PHYSICS-INFORMED NEURAL PLASTICITY: PDE SOLVERS THAT RESHAPE THEMSELVES

Anonymous authors
Paper under double-blind review

ABSTRACT

Physics-informed neural PDE solvers adapt their parameters to satisfy governing equations, yet their *representational structure typically remains fixed throughout training*. This rigidity is poorly matched to PDE solutions with strongly heterogeneous complexity across space and space–time, leaving capacity insufficient where the physics is difficult and redundant where it is simple. We introduce *physics-informed neural plasticity*, a paradigm in which the representation itself reshapes during optimization in response to unresolved physics. We instantiate this principle with *Representation Capacity Adaptation for PDEs* (ReCAP), a Gaussian-localized solver that dynamically redistributes capacity through local enrichment, residual-directed splitting, gate-based pruning, and function-aware merging. ReCAP uses responsibility-weighted error indicators and the geometry of residual energy to determine where and how to refine. To limit the disturbance introduced by splitting, we introduce *quiet-child refinement*, which initializes new components by transporting the parent representation while controlling instantaneous functional perturbation. We further establish conditional a posteriori reliability and structural-stability guarantees linking localized physics residuals to solution error and stable refinement. Across five challenging 3D and 4D PDE benchmarks against 11 physics-informed solvers, ReCAP achieves the lowest relative L^2 error on every problem, reducing error by 10.7%–27.5% relative to the strongest competing result. These results suggest that physics-informed solvers need not merely learn their parameters—they can learn how their representational capacity should be organized.

1 INTRODUCTION

Physics-informed neural networks (PINNs) approximate solutions to partial differential equations (PDEs) by optimizing differentiable function representations against governing equations and available initial or boundary data. Their mesh-free formulation and unified treatment of forward and inverse problems make them attractive when conventional discretization is difficult or solution data are scarce (Raissi et al., 2019; Karniadakis et al., 2021). Yet accurate training remains challenging for high-frequency, multiscale, and strongly nonlinear solutions, where spectral bias and optimization difficulties can leave localized structures unresolved even when the overall loss appears small (Rahaman et al., 2019; Wang et al., 2021; Krishnapriyan et al., 2021). This challenge is especially pronounced when solution complexity varies across space or space–time: smooth regions may coexist with oscillatory waves, localized peaks, or rapidly evolving fronts that demand markedly different approximation resources.

Yet most existing approaches share a key limitation: *the parameters learn, but the representation itself largely stays fixed.* Optimization strategies, loss balancing, causal training, and residual-adaptive sampling can substantially improve PINN training (Wu et al., 2023; Daw et al., 2023; Wang et al., 2024a), but they generally operate within a representational budget chosen before training. Adaptive sampling changes *where* the PDE is enforced, but not *what the model can represent locally*. As a result, capacity may remain redundant in already-resolved regions while being insufficient where the physics is hardest to resolve.

We argue that physics-informed PDE solvers should instead exhibit *neural plasticity*: the ability to reshape their representational capacity during training in response to unresolved physics. Rather

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

than optimizing a fixed representation, a plastic solver jointly learns the solution and reorganizes the capacity used to express it. Localized neural representations provide a natural basis for such plasticity. FBPINNs decompose the domain among overlapping subnetworks (Moseley et al., 2023); PIG learns the locations and shapes of Gaussian features while fixing their number at initialization (Kang et al., 2025); and AB-PINNs introduce learnable RBF-defined subdomains with residual-driven addition of local subnetworks (Botvinick-Greenhouse et al., 2026). These developments motivate a broader research question: how can physics-informed solvers coordinate *where capacity is needed, what form it should take, and when it has become redundant*, while limiting disruption to the learned solution?

To realize this principle, we introduce **Representation Capacity Adaptation for PDEs (ReCAP)**, a Gaussian-localized physics-informed solver whose representation can enrich, split, merge, and prune itself during training. Each component carries a learned center and scale, a contribution gate, and a local polynomial approximation. Component responsibilities turn global physics errors into local adaptation signals, allowing ReCAP to *redistribute* capacity rather than simply increase it. Difficult regions receive higher local polynomial order or additional localized components, while weak or functionally redundant components are removed or consolidated. These operations implement a common principle: adapt the local approximation space to unresolved physics through p-like enrichment, h-like refinement, and coarsening. For such plasticity to be useful, structural changes must be both *targeted and stable*. ReCAP uses responsibility-weighted physics errors to identify unresolved components and a residual-weighted covariance to orient splitting along the dominant spread of unresolved residual energy. Merging is function-aware, requiring both effective overlap and agreement of local predictions rather than geometric proximity alone. Structural refinement can nevertheless perturb an already learned solution. We therefore introduce a *quiet-child* update: the trained parent is retained, its polynomial is translated into the child's shifted and rescaled coordinates, and the child is introduced through a small gate. This creates new representational capacity while controlling its immediate effect on the learned function.

