(\omega_1^\top x), \cos(\omega_1^\top x), \ldots, \sin(\omega_F^\top x), \cos(\omega_F^\top x)], \tag{8}$$

where $\omega_f \in \mathbb{R}^d$ are fixed frequency vectors with linearly spaced magnitudes $\|\omega_f\|_2 = f\omega_{\max}/F$ for $f = 1, \ldots, F$, with $F = 10$ bands and $\omega_{\max} = 10\pi$. The trunk consists of $L = 4$ residual blocks with hidden dimension $d_h = 128$ and Tanh activations:

$$h^{(\ell)} = h^{(\ell-1)} + \tanh(W_2^{(\ell)} \tanh(W_1^{(\ell)} h^{(\ell-1)} + b_1^{(\ell)}) + b_2^{(\ell)}). \tag{9}$$

A final linear layer maps to the scalar output $u_\theta(x) \in \mathbb{R}$, or to $\mathbb{R}^{d_{\text{out}}}$ for systems. This trunk has 140,033 parameters and serves as the Vanilla baseline. In adapter variants, the same residual trunk is retained as the shared backbone; the adapters are additional loss-indexed low-rank modules inserted at each block.

C.2 Conflict Metrics and Profiled Summaries

The aggregate conflict score for loss k is $C_k = (1/(K-1)) \sum_{j \neq k} C_{kj}$, measuring how much loss k conflicts with the others on average. For conflict-aware mixing and per-layer analyses, we analogously define

$$C_k^{(\ell)} := \frac{1}{K-1} \sum_{j \neq k} C_{kj}^{(\ell)},$$

where $C_{ij}^{(\ell)}$ is obtained by applying Eq. 3 to the layer-ℓ parameter gradients $g_k^{(\ell)} := \nabla_{\theta^{(\ell)}} \mathcal{L}_k(\theta)$. For per-problem summaries, we also report the average pairwise magnitude ratio

$$\bar{r} := \frac{1}{\binom{K}{2}} \sum_{i<j} \frac{\max(\|g_i\|, \|g_j\|) + \varepsilon_g}{\min(\|g_i\|, \|g_j\|) + \varepsilon_g},$$

which is finite and at least 1 by construction.

Profiled summaries. For a profiling window $t = 1, \ldots, T_{\text{prof}}$, let $g_k^{(t)} := \nabla_\theta \mathcal{L}_k(\theta_t)$ and define the pairwise cosine with a zero-gradient convention

$$c_{ij}^{(t)} := \begin{cases} \cos(g_i^{(t)}, g_j^{(t)}), & \|g_i^{(t)}\| \|g_j^{(t)}\| > 0, \\ 0, & \text{otherwise}, \end{cases}$$

and define

$$N_t^- := \{(i,j) : 1 \leq i < j \leq K, c_{ij}^{(t)} < 0\}, \qquad f_{\text{neg}}^{(t)} := \frac{|N_t^-|}{\binom{K}{2}},$$

$$\bar{c}_-^{(t)} := \begin{cases} \frac{1}{|N_t^-|} \sum_{(i,j) \in N_t^-} c_{ij}^{(t)}, & |N_t^-| > 0, \\ 0, & |N_t^-| = 0, \end{cases} \qquad D^{(t)} := f_{\text{neg}}^{(t)} |\bar{c}_-^{(t)}|,$$

$$M^{(t)} := \frac{\text{std}_k(\|g_k^{(t)}\|)}{K^{-1} \sum_k \|g_k^{(t)}\| + \varepsilon_g}.$$

We report the profiled summaries

$$\widehat{f}_{\text{neg}} := \frac{1}{T_{\text{prof}}} \sum_{t=1}^{T_{\text{prof}}} f_{\text{neg}}^{(t)}, \quad \widehat{D} := \frac{1}{T_{\text{prof}}} \sum_{t=1}^{T_{\text{prof}}} D^{(t)}, \quad \widehat{M} := \frac{1}{T_{\text{prof}}} \sum_{t=1}^{T_{\text{prof}}} M^{(t)}.$$

We also report the empirical proxy

$$\widehat{R} := \frac{\widehat{D}}{\widehat{M} + \varepsilon_R},$$

where $\varepsilon_R = 10^{-8}$. Here $\widehat{M}$ is a coefficient-of-variation surrogate for magnitude imbalance, whereas Theorem 1 uses the max-to-mean ratio $M_{\text{eff}}(K)$. The profiled ratio $\widehat{R}$ should therefore be read as an empirical surrogate for the theorem-level quantity $\mathcal{U}_K$, not as a plug-in estimator of it. These profiled summaries are the quantities reported in the experiments.

C.3 Additional Adapter Details

Adapter orthogonality. To encourage the independence of the adapter row subspaces in $\mathbb{R}^{d_h}$, we regularize both self-orthogonality within each down-projection and cross-orthogonality between different down-projections:

$$\mathcal{R}_{\text{ortho}} = \lambda_{\text{self}} \sum_{k,\ell} \|D_k^{(\ell)} D_k^{(\ell)\top} - I_r\|_F^2 + \lambda_{\text{cross}} \sum_{\substack{i<j \\ \ell}} \|D_i^{(\ell)} D_j^{(\ell)\top}\|_F^2. \tag{10}$$

Algorithm 1 Per-loss adapter training with UAM as the default and CAM optional

Require: PDE problem with K losses, single-output network u_θ with K adapters

for epoch $t = 1, \ldots, T$ **do**
  Sample collocation, BC, and IC points
  Run the forward pass through the shared trunk and aggregated adapter correction $\rightarrow u_\theta(x)$ as in Eq. 6
  Compute losses $\{\mathcal{L}_k\}_{k=1}^K$ from the same u_θ
  For CAM and LCAM only, after warm-up and every τ epochs, update EMA conflict scores, refresh ρ, and recompute π
  Total loss: $\mathcal{L} = \sum_k \lambda_k \mathcal{L}_k + \mathcal{R}_{\text{ortho}}$
  Update θ via the chosen optimizer (AdamW by default), with gradient clipping
end for

The first term encourages the rows of each $D_k^{(\ell)}$ to be orthonormal, while the second penalizes overlap between the row spaces of different adapters. When the row spaces are exactly orthogonal and each $D_k^{(\ell)}$ has full row rank r, the cross term vanishes.

C.4 Conflict-Aware Adapter Mixing

The latent mixing scores $\rho_k^{(\ell)}$ can be fixed or dynamically adjusted based on gradient conflict. UAM uses the uniform rule $\pi_k^{(\ell)} = 1/K$. For global CAM, we use the aggregate conflict summary C_k and set a layer-independent score $\rho_k = \sigma(a_0 - b \cdot C_k)$, with $\rho_k^{(\ell)} \equiv \rho_k$ in the normalization above. For LCAM, we instead use layerwise conflict summaries and set $\rho_k^{(\ell)} = \sigma(a_0 + \delta_\ell - b \cdot C_k^{(\ell)})$ before converting these scores into the normalized adapter weights $\pi_k^{(\ell)}$. In both cases, larger conflict decreases ρ and therefore increases the corresponding adapter weight. Here $a_0 = 1.0$, $b = 2.0$, and the LCAM offset δ_ℓ interpolates from $+0.3$ at shallow layers, which biases toward sharing, to -0.3 at deep layers, which biases toward specialization. The conflict scores are exponential-moving-average (EMA) smoothed with $\beta = 0.95$. In practice, fixed UAM captures nearly all of the benefit. On rerun $K = 3$ stress tests in Helmholtz and Klein–Gordon, reported in Table 41, dynamic LCAM stays within single-digit percentage differences and fixed uniform mixing is slightly better in both cases. We therefore recommend UAM with uniform π as the default, while CAM remains a higher-complexity variant for settings in which fixed mixing is insufficient.

C.5 Training Algorithm

Training follows Algorithm 1. A warm-up of $T_{\text{warm}} = 200$ epochs with equal weights and uniform adapter mixing $\pi_k = 1/K$ establishes a stable baseline. The optional mixing step is used only by dynamic CAM and LCAM variants and is skipped for fixed UAM. After warm-up, every τ epochs, we compute the current aggregate conflict scores C_k for CAM or layerwise scores $C_k^{(\ell)}$ for LCAM, update their EMA, set the corresponding latent scores ρ using Section C.4 with clipping to $[\rho_{\min}, \rho_{\max}]$, and recompute π through the normalization in Section 2.3. The latent scores are bookkeeping variables for adapter mixing rather than AdamW-optimized model parameters.

D Supplementary Theory Details

We collect the supplementary theory details referenced in the main text. First, we give the supporting directional-to-magnitude results for the regime-transition analysis. We then give a blockwise NTK interpretation of per-loss adapters as structured perturbations of the shared representation. Finally, we justify the operator-structure arguments summarized in Remark 3.

D.1 Supporting Directional-to-Magnitude Results

Proposition 1 (Conditional concentration of D_K)**.** *Let $g_1, \dots, g_K$ be nonzero gradients in $\mathbb{R}^p$, write $\hat{g}_k = g_k/\|g_k\|$, and define*

$$X_{ij} := \max(0, -\hat{g}_i^\top \hat{g}_j), \qquad D_K := \frac{1}{\binom{K}{2}} \sum_{i<j} X_{ij}.$$

Suppose there exists a σ-algebra Λ such that, conditional on Λ, the family $\{X_{ij}\}_{i<j}$ is exchangeable with

$$\mathbb{E}[X_{12} \mid \Lambda] = \mu(\Lambda), \qquad \sup_{(i,j)\neq(k,\ell)} |\mathrm{Cov}(X_{ij}, X_{k\ell} \mid \Lambda)| \le c(\Lambda),$$

where the supremum is over distinct unordered pairs and $c(\Lambda) \ge 0$ is Λ-measurable. Then with $n_K := \binom{K}{2}$, we have

$$\mathbb{E}[D_K \mid \Lambda] = \mu(\Lambda),$$
$$\mathrm{Var}(D_K \mid \Lambda) \le \frac{1}{4n_K} + c(\Lambda),$$

and therefore

$$\mathrm{Var}(D_K) = \mathrm{Var}(\mu(\Lambda)) + \mathbb{E}[\mathrm{Var}(D_K \mid \Lambda)] \le \mathrm{Var}(\mu(\Lambda)) + \frac{1}{4n_K} + \mathbb{E}[c(\Lambda)]. \tag{11}$$

If, in addition, the directions $\hat{g}_k$ are independent and uniformly distributed on $V \cap \mathbb{S}^{p-1}$ for some fixed subspace $V \subseteq \mathbb{R}^p$ with $\dim(V) = d_{\mathrm{eff}}$, then in orthonormal coordinates on V they are independent and uniformly distributed on $\mathbb{S}^{d_{\mathrm{eff}}-1}$. For

$$\mu_d := \frac{\Gamma(d_{\mathrm{eff}}/2)}{2\sqrt{\pi}\Gamma((d_{\mathrm{eff}}+1)/2)} = \Theta(d_{\mathrm{eff}}^{-1/2}),$$

we have

$$\mathbb{E}[D_K] = \mu_d,$$
$$\mathrm{Var}(D_K) = \frac{\mathrm{Var}(X_{12})}{n_K} = \frac{1}{n_K}\left(\frac{1}{2d_{\mathrm{eff}}} - \mu_d^2\right) = O((K^2 d_{\mathrm{eff}})^{-1}).$$

Remark 1. *Eq. 11 is the quantity that is stable under latent regime variation. Exchangeability alone does not force D_K to concentrate without the additional term $\mathrm{Var}(\mu(\Lambda))$ vanishing. In particular, a short profiling window should be interpreted as estimating a marginal average of the directional-conflict level over that window, not a pointwise constant that is guaranteed to concentrate across all optimization states.*

Remark 2. *The next theorem is not a consequence of Proposition 1 under arbitrary latent regime variation. It is a separate stylized model in which the directional regime is fixed, so Λ is trivial, and gradient directions decouple from gradient magnitudes. This fixed-directional-regime model is the setting in which explicit rate calculations are available.*

Corollary 1 (Expected beneficial range)**.** *For $\tau > 0$, whenever the set is nonempty, define*

$$K_{\mathrm{E}}^\star(\tau) := \sup\{K \ge 2 : \mathbb{E}[\mathcal{U}_K] > \tau\}.$$

Under the hypotheses of Theorem 1, this set is nonempty for all sufficiently small τ. As $\tau \to 0^+$,

$$K_{\mathrm{E}}^\star(\tau) = \begin{cases} \exp(\Theta(d_{\mathrm{eff}}^{-1/2}/\tau)), & \textit{under the exponential-tail model,} \\ \Theta((\tau d_{\mathrm{eff}}^{1/2})^{-\alpha}), & \textit{under the Pareto-tail model.} \end{cases}$$

The constants hidden in $\Theta(\cdot)$ depend on the tail parameters and on μ_m, but not on τ or d_{eff}.

D.2 Blockwise NTK View for Per-Loss Adapters

This subsection develops the NTK perspective on per-loss adapters, building on the foundational NTK framework [11] and its PINN-specific analysis by Wang et al. [43].

For fixed mixing, write the parameter vector as $\theta = [\theta_{\text{trunk}}, \theta_1, \ldots, \theta_K]$, where θ_k denotes the parameters of the kth adapter block. For each loss k, let $s_k(x)$ denote a scalar quantity entering $\mathcal{L}_k$ and derived from the same shared output u_θ, such as the raw output, a boundary mismatch, or a PDE residual component. We define the loss-indexed block NTK entry, measuring parameter-space coupling between scalar channels i and j at inputs x and x', by

$$\Theta_{ij}(x, x') := \nabla_\theta s_i(x)^\top \nabla_\theta s_j(x') \tag{12}$$

for $1 \le i, j \le K$. When the latent mixing scores $\rho_k^{(\ell)}$ are treated as fixed with respect to θ during differentiation, and equivalently when the normalized weights $\pi_k^{(\ell)}$ are fixed, the block kernel splits additively by parameter block:

$$\Theta_{ij}(x, x') = \Theta_{ij}^{\text{shared}}(x, x') + \sum_{k=1}^{K} \Theta_{ij}^{(k)}(x, x'), \tag{13}$$

where

$$\Theta_{ij}^{\text{shared}} = \nabla_{\theta_{\text{trunk}}} s_i(x)^\top \nabla_{\theta_{\text{trunk}}} s_j(x'),$$
$$\Theta_{ij}^{(k)} = \nabla_{\theta_k} s_i(x)^\top \nabla_{\theta_k} s_j(x').$$

In the exact single-output architecture, the adapter terms $\Theta_{ij}^{(k)}$ need not vanish for $i \neq j$ because every s_i depends on the same shared output u_θ, which itself depends on all adapter blocks through Eq. 6. In the idealized exact-separation surrogate where s_i depends only on θ_{trunk} and θ_i, the adapter contribution instead reduces to a diagonal δ_{ij} term. Here, the term "exact" refers to the implemented shared-output adapter architecture with fixed mixing, for which the parameter-block decomposition above is exact but no diagonalization is assumed. The exact-separation surrogate is not the implemented model; it is an idealized limiting case in which cross-adapter sensitivities are neglected. Thus, the diagonal δ_{ij} form should be read as a mechanism-level approximation, while the actual architecture may retain off-diagonal adapter couplings that are encouraged, but not forced, to be small by subspace separation and orthogonality regularization. We use this blockwise picture as bookkeeping for the kernel-level interpretation of per-loss adapters. It identifies the adapter contribution as an additive perturbation of the shared kernel and clarifies why loss-indexed parameter blocks can relieve directional interference without introducing separate physical outputs.

D.3 Capacity Heuristics and Per-Layer Mixing

For architectural capacity, the saturation ratio $\Delta_{\text{sat}} := Kr/d_h$ is a useful heuristic proxy. At fixed width, increasing either the number of losses or the adapter rank leaves less room for loss-specific low-rank specialization, and exact cross-adapter orthogonality becomes harder to satisfy. Per-layer mixing can in principle exploit layer-wise variation in conflict, but in our experiments, its gains are modest at $K = 3$ and become more noticeable only as K grows. We therefore treat capacity, conflict-aware mixing, and orthogonality effects primarily through the rank, mixing, and regularization ablations rather than through separate standalone theorems.

D.4 PDE Structure and Gradient Conflict

Beyond the K-dependent regime structure, the type of PDE determines the conflict profile. The empirical results across the evaluation suites suggest a structural pattern:

Remark 3 (Operator-type heuristic for gradient coupling)**.** *For the PDE and Dirichlet-boundary setting made explicit in Appendix F, the operator class constrains the state-level influence pattern and therefore provides only a qualitative prior for parameter-gradient coupling. Uniform ellipticity is compatible with domain-wide state-level influence via Green and Poisson representations, so widespread PDE and BC gradient coupling is possible; however, its sign and magnitude still depend on the residual and error coefficients and on the network-Jacobian correlations. Scalar linear*

parabolic operators satisfying a comparison principle or a maximum principle are order-preserving in the state variable, so the operator alone does not determine the sign of PDE and BC parameter-gradient alignment. Negative alignment can already arise from residual and error signs together with network Jacobian correlations, and can be further amplified by nonlinearities, discretization, or non-monotone parameterizations. Hyperbolic operators localize state-level influence to a domain of dependence through finite propagation speed, but finite propagation alone does not determine the sign of the parameter-gradient cosine. Stiff or highly oscillatory terms can nevertheless change the effective length scales seen by the network. Hence operator type provides a heuristic prior on conflict structure, but not a universal sign or numerical value of f_{neg}.

This classification should therefore be read as a heuristic prior rather than a standalone decision rule. Elliptic problems are more compatible with widespread coupling, while parabolic problems require an additional nonlinearity or parameterization check. However, the following remark shows that operator type alone is still insufficient in high dimensions.

Remark 4 (Residual decomposition alone does not imply a high-dimensional elliptic monotonicity law)**.** *For the* d*-dimensional Poisson equation* $-\Delta u = f$ *on* $[0,1]^d$ *with manufactured solution* $u(x) = \prod_{i=1}^{d} \sin(\pi x_i)$*, the residual decomposition*

$$r_\theta(x) = -\sum_{i=1}^{d} \partial_{x_i}^2 u_\theta(x) - f(x), \tag{14}$$

with forcing amplitude $f(x) = d\pi^2 u(x)$*, together with the identity*

$$\nabla_\theta r_\theta(x) = -\sum_{i=1}^{d} \nabla_\theta \partial_{x_i}^2 u_\theta(x), \tag{15}$$

does not by itself imply any universal monotone-in-d *law for the instantaneous cosine between* $\nabla_\theta \mathcal{L}_{\mathrm{PDE}}$ *and* $\nabla_\theta \mathcal{L}_{\mathrm{BC}}$ *whenever those aggregated gradients are nonzero. In particular, if the per-coordinate sensitivities* $\nabla_\theta \partial_{x_i}^2 u_\theta(x)$ *are weakly correlated, then the normalized sum* $d^{-1}\sum_{i=1}^{d} \nabla_\theta \partial_{x_i}^2 u_\theta(x)$ *can become more concentrated with* d*, while the residual formula simultaneously contains an explicit forcing contribution of size* $O(d)$*. Because the network Laplacian term can either reinforce or cancel that forcing contribution, the residual amplitude need not scale monotonically with* d*. Therefore, the displayed decomposition alone does not justify a universal monotone-in-*d *law for the instantaneous PDE and BC cosine. Any extension of this warning to profiled statistics such as* $f_{\mathrm{neg}}(d)$ *requires additional assumptions on the boundary-gradient scaling and on the training trajectory used in the profiling procedure.*

Appendix F justifies both remarks. Accordingly, operator-type heuristics are insufficient. The conflict regime depends on the interplay between the differential operator, forcing term, and spatial dimension, motivating the lightweight profiling criterion in Section 4.4 as a general-purpose tool that subsumes operator-type heuristics.

E Proofs for the Regime-Transition Results

Proof of Proposition 1. By conditional exchangeability,

$$\mathbb{E}[D_K \mid \Lambda] = \frac{1}{n_K} \sum_{i<j} \mathbb{E}[X_{ij} \mid \Lambda] = \mu(\Lambda).$$

For the variance,

$$\mathrm{Var}(D_K \mid \Lambda) = \frac{1}{n_K^2} \sum_{i<j} \sum_{k<\ell} \mathrm{Cov}(X_{ij}, X_{k\ell} \mid \Lambda).$$

The diagonal contribution satisfies

$$\frac{1}{n_K^2} \sum_{i<j} \mathrm{Var}(X_{ij} \mid \Lambda) \le \frac{1}{n_K^2} \cdot n_K \cdot \frac{1}{4} = \frac{1}{4n_K},$$

because $0 \le X_{ij} \le 1$. The off-diagonal contribution satisfies

$$\frac{1}{n_K^2} \sum_{(i,j)\neq(k,\ell)} |\mathrm{Cov}(X_{ij}, X_{k\ell} \mid \Lambda)| \le \frac{n_K(n_K-1)}{n_K^2} c(\Lambda) \le c(\Lambda).$$

The conditional covariance bound proves the conditional variance bound. Eq. 11 is then the law of total variance.

For the spherical claim, let V be a fixed subspace of $\mathbb{R}^p$ with $\dim(V) = d := d_{\text{eff}}$, choose orthonormal coordinates on V, and work in that coordinate system. Then the directions become independent and uniformly distributed on $\mathbb{S}^{d-1}$. If $u, v \in \mathbb{S}^{d-1}$ are independent and uniform, and $c := u^\top v$, then the density of c on $[-1,1]$ is symmetric and proportional to $(1-z^2)^{(d-3)/2}$, so

$$\mathbb{E}[X_{12}] = \mathbb{E}[(-c)_+] = \frac{1}{2}\mathbb{E}|c| = \frac{\Gamma(d/2)}{2\sqrt{\pi}\Gamma((d+1)/2)} = \mu_d.$$

Also,

$$\mathbb{E}[X_{12}^2] = \mathbb{E}[c^2 \mathbf{1}_{\{c<0\}}] = \frac{1}{2}\mathbb{E}[c^2] = \frac{1}{2d},$$

because $\mathbb{E}[c^2] = 1/d$.

It remains to show that distinct X_{ij} are pairwise independent. If the pairs are disjoint, pairwise independence is immediate from independence of the underlying directions. If they share one index, say $(i,j) = (1,2)$ and $(k,\ell) = (1,3)$, then conditional on u_1 the random variables X_{12} and X_{13} are independent because u_2 and u_3 are independent. Moreover, by rotational invariance the conditional law of $u_1^\top u_2$ given u_1 does not depend on u_1, so for any bounded measurable φ and ψ,

$$\mathbb{E}[\varphi(X_{12})\psi(X_{13}) \mid u_1] = \mathbb{E}[\varphi(X_{12}) \mid u_1]\mathbb{E}[\psi(X_{13}) \mid u_1] = \mathbb{E}[\varphi(X_{12})]\mathbb{E}[\psi(X_{13})].$$

Taking expectations gives pairwise independence. Therefore

$$\mathrm{Var}(D_K) = \frac{1}{n_K^2}\sum_{i<j}\mathrm{Var}(X_{ij}) = \frac{\mathrm{Var}(X_{12})}{n_K} = \frac{1}{n_K}\left(\frac{1}{2d_{\text{eff}}} - \mu_d^2\right),$$

which is $O((K^2 d_{\text{eff}})^{-1})$. □

Proof of Theorem 1. Since D_K depends only on the directions $u_1, \dots, u_K$ and $\bar{m}/A_K$ depends only on the magnitudes $m_1, \dots, m_K$, the two terms are independent. Proposition 1 gives $\mathbb{E}[D_K] = \mu_d$, hence

$$\mathbb{E}[\mathcal{U}_K] = \mathbb{E}\left[D_K \frac{\bar{m}}{A_K}\right] = \mathbb{E}[D_K]\mathbb{E}\left[\frac{\bar{m}}{A_K}\right] = \mu_d \mathbb{E}\left[\frac{\bar{m}}{A_K}\right].$$

The factorization argument proves Eq. 7.

Exponential-tail magnitudes. Fix constants $c_-, c_+ > 0$ such that

$$0 < c_- < \frac{\nu}{b}, \qquad c_+ > \frac{\nu}{a}.$$

Set

$$q_K^- := c_- \log K, \qquad q_K^+ := c_+ \log K.$$

For sufficiently large K, both q_K^- and q_K^+ exceed t_0, so the tail bounds apply. By the lower tail bound,

$$\begin{aligned}\Pr(A_K < q_K^-) &= \Pr(m_1 \le q_K^-)^K \\ &\le \left(1 - \exp\left(-\frac{b q_K^-}{\nu}\right)\right)^K \\ &\le \exp\left(-K^{1-bc_-/\nu}\right) = o((\log K)^{-1}).\end{aligned}$$

Therefore

$$\begin{aligned}\mathbb{E}\left[\frac{\bar{m}}{A_K}\right] &\le \mathbb{E}\left[\frac{\bar{m}}{q_K^-}\mathbf{1}_{\{A_K \ge q_K^-\}}\right] + \Pr(A_K < q_K^-) \\ &\le \frac{\mathbb{E}[\bar{m}]}{q_K^-} + o((\log K)^{-1}) \\ &= O((\log K)^{-1}).\end{aligned}$$

For the lower bound, the upper tail bound gives

$$\begin{aligned}\Pr(A_K > q_K^+) &\le K \Pr(m_1 > q_K^+) \\ &\le K^{1-ac_+/\nu} = o(1).\end{aligned}$$

Also $\bar{m} \to \mu_m$ in probability by the law of large numbers, so

$$\Pr\left(\bar{m} \ge \frac{\mu_m}{2}\right) \to 1.$$

Hence, by the union bound,

$$\Pr\left(\bar{m} \ge \frac{\mu_m}{2}, A_K \le q_K^+\right) \ge 1 - \Pr\left(\bar{m} < \frac{\mu_m}{2}\right) - \Pr(A_K > q_K^+) = 1 - o(1).$$

Hence

$$\begin{aligned}\mathbb{E}\left[\frac{\bar{m}}{A_K}\right] &\ge \frac{\mu_m}{2q_K^+} \Pr\left(\bar{m} \ge \frac{\mu_m}{2}, A_K \le q_K^+\right) \\ &= \Omega((\log K)^{-1}).\end{aligned}$$

Combining the two bounds with $\mu_d = \Theta(d_{\text{eff}}^{-1/2})$ proves

$$\mathbb{E}[\mathcal{U}_K] = \Theta(d_{\text{eff}}^{-1/2}/\log K).$$

Pareto-tail magnitudes. Let $A_K = \max_{1 \le k \le K} m_k$ and fix any $q \in (1, \alpha)$. Write $r := q/(q-1)$. By Hölder's inequality,

$$\mathbb{E}\left[\frac{\bar{m}}{A_K}\right] \le \mathbb{E}[\bar{m}^q]^{1/q} \mathbb{E}[A_K^{-r}]^{1/r}.$$

Since $q < \alpha$, we have $\mathbb{E}[m_1^q] < \infty$. By the convexity of $t \mapsto t^q$ for $q \ge 1$,

$$\mathbb{E}[\bar{m}^q] \le \frac{1}{K}\sum_{k=1}^{K} \mathbb{E}[m_k^q] = \mathbb{E}[m_1^q] < \infty.$$

It remains to evaluate $\mathbb{E}[A_K^{-r}]$. For Pareto tails,

$$\Pr(A_K \le t) = \left(1 - \left(\frac{t_0}{t}\right)^{\alpha}\right)^K, \qquad t \ge t_0,$$

so A_K has density

$$f_{A_K}(t) = K\alpha t_0^{\alpha} t^{-\alpha-1}\left(1 - \left(\frac{t_0}{t}\right)^{\alpha}\right)^{K-1}, \qquad t \ge t_0.$$

Therefore

$$\mathbb{E}[A_K^{-r}] = K\alpha t_0^{\alpha} \int_{t_0}^{\infty} t^{-r-\alpha-1}\left(1 - \left(\frac{t_0}{t}\right)^{\alpha}\right)^{K-1} dt.$$

With the substitution $y = (t_0/t)^{\alpha}$, the integral becomes

$$\begin{aligned}\mathbb{E}[A_K^{-r}] &= K t_0^{-r} \int_0^1 y^{r/\alpha}(1-y)^{K-1} dy \\ &= K t_0^{-r} B\left(1 + \frac{r}{\alpha}, K\right) \\ &= t_0^{-r}\Gamma\left(1 + \frac{r}{\alpha}\right)\frac{\Gamma(K+1)}{\Gamma(K+1+r/\alpha)} = \Theta(K^{-r/\alpha}).\end{aligned}$$

Hence

$$\mathbb{E}\left[\frac{\bar{m}}{A_K}\right] = O(K^{-1/\alpha}).$$

For the lower bound, fix any constant $C > t_0$. Then

$$\Pr(A_K \le CK^{1/\alpha}) = \left(1 - \frac{t_0^\alpha}{C^\alpha K}\right)^K \longrightarrow \exp\left(-\frac{t_0^\alpha}{C^\alpha}\right) > 0.$$

Also $\Pr(\bar{m} \ge \mu_m/2) \to 1$ by the law of large numbers. Hence

$$\begin{aligned}\Pr\left(\bar{m} \ge \frac{\mu_m}{2}, A_K \le CK^{1/\alpha}\right) &\ge \Pr(A_K \le CK^{1/\alpha}) - \Pr\left(\bar{m} < \frac{\mu_m}{2}\right) \\ &\longrightarrow \exp\left(-\frac{t_0^\alpha}{C^\alpha}\right) > 0.\end{aligned}$$

Thus

$$\begin{aligned}\mathbb{E}\left[\frac{\bar{m}}{A_K}\right] &\ge \frac{\mu_m}{2CK^{1/\alpha}} \Pr\left(\bar{m} \ge \frac{\mu_m}{2}, A_K \le CK^{1/\alpha}\right) \\ &= \Omega(K^{-1/\alpha}).\end{aligned}$$

Combining upper and lower bounds with $\mu_d = \Theta(d_{\text{eff}}^{-1/2})$ yields

$$\mathbb{E}[\mathcal{U}_K] = \Theta(d_{\text{eff}}^{-1/2} K^{-1/\alpha}).$$

□

Proof of Corollary 1. Under the exponential-tail model,

$$\mathbb{E}[\mathcal{U}_K] = \Theta(d_{\text{eff}}^{-1/2}/\log K)$$

as $K \to \infty$. Hence, there exist constants $c_1, c_2 > 0$ and $K_0 \ge 2$ such that for all $K \ge K_0$,

$$c_1 \frac{d_{\text{eff}}^{-1/2}}{\log K} \le \mathbb{E}[\mathcal{U}_K] \le c_2 \frac{d_{\text{eff}}^{-1/2}}{\log K}.$$

For sufficiently small τ, define

$$\begin{aligned}K_-(\tau) &:= \left\lfloor \exp\left(\frac{c_1 d_{\text{eff}}^{-1/2}}{2\tau}\right)\right\rfloor, \\ K_+(\tau) &:= \left\lceil \exp\left(\frac{2c_2 d_{\text{eff}}^{-1/2}}{\tau}\right)\right\rceil.\end{aligned}$$

When τ is small enough, $K_-(\tau) \ge K_0$, and the lower bound gives $\mathbb{E}[\mathcal{U}_{K_-(\tau)}] \ge 2\tau > \tau$, so the defining set for $K_{\text{E}}^\star(\tau)$ is nonempty. Likewise, for every $K \ge K_+(\tau)$, the upper bound gives $\mathbb{E}[\mathcal{U}_K] \le \tau/2$, so no such K belongs to the set. Therefore

$$K_-(\tau) \le K_{\text{E}}^\star(\tau) < K_+(\tau),$$

which implies

$$K_{\text{E}}^\star(\tau) = \exp(\Theta(d_{\text{eff}}^{-1/2}/\tau))$$

as $\tau \to 0^+$.

Under the Pareto-tail model,

$$\mathbb{E}[\mathcal{U}_K] = \Theta(d_{\text{eff}}^{-1/2} K^{-1/\alpha})$$

as $K \to \infty$. Therefore, there exist constants $c_1', c_2' > 0$ and $K_0' \ge 2$ such that for all $K \ge K_0'$,

$$c_1' d_{\text{eff}}^{-1/2} K^{-1/\alpha} \le \mathbb{E}[\mathcal{U}_K] \le c_2' d_{\text{eff}}^{-1/2} K^{-1/\alpha}.$$

For sufficiently small τ, define

$$\widetilde{K}_-(\tau) := \left\lfloor \left(\frac{c_1' d_{\text{eff}}^{-1/2}}{2\tau} \right)^{\alpha} \right\rfloor,$$
$$\widetilde{K}_+(\tau) := \left\lceil \left(\frac{2c_2' d_{\text{eff}}^{-1/2}}{\tau} \right)^{\alpha} \right\rceil.$$

When τ is small enough, $\widetilde{K}_-(\tau) \geq K_0'$, and the lower bound gives $\mathbb{E}[\mathcal{U}_{\widetilde{K}_-(\tau)}] \geq 2\tau$. For every $K \geq \widetilde{K}_+(\tau)$, the upper bound gives $\mathbb{E}[\mathcal{U}_K] \leq \tau/2$. Therefore

$$\widetilde{K}_-(\tau) \leq K_{\text{E}}^\star(\tau) < \widetilde{K}_+(\tau),$$

and hence

$$K_{\text{E}}^\star(\tau) = \Theta((\tau d_{\text{eff}}^{1/2})^{-\alpha})$$

as $\tau \to 0^+$. The preceding endpoint bounds prove the two asymptotic forms. □

Empirical interpretation. Theorem 1 should be read as a two-model rate statement in a fixed stylized model, and Corollary 1 is an expectation-level consequence of that model, not a deterministic threshold for every realization. The theorem is not an exhaustive taxonomy of all light-tailed distributions. The relevant empirical issue is whether the profiled per-loss gradient magnitudes behave closer to a logarithmic-growth maximum regime or to a power-growth maximum regime over the observed K range. If neither model fits the data well, the qualitative monotonicity conclusion may still be useful, but the explicit rates should not be over-interpreted.

F Justification of the PDE-Structure Remarks

We justify Remarks 3 and 4 by formalizing how the PDE operator type constrains the gradient-coupling structure between $\mathcal{L}_{\text{PDE}}$ and a Dirichlet BC loss. This appendix therefore specializes the general constraint-residual notation introduced in Section 2.1 to the Dirichlet BC case $\Gamma_m = \partial\Omega$ with $\mathcal{R}_m[u](x) = u(x) - g(x)$ so that the BC residual is explicit. Throughout, let $u_\theta : \Omega \to \mathbb{R}$ be a neural network parameterized by $\theta \in \mathbb{R}^p$, and define the Jacobian $J_\theta(x) = \partial u_\theta(x)/\partial\theta \in \mathbb{R}^{1\times p}$. For the pointwise half-squared losses

$$\ell_{\text{PDE},i} := \frac{1}{2}(\mathcal{F}u_\theta(x_i) - f(x_i))^2,$$
$$\ell_{\text{BC},j} := \frac{1}{2}(u_\theta(x_j') - g(x_j'))^2,$$

the corresponding gradients are:

$$\nabla_\theta \ell_{\text{PDE},i} = \underbrace{(\mathcal{F}u_\theta(x_i) - f(x_i))}_{r_i} \cdot \nabla_\theta[\mathcal{F}u_\theta(x_i)], \tag{16}$$

$$\nabla_\theta \ell_{\text{BC},j} = \underbrace{(u_\theta(x_j') - g(x_j'))}_{e_j} \cdot J_\theta(x_j')^\top, \tag{17}$$

where $\{x_i\}_{i=1}^{N_r} \subset \Omega$ and $\{x_j'\}_{j=1}^{N_b} \subset \partial\Omega$ are interior and boundary collocation points, and the aggregated losses are $\mathcal{L}_{\text{PDE}} = \sum_i \ell_{\text{PDE},i}$ and $\mathcal{L}_{\text{BC}} = \sum_j \ell_{\text{BC},j}$ up to any common positive averaging factors.

Lemma 1 (Elliptic PDE and Dirichlet-boundary gradient coupling)**.** *Let $\mathcal{F} = -\nabla \cdot (A(x)\nabla)$ be uniformly elliptic on a bounded domain $\Omega \subset \mathbb{R}^d$ with smooth boundary. If u_θ is twice differentiable in x and θ, then for any interior point $x \in \Omega$ and boundary point $x' \in \partial\Omega$:*

$$\nabla_\theta[\mathcal{F}u_\theta(x)] = \mathcal{F}_x[J_\theta(x)^\top],$$

where $\mathcal{F}_x$ applies the spatial differential operator to each component of J_θ. Moreover, provided $\nabla_\theta \mathcal{L}_{\text{PDE}} \neq 0$ and $\nabla_\theta \mathcal{L}_{\text{BC}} \neq 0$, the cosine similarity between the aggregated PDE and Dirichlet-boundary gradients satisfies:

$$\cos(\nabla_\theta \mathcal{L}_{\text{PDE}}, \nabla_\theta \mathcal{L}_{\text{BC}}) = \frac{\sum_i \sum_j r_i e_j \langle \mathcal{F}_x[J_\theta(x_i)^\top], J_\theta(x_j')^\top \rangle}{\|\nabla_\theta \mathcal{L}_{\text{PDE}}\| \cdot \|\nabla_\theta \mathcal{L}_{\text{BC}}\|}.$$

Proof. The first claim follows from the linearity of $\mathcal{F}$ in u and the smoothness of u_θ in (x, θ), which permits interchange of $\mathcal{F}_x$ and $\partial/\partial\theta$. The cosine similarity expression follows by expanding $\nabla_\theta \mathcal{L}_{\mathrm{PDE}} = \sum_i r_i \mathcal{F}_x[J_\theta(x_i)^\top]$ and $\nabla_\theta \mathcal{L}_{\mathrm{BC}} = \sum_j e_j J_\theta(x'_j)^\top$; any common positive averaging factors cancel from the cosine. □

Justification of Remark 3. For elliptic operators, a uniformly elliptic Dirichlet problem has a solution operator that is compatible with domain-wide state-level influence through Green-function or Poisson-kernel representations whose kernels have global spatial support. At the gradient level, Lemma 1 gives

$$\nabla_\theta[\mathcal{F}u_\theta(x)] = \mathcal{F}_x[J_\theta(x)^\top].$$

Therefore, the cosine between PDE and boundary gradients is controlled by the bilinear terms

$$\left\langle \mathcal{F}_x[J_\theta(x_i)^\top], J_\theta(x'_j)^\top \right\rangle.$$

Ellipticity places no locality restriction on these couplings, but their sign and magnitude still depend on the learned Jacobian correlations rather than on ellipticity alone. This elliptic coupling argument proves the qualitative compatible-with-widespread-coupling claim.

For parabolic operators, scalar linear equations satisfying a comparison principle or a maximum principle have order-preserving state evolution. Larger BC or IC data cannot create a smaller solution later in time. Hence, the operator itself does not determine the sign of the parameter-gradient cosine. Negative alignment can already arise through the residual and error signs together with the Jacobian-correlation terms in the cosine formula, and can be further amplified by nonlinear advection, reaction terms, discretization artifacts, or non-monotone parameterizations of u_θ. This parabolic conclusion is exactly the qualitative statement in the remark.

For hyperbolic operators, the domain-of-dependence theorem implies finite propagation speed. The state at (x, t) depends only on data inside the backward characteristic cone. Consequently, the state-level influence pattern is localized to causally connected regions. This finite-propagation structure constrains where PDE and boundary information can interact, unlike the elliptic case, but it does not by itself determine the sign of the parameter-gradient cosine. Stiff or highly oscillatory terms do not remove finite propagation, but they can shorten the relevant physical length scales and thereby change the effective coupling range seen by a finite-width network. This finite-propagation argument proves the qualitative hyperbolic claim. □

Justification of Remark 4. For the d-dimensional Poisson equation with $u^*(x) = \prod_{i=1}^d \sin(\pi x_i)$, the forcing term is $f(x) = d\pi^2 u^*(x)$.

We begin with the residual-gradient decomposition. At a collocation point x_j,

$$r_\theta(x_j) = -\sum_{i=1}^d \partial^2_{x_i} u_\theta(x_j) - f(x_j),$$

and because f is fixed data,

$$\nabla_\theta r_\theta(x_j) = -\sum_{i=1}^d \nabla_\theta \partial^2_{x_i} u_\theta(x_j).$$

Thus, the direction of the PDE residual gradient is determined by a sum of d parameter-sensitivity terms, while the residual formula also contains the forcing term $f(x) = d\pi^2 u^*(x)$, which scales linearly with d.

Two competing effects then matter. If the vectors $\nabla_\theta \partial^2_{x_i} u_\theta(x_j)$ are weakly correlated across coordinates, then the normalized sum $d^{-1} \sum_i \nabla_\theta \partial^2_{x_i} u_\theta(x_j)$ can become more concentrated as d grows, which tends to stabilize the residual-gradient direction after rescaling. At the same time, the scalar residual multiplier $r_\theta(x_j)$ depends on the competition between $-\sum_i \partial^2_{x_i} u_\theta(x_j)$ and the explicit $O(d)$ forcing term, so its magnitude can either grow, shrink, or partially cancel depending on the approximation error. Since the PDE-loss gradient has the form

$$\nabla_\theta \mathcal{L}_{\mathrm{PDE}} \propto \sum_j r_\theta(x_j) \nabla_\theta r_\theta(x_j),$$

its cosine with the boundary gradient depends on both effects simultaneously.

Taken together, the conclusion is as follows. The concentration effect can push the coupling toward more stable alignment, while the residual-amplitude term can amplify or damp whatever misalignment remains. Because these effects act in opposite directions, the residual-gradient decomposition by itself does not fix the net d-dependence of the instantaneous PDE and BC cosine. Without further control of the boundary-gradient scaling or of the training trajectory that defines profiled quantities, this decomposition-level argument also does not yield a monotone law for $f_{\mathrm{neg}}(d)$. Determining whether the trajectory is decreasing, increasing, or non-monotone requires additional assumptions or explicit constructions tied to the forcing family and parameter-sensitivity structure. □

G Experimental Setup

G.1 PDE Benchmarks

We evaluate on 60+ PDE configurations, counting distinct PDE families together with parameter, BC, dimension, inverse, and multi-physics variants:

Forward problems ($K = 3$). Burgers, Allen–Cahn, Helmholtz, Conv-Diff, Klein–Gordon, and Navier–Stokes make up this six-PDE set. Each PDE uses PDE residual, BC, and IC losses. The additional forward set includes 3D Helmholtz, reaction-diffusion, Taylor–Green vortex, and Darcy flow.

Expanded criterion validation for $K = 3$ and $K = 4$. Twenty additional forward-PDE configurations broaden the empirical coverage beyond the main forward suite. The twelve parameter variations are Helmholtz high-frequency (Helmholtz-HF, $\omega = 10$), Burgers high-viscosity (Burgers-HV, $\nu = 0.1$), two-dimensional (2D) Poisson (Poisson-2D), one-dimensional (1D) heat (Heat-1D), Korteweg–de Vries (KdV), 2D Laplace (Laplace-2D), sharp Allen–Cahn (Allen–Cahn-Sharp), heavy Klein–Gordon (KG-Heavy, $m = 2$), Fisher–Kolmogorov–Petrovsky–Piskunov (Fisher–KPP), Telegrapher, Fokker–Planck, and Burgers-Neumann. The eight new operator families are Cahn–Hilliard with fourth-order parabolic structure, Kuramoto–Sivashinsky with chaotic fourth-order structure, Sine–Gordon as a nonlinear wave equation, Gray–Scott as a two-component reaction-diffusion system with $K = 4$, Porous Medium as nonlinear diffusion, Burgers–Fisher as reaction–convection–diffusion, 2D convection (Convection-2D) as a steady problem, and Biharmonic as a fourth-order elliptic problem with $\nabla^4 u = f$. These span all major operator classes, namely elliptic, parabolic, hyperbolic, fourth-order, and multi-component settings, and they substantially broaden forward-problem coverage.

Inverse problems with $K = 3$ and $K = 4$. Inverse Burgers with parameter ν, inverse heat with parameter κ and $K = 4$, and inverse Poisson with parameter α include learnable physical parameters that create the gradient heterogeneity needed to understand surgery failures.

Multi-physics with K from 4 to 6. The multi-physics suite includes thermoelastic coupling with $K = 4$ terms for the heat PDE, wave PDE, BC, and IC; a $K = 5$ heat–elasticity variant with an explicit coupling loss; reactive transport with $K = 5$ losses for two-species advection–diffusion–reaction and genuine coupling conflict from the $k_r c_1 c_2$ reaction term; and simplified MHD with $K = 6$ losses for mass conservation, momentum, induction, the divergence-free constraint, BC, and IC.

G.2 K-Scaling Experiments

To evaluate how performance changes with K in the reported results, we focus on natural multi-loss systems: $K = 4$ thermoelastic coupling, $K = 5$ heat–elasticity and reactive transport, and $K = 6$ MHD. These systems represent genuinely distinct physics at each loss term rather than artificial residual splitting, directly addressing the concern that K-scaling validation may rely on constructed decompositions. The regime-transition framework is used empirically. Once profiling shows that the surrogate $\widehat{R}$ has dropped below the adapter-favoring range, scalar reweighting is expected empirically to dominate in these settings.

G.3 Methods Compared

We compare a broad method suite that includes Vanilla with 140K parameters and Vanilla-Wide with 162K capacity-matched parameters as baselines; gradient-surgery and gradient-alignment methods PCGrad, CAGrad, NashMTL, ConFIG [25], and AlignedMTL [36]; loss-weighting methods GradNorm, FAMO, uncertainty weighting [16], causal training [42], and self-adaptive weights [29]; and domain decomposition through XPINN [12]. ConFIG constructs conflict-free updates, whereas AlignedMTL performs gradient alignment based on SVD or eigendecomposition. The XPINN baselines use 4 and 8 subdomains with roughly comparable capacity, with interface and flux continuity constraints, providing a direct comparison between gradient-space and input-space decomposition. We also compare our architectural variants FAMO+UAM, which is the recommended default with 161K parameters and a 15% parameter overhead, and GN+UAM.

Additional variants such as UAM, CAM, LCAM, and their loss-weighting combinations serve as comparison controls. We also test architecture generality by adding per-loss adapters to Modified MLP [41] on 6 PDEs and to a Transformer backbone [50] on 4 PDEs, as discussed in Appendix M.4.

Causal training [42] enforces temporal causality via exponentially weighted residual chunks, and self-adaptive weights [29] learn loss weights as differentiable parameters. NTK weighting [43] is evaluated on a representative subset of four PDEs described in Section I.6 using 3 seeds, with reduced collocation points for computational feasibility.

For inverse problems, we additionally report PCGrad-Grouped and CAGrad-Grouped, which apply surgery only to network parameters while updating learnable physical parameters with vanilla updates.

G.4 Evaluation and Reproducibility Protocol

Evaluation. We report relative L_2 error, namely, $\|u_\theta - u_{\text{ref}}\|_2 / \|u_{\text{ref}}\|_2$, on a held-out grid. Methods are ranked from 1 to M per PDE, where 1 is best and failed methods receive the worst rank. Forward-benchmark comparisons are additionally summarized with Bayesian bootstrap comparisons using a Dirichlet prior, which quantify the probability of improvement $P(A < B)$. In the aggregate significance tables, effect sizes are reported as median relative improvement: $\Delta = \text{median}\left((L_2^{\text{base}} - L_2^{\text{method}})/L_2^{\text{base}}\right)$. For a small number of headline pairwise summaries in the prose, we additionally report Cohen's d on the seed-level L_2 values.

Reproducibility Protocol. Seed counts vary by suite and are stated in each table or caption. The main 10K forward benchmarks use 5 random seeds from 0 to 4. Many supplementary or higher-cost studies use 3 seeds, and targeted sensitivity checks use fewer when explicitly noted. The selected forward comparisons in Table 34 rerun seeds 5 to 7 in addition to the default seeds 0 to 4, giving 8 total seeds for those entries. The original benchmark suites run from a PyTorch 2.8 codebase on graphics processing unit (GPU) hardware, with suite-specific fixed seeds and deterministic collocation sampling. Additional validation protocols and their execution environment are stated in Appendix J. The default training setup uses 10,000 epochs, the AdamW optimizer with a learning rate of 10^{-3} and a weight decay of 10^{-5}, cosine annealing with a 200-epoch warm-up, and gradient clipping at a norm of 1.0. The implementation is hardware-agnostic across standard PyTorch GPU backends. PDE-specific settings and all hyperparameters are listed in Appendix H.

H Architecture and Hyperparameter Details

H.1 Default Hyperparameters

Unless otherwise stated, all experiments use a hidden dimension $d_h = 128$, $L = 4$ residual blocks, $F = 10$ Fourier bands with maximum frequency 10π, and an adapter rank $r = 16$. We train with AdamW at a learning rate of 10^{-3} and a weight decay of 10^{-5}, with 200 warm-up epochs. Klein–Gordon uses a higher learning rate of 5×10^{-3}. Adapter-specific settings were determined via a grid search on Burgers and then fixed across all other benchmarks. The settings are EMA momentum $\beta_{\text{ema}} = 0.95$, mixing update frequency $\tau = 20$, mixing bias $a_0 = 1.0$, mixing sensitivity $b = 2.0$, latent mixing score range $[\rho_{\min}, \rho_{\max}] = [0.15, 0.95]$, depth prior range $[\delta_1, \delta_L] = [+0.3, -0.3]$, orthogonality weight $\lambda_{\text{ortho}} = 0.01$, cross-orthogonality weight $\lambda_{\text{cross}} = 0.001$, a gradient clip norm

of 1.0, and magnitude-aware conflict scoring enabled. Per-layer methods use a gradient clip norm of 0.8.

H.2 PDE-Specific Settings

The default training budget is 10,000 epochs for the main forward suite and most supplementary evaluations unless otherwise stated. Klein–Gordon requires a higher learning rate of 5×10^{-3} because its u_{tt} term produces initial PDE residuals $\sim 10^4$. For per-layer methods on Klein–Gordon, we use a learning rate of 10^{-3} with gradient clipping at a norm of 0.5 and warm-up extended to 500 epochs. 3D problems use 10,000 epochs with 2,000 collocation points unless otherwise stated. The dedicated Helmholtz-3D validation in Table 23 uses 15,000 epochs. Inverse problems use 10,000 epochs with learnable parameters initialized via log-parameterization. Multi-physics and K-scaling suites use experiment-specific budgets. The corresponding tables cover 5,000 to 15,000 epochs, so their captions should be treated as definitive for those experiments.

H.3 Experiment-wise Dataset, Task, and Budget Summary

Main forward benchmarks are Burgers, Helmholtz, Allen–Cahn, Conv-Diff, and Klein–Gordon. These synthetic forward-PDE benchmarks recover the solution field using only physics losses. The runs use 5 seeds, 10K epochs, $n_{\text{coll}} = 2000$, $n_{\text{bc}} = n_{\text{ic}} = 400$, 10^4 test points, and the default backbone with $d_h = 128$, $L = 4$, $F = 10$, and $r = 16$. Klein–Gordon uses a learning rate of 5×10^{-3}, whereas the others use a learning rate of 10^{-3}.

Inverse problems include inverse Burgers, inverse heat, and inverse Poisson. These settings require joint recovery of the state and one unknown coefficient, namely, ν, κ, or α. We keep the same network and sampling budget, use 10K epochs with log-parameterized scalar initialization, and report 3 to 5 seeds depending on the method.

The natural multi-physics and K-scaling suite includes the natural $K = 4$ thermoelastic benchmark, a coupled heat–wave problem on $[-1, 1] \times [0, 1]$ with a finite-difference reference solution on a 200×200 grid. These runs use 2K epochs, 1.5K collocation points, and 5 seeds. The natural $K = 5$ and $K = 6$ studies use reactive transport and simplified MHD with finite-difference references, 15K epochs, $n_{\text{coll}} = 2000$, $n_{\text{bc}} = n_{\text{ic}} = 400$, and 5 seeds.

The expanded forward validation adds 20 forward PDE configurations and parameter variations, including Poisson-2D, Heat-1D, KdV, Laplace-2D, 1D wave (Wave-1D), Fisher–KPP, Telegrapher, Fokker–Planck, Cahn–Hilliard, Kuramoto–Sivashinsky, Sine–Gordon, Gray–Scott, Porous Medium, Burgers–Fisher, Convection-2D, and Biharmonic. Unless a table caption states otherwise, these runs use 3 seeds, 10K epochs, $d_h = 128$, $r = 16$, and the same collocation budget as the main suite.

Architecture transfer evaluates the generality of Modified MLP across 6 PDEs, namely, Helmholtz, Burgers, Klein–Gordon, Conv-Diff, Allen–Cahn, and Wave-1D, with 3 seeds and 10K epochs. It evaluates the generality of the Transformer backbone across 4 PDEs, namely, Helmholtz, Burgers, Allen–Cahn, and Klein–Gordon, with 3 seeds, 15K epochs, $d_h = 128$, $r = 16$, and $n_{\text{coll}} = 2000$.

MTL transfer uses two-task benchmarks built from the Modified National Institute of Standards and Technology (MNIST) and Fashion-MNIST datasets. The benchmarks are Multi-MNIST and Multi-FashionMNIST, respectively. Both overlay two digits or garments in a single 36×36 image and predict the left and right labels. They use 60K training samples, 10K test samples, 5 seeds, 100 epochs, a batch size of 256, a hidden width of 256, an adapter rank of 16, and Adam with a learning rate of 10^{-3}. The Canadian Institute for Advanced Research 100-class dataset (CIFAR-100) transfer experiment uses fine-label and coarse-superclass prediction as a two-task classification problem with a shared residual network (ResNet)-style encoder. It uses 3 seeds, 50 epochs, a batch size of 128, a hidden width of 512, an adapter rank of 32, and Adam with a learning rate of 10^{-3}.

H.4 Baseline Implementation Details

Unless otherwise stated, all baselines use the same trunk architecture, data sampling, outer AdamW optimizer, scheduler, and training budget. Method-specific inner updates, auxiliary optimizers, parameter grouping, and learning-rate sweeps are stated here or in the corresponding sensitivity sections.

FAMO [24] maintains logits $z \in \mathbb{R}^K$ updated via dual ascent. The model loss is $\sum_k w_k \mathcal{L}_k$, where $w = \mathrm{softmax}(z)$ with a lower clamp of 0.01 followed by renormalization. After each step, the logits are updated componentwise as $z_k \leftarrow z_k + \gamma(\mathcal{L}_k^{(t)} - \mathcal{L}_k^{(t-1)})$ with $\gamma = 0.01$, which focuses weight on losses that are worsening.

GradNorm follows the reference algorithm [2]. Task weights $w \in \mathbb{R}^K$, initialized to 1, are represented as an `nn.Parameter` and optimized by a separate Adam optimizer with $\eta_w = 0.025$. At each step, gradient norms $G_W^{(k)} = w_k \|\nabla_{W_\ell} \mathcal{L}_k\|_2$ are computed with respect to the last shared layer W_ℓ. Their average is $\bar{G} = K^{-1} \sum_k G_W^{(k)}$, the normalized loss ratios are $\tilde{\mathcal{L}}_k(t) = \mathcal{L}_k(t)/\mathcal{L}_k(0)$ and $\xi_k(t) = \tilde{\mathcal{L}}_k(t)/((1/K)\sum_j \tilde{\mathcal{L}}_j(t))$, and the targets are $G_{\text{target}}^{(k)} = \bar{G}\xi_k(t)^\alpha$ with $\alpha = 1.5$. The GradNorm loss $L_{\text{grad}} = \sum_k |G_W^{(k)} - G_{\text{target}}^{(k)}|$ is then minimized by backpropagation through w.

PCGrad [49] and CAGrad [23] use per-loss gradients over the same trainable parameter vector for the forward benchmarks. For inverse-problem grouped variants, projection or conflict-averse updates are applied only to network parameters, while learnable physical scalars receive vanilla AdamW updates as described in the parameter-grouped surgery section.

NashMTL uses projected gradient ascent in log-space to maximize the Nash product [32]. The Gram matrix $G_{ij} = \langle g_i, g_j \rangle$ is regularized as $G + 0.01 \cdot \bar{G}_{ii} \cdot I$. We update the weights through $\alpha \leftarrow \alpha \cdot \exp(\eta_\alpha \nabla_\alpha)$ with clamped increments satisfying $|\Delta| \leq 0.5$, use 25 inner iterations, and warm-start from the previous solution. A numerical-instability fallback reverts to uniform weights. This is a GPU-native adaptation of NashMTL that uses gradient-based optimization rather than the original solver-based implementation by Navon et al. [32], avoiding expensive host round-trips on GPU hardware.

AlignedMTL [36] computes the top-k SVD of the gradient matrix and projects the average gradient onto the principal subspace. ConFIG uses iterative projection to find a conflict-free direction, with geometric mean scaling [25].

Uncertainty weighting [16] learns per-loss log-variance parameters with the standard $\exp(-s_k)\mathcal{L}_k + s_k$ formulation. Causal training [42] uses exponentially weighted residual chunks, and self-adaptive weighting [29, 47] learns differentiable loss weights. NTK weighting and XPINN settings are described in their dedicated comparison sections. These shared defaults and explicitly stated method-specific deviations define the reproducibility protocol used for the comparisons.

H.5 Model Sizes

Vanilla has 140,033 parameters. Vanilla-Wide has 162,013 parameters with $d_h \approx 138$ to match adapter capacity. UAM has 160,772 parameters with fixed adapters and a 15% parameter overhead. CAM has 160,776 parameters with global conflict-aware mixing and a 15% parameter overhead. LCAM has 197,661 parameters with per-layer conflict-aware mixing scores and a 41% parameter overhead. These adapter parameters remain active in the forward pass at inference, so the listed parameter overhead is not a training-only cost. For higher-K settings, adapter count scales linearly with K.

H.6 Natural $K = 4$ Thermoelastic System

The natural $K = 4$ benchmark couples a heat equation with a wave equation on $[-1, 1] \times [0, 1]$:

$$u_t = Du_{xx} + \alpha v, \tag{18}$$

$$v_{tt} = c^2 v_{xx} + \beta u, \tag{19}$$

where $D = 0.01$, $\alpha = 0.5$, $c = 1.0$, and $\beta = 0.3$. The system uses ICs $u(x, 0) = \sin(\pi x)$, $v(x, 0) = \cos(\pi x/2)$, $v_t(x, 0) = 0$, and Dirichlet BCs $u(\pm 1, t) = v(\pm 1, t) = 0$. The reference solution is computed by finite differences on a 200×200 grid, using Crank–Nicolson for the heat equation and Verlet integration for the wave equation. This thermoelastic formulation produces four natural loss terms. These are $\mathcal{L}_{\text{heat}}$ for the heat PDE residual, $\mathcal{L}_{\text{wave}}$ for the wave PDE residual, $\mathcal{L}_{\text{BC}}$ for the BCs, and $\mathcal{L}_{\text{IC}}$ for the ICs of both fields. Training uses 2,000 epochs with 1,500 collocation points.

Algorithm 2 Regime-aware method selection

Require: A PDE problem with K losses, planned optimizer/LR, budget policy $B_{\max}$

if problem has learnable physical parameters **then**
  if $K = 3$ **then**
    return FAMO or GradNorm without adapters
  else if $K = 4$ **then**
    return FAMO+UAM {$K = 4$ inverse benefits empirically}
  end if
end if
Run $T_{\text{prof}} = 1000$ Vanilla steps using the planned optimizer/LR
Compute $\widehat{f}_{\text{neg}}$ and the first-/last-third means. Set $P = \bar{f}^{\text{late}}_{\text{neg}} / \max(\bar{f}^{\text{early}}_{\text{neg}}, 10^{-8})$ and fit the slope against normalized time $(t-1)/(T_{\text{prof}}-1)$; also report $\widehat{D}, \widehat{M}$
Evaluate Vanilla profiling L_2 error ϵ_{van}
if $\epsilon_{\text{van}} < 10^{-3}$ **then**
  return FAMO alone {an easy problem, adapters unnecessary}
end if
if $\widehat{f}_{\text{neg}} < 0.05$ **then**
  return FAMO alone {negligible directional conflict}
else if $P > 0.8$ **then**
  return FAMO+UAM if $B_{\max}$ permits its approximately $1.3\times$ training cost; otherwise FAMO
else if $P < 0.5$ **and** slope < -0.02 **then**
  return FAMO alone {transient conflict resolves}
else
  Report a borderline profile and its continuous diagnostics
  return FAMO for a strict budget; otherwise provisional FAMO+UAM or aggregate three profiles
end if

H.7 Parameter-Grouped Gradient Surgery

The PCGrad-Grouped and CAGrad-Grouped variants partition the parameter vector θ into network parameters θ_{net}, which include hidden weights, biases, and embeddings, and physical parameters θ_{phys}, which include learnable scalars such as viscosity ν, diffusivity κ, and source coefficient α. Gradient projection with PCGrad or the conflict-aware update with CAGrad is applied only to $\nabla_{\theta_{\text{net}}} \mathcal{L}_k$. Physical parameters receive vanilla gradient updates. This grouped-update scheme avoids the catastrophic scale mismatch between $\|\nabla_{\theta_{\text{net}}} \mathcal{L}_k\| \sim 10^{-2}$ and $\|\nabla_{\theta_{\text{phys}}} \mathcal{L}_k\| \sim 10^{2}$.

H.8 ConFIG Baseline

ConFIG [25] computes a conflict-free gradient direction by iterative pairwise projections with 20 iterations and scales the update by the geometric mean of per-loss gradient norms. It uses the same learning rate, weight decay, and architecture as the vanilla baseline. ConFIG incurs roughly $1.7\times$ to $1.9\times$ empirical training-time overhead in our setup, as Table 8 shows, due to the extra per-loss gradient computation and iterative projection.

I Supplementary Cost, Scalability, and Diagnostic Analyses

I.1 Regime-Aware Method Selection

Algorithm 2 therefore does not treat K alone as sufficient evidence for method selection; the natural higher-K multi-physics results provide empirical guidance, while the final choice is made by the profiling diagnostics.

Profiling criterion. The primary discriminator is the window-average negative-pair fraction $\widehat{f}_{\text{neg}} := T_{\text{prof}}^{-1} \sum_t f^{(t)}_{\text{neg}}$ from a 1000-step Vanilla profile under the selected optimizer and LR: $\widehat{f}_{\text{neg}} < 0.05$ indicates negligible conflict. We write $\bar{f}^{\text{early}}_{\text{neg}}$ and $\bar{f}^{\text{late}}_{\text{neg}}$ for the averages over the first and last thirds

Table 6: Diagnostic regimes and recommended interventions.

Profile	Choice	Appropriate Methods
Low conflict or easy profile	FAMO only	[41, 24]
High $\widehat{M}$, low $\widehat{R}$	FAMO or GradNorm	[2, 24]
Transient: $P < 0.5$, slope < -0.02	Reweight only	[2, 24]
Persistent: $P > 0.8$	FAMO+UAM	[8, 10, 45]
Inverse or heterogeneous	Avoid full-space surgery	[49, 23, 32]

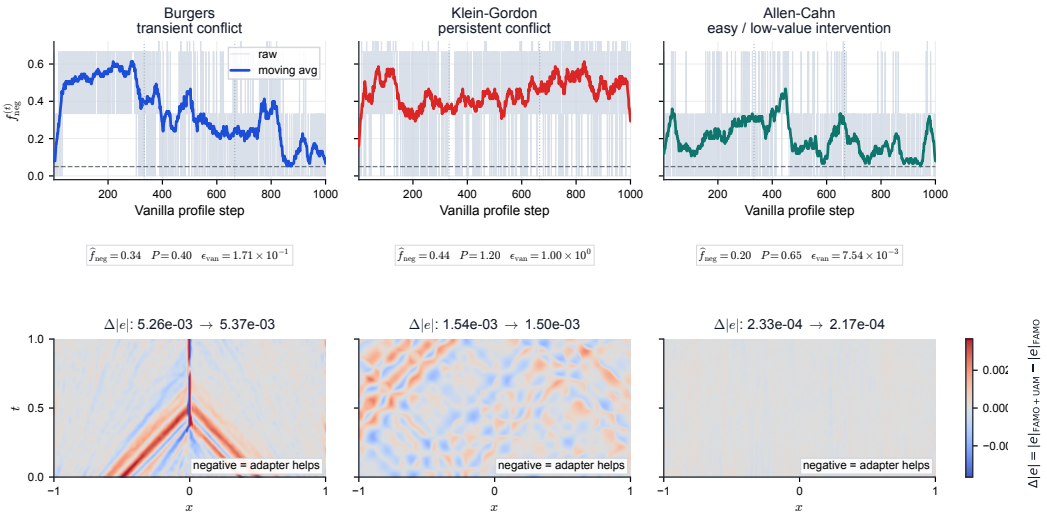

Figure 3: Profiling-to-outcome bridge on three representative forward PDEs. The top row shows the 1000-step Vanilla negative-pair trajectory $f_{\text{neg}}^{(t)}$ together with the profile summary $(\widehat{f}_{\text{neg}}, P, \epsilon_{\text{van}})$, and the bottom row visualizes the local error change $\Delta|e| = |e|_{\text{FAMO+UAM}} - |e|_{\text{FAMO}}$, where negative values indicate regions in which the adapter variant helps. Burgers exhibits transient conflict with $P = 0.40$, and the adapter effect stays confined to a thin shock-adjacent band, yielding essentially tied final errors of 5.26×10^{-3} and 5.37×10^{-3}. Klein–Gordon maintains persistent conflict with $P = 1.20$ and a poor profiled Vanilla error $\epsilon_{\text{van}} = 1.00$, and the adapter-induced change concentrates along oscillatory ridges, improving the final error from 1.54×10^{-3} to 1.50×10^{-3}. Allen–Cahn shows weaker conflict with $\widehat{f}_{\text{neg}} = 0.20$ and a much smaller profiled Vanilla error $\epsilon_{\text{van}} = 7.54 \times 10^{-3}$, and correspondingly the error-difference map is visually negligible.

of that profile, define $P = \bar{f}_{\text{neg}}^{\text{late}} / \max(\bar{f}_{\text{neg}}^{\text{early}}, 10^{-8})$, and fit the slope against normalized time $(t-1)/(T_{\text{prof}}-1)$. The 100-profile validation in Appendix J.3 retains the original thresholds and UAM fallback; its votes are not changed by the additional budget-aware reporting policy. We use these quantities as a deployment heuristic for choosing between FAMO and FAMO+UAM. Broader comparisons against Vanilla, UAM, or surgery baselines are reported separately in the main result tables. When conflict is present, $P > 0.8$ is treated as evidence of persistent conflict, while $P < 0.5$ with negative slope indicates transient conflict likely to resolve without architectural separation. The full procedure in Algorithm 2 should therefore be read as a practical decision rule, not as a guarantee of global optimality. It combines an easy-problem filter with persistence-based classification.

I.2 Profiling-to-Outcome Bridge

Figure 3 connects the 1000-step profiling signal used by Algorithm 2 to the final local error difference between FAMO and FAMO+UAM on representative PDEs. It shows why the selection rule depends on persistence rather than merely detecting negative cosine pairs. Persistent conflict yields spatially structured regions where adapters help, while transient or low-impact conflict leaves little qualitative change.

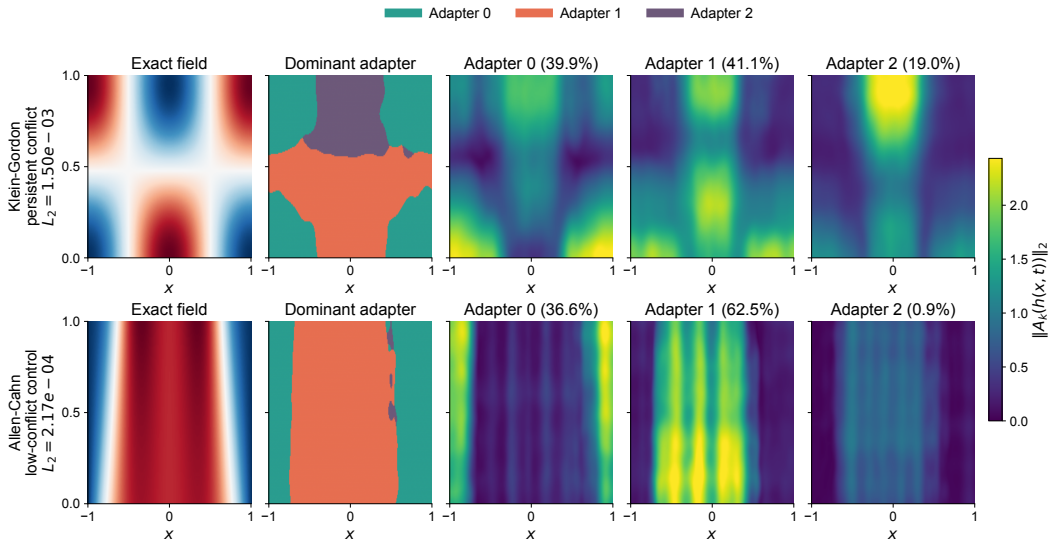

Figure 4: Illustrative spatial specialization of learned per-loss adapters on a persistent-conflict case and a low-conflict control; branch colors are local to each trained model. Each row shows the exact field, the dominant adapter index at each space-time point, and the contribution norms $\|A_k(h(x,t))\|_2$ of the three final adapters in the trained FAMO+UAM model. On Klein–Gordon, all three adapters occupy non-trivial regions of the slab, with shares of 39.9%, 41.1%, and 19.0%, and split the dynamics into boundary- and interior-sensitive correction zones. On Allen–Cahn, by contrast, one adapter dominates 62.5% of the domain and the third adapter is almost unused at 0.9%. Thus, in the adapter-beneficial regime, the additional low-rank branches do more than uniformly rescale the shared trunk; they learn spatially distinct corrections for the same physical field.

I.3 Adapter Specialization Visualization

Figure 4 complements the shared-output adapter construction in Section 2.3 and the blockwise NTK interpretation in Appendix D. It illustrates spatially distinct corrections in an adapter-beneficial case and more concentrated branch contributions in a low-conflict control. Branch identities are permutation-symmetric; the seed-aggregated measurements below quantify correction size without assigning fixed physical meaning to individual colors.

Correction size and initialization. The spatial maps in Figure 4 are illustrative: adapter indices are permutation-symmetric, so a branch color does not have a seed-invariant physical identity. We therefore complement them with permutation-invariant, seed-aggregated measurements. Adapter up-projections are zero-initialized; across all five main PDEs and seeds 0–2, the initial correction/trunk ratio, output change, and active-versus-removed prediction difference are exactly zero.

Table 7 reports the learned mixed correction relative to the trunk and the relative prediction change when that correction is removed, using the fixed 10K AdamW+FAMO+UAM checkpoints. In Helmholtz and Klein–Gordon, corrections are 78–91% of the trunk norm and change predictions by 25–29% on average. The Allen–Cahn control uses a 5% feature correction. Across the five main PDEs, removing the trained correction worsens every tested seed by at least $23.9\times$. Removal establishes functional use after co-adaptation; the matched gate-off/on training comparisons in Appendix J.4 separately assess the accuracy effect of introducing the pathway.

I.4 Computational Cost and Scalability

Table 8 shows that FAMO+UAM incurs only a $1.3\times$ runtime cost relative to Vanilla, whereas GN+UAM rises to $1.8\times$ because of the extra gradient-norm computation. LCAM variants are substantially more expensive, with overhead ranging from $2.4\times$ to $3.0\times$. We therefore treat FAMO+UAM as the default cost-conscious adapter configuration rather than claiming a universal cost optimum

Table 7: Permutation-invariant adapter magnitudes at 10K updates, mean ± sample std over three seeds. The feature ratio measures the mixed correction relative to the trunk; output change is relative to the active model prediction.

PDE	Correction/trunk norm	Relative output change
Helmholtz	0.912 ± 0.149	0.246 ± 0.124
Klein–Gordon	0.777 ± 0.123	0.287 ± 0.177
Allen–Cahn	0.052 ± 0.017	0.027 ± 0.016

Table 8: Computational cost comparison at $d_h = 128$, 10K epochs. Wall-clock time (seconds) and overhead relative to Vanilla are shown separately for each PDE. All runs use 5 seeds except where noted. Parameter counts reflect the shared trunk plus any per-loss adapter modules. GradNorm timings reflect the reference algorithm [2] with gradient-norm computation with respect to the last shared layer.

		Helmholtz		Burgers	
Method	# Params.	Time (s)	Overhead	Time (s)	Overhead
Shared-output baselines					
Vanilla	140,033	697	1.00×	440	1.00×
FAMO	140,033	692	0.99×	456	1.04×
GradNorm	140,033	929	1.33×	613	1.39×
ConFIG	140,033	1201	1.72×	848	1.93×
Uniform adapter mixing (ours)					
UAM	160,772	933	1.34×	607	1.38×
FAMO+UAM	160,772	929	1.33×	615	1.40×
GN+UAM	160,772	1262	1.81×	820	1.86×
Layerwise conflict-aware adapter mixing (ours)					
FAMO+LCAM	197,661	1689	2.42×	1160	2.64×
GN+LCAM	197,661	1963	2.82×	1336	3.04×

across all budgets and PDEs. Figure 7 shows that higher rank mainly helps on persistent-conflict PDEs, while transient or no-conflict cases are largely rank-insensitive.

Adaptive rank selection. The default rank $r = 16$ performs well across benchmarks, and the saturation ratio $\Delta_{\text{sat}} = Kr/d_h$ is a useful capacity proxy. One simple profiling-based heuristic is

$$r_{\text{heur}} = \left\lceil \frac{d_h \widehat{f}_{\text{neg}}}{K} \right\rceil,$$

where $\widehat{f}_{\text{neg}}$ is the profiled average fraction of negative pairwise cosines. This rule is a capacity-matching heuristic rather than a direct theorem consequence, chosen so that $\Delta_{\text{sat}} \approx \widehat{f}_{\text{neg}}$. In persistent-conflict cases, the rule suggests a nontrivial adapter rank, whereas in no-conflict cases it collapses to zero, so no adapters are used. Figure 7 shows that larger ranks help mainly on persistent-conflict PDEs, while transient or no-conflict cases are comparatively insensitive. We recommend $r = 16$ as a conservative mid-range default; practitioners with a profiling step can use r_{heur} as a coarse starting point.

I.5 Training Dynamics Visualization

Figure 5 visualizes the training dynamics for five PDEs across the conflict regimes and shows when the performance gap appears and widens. On Helmholtz, which exhibits persistent conflict with $P > 0.8$, FAMO+UAM separates from Vanilla within the first 2000 epochs and reaches a $52.2\times$ gain in terminal median error at 20K epochs. Poisson-2D exhibits the same pattern. FAMO+UAM reaches 2.19×10^{-4} compared with Vanilla's 7.41×10^{-3}, which gives a $33.8\times$ improvement at 20K and confirms that persistent conflict drives sustained adapter benefit. On Allen–Cahn, where no conflict is detected, all methods converge to similar errors around 5×10^{-5}, with a much smaller accuracy gap

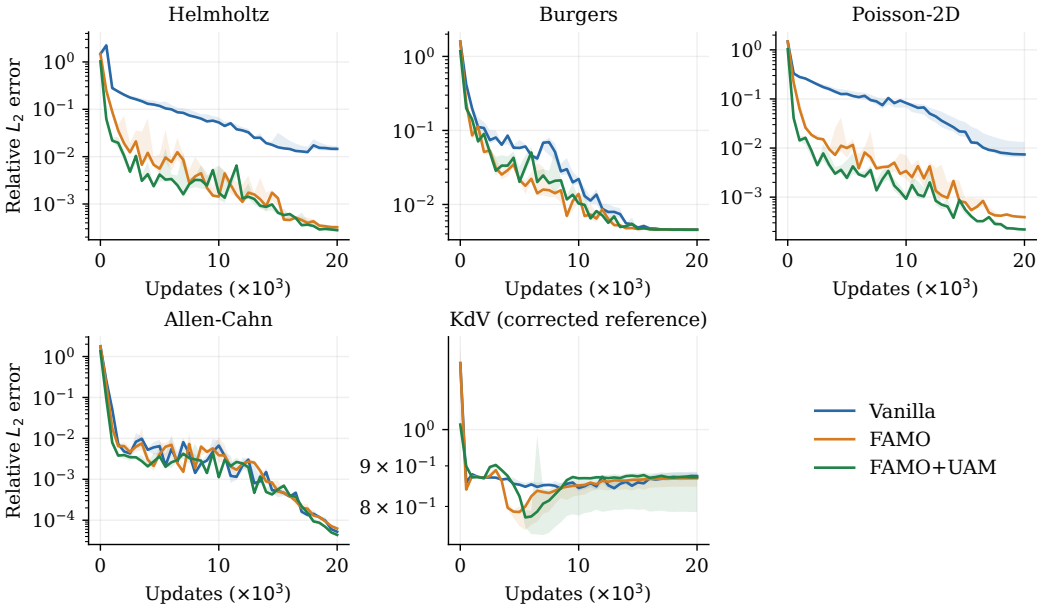

Figure 5: Training curves of relative L_2 error for five PDEs, with 20K updates, three seeds, median curves, and interquartile range (IQR) shading. On Helmholtz and Poisson-2D, FAMO+UAM separates from Vanilla early and reaches $52.2\times$ and $33.8\times$ gains at 20K (ratios of terminal medians). Allen–Cahn and Burgers converge similarly across methods. The KdV panel uses the corrected positive-soliton reference and the separate 20K runs in Appendix M.1.3; all three methods retain large errors.

than in the persistent-conflict cases. On Burgers, which exhibits mixed but transient conflict with $P = 0.63$, all methods converge to nearly identical errors of about 4.5×10^{-3}, with early separation that vanishes at extended training. With the corrected reference, the KdV 20K mean errors remain 0.875, 0.856, and 0.823 for Vanilla, FAMO, and FAMO+UAM, respectively. It remains difficult under the evaluated backbone and training configuration (Appendix M.1.3).

I.6 NTK Weighting Comparison

NTK-based loss weighting [43] adjusts loss weights via the NTK trace to balance learning speeds. We compare NTK weighting against Vanilla, FAMO, and FAMO+UAM on 4 representative PDEs spanning different conflict regimes, as reported in Table 9. NTK weighting adds negligible training overhead, about $1.06\times$ with periodic kernel recomputation every 100 steps, making it a low-cost alternative to FAMO+UAM, whose training-time cost is about $1.3\times$.

Results confirm that the intervention class should match the conflict type. On the high-conflict Helmholtz PDE, FAMO+UAM achieves 5.9×10^{-4} while NTK weighting reaches 7.5×10^{-4}, showing that persistent directional conflict can require architectural intervention rather than scalar reweighting. On Klein–Gordon, FAMO+UAM (1.57×10^{-3}) and NTK (1.58×10^{-3}) are comparable. On Burgers, which shows moderate and transient conflict, NTK weighting reaches 4.8×10^{-3} and slightly outperforms FAMO+UAM at 6.5×10^{-3}, which is consistent with the regime prediction that magnitude rebalancing suffices for transient conflict. On Allen–Cahn, where no conflict is detected, Vanilla reaches 1.6×10^{-4} and outperforms all intervention methods, including NTK at 1.0×10^{-3}, confirming that unnecessary intervention can be harmful. Figure 6 visualizes the training dynamics. On Helmholtz, FAMO+UAM separates from NTK weighting within the first 2000 epochs, confirming that the architectural advantage emerges early and persists.

Table 9: NTK weighting comparison on four representative PDEs. Lower relative L_2 error is better.

PDE	Vanilla	FAMO	NTK Weighting	FAMO+UAM (Ours)
Helmholtz	4.38×10^{-2}	1.25×10^{-3}	7.51×10^{-4}	$\mathbf{5.93 \times 10^{-4}}$
Klein–Gordon	4.62×10^{-2}	2.09×10^{-3}	1.58×10^{-3}	$\mathbf{1.57 \times 10^{-3}}$
Burgers	1.36×10^{-2}	8.33×10^{-3}	$\mathbf{4.83 \times 10^{-3}}$	6.46×10^{-3}
Allen–Cahn	$\mathbf{1.61 \times 10^{-4}}$	2.22×10^{-4}	1.01×10^{-3}	1.92×10^{-4}

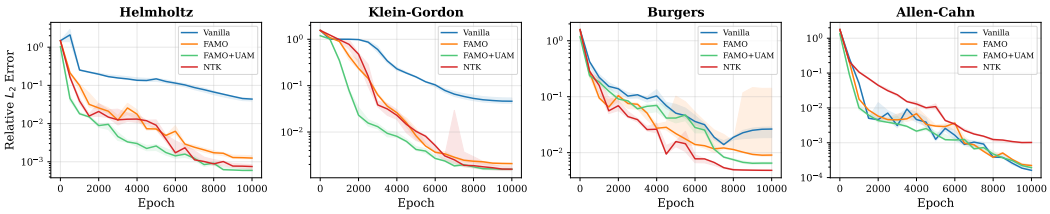

Figure 6: Training dynamics for the NTK-weighting comparison with 4 PDEs, 3 seeds, and IQR shading. On Helmholtz and Klein–Gordon, FAMO+UAM converges faster and achieves lower final error than NTK weighting. On Burgers, which shows transient conflict, NTK weighting matches or slightly outperforms FAMO+UAM. On Allen–Cahn, where no conflict is detected, Vanilla dominates.

I.7 Adaptive Rank Selection Validation

We use the saturation ratio $\Delta_{\text{sat}} = Kr/d_h$ as a simple capacity proxy for how much adapter specialization a fixed-width trunk can support. The profiling-based rule

$$r_{\text{heur}} = \left\lceil \frac{d_h \widehat{f}_{\text{neg}}}{K} \right\rceil$$

is a capacity proxy rather than a theoretical guarantee. Figure 7 therefore provides end-to-end empirical validation across three PDEs spanning different conflict regimes. On Helmholtz, which exhibits persistent conflict, performance improves monotonically with rank: $r = 4$ gives 6.95×10^{-4}, $r = 8$ gives 5.63×10^{-4}, $r = 16$ gives 3.72×10^{-4}, $r = 32$ gives 3.33×10^{-4}, and $r = 64$ gives 3.06×10^{-4}. This Helmholtz rank sweep spans $2.3\times$, with diminishing returns above $r = 32$. On Burgers, which exhibits transient conflict, all five ranks yield indistinguishable errors from 4.57×10^{-3} to 4.64×10^{-3}. This narrow range corresponds to only a $1.0\times$ span and is consistent with the regime prediction that scalar reweighting suffices. On Allen–Cahn, where no conflict is detected, rank has minimal effect. Errors range from 6.7×10^{-5} to 8.6×10^{-5}, which corresponds to a $1.3\times$ span and confirms that adapters are unnecessary. These results support the heuristic capacity picture. Higher-conflict PDEs benefit from larger adapter subspaces, and the $2.3\times$ gap between $r = 4$ and $r = 64$ on Helmholtz shows that underpowered adapters can leave performance on the table.

I.8 Scalability and Ablation Tables

Tables 10 and 11 test whether the adapter benefit persists across changes in network capacity and collocation density rather than depending on a single training configuration. Several observations follow from these stress tests. First, increasing trunk capacity does not turn the adapter into a uniformly beneficial add-on. On low-conflict Allen–Cahn, Vanilla remains competitive or best, which is consistent with the profiling rule that recommends no architectural intervention when directional conflict is absent. Second, Burgers shows the expected transient-conflict pattern. Across widths, depths, and collocation budgets, FAMO and adapter variants are close, and the best entry alternates rather than being dominated by the adapter; this indicates that scalar loss balancing and ordinary approximation capacity capture most of the remaining difficulty once the early conflict decays. Third, Helmholtz behaves differently. The best adapter-and-weighting variants stay between 10^{-4} and 10^{-3} error across network sizes, while Vanilla remains one to two orders of magnitude worse in most configurations. The collocation sweep gives the clearest signal: FAMO+UAM stays between 1.9×10^{-3} and 3.7×10^{-3} from 1K to 20K collocation points, whereas Vanilla stays near 10^{-1}. Thus, the Helmholtz improvement is not explained by a fortunate width and depth choice or by

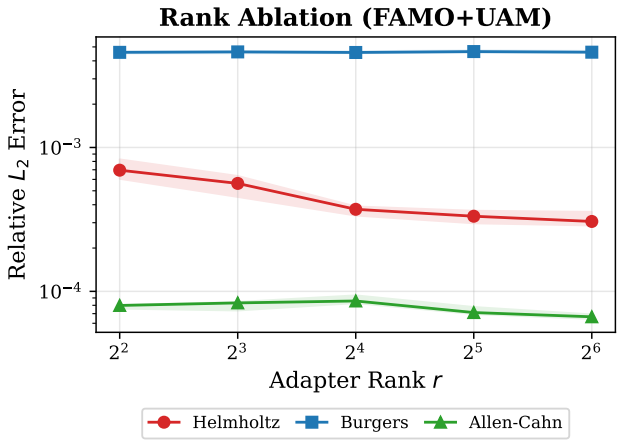

Figure 7: Adapter rank ablation across three PDEs spanning different conflict regimes. Optimal rank correlates with conflict persistence. Helmholtz benefits from higher rank, while Allen–Cahn is largely rank-insensitive.

Table 10: Scalability analysis of network size effects on adapter benefits. Relative L_2 error is averaged over 3 seeds. Bold indicates the best result for each configuration. †GN+UAM scalability data uses a simplified GradNorm variant, whereas the main-forward results in Tables 1 and 2 use the reference algorithm.

PDE	Config	Vanilla	FAMO	FAMO+UAM	GN+UAM†
Allen–Cahn	$d_h = 128, L = 4$	$\mathbf{5 \times 10^{-4}}$	7×10^{-4}	7×10^{-4}	7×10^{-4}
	$d_h = 128, L = 8$	$\mathbf{4 \times 10^{-4}}$	5×10^{-4}	6×10^{-4}	5×10^{-4}
	$d_h = 256, L = 4$	$\mathbf{2 \times 10^{-4}}$	3×10^{-4}	$\mathbf{2 \times 10^{-4}}$	$\mathbf{2 \times 10^{-4}}$
	$d_h = 256, L = 8$	$\mathbf{1 \times 10^{-4}}$	2×10^{-4}	2×10^{-4}	5×10^{-4}
	$d_h = 512, L = 4$	$\mathbf{1 \times 10^{-4}}$	$\mathbf{1 \times 10^{-4}}$	$\mathbf{1 \times 10^{-4}}$	4×10^{-4}
Burgers	$d_h = 128, L = 4$	3.17×10^{-2}	1.08×10^{-2}	$\mathbf{1.04 \times 10^{-2}}$	1.06×10^{-2}
	$d_h = 128, L = 8$	6.14×10^{-2}	1.33×10^{-2}	$\mathbf{9.5 \times 10^{-3}}$	1.00×10^{-2}
	$d_h = 256, L = 4$	4.41×10^{-2}	$\mathbf{5.8 \times 10^{-3}}$	1.03×10^{-2}	9.4×10^{-3}
	$d_h = 256, L = 8$	2.07×10^{-2}	$\mathbf{7.4 \times 10^{-3}}$	$\mathbf{7.4 \times 10^{-3}}$	1.10×10^{-2}
	$d_h = 512, L = 4$	2.93×10^{-2}	1.35×10^{-2}	$\mathbf{9.9 \times 10^{-3}}$	1.01×10^{-2}
Helmholtz	$d_h = 128, L = 4$	1.248×10^{-1}	5.3×10^{-3}	2.0×10^{-3}	$\mathbf{4 \times 10^{-4}}$
	$d_h = 128, L = 8$	9.62×10^{-2}	2.2×10^{-3}	1.9×10^{-3}	$\mathbf{3 \times 10^{-4}}$
	$d_h = 256, L = 4$	4.42×10^{-2}	1.0×10^{-3}	5×10^{-4}	$\mathbf{2 \times 10^{-4}}$
	$d_h = 256, L = 8$	4.48×10^{-2}	8×10^{-4}	4×10^{-4}	$\mathbf{1 \times 10^{-4}}$
	$d_h = 512, L = 4$	9.6×10^{-3}	6×10^{-4}	6×10^{-4}	$\mathbf{3 \times 10^{-4}}$

sparse residual sampling; it reflects a persistent directional-conflict bottleneck that benefits from loss-indexed parameter subspaces.

J Optimizer, Budget, and Diagnostic Validation

J.1 Optimizer Substitution Protocol

Table 3 uses the frozen benchmark implementation with the official SOAP optimizer[1] and the PINN settings reported by Wang et al. [45]: LR 10^{-3}, $(\beta_1, \beta_2) = (0.99, 0.999)$, and preconditioner refresh every two steps. Width, depth, collocation sampling, gradient clipping, the 200-step warm-up/cosine schedule, and the 10K-update horizon are held fixed. The high-frequency stress problem uses $\omega = 20$; it is distinct from the $\omega = 10$ variant in the original expanded suite. KG-Heavy uses $m = 2$.

[1] `https://github.com/nikhilvyas/SOAP`, commit `a1e553530fde97d0e6b307d7c82ac6d38b072340`.

Table 11: Scalability with collocation points. Relative L_2 error, averaged over 3 seeds. On Helmholtz, FAMO+UAM maintains roughly $34\times$ to $72\times$ improvement over Vanilla across all tested N_{coll} values, where $N_{\text{coll}} \in \{1\text{K}, 2\text{K}, 5\text{K}, 10\text{K}, 20\text{K}\}$, confirming robustness to collocation density.

PDE	N_{coll}	Vanilla	FAMO	FAMO+UAM
Burgers	1,000	2.60×10^{-2}	$\mathbf{1.33 \times 10^{-2}}$	1.50×10^{-2}
	2,000	2.89×10^{-2}	1.43×10^{-2}	$\mathbf{7.7 \times 10^{-3}}$
	5,000	2.03×10^{-2}	1.07×10^{-2}	$\mathbf{5.0 \times 10^{-3}}$
	10,000	1.83×10^{-2}	8.0×10^{-3}	$\mathbf{4.7 \times 10^{-3}}$
	20,000	1.35×10^{-2}	7.7×10^{-3}	$\mathbf{4.7 \times 10^{-3}}$
Helmholtz	1,000	1.070×10^{-1}	7.1×10^{-3}	$\mathbf{1.9 \times 10^{-3}}$
	2,000	1.275×10^{-1}	4.8×10^{-3}	$\mathbf{3.7 \times 10^{-3}}$
	5,000	1.372×10^{-1}	5.3×10^{-3}	$\mathbf{1.9 \times 10^{-3}}$
	10,000	1.092×10^{-1}	4.0×10^{-3}	$\mathbf{2.0 \times 10^{-3}}$
	20,000	1.510×10^{-1}	6.1×10^{-3}	$\mathbf{2.3 \times 10^{-3}}$

Table 12: SOAP+FAMO+UAM within each seed's SOAP-only 10K time. Relative L_2 is mean $\pm$ sample std over three seeds; updates and budgets are seed means. Gain is the ratio of arithmetic mean errors, SOAP/complete.

PDE	Relative L_2	Updates	Budget (s)	Gain
Helmholtz	$(2.51 \pm 0.36) \times 10^{-4}$	7,417	694.7	$16.38\times$
Klein–Gordon	$(8.30 \pm 4.65) \times 10^{-4}$	7,000	751.0	$9.14\times$
Allen–Cahn	$(2.53 \pm 0.47) \times 10^{-4}$	6,667	500.4	$0.14\times$
Helmholtz-HF	$(4.03 \pm 2.09) \times 10^{-3}$	6,667	655.7	$208.27\times$
KG-Heavy	$(8.67 \pm 1.80) \times 10^{-4}$	6,750	725.7	$6.48\times$

These are complete-configuration comparisons on the submitted models (140,033 baseline parameters and 160,772 with UAM). The matched linear-readout controls below instead hold parameter count and readout fixed to identify the adapter-gate effect. The optimizer-substitution experiment tests complementarity within this PINN pipeline; the backbone, constraints, curricula, and longer training horizons of Wang et al. [45] are not replicated.

All entries in Table 3 use seeds 0–2, terminal relative L_2 at the fixed final checkpoint, and an RNG-isolated test set. We report arithmetic mean and sample standard deviation; neither checkpoints nor hyperparameters are selected on these test errors. The additional optimizer, gate, diagnostic, and targeted Helmholtz runs use PyTorch 2.5.1 on NVIDIA GH200 GPUs. The original benchmark tables retain their own reported protocols. All comparisons of elapsed times below are made within the corresponding suite and hardware setting.

J.2 Accuracy under Equal Training Time

For Table 12, each PDE/seed receives that seed's measured SOAP-only 10K training time. We select the last scheduled checkpoint within the deadline using elapsed time only, without compressing or retuning the schedule. The complete configuration reaches approximately 6.7K–7.4K updates and preserves the gains on the four persistent-conflict problems. SOAP alone remains preferable on Allen–Cahn. These budgets measure training after configuration selection; the separate 1000-step diagnostic is not charged to these training-time comparisons.

Table 13 directly includes the more expensive LCAM variants. Here the deadline is each seed's AdamW-Vanilla 10K training time, and all four methods retain the submitted AdamW settings. FAMO is the lowest-error choice at this deadline on Burgers and Helmholtz, while FAMO+UAM has the lowest mean error on Klein–Gordon. The latter also improves Vanilla by $61.4\times$; on Helmholtz its gain over Vanilla is $6.69\times$. LCAM reaches only 3.25K–3.75K updates, and its additional cost is not recovered in these comparisons. This motivates adapter-free FAMO for strict or uncertain budgets and UAM when persistent-conflict improvements justify its approximately $1.3\times$ cost.

Table 13: Equal AdamW-Vanilla-10K training time. Terminal error at the last time-feasible checkpoint, mean ± sample std over seeds 0–2; mean completed updates and seed-matched budget.

PDE	Method	Relative L_2	Updates	Budget (s)
Burgers	Vanilla	$(5.48 \pm 0.75) \times 10^{-3}$	10,000	325.6
Burgers	FAMO	$(4.78 \pm 0.44) \times 10^{-3}$	10,000	325.6
Burgers	FAMO+UAM	$(6.64 \pm 1.84) \times 10^{-3}$	7,250	325.6
Burgers	FAMO+LCAM	$(3.29 \pm 0.92) \times 10^{-2}$	3,250	325.6
Helmholtz	Vanilla	$(1.01 \pm 0.69) \times 10^{-2}$	10,000	483.4
Helmholtz	FAMO	$(1.13 \pm 0.14) \times 10^{-3}$	9,417	483.4
Helmholtz	FAMO+UAM	$(1.51 \pm 0.80) \times 10^{-3}$	6,750	483.4
Helmholtz	FAMO+LCAM	$(4.20 \pm 1.01) \times 10^{-3}$	3,333	483.4
Klein–Gordon	Vanilla	$(4.42 \pm 3.10) \times 10^{-2}$	10,000	596.4
Klein–Gordon	FAMO	$(7.70 \pm 3.25) \times 10^{-4}$	10,000	596.4
Klein–Gordon	FAMO+UAM	$(7.19 \pm 0.91) \times 10^{-4}$	7,583	596.4
Klein–Gordon	FAMO+LCAM	$(6.32 \pm 0.76) \times 10^{-3}$	3,750	596.4

Table 14: Natural $K = 4$ thermoelasticity at a common 190.5-s training deadline, five seeds. Errors are mean ± sample std; updates are seed means. † denotes the suite's simplified GradNorm variant.

Method	Updates	Relative L_2
Vanilla	1920	$(5.14 \pm 0.84) \times 10^{-1}$
FAMO	1800	$(1.16 \pm 0.35) \times 10^{-1}$
FAMO+UAM	1320	$(1.25 \pm 0.12) \times 10^{-1}$
GN+UAM†	1320	$\mathbf{(8.67 \pm 1.12) \times 10^{-2}}$
FAMO+LCAM	600	$(3.81 \pm 0.34) \times 10^{-1}$
GN+LCAM†	600	$(3.14 \pm 0.55) \times 10^{-1}$

Natural multi-physics. A separate saved-trajectory comparison uses the natural $K = 4$ thermoelastic suite and a common deadline of 190.5 s, the median Vanilla time for its nominal 2K-update budget. Table 14 reports the last checkpoint at this deadline, rather than the best error observed before it. The five-seed GN+UAM result improves Vanilla by $5.94\times$, while the LCAM variants reach only 600 updates. This comparison retains the simplified GradNorm variant used for this suite in Table 21; it is separate from the seed-matched 10K forward controls above.

J.3 Seed, Learning-Rate, and Optimizer Sensitivity of Profiling

We evaluated 100 independent 1000-step profiles: five main PDEs, five initialization/sampling seeds, AdamW at $0.3\times$, $1\times$, and $3\times$ the nominal LR, and SOAP at the nominal LR. The nominal LR is 10^{-3} except for Klein–Gordon (5×10^{-3}). These short profiles use a constant LR, an unmodified PINN, raw per-loss gradients at every update, and the original thresholds. SOAP uses its default $\beta = (0.9, 0.999)$ and refresh 2 in this sensitivity suite. This is distinct from the common-LR, $\beta_1 = 0.99$ optimizer-substitution test in Table 3. Thresholds were fixed before running these profiles and were not recalibrated using downstream errors.

Table 15: Five-seed modal recommendation and votes for the frozen profiling rule. UAM means FAMO+UAM. The original ambiguous-case fallback is UAM; the budget-aware borderline reporting described in Section 4.4 is not applied retroactively to these votes.

PDE	AdamW $0.3\times$	AdamW $1\times$	AdamW $3\times$	SOAP $1\times$
Helmholtz	UAM 5/5	UAM 5/5	UAM 5/5	UAM 5/5
Klein–Gordon	UAM 5/5	UAM 5/5	UAM 5/5	UAM 5/5
Burgers	UAM 5/5	UAM 3/5	UAM 3/5	UAM 5/5
Conv-Diff	UAM 3/5	FAMO 5/5	FAMO 5/5	FAMO 5/5
Allen–Cahn	UAM 4/5	FAMO 4/5	FAMO 3/5	UAM 5/5

Table 16: Continuous profile diagnostics, mean ± sample std over five seeds per condition. A/S denote AdamW/SOAP. Persistence uses the first and last 333 steps; slope uses time normalized to $[0, 1]$.

PDE	Optimizer/LR	$\widehat{f}_{\mathrm{neg}}$	$\widehat{D}$	$\widehat{M}$	P	Slope
Helmholtz	A 0.3×	0.36 ± 0.04	0.03 ± 0.01	1.40 ± 0.01	1.24 ± 0.26	0.11 ± 0.11
Helmholtz	A 1×	0.46 ± 0.05	0.05 ± 0.03	1.39 ± 0.00	1.39 ± 0.25	0.23 ± 0.12
Helmholtz	A 3×	0.52 ± 0.01	0.10 ± 0.02	1.39 ± 0.00	1.22 ± 0.15	0.18 ± 0.08
Helmholtz	S 1×	0.35 ± 0.02	0.03 ± 0.01	1.40 ± 0.00	1.47 ± 0.30	0.20 ± 0.06
Klein–Gordon	A 0.3×	0.43 ± 0.04	0.09 ± 0.03	1.30 ± 0.02	1.59 ± 0.26	0.28 ± 0.07
Klein–Gordon	A 1×	0.52 ± 0.02	0.13 ± 0.03	1.03 ± 0.03	1.38 ± 0.19	0.21 ± 0.08
Klein–Gordon	A 3×	0.51 ± 0.03	0.28 ± 0.13	0.83 ± 0.12	1.40 ± 0.12	0.24 ± 0.04
Klein–Gordon	S 1×	0.52 ± 0.01	0.07 ± 0.00	1.17 ± 0.01	0.97 ± 0.11	-0.00 ± 0.08
Burgers	A 0.3×	0.57 ± 0.04	0.16 ± 0.02	1.21 ± 0.03	1.03 ± 0.07	0.03 ± 0.06
Burgers	A 1×	0.41 ± 0.07	0.15 ± 0.04	1.06 ± 0.05	0.55 ± 0.19	-0.34 ± 0.13
Burgers	A 3×	0.22 ± 0.11	0.07 ± 0.05	0.98 ± 0.07	0.56 ± 0.34	-0.22 ± 0.12
Burgers	S 1×	0.46 ± 0.02	0.14 ± 0.03	1.09 ± 0.05	0.80 ± 0.04	-0.14 ± 0.04
Conv-Diff	A 0.3×	0.64 ± 0.08	0.10 ± 0.03	1.01 ± 0.04	0.58 ± 0.15	-0.47 ± 0.22
Conv-Diff	A 1×	0.44 ± 0.07	0.09 ± 0.02	1.06 ± 0.04	0.22 ± 0.08	-0.82 ± 0.19
Conv-Diff	A 3×	0.31 ± 0.05	0.08 ± 0.01	0.54 ± 0.10	0.32 ± 0.15	-0.62 ± 0.15
Conv-Diff	S 1×	0.30 ± 0.03	0.05 ± 0.01	1.17 ± 0.01	0.33 ± 0.03	-0.58 ± 0.06
Allen–Cahn	A 0.3×	0.33 ± 0.05	0.04 ± 0.01	1.35 ± 0.01	0.73 ± 0.21	-0.16 ± 0.13
Allen–Cahn	A 1×	0.24 ± 0.03	0.05 ± 0.02	1.05 ± 0.09	0.34 ± 0.17	-0.31 ± 0.07
Allen–Cahn	A 3×	0.14 ± 0.05	0.06 ± 0.03	0.48 ± 0.12	0.79 ± 0.79	-0.16 ± 0.18
Allen–Cahn	S 1×	0.24 ± 0.03	0.03 ± 0.01	1.23 ± 0.01	0.72 ± 0.05	-0.14 ± 0.03

Helmholtz and Klein–Gordon select UAM in all 40 optimizer/LR/seed profiles; their persistence ranges are $[0.970, 1.995]$ and $[0.862, 2.037]$, respectively. At nominal AdamW LR, 22/25 seed votes agree with the modal choice. Across all four conditions, the modal choice is unchanged for three of the five PDEs. These are decision-reproducibility statistics, not selector-accuracy estimates.

The lower AdamW LR changes the majority choice for Conv-Diff and Allen–Cahn, since a fixed number of updates covers less optimization progress. SOAP also changes the Allen–Cahn mode to UAM, although its complete-configuration comparison favors SOAP alone. Thus Allen–Cahn is an informative low-benefit control, and the short raw-gradient profile is configuration-dependent. For borderline profiles, report the continuous diagnostics and seed votes, prefer the cheaper non-adapter branch under a strict budget, or aggregate three short profiles when greater confidence is needed. The latter is an optional deployment policy; Table 15 reports the original single-profile decisions.

J.4 Matched Adapter Gates and Conflict Trajectories

To isolate the adapter pathway, both gate-off/on models use a linear shared readout and have exactly 152,324 parameters, bitwise-identical initial states and outputs, identical sampling streams, FAMO, and the same adapter-evaluation workload. Only the gate multiplying the aggregate adapter correction differs. AdamW uses the submitted per-PDE LR; SOAP uses LR 10^{-3}, $\beta = (0.99, 0.999)$, and refresh 2. Each condition uses three paired seeds and 10K updates.

Figures 8 and 9 show raw, unweighted shared-trunk conflict, recorded before weighting or optimizer transformation. Measurements include initialization and every 500 steps. We call the arithmetic trajectory mean of $D = f_{\mathrm{neg}}|\bar{c}_-|$ the directional AUC. For the original compact early/mid/late summaries, the windows are the measured steps up to 1000, 4500–5500, and 9000–10000, respectively; these differ from the thirds of the short profiling window.

The gate reduces integrated destructive strength by 16%/43% on Helmholtz/Klein–Gordon under AdamW and by 33% on Klein–Gordon under SOAP. On SOAP/Klein–Gordon, late negative-pair frequency increases while negative cosines weaken and accuracy improves. The relevant signal therefore includes frequency, strength, magnitude, and persistence, rather than the sign alone. SOAP/Helmholtz is essentially neutral in the matched gate comparison, showing that complementarity depends on the regime.

Table 17: Strict gate intervention: raw directional AUC and paired geometric error gain (off/on), three seeds. Parentheses give the number of seeds with lower terminal error when the gate is on.

Optimizer	PDE	D off	D on	Error gain (wins)
AdamW	Helmholtz	0.0377	0.0318	1.22× (2/3)
AdamW	Klein–Gordon	0.0410	0.0234	1.41× (2/3)
SOAP	Helmholtz	0.0275	0.0265	0.96× (1/3)
SOAP	Klein–Gordon	0.0497	0.0332	1.65× (3/3)

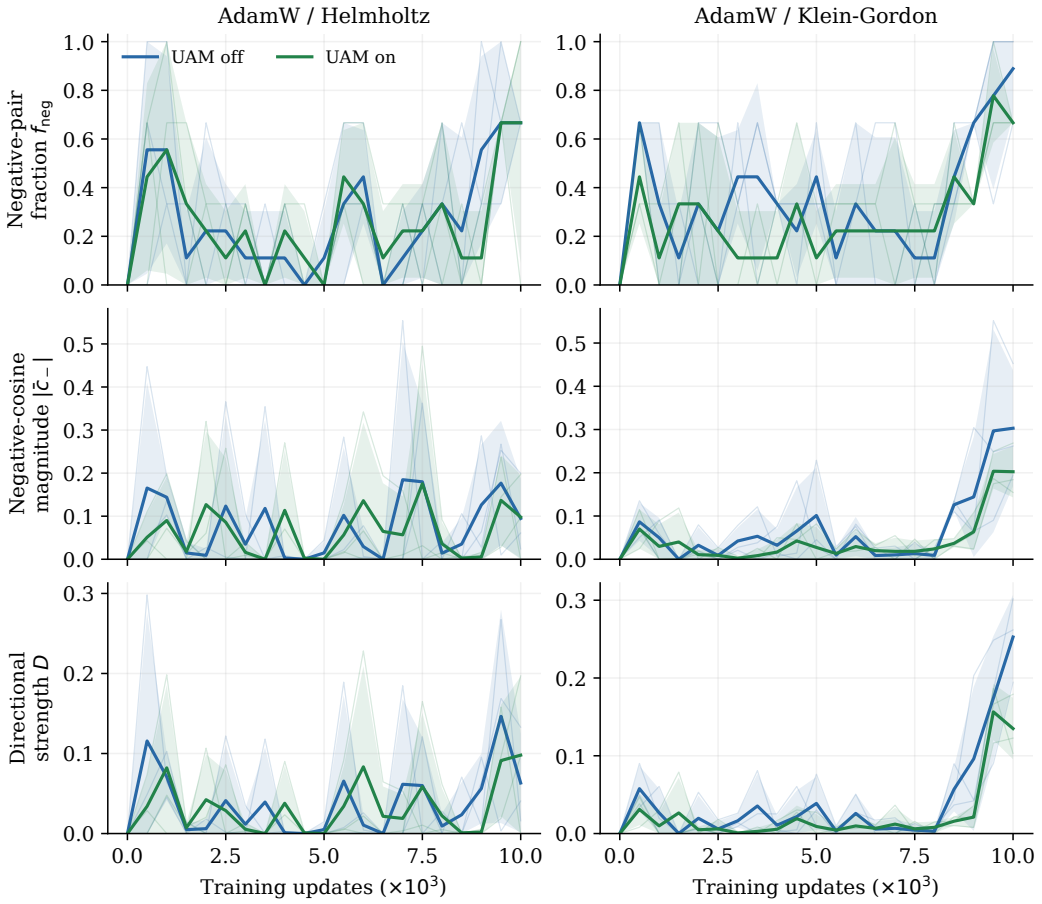

Figure 8: Raw common-trunk conflict under AdamW with FAMO fixed and only the UAM gate changed. Thin lines show the three individual seeds, thick lines their mean, and shading ± one sample standard deviation (clipped to the metric range for display). Measurements are taken every 500 updates, with initialization also shown.

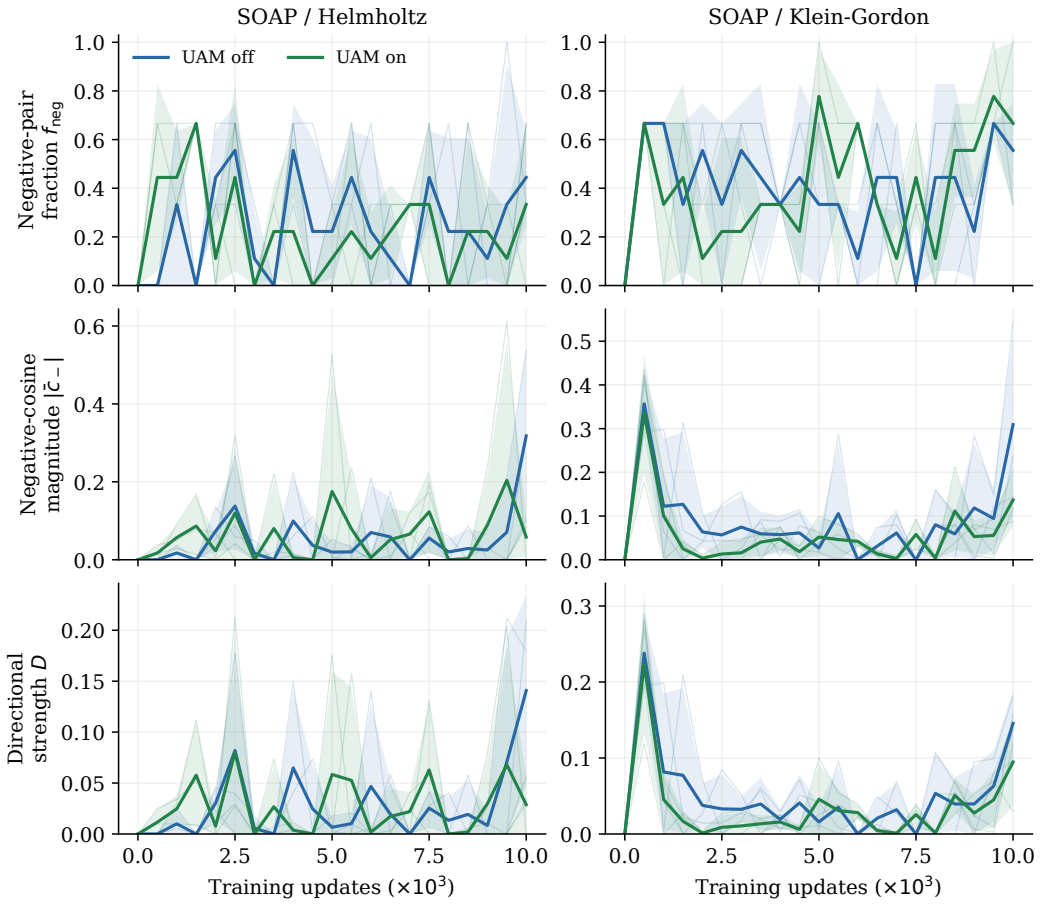

Figure 9: Matched gate trajectories under SOAP with the same plotting conventions as Figure 8. Raw negative-pair frequency can remain high even when the average negative-cosine magnitude and directional strength decrease. These raw metrics are distinct from the optimizer-transformed alignment in Table 18.

J.5 Raw-Gradient and Optimizer-Transformed Alignment

An instrumented rerun of the matched 2×2 AdamW/SOAP $\times$ UAM-off/on configuration records both metric conventions at the same model states on Helmholtz and Klein–Gordon, with three paired seeds and FAMO held fixed. The raw diagnostic is

$$D_t = \binom{K}{2}^{-1} \sum_{i<j} [-\cos(g_i^{\text{trunk}}, g_j^{\text{trunk}})]_+. \tag{20}$$

For the optimizer-coordinate comparison, form the current FAMO-weighted full-parameter components $\widetilde{g}_k = w_{k,t} \nabla_{\theta_{\text{full}}} \mathcal{L}_k$ and evaluate each candidate direction $h_k = \mathcal{T}_t(\widetilde{g}_k)$ independently from the same current optimizer state. Here $\mathcal{T}_t$ includes the optimizer's current moment transformation; the optimizer state is not advanced between candidate components. The intra-step score of Wang et al. [45] is

$$\mathcal{A}_{\text{intra}} = 2 \left\| \frac{1}{K} \sum_k \frac{h_k}{\|h_k\|} \right\|^2 - 1. \tag{21}$$

The directional AUC is lower-is-better and retains only negative strength; $\mathcal{A}_{\text{intra}}$ is higher-is-better and includes both positive and negative alignment. Table 18 averages post-initialization checkpoints (500–10000 in steps of 500), excluding the uninitialized optimizer state.

Table 18: Alignment in two coordinate systems, averaged over two PDEs, three paired seeds, and post-initialization checkpoints. Transformed columns use current FAMO-weighted full-parameter components.

	Raw trunk		Optimizer-transformed		
Configuration	$f_{\text{neg}} \downarrow$	$D \downarrow$	$\mathcal{A}_{\text{intra}} \uparrow$	$f_{\text{neg}} \downarrow$	$D \downarrow$
AdamW+FAMO, UAM off	.328	.0413	.338	.164	.0255
AdamW+FAMO+UAM	.281	.0290	.283	.211	.0360
SOAP+FAMO, UAM off	.333	.0406	.958	.000	.0000
SOAP+FAMO+UAM	.328	.0313	.957	.000	.0000

Table 19: Terminal accuracy in the instrumented matched factorial. Errors are geometric means over the six PDE-seed pairs; gains are relative to AdamW+FAMO with the gate off.

Configuration	Relative L_2	Gain
AdamW+FAMO, UAM off	9.49×10^{-4}	$1.00\times$
AdamW+FAMO+UAM	7.23×10^{-4}	$1.31\times$
SOAP+FAMO, UAM off	5.54×10^{-4}	$1.71\times$
SOAP+FAMO+UAM	4.45×10^{-4}	$2.13\times$

SOAP has no negative transformed pair at any measured post-initialization checkpoint in these weighted controls, while the raw diagnostics remain nonzero. Under the separate unweighted full-parameter metric convention, replacing AdamW with SOAP in the gate-off control increases trajectory-mean $\mathcal{A}_{\text{intra}}$ from -0.181 to 0.308 and reduces transformed D by 92.7%. This supports SOAP's qualitative alignment benefit under both measurement conventions.

Under AdamW, UAM reduces raw D by 29.9% and improves accuracy by $1.31\times$. Activating UAM on top of SOAP improves geometric-mean accuracy by $1.25\times$ (4/6 paired wins, including 3/3 on Klein–Gordon). UAM is an intermediate intervention in where and how extensively geometry is changed: its low-rank Jacobian directions are richer than scalar weighting and more localized than full-network quasi-second-order conditioning. This does not require its alignment score to lie numerically between the other methods, and the combined mechanisms remain useful even when transformed negative pairs have vanished.

Why negative signs alone are insufficient. For a positive-weighted first-order step $\Delta\theta = -\eta \sum_j \lambda_j g_j$,

$$\Delta\mathcal{L}_i = -\eta \left[\lambda_i \|g_i\|^2 + \sum_{j \neq i} \lambda_j \|g_i\| \|g_j\| \cos(g_i, g_j) \right] + O(\eta^2). \tag{22}$$

A negative cosine contributes an adverse cross-term, but a first-order loss increase requires the adverse terms to outweigh the self-term and positive cross-terms. For two unweighted losses, this requires $\cos(g_i, g_j) < -\|g_i\|/\|g_j\|$. Frequency measures how often negative pairs occur; D measures their aggregate strength; $\widehat{M}$ records scale asymmetry; and persistence separates sustained interference from transient events. This is the rationale for interpreting the joint profile rather than any one sign statistic.

K Supplementary Benchmark Tables

This section collects supplementary benchmark tables that extend the main forward and natural $K = 4$ evaluations used to support the regime classification.

L Statistical Comparison

Table 22 reports Bayesian bootstrap comparisons with 10,000 Dirichlet-weighted resamples over 25 matched PDE-seed pairs for each method comparison and shows that FAMO+UAM consistently

Table 20: Darcy flow 2D benchmark. We report L_2 error as mean $\pm$ std over 3 seeds at 5K epochs. GN+UAM achieves the best result with 5.6×10^{-2}, which is a $4.5\times$ improvement over Vanilla and confirms adapter benefits on this standard elliptic PDE benchmark with $K = 3$ losses and extreme magnitude imbalance $\bar{r} > 8,500$. †GradNorm uses a simplified variant, whereas the main-forward results in Tables 1 and 2 use the reference algorithm.

Method	L_2 Error	# Params.
Vanilla	$(2.48 \pm 0.29) \times 10^{-1}$	140K
FAMO	$(1.44 \pm 0.43) \times 10^{-1}$	140K
FAMO+UAM	$(9.7 \pm 0.8) \times 10^{-2}$	161K
FAMO+LCAM	$(8.8 \pm 0.9) \times 10^{-2}$	198K
GN+UAM†	$\mathbf{(5.6 \pm 0.8) \times 10^{-2}}$	161K

Table 21: Natural $K = 4$ coupled thermoelastic benchmark. Mean $\pm$ std L_2 error over 5 seeds. This naturally coupled system of heat and wave equations provides a $K = 4$ validation where the adapter-and-weighting combination outperforms either component alone. Best values are shown in bold. †GradNorm uses a simplified variant; the main-forward results in Tables 1 and 2 use the reference algorithm.

Method	L_2 Error	Rank
Vanilla	$(5.11 \pm 0.83) \times 10^{-1}$	10
FAMO	$(1.04 \pm 0.17) \times 10^{-1}$	5
GradNorm†	$(2.28 \pm 0.84) \times 10^{-1}$	6
UAM	$(4.15 \pm 0.58) \times 10^{-1}$	7
FAMO+UAM	$(7.95 \pm 2.38) \times 10^{-2}$	4
GN+UAM†	$(5.84 \pm 1.31) \times 10^{-2}$	2
CAM	$(4.57 \pm 0.92) \times 10^{-1}$	9
LCAM	$(4.34 \pm 1.06) \times 10^{-1}$	8
FAMO+LCAM	$(7.79 \pm 2.1) \times 10^{-2}$	3
GN+LCAM†	$\mathbf{(4.78 \pm 0.78) \times 10^{-2}}$	**1**

Table 22: Bayesian pairwise comparison on forward benchmarks with 5 PDEs, $d_h = 128$, and 10K epochs. P(A<B) denotes the Dirichlet-weighted bootstrap probability that Method A achieves lower L_2 error than Method B, using $n = 10,000$ resamples. Each comparison uses 25 matched PDE-seed error pairs. Probabilities above 0.95 are bold.

Method A	Method B	P(A<B)
FAMO+UAM	FAMO	**0.991**
FAMO+UAM	GN+UAM	**1.000**
FAMO+UAM	GradNorm	**1.000**
FAMO+UAM	FAMO+LCAM	0.824
FAMO+UAM	GN+LCAM	**1.000**
FAMO+UAM	Vanilla	**1.000**
FAMO	GN+UAM	**1.000**
FAMO	GradNorm	**1.000**
FAMO	FAMO+LCAM	0.053
FAMO	GN+LCAM	**1.000**
FAMO	Vanilla	**1.000**
GN+UAM	GradNorm	**1.000**
GN+UAM	FAMO+LCAM	0.000
GN+UAM	GN+LCAM	**1.000**
GN+UAM	Vanilla	**0.993**
GradNorm	FAMO+LCAM	0.000
GradNorm	GN+LCAM	0.000
GradNorm	Vanilla	0.063
FAMO+LCAM	GN+LCAM	**1.000**
FAMO+LCAM	Vanilla	**1.000**
GN+LCAM	Vanilla	0.773

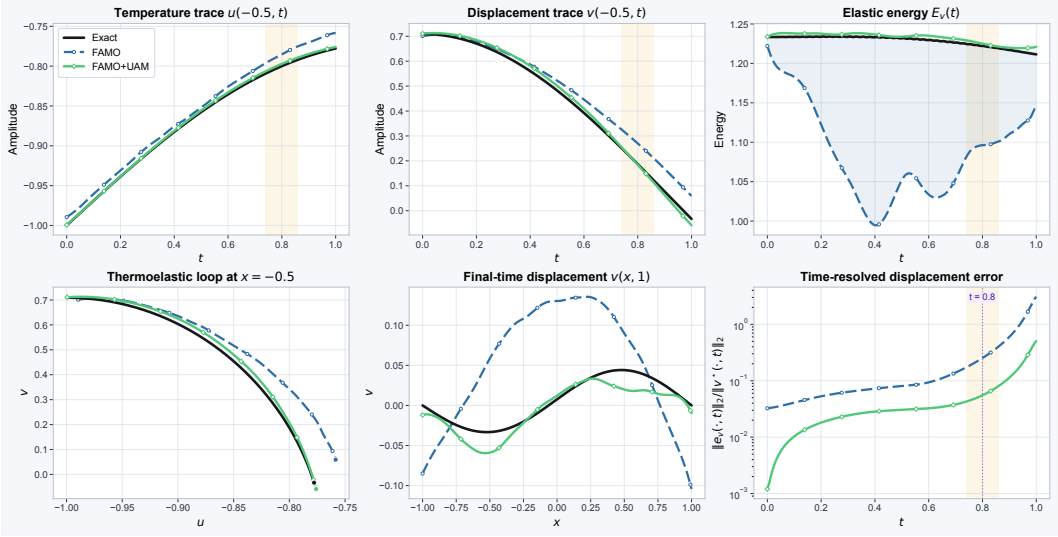

Figure 10: Energy exchange and branchwise dynamics on the natural $K = 4$ thermoelastic system. The top row compares a representative temperature trace $u(-0.5, t)$, displacement trace $v(-0.5, t)$, and elastic energy $E_v(t) = 1/2 \int (v_t^2 + c^2 v_x^2) dx$; the bottom row shows the local thermoelastic loop $(u(-0.5, t), v(-0.5, t))$, the final-time displacement profile, and the time-resolved displacement-profile error. The qualitative signal is branchwise rather than uniform. FAMO already tracks the thermal branch closely, but it keeps the displacement branch too positive at late time and underestimates the elastic-energy envelope, whereas FAMO+UAM restores the signed displacement swing and follows the coupled loop more faithfully. On this representative run, the mean relative elastic-energy error drops from 1.15×10^{-1} to 3.17×10^{-3}, and the displacement-profile error at $t \approx 0.8$ drops from 2.51×10^{-1} to 5.48×10^{-2}.

Table 23: 3D validation on Helmholtz-3D over $[0, 1]^3$ with $K = 3$, 5 seeds, $d_h = 128$, and 15K epochs. We report relative L_2 error as mean $\pm$ std, where lower is better. The adapter benefit persists in 3D with the same regime structure as in 2D.

Method	Relative L_2	Ratio
Vanilla	$(4.13 \pm 3) \times 10^{-2}$	1.0×
FAMO	$(3.58 \pm 2.7) \times 10^{-3}$	11.5×
PCGrad	$(1.80 \pm 0.59) \times 10^{-3}$	22.9×
ConFIG	$(2.21 \pm 0.14) \times 10^{-3}$	18.7×
FAMO+UAM	$\mathbf{(1.45 \pm 0.18) \times 10^{-3}}$	28.5×

outperforms Vanilla with $P > 0.99$, as well as GradNorm and most competitors on the 5 forward PDEs. The comparison between FAMO+UAM and FAMO remains close because FAMO alone is already strong on the less challenging PDEs. The gap widens on high-conflict problems like Klein–Gordon, where FAMO+UAM reaches 29× over Vanilla and FAMO reaches 19×.

M Extended Validation and Robustness

M.1 Extended Forward Validation

M.1.1 3D Validation

On 3D Helmholtz, shown in Table 23, FAMO+UAM achieves a 28.5× improvement over Vanilla. On Taylor–Green, which shows low conflict, Vanilla matches adapters and confirms that benefit tracks conflict rather than dimensionality.

The 25-pair Bayesian bootstrap comparisons in Table 22 support the same pattern. FAMO+UAM is favored over Vanilla and most competitors on the forward suite.

Table 24: Expanded PDE benchmarks on 12 additional forward PDEs. We report median L_2 error with 3 seeds, $d_h = 128$, and 10K epochs unless noted. Lower is better. The operator-type column spells out elliptic, parabolic, hyperbolic, and damped wave. Bold indicates the best method for each PDE. The last column reports the Vanilla-to-best ratio.

PDE	Type	Vanilla	FAMO	FAMO+UAM	Ratio
Adapter benefit ($\geq 1.5\times$)					
Poisson-2D	Elliptic	4.21×10^{-2}	1.24×10^{-3}	$\mathbf{4.78 \times 10^{-4}}$	$88\times$
KG-Heavy	Hyperbolic	5.95×10^{-2}	2.64×10^{-3}	$\mathbf{1.52 \times 10^{-3}}$	$39\times$
Laplace-2D	Elliptic	1.30×10^{-2}	$\mathbf{5.81 \times 10^{-4}}$	5.83×10^{-4}	$22\times$
Telegrapher	Damped wave	7.02×10^{-1}	$\mathbf{1.58 \times 10^{-1}}$	$\mathbf{1.58 \times 10^{-1}}$	$4.5\times$
Helmholtz-HF*	Elliptic	9.39×10^{-1}	8.58×10^{-1}	$\mathbf{3.24 \times 10^{-1}}$	$2.9\times$
Fisher–KPP	Parabolic	6.11×10^{-3}	5.27×10^{-3}	$\mathbf{2.95 \times 10^{-3}}$	$2.1\times$
Burgers-HV	Parabolic	2.78×10^{-4}	1.95×10^{-4}	$\mathbf{1.53 \times 10^{-4}}$	$1.8\times$
No adapter benefit					
KdV‡	Dispersive	7.64×10^{-1}	6.43×10^{-1}	$\mathbf{6.30 \times 10^{-1}}$	$1.21\times$
Burgers-Neumann	Hyperbolic	2.76×10^{-2}	$\mathbf{2.74 \times 10^{-2}}$	2.99×10^{-2}	$1.01\times$
Heat-1D	Parabolic	4.77×10^{-5}	5.95×10^{-5}	$\mathbf{4.69 \times 10^{-5}}$	$1.02\times$
Allen–Cahn-Sharp	Parabolic	$\mathbf{1.01 \times 10^{-3}}$	1.14×10^{-3}	1.15×10^{-3}	$1.00\times$
Fokker–Planck†	Parabolic	$\mathbf{4.47 \times 10^{-3}}$	7.26×10^{-1}	8.26×10^{-1}	$1.00\times$

*15K epochs. †FAMO catastrophically fails and becomes $162\times$ worse than Vanilla. ‡Reference-corrected 10K medians; protocol and learning rates are stated in Appendix M.1.3.

Table 25: Corrected KdV reference, three seeds. Mean $\pm$ sample std at each fixed training horizon. The 10K median is included to match Table 24; dashes denote configurations not evaluated at 20K.

Method	LR	10K mean $\pm$ std	10K median	20K mean $\pm$ std
Vanilla (AdamW)	3×10^{-3}	0.744 ± 0.055	0.764	0.875 ± 0.020
FAMO	10^{-3}	0.660 ± 0.108	0.643	0.856 ± 0.026
FAMO+UAM	10^{-3}	0.707 ± 0.142	0.630	0.823 ± 0.104
SOAP	3×10^{-3}	0.688 ± 0.167	0.611	—
SOAP+FAMO+UAM	3×10^{-3}	0.648 ± 0.175	0.652	—

M.1.2 Additional Forward Benchmarks

Table 24 broadens the forward evaluation across operator families. The gains remain strongly PDE-dependent. FAMO+UAM is clearly best on Poisson-2D, KG-Heavy, Helmholtz-HF, Fisher–KPP, and Burgers-HV, while several easier or magnitude-dominated cases show little or no benefit. Fokker–Planck exposes a FAMO instability, with a $162\times$ degradation from near-singular loss Hessians, which motivates log-space dual updates in Appendix M.3.

M.1.3 Reference-Consistent KdV Evaluation

The KdV operator is $u_t + 6uu_x + u_{xxx} = 0$, with reference soliton $u(x,t) = 2\,\mathrm{sech}^2(x - 4t)$. A reference-consistency audit identified a sign error in the earlier auxiliary target: the negative soliton does not satisfy this operator. On 20,000 audit points, the residual RMS is 9.722 for the negative target and 1.687×10^{-15} for the positive target. We use the positive solution consistently for ICs, BCs, and evaluation. The KdV row in Table 24 and the KdV panel in Figure 5 use these corrected results.

Table 25 reports the completed three-seed controls. The 10K study uses disjoint seed-11 LR selection for Vanilla/SOAP and retains the submitted 10^{-3} LR for AdamW FAMO/UAM; the 20K study uses the same listed AdamW LRs and a 20K cosine schedule. SOAP uses $\beta = (0.9, 0.999)$ and refresh 2. Other architecture, sampling, warm-up, and clipping settings are held to the benchmark implementation. Every value is evaluated at the fixed final checkpoint on an RNG-isolated test set. The 10K and 20K columns are separate training runs with their respective schedule horizons.

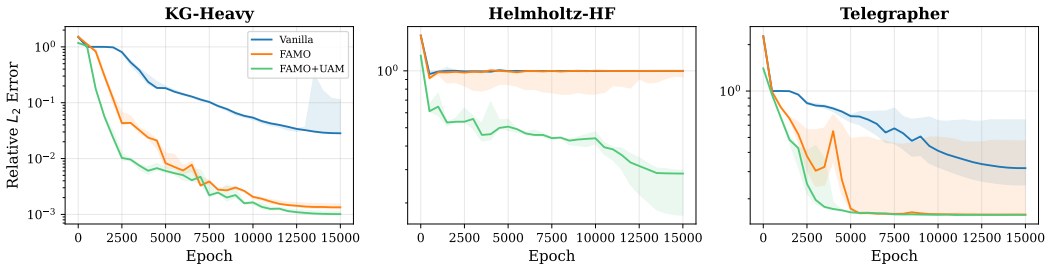

Figure 11: Training curves for three criterion-expansion PDEs with 3 seeds and IQR shading. KG-Heavy and Helmholtz-HF show persistent adapter benefit, with gains of $27.8\times$ and $3.4\times$, and the gap widens monotonically.

Table 26: Helmholtz improvement across scales. We report relative L_2 error for the best multi-loss method compared with Vanilla at different widths, epochs, and dimensions. Across the 10K and 15K settings, the improvement ratio consistently exceeds $19\times$, which confirms that the benefit is structural because it reflects conflict mitigation rather than capacity. Best values are shown in bold.

Config	Seeds	Epochs	Vanilla	Best method	Ratio
$d_h = 128$ (2D)	5	10K	2.8×10^{-2}	$\mathbf{1.0 \times 10^{-3}}$ (GN+UAM)	$28\times$
$d_h = 128$ (2D)	3	15K	1.2×10^{-2}	$\mathbf{4.4 \times 10^{-4}}$ (FAMO+UAM)	$27\times$
$d_h = 192$ (2D)	3	15K	7.3×10^{-3}	$\mathbf{2.6 \times 10^{-4}}$ (FAMO+UAM)	$28\times$
$d_h = 256$ (2D)	3	15K	8.1×10^{-3}	$\mathbf{2.2 \times 10^{-4}}$ (FAMO+UAM)	$36\times$
$d_h = 384$ (2D)	3	15K	3.8×10^{-3}	$\mathbf{2.0 \times 10^{-4}}$ (FAMO+UAM)	$19\times$
$d_h = 128$ (3D)	5	15K	4.1×10^{-2}	$\mathbf{1.5 \times 10^{-3}}$ (FAMO+UAM)	$28.5\times$

Although the corrected 10K group means are below 0.8, the relative errors remain large. At 20K, FAMO+UAM has the lowest mean error of the three evaluated methods, but none achieves an accurate solution under this backbone and training configuration. The corrected evidence therefore retains KdV as a difficult benchmark without attributing the earlier inconsistent-reference result to an architectural limitation.

M.1.4 Criterion Expansion Validation

To complement the 10K expanded forward-benchmark table with longer-horizon runs, we evaluate 6 representative PDEs at 15K epochs with 3 seeds each, as shown in Figure 11. These are separate 15K validation runs, not part of the 10K profiling and variant analyses discussed in Section 4.4. Results confirm the regime characterization. KG-Heavy, a hyperbolic case with $K = 3$, shows $27.8\times$ adapter benefit. Helmholtz-HF, the high-frequency variant, shows $3.4\times$, and Telegrapher shows $2.0\times$. Fisher–KPP is mixed at 15K. FAMO+UAM achieves a $1.6\times$ improvement over Vanilla, but FAMO attains the best median overall, with 3.10×10^{-3} compared with 3.39×10^{-3} for FAMO+UAM. The sharp Allen–Cahn variant, with a ratio of $1.0\times$, confirms no benefit for reaction-dominated PDEs. Fokker–Planck remains outside the adapter-beneficial regime. FAMO fails catastrophically, with median 3.4×10^{-1} compared with Vanilla's 1.6×10^{-3}, and FAMO+UAM is worse still. This case should therefore be read as a cautionary limitation of applying the default FAMO+UAM combination without screening for loss-scale stiffness rather than as evidence against the diagnostic-first framework.

M.1.5 Helmholtz Scaling Across Width, Dimension, and Training Budget

Helmholtz serves as a stress test for multi-loss methods. As shown in Table 26, the best listed multi-loss method improves over Vanilla by at least $19\times$ across six configurations spanning 2D and 3D settings, $d_h \in \{128, 192, 256, 384\}$, and 10K or 15K epochs, with a peak of $36\times$ at $d_h = 256$ and 15K epochs. The 3D row, drawn from Table 23, achieves $28.5\times$, indicating that the empirical Helmholtz benefit persists beyond 2D and is consistent with the same qualitative persistent-conflict regime, while Remark 3 should not be read as implying dimension-invariant effect sizes.

Table 27: Additional PDE evaluation with $d_h = 128$. Persistent-conflict PDEs show clear adapter gains, whereas low-conflict or transient-conflict cases show little or no adapter benefit, consistent with regime-dependent selection. Best values are bold and second-best values are underlined.

PDE	Vanilla	FAMO	FAMO+UAM	GN+UAM
10K evaluation subset with 3 to 5 seeds				
Klein–Gordon	$(4.6 \pm 1) \times 10^{-2}$	$\underline{(2.46 \pm 0.39) \times 10^{-3}}$	$\mathbf{(1.62 \pm 0.38) \times 10^{-3}}$	$(4.8 \pm 1.1) \times 10^{-2}$
Reaction-Diffusion	$\mathbf{(8.94 \pm 0.68) \times 10^{-4}}$ [3]	$\underline{(1.08 \pm 0.063) \times 10^{-3}}$	$(1.12 \pm 0.055) \times 10^{-3}$	$(3.59 \pm 0.35) \times 10^{-3}$
Darcy flow	$(1.34 \pm 0.49) \times 10^{-1}$ [3]	$(7.5 \pm 0.8) \times 10^{-2}$	$\underline{(6.9 \pm 0.3) \times 10^{-2}}$	$\mathbf{(6.8 \pm 0.5) \times 10^{-2}}$
Navier–Stokes	$(2.41 \pm 1.2) \times 10^{-4}$ [3]	$\mathbf{(1.75 \pm 0.23) \times 10^{-4}}$	$\underline{(2.45 \pm 0.2) \times 10^{-4}}$	$(4.34 \pm 0.63) \times 10^{-4}$
3-seed extended subset with 15K to 20K epochs				
Wave-1D	$(2.69 \pm 0.41) \times 10^{-2}$	$\underline{(2.11 \pm 0.53) \times 10^{-3}}$	$\mathbf{(1.44 \pm 0.67) \times 10^{-3}}$	NA
Advection	$\mathbf{(1.79 \pm 0.24) \times 10^{-3}}$	$\underline{(2.09 \pm 0.23) \times 10^{-3}}$	$(2.21 \pm 0.44) \times 10^{-3}$	NA
Eikonal (15K)	$\mathbf{(2.22 \pm 0.02) \times 10^{-4}}$	$(3.16 \pm 0.1) \times 10^{-4}$	$\underline{(2.69 \pm 0.076) \times 10^{-4}}$	NA
Eikonal (20K)	$\mathbf{(1.61 \pm 0.034) \times 10^{-4}}$	NA	$\underline{(1.80 \pm 0.071) \times 10^{-4}}$	NA
Poisson-Multiscale	$(1.74 \pm 0.09) \times 10^{-1}$	$\underline{(2.09 \pm 2.3) \times 10^{-2}}$	$\mathbf{(3.65 \pm 0.046) \times 10^{-3}}$	NA

Superscript $[n]$ denotes seed count when <5, and NA denotes not evaluated.

NA denotes not evaluated.

Table 28: Profile-score diagnostic validation on MTL benchmarks ($K = 2$). CIFAR-100 uses a ResNet-18 shared encoder.

Method	Multi-MNIST			Multi-FashionMNIST			CIFAR-100 (Fine+Coarse)		
	Task L	Task R	Avg	Task L	Task R	Avg	Fine	Coarse	Avg
Vanilla	98.6	98.2	98.40±0.17	88.4	87.6	87.98±0.28	48.3	63.2	55.75±0.19
PCGrad	98.6	98.3	98.45±0.18	88.9	88.1	**88.48**±0.21	50.2	65.0	**57.60**±0.23
FAMO	98.7	98.3	**98.47**±0.24	88.9	87.9	88.39±0.10	47.6	62.5	55.05±0.19
UAM	98.6	98.2	98.40±0.12	88.7	88.1	88.40±0.22	47.2	63.5	55.35±0.39
FAMO+UAM	98.6	98.2	98.40±0.12	88.9	87.9	88.40±0.14	47.4	63.4	55.40±0.16

Profile statistics: $\widehat{D} = 0.035$, $\widehat{M} = 0.55$, $f_{\text{neg}} = 0.37$ for Multi-MNIST, and $\widehat{D} = 0.056$, $\widehat{M} = 0.30$, $f_{\text{neg}} = 0.41$ for Multi-FashionMNIST. $\widehat{D} < 0.01$, $\widehat{M} = 0.17$, and $f_{\text{neg}} < 0.05$ for CIFAR-100, which places all three in the magnitude regime. The prediction is that adapters are unnecessary, and this prediction is supported because no adapter-based variant beats the strongest non-adapter baseline on Multi-MNIST, Multi-FashionMNIST, or CIFAR-100.

M.1.6 Additional PDE Generalization Checks

Across the supplementary PDE checks in Table 27, which use 3 to 5 seeds, the conflict profile remains directionally consistent. Clear adapter-beneficial cases include Klein–Gordon at $29\times$, Poisson-Multiscale at $48\times$, Wave-1D at $19\times$, and Darcy flow at $2.0\times$, while low-conflict or transient cases show little or no adapter advantage in Reaction-Diffusion, Navier–Stokes, Advection, and Eikonal. On Reaction-Diffusion, GN+UAM degrades $4\times$, giving the clearest case where the criterion correctly recommends against adapters.

M.2 Diagnostic Transfer to Standard MTL Benchmarks

The profiled ratio $\widehat{R} = \widehat{D}/(\widehat{M} + \varepsilon_R)$ is domain-agnostic in the sense that it depends only on loss-specific gradient geometry, not on PDE-specific structure. It compares implementation-level summaries of directional conflict and a coefficient-of-variation surrogate for magnitude imbalance between loss-specific gradients using the same profiling pipeline as in the PINN experiments. We test whether it correctly predicts the appropriate intervention class. This validation focuses on whether the regime classification transfers across domains, not on whether adapters improve MTL. We evaluate three standard $K = 2$ MTL benchmarks spanning magnitude-dominated settings, namely, Multi-MNIST, Multi-FashionMNIST [35], and CIFAR-100. Across all three benchmarks, the diagnostic is directionally consistent. No adapter-based variant is the best method on any benchmark, as shown in Table 28.

These MTL checks serve as an external sanity test for the do-not-intervene side of the rule. When magnitude imbalance dominates and task-specific heads already absorb much of the interference, architectural separation provides no consistent benefit.

Table 29: Architecture generality for the Transformer backbone. We report median relative L_2 error across 3 seeds at 15K epochs. The regime prediction transfers. The adapter-beneficial cases of Helmholtz, Burgers, and Klein–Gordon show clear gains on the Transformer architecture, while Allen–Cahn, a no-conflict control, remains essentially unchanged as predicted.

	MLP		Transformer	
PDE	Vanilla	FAMO+UAM	Vanilla	FAMO+UAM
Helmholtz	4.38×10^{-2}	5.93×10^{-4}	1.21×10^{-1}	$\mathbf{5.36 \times 10^{-4}}$
Burgers	1.36×10^{-2}	6.46×10^{-3}	1.17×10^{-2}	$\mathbf{4.83 \times 10^{-3}}$
Klein–Gordon	4.62×10^{-2}	$\mathbf{1.57 \times 10^{-3}}$	2.09×10^{-2}	1.82×10^{-3}
Allen–Cahn	1.61×10^{-4}	1.92×10^{-4}	$\mathbf{5.83 \times 10^{-5}}$	5.96×10^{-5}

M.3 Mitigating Catastrophic Weight Divergence with Bounded FAMO

The Fokker–Planck FAMO failure, which corresponds to a $162\times$ degradation at 10K epochs, reveals that FAMO's componentwise absolute-scale dual update

$$z_k \leftarrow z_k + \gamma(\mathcal{L}_k^{(t)} - \mathcal{L}_k^{(t-1)})$$

can diverge when losses span multiple orders of magnitude. A log-space variant,

$$z_k \leftarrow z_k + \gamma(\log(\mathcal{L}_k^{(t)} + \varepsilon_L) - \log(\mathcal{L}_k^{(t-1)} + \varepsilon_L)),$$

where $\varepsilon_L = 10^{-12}$, provides scale-invariant updates and achieves $L_2 = 1.1 \times 10^{-2}$ on Fokker–Planck. The log-space update gives a $64\times$ improvement over FAMO with 3 seeds and changes well-behaved PDEs by less than 5%. The bounded variant therefore substantially mitigates but does not fully eliminate the instability. It improves sharply over standard FAMO yet still trails the best stable baselines, and pairing it with adapters did not remove the remaining gap in our runs.

M.4 Architecture Generality

Modified MLP backbone. To verify architecture-agnostic regime transfer, we evaluate Modified MLP [41], a multiplicative-gating architecture widely used in PINN practice, on 6 PDEs with 3 seeds and 10K epochs. The pattern of adapter benefits transfers. Helmholtz shows a $734\times$ improvement with FAMO+UAM, and Klein–Gordon shows $154\times$. Wave-1D shows $495\times$ because adapters rescue non-converging vanilla Modified MLP, and Burgers shows $2.6\times$. Allen–Cahn and Conv-Diff show no benefit above $1.1\times$, which matches the regime predictions. This Modified MLP transfer result confirms that the profiling criterion operates on gradient statistics, not model structure.

Transformer backbone. To further verify architecture-agnostic regime transfer, we evaluate using a Transformer-based PINN backbone, a self-attention architecture that uses a learnable positional embedding over Fourier-encoded point features. This Transformer evaluation tests a fundamentally different inductive bias, namely global attention rather than local MLP processing, and the design is inspired by PINNsFormer [50]. We evaluate 4 PDEs, namely Helmholtz, Burgers, Allen–Cahn, and Klein–Gordon, across 2 configurations, Vanilla Transformer and FAMO+UAM Transformer, with 3 seeds each for a total of 24 experiments.

Table 29 reports the median L_2 errors. The regime prediction transfers cleanly. On Helmholtz with persistent conflict, FAMO+UAM Transformer achieves 5.4×10^{-4} compared with Vanilla Transformer's 1.2×10^{-1}, which gives a $225.8\times$ improvement. On Klein–Gordon, FAMO+UAM Transformer achieves 1.8×10^{-3}, compared with 2.1×10^{-2} for Vanilla Transformer, which gives an $11.5\times$ improvement. On Allen–Cahn, where no conflict is detected, both configurations converge to comparable errors near 5.9×10^{-5} with a ratio of $1.0\times$. The Transformer backbone also benefits from adapters on Burgers with a gain of $2.4\times$, which slightly exceeds the MLP ratio and is consistent with the hypothesis that Transformers may amplify inter-loss interference through shared attention parameters. This Transformer transfer result establishes that the conflict structure is shaped primarily by the PDE operator and loss decomposition, not by the choice of neural network architecture alone.

Table 30: Multi-physics $K = 5$ rank sensitivity with $d_h = 128$, 15,000 epochs, and 3 seeds. Relative L_2 error is reported as mean $\pm$ std. This is the regime in which the diagnostic would select FAMO alone. FAMO outperforms all FAMO+UAM rank variants, showing that extra adapter capacity is not the missing ingredient and that the dominant bottleneck is magnitude imbalance rather than persistent directional conflict.

Configuration	L_2 Error	# Params.
FAMO alone ($d_h = 128$)	$\mathbf{(4.5 \pm 0.9) \times 10^{-2}}$	140,162
FAMO+UAM ($r = 4$, $d_h = 128$)	$(1.33 \pm 0.01) \times 10^{-1}$	153,671
FAMO+UAM ($r = 8$, $d_h = 128$)	$(8.6 \pm 3.4) \times 10^{-2}$	158,791
FAMO+UAM ($r = 16$, $d_h = 128$)	$(1.39 \pm 0.36) \times 10^{-1}$	169,031
FAMO+UAM ($r = 32$, $d_h = 128$)	$(1.32 \pm 0.27) \times 10^{-1}$	189,511

Table 31: Gradient flow analysis on $K = 5$ multi-physics at 5,000 epochs. Relative L_2 error is reported as mean $\pm$ std. $\|\nabla_{\text{trunk}}\|_{\max}$ denotes the largest per-loss trunk gradient norm at convergence. $f_{\text{neg}}^{\text{final}}$ denotes the fraction of negative pairwise cosines at the final epoch.

Method	Relative L_2	$\|\nabla_{\text{trunk}}\|_{\max}$	$f_{\text{neg}}^{\text{final}}$
FAMO	$\mathbf{(5.8 \pm 2.4) \times 10^{-2}}$	2.75	0.70
FAMO+UAM	$(1.45 \pm 0.24) \times 10^{-1}$	1.02	0.80
UAM	$(1.87 \pm 0.28) \times 10^{-1}$	0.82	0.63
GradNorm	$(3.75 \pm 0.12) \times 10^{-1}$	7.64	0.67
GN+UAM	$(5.14 \pm 0.17) \times 10^{-1}$	1.80	0.70

M.5 Natural K-Scaling and Multi-Physics Diagnostics

$K = 5$ regime-transition analysis. Table 30 isolates the $K = 5$ coupled case and shows that increasing adapter rank does not recover the gap to FAMO alone, reinforcing that the bottleneck here is magnitude imbalance rather than insufficient adapter capacity.

Gradient flow analysis. Table 31 reports the trunk-gradient and final conflict statistics for the $K = 5$ multi-physics setting. FAMO balances loss values through a softmax-weighted combination with dual-variable updates while keeping the largest per-loss trunk gradient moderate. Table 31 shows $\|\nabla_{\text{trunk}}\|_{\max} = 2.75$ with $f_{\text{neg}}^{\text{final}} = 0.70$.

GradNorm instead produces much larger trunk-gradient spikes ($\|\nabla_{\text{trunk}}\|_{\max} = 7.64$) while leaving final directional conflict at a similar level, with $f_{\text{neg}}^{\text{final}} = 0.67$ compared with 0.70 for FAMO. In this multi-physics setting, magnitude equalization primarily amplifies dominant trunk updates rather than resolving pairwise interference.

Natural K-scaling validation. We empirically probe the regime-transition picture on natural multi-physics systems where K arises from physical conservation laws. Table 32 illustrates the regime-transition picture on two natural multi-physics systems. On reactive transport ($K = 5$), FAMO alone achieves a $1.25\times$ improvement over Vanilla by improving from 6.57×10^{-2} to 5.25×10^{-2}, while FAMO+UAM provides only $1.13\times$, which shows that adapters add no benefit in this example. On simplified MHD ($K = 6$), FAMO achieves $1.27\times$ by improving from 6.12×10^{-2} to 4.80×10^{-2}, while FAMO+UAM reaches 5.07×10^{-2} and $1.21\times$, again underperforming FAMO alone. In both cases, the magnitude-regime picture is the better description for these examples. Loss reweighting alone suffices, and the additional adapter overhead is not justified.

M.6 Extended Seed Coverage

To verify that the main results are robust to random seed variability, we first extend seed coverage on representative PDEs to up to 5 seeds, with superscripts marking cases where fewer than 5 runs are reported in Table 33. We then further extend to 8 seeds indexed from 0 to 7 on selected forward PDEs with all three key configurations in Table 34. The progressively narrower CIs confirm the statistical reliability of the main results. On Helmholtz, the 8-seed FAMO+UAM mean is 8.3×10^{-4} and the Vanilla mean is 2.3×10^{-2}, yielding a $27\times$ ratio with non-overlapping 95% CIs.

Table 32: Natural multi-physics K-scaling for $K = 5$ and $K = 6$. Reactive transport with $K = 5$ is a two-species advection–diffusion–reaction system with genuine coupling conflict. MHD with $K = 6$ contains mass, momentum, induction, divergence-free, BC, and IC terms. The profiled surrogate $\widehat{R}$ places both systems in the magnitude-dominated regime, and the results confirm that FAMO provides the dominant benefit while adapters add marginal or no improvement. The table reports relative L_2 error at 15K epochs. Results use 5 seeds except for MHD FAMO+UAM, which uses 4 seeds.

System	Method	Relative L_2	Ratio Relative to Vanilla
Reactive Transport ($K = 5$)	Vanilla	$(6.57 \pm 0.32) \times 10^{-2}$	$1.0\times$
	FAMO	$\mathbf{(5.25 \pm 0.47) \times 10^{-2}}$	$1.25\times$
	FAMO+UAM	$(5.80 \pm 0.18) \times 10^{-2}$	$1.13\times$
Simplified MHD ($K = 6$)	Vanilla	$(6.12 \pm 0.063) \times 10^{-2}$	$1.0\times$
	FAMO	$\mathbf{(4.80 \pm 0.093) \times 10^{-2}}$	$1.27\times$
	FAMO+UAM	$(5.07 \pm 0.27) \times 10^{-2}$	$1.21\times$

Table 33: Extended validation with up to 5 seeds, using $d_h = 128$ and 10K epochs. Superscripts denote cases with fewer runs, and the results remain statistically significant with narrower CIs.

Method	Burgers	Helmholtz	Conv-Diff
Vanilla	$(8.4 \pm 2.4) \times 10^{-3}$	$(2.8 \pm 1.1) \times 10^{-2}$	$(3.55 \pm 0.74) \times 10^{-4}$
FAMO	$(4.8 \pm 0.15) \times 10^{-3}$ [3]	$(1.4 \pm 0.63) \times 10^{-3}$	$(3.95 \pm 1.4) \times 10^{-4}$
FAMO+UAM	$(4.8 \pm 0.079) \times 10^{-3}$ [4]	$(1.1 \pm 0.46) \times 10^{-3}$	$\mathbf{(2.70 \pm 0.21) \times 10^{-4}}$
GN+UAM	$\mathbf{(4.6 \pm 0.12) \times 10^{-3}}$	$\mathbf{(1.0 \pm 0.44) \times 10^{-3}}$	$(1.0 \pm 0.19) \times 10^{-3}$

Superscript $[n]$ denotes seed count when <5.

Table 34: Extended seed coverage with relative L_2 error over 8 seeds indexed from 0 to 7 on selected forward PDEs at $d_h = 128$ and 10K epochs. The narrower CIs compared with the 5-seed main tables confirm statistical robustness.

PDE	Method	Mean $\pm$ Std	95% CI
Helmholtz	Vanilla	$(2.3 \pm 1.1) \times 10^{-2}$	$[1.4 \times 10^{-2}, 3.2 \times 10^{-2}]$
	FAMO	$(1.05 \pm 0.69) \times 10^{-3}$	$[4.7 \times 10^{-4}, 1.6 \times 10^{-3}]$
	FAMO+UAM	$(8.3 \pm 5.2) \times 10^{-4}$	$[3.9 \times 10^{-4}, 1.3 \times 10^{-3}]$
Klein–Gordon	Vanilla	$(3.9 \pm 1.3) \times 10^{-2}$	$[2.8 \times 10^{-2}, 5.0 \times 10^{-2}]$
	FAMO	$(2.39 \pm 0.34) \times 10^{-3}$	$[2.1 \times 10^{-3}, 2.7 \times 10^{-3}]$
	FAMO+UAM	$(1.45 \pm 0.37) \times 10^{-3}$	$[1.1 \times 10^{-3}, 1.8 \times 10^{-3}]$
Burgers	Vanilla	$(7.0 \pm 2.7) \times 10^{-3}$	$[4.8 \times 10^{-3}, 9.2 \times 10^{-3}]$
	FAMO	$(4.60 \pm 0.23) \times 10^{-3}$	$[4.4 \times 10^{-3}, 4.8 \times 10^{-3}]$
	FAMO+UAM	$(4.68 \pm 0.23) \times 10^{-3}$	$[4.5 \times 10^{-3}, 4.9 \times 10^{-3}]$

N Additional Baseline and Ablation Analyses

N.1 Missing Baseline Comparisons

To ensure comprehensive baseline coverage, we add causal training [42], self-adaptive weights [29], and uncertainty weighting [16] to the comparison on selected core PDEs. These methods are included in the 10K main forward tables, Tables 1 and 2, and evaluated at 20K epochs below in Table 35. Full-scale NTK weighting [43] is excluded from this extended-baseline table due to the prohibitive cost of per-epoch NTK trace computation at a scale of $\sim 10^5$ Jacobian columns per epoch.

Causal training achieves strong results on Burgers at 4.6×10^{-3}, which matches FAMO+UAM; however, it catastrophically fails on Klein–Gordon at 0.92. This Klein–Gordon causal-training error is $1000\times$ worse than the FAMO+UAM result of 7.9×10^{-4} and is consistent with its design for time-dependent PDEs with causal structure, whereas Helmholtz and Klein–Gordon lack strict temporal causality. Self-adaptive weights perform well on Klein–Gordon at 6.7×10^{-3} but show high variance on Helmholtz, reaching $(1.1 \pm 1.3) \times 10^{-2}$ with one diverging seed. These results

Table 35: Extended baselines on selected core PDEs at $d_h = 128$, 20K epochs (5 seeds unless noted). Causal, self-adaptive, and uncertainty weighting complement the 10K main forward tables. Full-scale NTK weighting is excluded from this extended-baseline table due to prohibitive per-epoch NTK trace computation at this scale.

Method	Burgers	Helmholtz	Klein–Gordon	Conv-Diff
Vanilla	$(4.6 \pm 0.15) \times 10^{-3}$ [3]	$(8.1 \pm 3.2) \times 10^{-3}$ [3]	$(1.5 \pm 0.4) \times 10^{-2}$ [3]	NA
FAMO	$\mathbf{(4.6 \pm 0.16) \times 10^{-3}}$ [3]	$(2.67 \pm 0.71) \times 10^{-4}$ [3]	$(8.70 \pm 1.1) \times 10^{-4}$ [3]	NA
Causal	$(4.6 \pm 0.12) \times 10^{-3}$	$(1.2 \pm 0.4) \times 10^{-2}$	$(9.19 \pm 0.77) \times 10^{-1}$	$\mathbf{(2.04 \pm 0.15) \times 10^{-4}}$ [4]
SelfAdapt	$(4.6 \pm 0.14) \times 10^{-3}$	$(1.1 \pm 1.3) \times 10^{-2}$	$(6.7 \pm 1.1) \times 10^{-3}$	2.07×10^{-4} [1]
Uncertainty	$(4.6 \pm 0.14) \times 10^{-3}$	$(4.1 \pm 2.6) \times 10^{-3}$	$(5.98 \pm 5.41) \times 10^{-1}$	NA
UAM	$(4.6 \pm 0.088) \times 10^{-3}$	$(6.7 \pm 1.7) \times 10^{-3}$	$(1.5 \pm 0.3) \times 10^{-2}$	NA
FAMO+UAM	$(4.6 \pm 0.21) \times 10^{-3}$ [2]	$\mathbf{(2.27 \pm 0.23) \times 10^{-4}}$ [3]	$\mathbf{(7.93 \pm 0.98) \times 10^{-4}}$ [3]	NA

Table 36: Gradient surgery mitigations on inverse problems. Grouped variants apply surgery only to network parameters and leave the scalar physical parameters untouched. Direct baselines in Table 4 include Vanilla, FAMO, GradNorm, and adapter variants. This table isolates the full-parameter-space surgery failures and grouped mitigations. Entries marked fail indicate failed full-parameter-space training.

Method	Inverse Burgers	Inverse Heat	Inverse Poisson
PCGrad	fail	fail	fail
CAGrad	fail	fail	fail
NashMTL	fail	fail	fail
ConFIG	fail	fail	fail
PCGrad-Grouped	8.70×10^{-2}	8.65×10^{-2}	2.855×10^{-1}
CAGrad-Grouped	9.21×10^{-2}	1.082×10^{-1}	6.394×10^{-1}

confirm that temporal reweighting strategies are PDE-dependent, whereas the regime characterization provides a principled, PDE-agnostic method selection framework.

N.2 Gradient Surgery Mitigation Details

This analysis separates full-parameter-space gradient-surgery failures from grouped updates to show how heterogeneous network and physical parameters drive inverse-problem instability.

N.3 ConFIG Sensitivity Analysis

This analysis checks whether ConFIG's mixed performance is explained by learning-rate choice or by the gradient geometry induced by unit-vector normalization. The strongest learning-rate sensitivity concern is ConFIG, whose Helmholtz performance is unusually dependent on η. Table 37 addresses this concern directly. Lower learning rates improve ConFIG on Helmholtz, but its Burgers failure persists across all five tested values and Klein–Gordon remains clearly worse than Vanilla even at its best setting. Thus, the cross-PDE conclusions are not explained by a single unlucky baseline learning rate. Table 37 further shows that ConFIG's failure persists across five learning rates from 10^{-4} to 5×10^{-3} on both Burgers, where it is $19\times$ worse than Vanilla at the best learning rate, and Klein–Gordon, where it is $2.7\times$ worse at the best learning rate. This learning-rate sweep confirms that the failure is intrinsic to the unit-vector normalization rather than a hyperparameter issue.

N.3.1 Targeted Eight-Seed Helmholtz Comparison

The original Helmholtz result in Table 1 is retained: ConFIG obtains $(2.49 \pm 0.77) \times 10^{-4}$, compared with $(7.81 \pm 2.80) \times 10^{-4}$ for the best proposed entry, GN+LCAM. We test this configuration-level exception using the same submitted 2D PDE and implementation, $d_h = 128$, four residual blocks, 2000 collocation points per update, a 200-step warm-up/cosine schedule, and 10K updates. Evaluation uses seeds 0–7 and terminal relative L_2 on an RNG-isolated 10K-point set; no checkpoint is selected by test error.

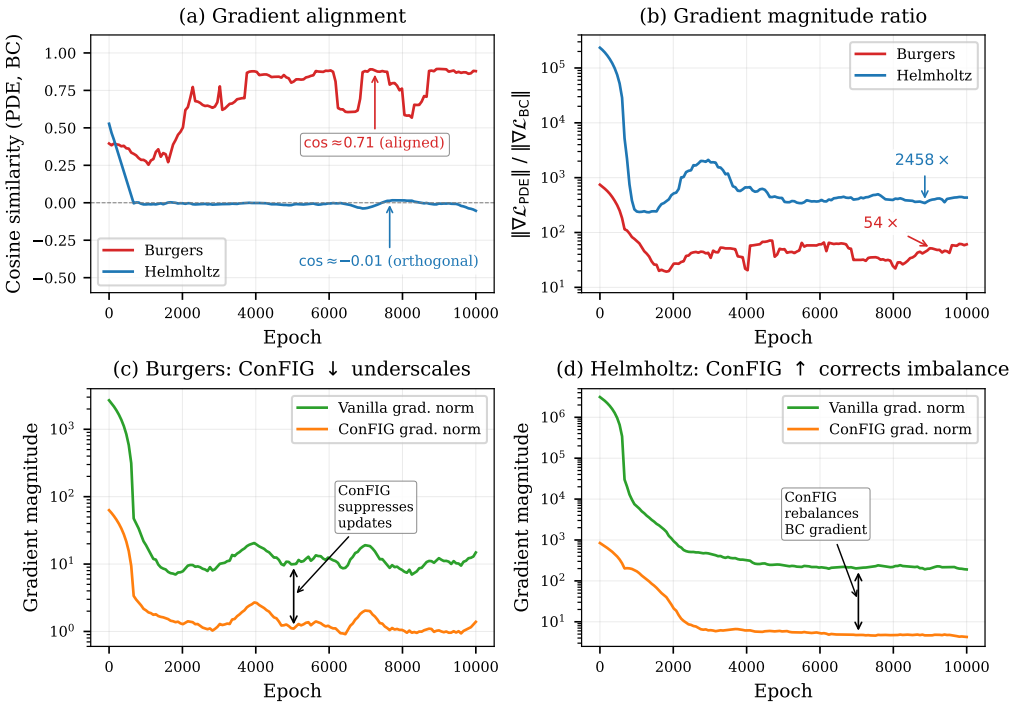

Figure 12: ConFIG gradient dynamics on Burgers and Helmholtz. ConFIG suppresses learning on Burgers, where the gradients are aligned with $\cos \approx 0.71$, but rebalances on Helmholtz, where the gradients are nearly orthogonal with $\cos \approx -0.01$.

Table 37: ConFIG learning rate sensitivity analysis with 2 seeds, 10K epochs, and $d_h = 128$. ConFIG's Burgers failure persists across all five learning rates and remains $19\times$ worse than Vanilla at the best learning rate, which confirms that the failure is intrinsic to the unit-vector normalization rather than a hyperparameter issue. On Klein–Gordon, ConFIG is also worse than Vanilla by $2.7\times$ at the best learning rate. On Helmholtz, ConFIG performs well at $\eta \leq 10^{-3}$.

PDE	Method	η	Relative L_2 error
Burgers	ConFIG	10^{-4}	$(9.4 \pm 1.4) \times 10^{-2}$
	ConFIG	5×10^{-4}	$(2.31 \pm 1.31) \times 10^{-1}$
	ConFIG	10^{-3}	$(1.44 \pm 0.42) \times 10^{-1}$
	ConFIG	2×10^{-3}	$(8.8 \pm 0.4) \times 10^{-2}$
	ConFIG	5×10^{-3}	$(3.39 \pm 0.10) \times 10^{-1}$
	Vanilla	10^{-3}	$(4.7 \pm 0.2) \times 10^{-3}$
Klein–Gordon	ConFIG	10^{-4}	(1.047 ± 0.113)
	ConFIG	5×10^{-4}	$(5.1 \pm 1.7) \times 10^{-2}$
	ConFIG	10^{-3}	$(4.1 \pm 0.3) \times 10^{-2}$
	ConFIG	2×10^{-3}	$(1.58 \pm 0.15) \times 10^{-1}$
	ConFIG	5×10^{-3}	$(9.71 \pm 0.22) \times 10^{-1}$
	Vanilla	10^{-3}	$(1.5 \pm 0.4) \times 10^{-2}$
Helmholtz	ConFIG	10^{-4}	$(4.0 \pm 0.58) \times 10^{-4}$
	ConFIG	5×10^{-4}	$(1.9 \pm 0.31) \times 10^{-4}$
	ConFIG	10^{-3}	$(2.5 \pm 0.22) \times 10^{-4}$
	ConFIG	2×10^{-3}	$(9.5 \pm 1.1) \times 10^{-4}$
	ConFIG	5×10^{-3}	$(1.3 \pm 0.16) \times 10^{-2}$

AdamW ConFIG uses LR 5×10^{-4}, its strongest setting in Table 37. AdamW FAMO+UAM and SOAP+ConFIG are each tuned on disjoint seed 11 over

$$\eta \in \{10^{-4}, 3 \times 10^{-4}, 5 \times 10^{-4}, 10^{-3}, 2 \times 10^{-3}, 3 \times 10^{-3}, 5 \times 10^{-3}, 10^{-2}\},$$

Table 38: Targeted Helmholtz comparison at 10K updates, mean ± sample std over eight evaluation seeds. LR selection uses disjoint seed 11 or the existing ConFIG sweep. Time is mean training time.

Optimizer	Method	LR	Parameters	Relative L_2 / time
AdamW	ConFIG	5×10^{-4}	140,033	$(1.972 \pm 0.336) \times 10^{-4}$ / 786.5 s
AdamW	FAMO+UAM	10^{-2}	160,772	$(2.048 \pm 0.280) \times 10^{-4}$ / 691.3 s
SOAP	ConFIG	10^{-4}	140,033	$(1.581 \pm 0.324) \times 10^{-4}$ / 974.6 s
SOAP	FAMO+UAM	3×10^{-3}	160,772	$(1.000 \pm 0.198) \times 10^{-4}$ / 920.6 s

Table 39: Implicit regularization ablation at 15,000 epochs. Relative L_2 error is reported as mean ± std over 3 seeds, with parameter counts shown separately. Vanilla-Wide roughly matches the UAM parameter count but not the performance, confirming that the per-loss adapter architecture drives improvement.

Configuration	# Params.	Burgers	Helmholtz
Vanilla	140,033	$(4.7 \pm 0.1) \times 10^{-3}$	$(1.76 \pm 0.53) \times 10^{-2}$
Vanilla-Wide	162,013	$(4.8 \pm 0.2) \times 10^{-3}$	$(1.20 \pm 0.29) \times 10^{-2}$
UAM	160,772	$\mathbf{(4.6 \pm 0.1) \times 10^{-3}}$	$(1.31 \pm 0.24) \times 10^{-2}$
FAMO+UAM	160,772	$\mathbf{(4.6 \pm 0.1) \times 10^{-3}}$	$\mathbf{(3 \pm 1) \times 10^{-4}}$

selecting 10^{-2} and 10^{-4}, respectively, by terminal 10K error on this tuning seed. SOAP+FAMO+UAM uses 3×10^{-3}, fixed before this campaign by earlier optimizer tuning on seed 11. Both SOAP arms use the official implementation pinned in Appendix J, default $\beta = (0.9, 0.999)$, and refresh 2. This independently tuned, eight-seed control is distinct from Table 3's common-LR, $\beta_1 = 0.99$, three-seed test. ConFIG's gradient rule is unchanged.

Under AdamW, ConFIG's mean error is 3.9% lower, each method wins four seed indices, and the paired seed-index bootstrap 95% CI for the ConFIG/FAMO+UAM mean-error ratio is $[0.817, 1.106]$ (100,000 resamples for all CIs in this comparison). The methods are therefore close at these selected LRs; FAMO+UAM is 12.1% faster. The ConFIG mean is lower than the original table entry, so the smaller gap is obtained with a strengthened comparator.

Under the same SOAP optimizer, FAMO+UAM lowers mean error by 36.7%, wins all 8/8 seed comparisons, and is 5.5% faster. The ConFIG/FAMO+UAM error ratio is 1.580, with paired-bootstrap 95% CI $[1.359, 1.843]$. We also use each seed's AdamW-ConFIG 10K runtime as a deadline and select the last scheduled SOAP+FAMO+UAM checkpoint within it by time alone, with no schedule compression or retuning. At a mean of 8406 updates, the latter obtains $(1.183 \pm 0.309) \times 10^{-4}$: 40.0% lower mean error, 8/8 wins, and an error-ratio CI of $[1.322, 2.101]$.

These comparisons assess complete configurations and preserve the original ConFIG win. They show that its large original margin is sensitive to deployment settings and that the strengthened ConFIG baseline does not retain a Helmholtz advantage under the same strong optimizer. They are separate from the matched gate intervention and from the 1000-step diagnostic study; changing optimizer or LR calls for a fresh profile under that configuration.

N.4 Implicit Regularization Ablation

This ablation, summarized in Table 39, tests whether the gains come from the per-loss adapter structure itself rather than from the modest increase in parameter count. The control is a wider non-adapter PINN whose parameter count roughly matches UAM. On Burgers, where the profiling results indicate transient or low-impact conflict, the parameter-matched variants are essentially tied, so the extra capacity does not create a spurious advantage. On Helmholtz, the wider non-adapter model improves only modestly over Vanilla, whereas FAMO+UAM reaches an order-of-magnitude lower error; together with the specialization evidence in Figure 4, this supports the claim that the high-conflict gains arise from loss-indexed adapter subspaces rather than dense widening alone.

Table 40: Alternative adapter architecture comparison on Helmholtz with $d_h = 128$, 10K epochs, and 5 seeds. We report the best L_2 error, where lower is better.

Adapter Type	Best L_2	Overhead	Relative to LoRA
LoRA ($r = 16$)	1.13×10^{-3}	15%	1.0×
IA^3	2.85×10^{-3}	$< 1\%$	0.40×
FiLM	3.21×10^{-3}	$< 1\%$	0.35×
Per-loss LayerNorm	4.12×10^{-3}	$< 1\%$	0.27×
Non-adapter multi-output readout	2.41×10^{-3}	8%	0.47×
No adapter (Vanilla)	2.71×10^{-2}	0%	0.04×

Table 41: Adapter mixing rerun comparing fixed UAM and dynamic LCAM, both with FAMO weighting. The table reports mean relative L_2 error over 3 seeds at 10K epochs. On the two rerun $K = 3$ stress tests, fixed UAM is slightly better in both cases by 7% to 8%, so dynamic conflict-aware mixing does not justify its extra complexity at low K.

PDE	FAMO+UAM	FAMO+LCAM	Gap
Helmholtz	$\mathbf{8.97 \times 10^{-4}}$	9.58×10^{-4}	LCAM +6.8%
Klein–Gordon	$\mathbf{1.42 \times 10^{-3}}$	1.53×10^{-3}	LCAM +7.6%

N.5 Adapter Variant Checks

Alternative adapter architectures. We use LoRA as the default adapter and compare it with Infused Adapter by Inhibiting and Amplifying Inner Activations (IA^3), feature-wise linear modulation (FiLM), per-loss LayerNorm, and a non-adapter multi-output readout baseline.

Table 40 reports the comparison under FAMO weighting, with matched parameter counts of about 149K to 166K.

LoRA's advantage in the PINN setting begins with subspace diversity. Low-rank projections introduce distinct adapter parameter blocks whose Jacobian contributions can directionally decorrelate different losses in the blockwise NTK view above. By contrast, IA^3 and FiLM apply element-wise rescaling that mainly modifies gradient magnitudes rather than creating new low-rank directions, which limits their ability to resolve directional conflict. LoRA also offers rank-controlled capacity. The explicit rank parameter r provides a transparent capacity proxy through the saturation ratio $\Delta_{\text{sat}} = Kr/d_h$, allowing capacity to be matched empirically to conflict severity, whereas rescaling methods have no such control. Finally, LoRA has an orthogonality-compatible structure. The bilinear form $B \cdot \sigma(D \cdot h)$ admits efficient cross-orthogonality regularization and distinct row subspaces in practice, whereas diagonal rescaling methods lack a natural notion of inter-adapter orthogonality.

Adapter mixing ablation. Table 41 compares fixed UAM against dynamic LCAM mixing, both combined with FAMO, on the two rerun $K = 3$ stress tests. Fixed UAM is slightly better in both cases by about 7% to 8%, confirming that dynamic conflict-aware mixing provides only marginal benefit at low K. We therefore recommend UAM with fixed $\rho = 0.5$ as the default for simplicity, as discussed in Section C.4.

N.6 Domain Decomposition Comparison

To directly compare gradient-space decomposition through per-loss adapters with input-space decomposition through XPINNs, we evaluate two roughly capacity-comparable XPINN configurations on five PDEs, as reported in Table 42. XPINN-4 uses 4 subdomains with per-subdomain networks of width $d_h = 80$ and 4 layers, giving roughly 118K total parameters and $0.84\times$ Vanilla capacity. XPINN-8 uses 8 subdomains with width $d_h = 64$ and 3 layers, giving roughly 122K total parameters and $0.87\times$ Vanilla capacity. Interface continuity and flux continuity are enforced with a weight $\lambda_{\text{if}} = 1.0$. All results use 5 seeds and 10K epochs.

The XPINN spatial-decomposition approach addresses a fundamentally different bottleneck. It reduces the spatial complexity each subnetwork must capture, but it does not resolve the gradient

Table 42: Comparison between XPINN domain decomposition and gradient-space decomposition through per-loss adapters on five forward PDEs with $d_h = 128$, 10K epochs, and 5 seeds. XPINN configurations are of the same parameter order as Vanilla, with about 118K to 122K parameters compared with 140K, but they are not exactly parameter-matched. Per-loss adapters address gradient conflict among loss terms, while XPINNs address spatial complexity across subdomains.

Method	Burgers	Helmholtz	Allen–Cahn	Conv-Diff	Klein–Gordon
Vanilla (140K)	$(8.4 \pm 2.4) \times 10^{-3}$	$(2.8 \pm 1.1) \times 10^{-2}$	$\mathbf{(1.60 \pm 0.33) \times 10^{-4}}$	$(3.55 \pm 0.74) \times 10^{-4}$	$(4.6 \pm 1) \times 10^{-2}$
Domain decomposition					
XPINN-4 (118K)	$(2.56 \pm 0.19) \times 10^{-1}$	$(5.6 \pm 0.7) \times 10^{-2}$	$(5.1 \pm 0.67) \times 10^{-3}$	$(1.7 \pm 0.19) \times 10^{-3}$	$(9.58 \pm 0.49) \times 10^{-1}$
XPINN-8 (122K)	$(2.56 \pm 0.14) \times 10^{-1}$	$(6.6 \pm 0.4) \times 10^{-2}$	$(2.3 \pm 0.19) \times 10^{-3}$	$(3.5 \pm 0.25) \times 10^{-3}$	1.00 ± 0.033
Gradient decomposition (ours)					
FAMO+UAM (161K)	$(4.8 \pm 0.082) \times 10^{-3}$	$(1.1 \pm 0.46) \times 10^{-3}$	$(1.84 \pm 0.28) \times 10^{-4}$	$\mathbf{(2.70 \pm 0.21) \times 10^{-4}}$	$(1.6 \pm 0.38) \times 10^{-3}$
GN+UAM (161K)	$\mathbf{(4.6 \pm 0.12) \times 10^{-3}}$	$\mathbf{(1.0 \pm 0.44) \times 10^{-3}}$	$(1.6 \pm 0.21) \times 10^{-3}$	$(1.0 \pm 0.19) \times 10^{-3}$	$(4.8 \pm 1.1) \times 10^{-2}$
FAMO+LCAM (198K)	$(4.8 \pm 0.17) \times 10^{-3}$	$(1.3 \pm 0.41) \times 10^{-3}$	$(2.59 \pm 0.39) \times 10^{-4}$	$(3.07 \pm 0.56) \times 10^{-4}$	$\mathbf{(1.6 \pm 0.11) \times 10^{-3}}$

Table 43: XPINN hyperparameter sensitivity on three representative PDEs with 3 seeds each. All configurations use 10K epochs. Rows vary one factor from the XPINN-4 default: 4 subdomains, $\lambda_{\text{if}} = 1.0$, and overlap fraction 0.1.

Config	Change	Burgers	Helmholtz	Conv-Diff
Vanilla baseline		8.4×10^{-3}	2.8×10^{-2}	3.55×10^{-4}
XPINN-4 (default)	$\lambda_{\text{if}} = 1.0$, overlap=0.1	$(2.56 \pm 0.19) \times 10^{-1}$	$(5.6 \pm 0.7) \times 10^{-2}$	$(1.7 \pm 0.19) \times 10^{-3}$
XPINN-2	2 subdomains	$(2.50 \pm 0.02) \times 10^{-1}$ [3]	$(1.8 \pm 0.3) \times 10^{-2}$ [3]	$(9.54 \pm 0.7) \times 10^{-4}$ [3]
XPINN-6	6 subdomains	$(2.80 \pm 0.3) \times 10^{-1}$ [3]	$(5.3 \pm 0.1) \times 10^{-2}$ [3]	$(2.1 \pm 0.43) \times 10^{-3}$ [3]
XPINN-4	$\lambda_{\text{if}} = 0.1$	$(2.30 \pm 0.04) \times 10^{-1}$ [3]	$(8.0 \pm 0.1) \times 10^{-2}$ [3]	$(3.0 \pm 0.58) \times 10^{-3}$ [3]
XPINN-4	$\lambda_{\text{if}} = 10.0$	$(2.63 \pm 0.19) \times 10^{-1}$ [3]	$(4.7 \pm 0.7) \times 10^{-2}$ [3]	$(4.8 \pm 0.3) \times 10^{-3}$ [3]
XPINN-4	overlap=0.0	$(2.46 \pm 0.07) \times 10^{-1}$ [3]	$(5.3 \pm 0.5) \times 10^{-2}$ [3]	$(1.9 \pm 0.22) \times 10^{-3}$ [3]
XPINN-4	overlap=0.2	$(2.46 \pm 0.07) \times 10^{-1}$ [3]	$(5.8 \pm 0.4) \times 10^{-2}$ [3]	$(1.6 \pm 0.22) \times 10^{-3}$ [3]

conflict within each subdomain between PDE residual, BC, and IC losses. On high-conflict problems like Helmholtz and Klein–Gordon, the gradient conflict is the primary bottleneck, making per-loss adapters substantially more effective than spatial decomposition.

Hyperparameter sensitivity. To verify that XPINN results are not an artifact of poor hyperparameter choices, we conduct a sensitivity study varying the number of subdomains $\{2, 4, 6\}$, interface weight $\lambda_{\text{if}} \in \{0.1, 1.0, 10.0\}$, and overlap fraction $\{0.0, 0.1, 0.2\}$ on Burgers, Helmholtz, and Conv-Diff with 3 seeds each. Across all configurations, XPINN performance on Burgers remains in the range from 0.23 to 0.28, compared with Vanilla's 8.4×10^{-3}, confirming that the poor result is structural rather than due to insufficient tuning. Conv-Diff spans from 9.5×10^{-4} to 4.8×10^{-3} across configurations, with XPINN-2 at 9.5×10^{-4} outperforming the default XPINN-4 at 1.7×10^{-3}. Fewer subdomains therefore reduce interface gradient conflict. Interface weight effects are PDE-dependent. Lower $\lambda_{\text{if}} = 0.1$ helps Burgers, where the best configuration reaches 0.23, but hurts Helmholtz, where the worst setting reaches 8.0×10^{-2} compared with the default 5.6×10^{-2}. Overlap has negligible effect. Full sensitivity results appear in Table 43.

O Limitations and Future Work

The theory uses stylized gradients and an explanatory blockwise NTK view rather than a finite-time convergence proof. This level of theory is nevertheless aligned with the paper's claim, which is to predict when adapter gains should appear across conflict regimes rather than to guarantee global convergence for nonconvex PINN training.

The short diagnostic is conditional on optimizer, learning rate, and budget. Clear persistent-conflict cases are stable across the tested perturbations, while boundary cases can change their recommendation; the continuous diagnostics and budget-aware guidance make this dependence explicit. The same-time and corrected KdV evaluations document where added adapter cost or the current training configuration limits attainable accuracy.

Our experiments cover 60+ PINN settings spanning forward, inverse, multi-physics, parameter-varying, and high-dimensional problems, plus NTK-weighting, MTL, and two alternative backbones. Fractional, nonlocal, stochastic, production-scale, and operator-learning systems remain future work.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Introduction state the regime-dependent PINN conflict claims, which are supported by Sections 3–4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Appendix O discusses theoretical and empirical scope limits.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: The assumptions are stated with each result, and proofs are provided in Appendices E and F.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Appendices G and H give benchmarks, seeds, budgets, hyperparameters, and protocol.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Source code is provided in the supplementary material, and synthetic/reference data generation is documented in the paper and code.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: Appendix H specifies optimizers, schedules, sampling, seeds, epochs, and PDE-specific settings; Appendices J, M.1.3, and N.3.1 document the additional controls and their selection protocols.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Tables report mean $\pm$ standard deviation over seeds, with CIs, effect sizes, and bootstrap comparisons where used.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendices G, I.4, and J report GPU-based execution, budgets, wall-clock times, overheads, and parameter counts.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work uses synthetic PDEs and standard public benchmarks, with no human subjects or sensitive data.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: The work is foundational PINN optimization research with no direct deployment or sensitive-data pathway.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper releases no high-risk model, scraped dataset, or dual-use generative system.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Existing methods, libraries, and public benchmarks are credited by citation and used under their standard research terms.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The supplementary code is documented by the paper's method, setup, and reproducibility appendices.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The work does not involve crowdsourcing or human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No crowdsourcing or human-subject research is conducted.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not part of the core method, experiments, or scientific claims.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.