We complement ReCAP with theoretical analysis connecting local adaptation signals to global solution accuracy and characterizing the stability of structural refinement. Across challenging three- and four-dimensional PDE benchmarks, our experiments further show that dynamically redistributing representational capacity can outperform both fixed-capacity and competing physics-informed solvers. Together, these results support neural plasticity as a principled approach to adapting *where and how* capacity is deployed during physics-informed learning. Our contributions are:

- We formulate *physics-informed neural plasticity*, where representational capacity becomes an adaptive variable of PDE optimization rather than a fixed design choice, and introduce ReCAP to jointly adapt local approximation order, component geometry, and component count.
- We develop a responsibility-driven adaptation mechanism that couples local p-enrichment with residual-directed h-like splitting. Splitting is oriented by the leading eigenvector of a responsibility- and residual-weighted covariance, directing new capacity along the dominant spread of unresolved physics.
- We introduce function-aware coarsening and *quiet-child* refinement: merging requires both effective overlap and agreement of local predictions, while polynomial coordinate translation and gated child insertion create new capacity with controlled instantaneous perturbation of the learned solution.
- We establish conditional a posteriori reliability connecting mass-weighted local physics indicators to solution error under PDE stability and partition-of-unity responsibilities. For normalized mixtures with a linear transport operator, we further derive a gate-dependent bound on the instantaneous loss increase caused by additive quiet-child refinement, clarifying how capacity can grow with controlled disruption.
- Across five 3D/4D PDE benchmarks and 11 competing physics-informed solvers, ReCAP achieves the lowest relative L^2 error on every benchmark.

2 Related Work

Physics-informed optimization and sampling. PINNs enforce differential equations through a training objective (Raissi et al., 2019; Karniadakis et al., 2021). Gradient imbalance, spectral bias, and difficult optimization landscapes motivate improved training schemes and architectures (Wang

et al., 2021; Rahaman et al., 2019; Krishnapriyan et al., 2021; Cho et al., 2023; Wang et al., 2023; 2024a). Residual-adaptive sampling and retain-resample-release sampling direct equation evaluations toward poorly resolved regions (Wu et al., 2023; Daw et al., 2023). These methods alter training signals or their optimization; ReCAP additionally changes the available local basis. Sampling and representation adaptation are complementary, and the experiments examine their cumulative effect.

Local representations and adaptive decomposition. Radial basis functions provide trainable localized approximations (Broomhead & Lowe, 1988). Domain-decomposed PINNs, including XPINNs and FBPINNs, organize approximation over local regions (Jagtap & Karniadakis, 2020; Moseley et al., 2023). PIG combines trainable Gaussian features with a lightweight neural network (Kang et al., 2025), while AB-PINNs learn local subdomains and support residual-driven addition of new ones (Botvinick-Greenhouse et al., 2026). ReCAP builds on this local-representation perspective by coordinating component insertion and removal with local polynomial-order changes. Its splitting rule uses residual geometry, its merge criterion checks local functional agreement, and its child initialization accounts explicitly for the coordinate dependence of polynomial coefficients.

Adaptive approximation spaces. Classical hp methods allocate degrees of freedom by changing element size and approximation order (Babuška & Suri, 1994); hp-VPINNs bring domain decomposition and hp refinement into variational physics-informed learning (Kharazmi et al., 2021). ReCAP uses analogous operations on learned Gaussian components rather than a fixed mesh. This analogy motivates the algorithm but does not transfer finite-element convergence guarantees to it. In particular, the residual scores used for adaptation are not local solution-error bounds: their reliability requires a stability estimate for the PDE and appropriate mass weighting, as made explicit in Section 3.4.

3 Methodology

ReCAP realizes *physics-informed neural plasticity* by adapting the approximation space together with its trainable parameters. At each stage, the structural state $\mathcal{S} = (M, p_1, \ldots, p_M)$ specifies the number of Gaussian components and their active polynomial orders; the continuous parameters Θ specify their locations, scales, gates, and local approximations. Training alternates between optimizing Θ at fixed $\mathcal{S}$ and updating the representation using localized physics errors. This separates two complementary tasks: fitting the current approximation space and changing where that space has capacity.

The adaptation rules act at different levels. Polynomial enrichment increases local expressivity without adding components; splitting introduces a more localized component; pruning and merging reduce weakly active or redundant components. The base objective $\mathcal{L}_{\text{base}}$ determines how the PDE and its conditions are enforced, whereas the adaptation scores determine which components are modified. These scores need not be additional loss terms. Figure 1 summarizes the loop. Appendix A details the component updates, and Appendix B contains the theoretical statements and proofs.

3.1 Representation and local error indicators

On a spatial or spatiotemporal domain $\Omega \subset \mathbb{R}^d$, consider $\mathcal{N}[u](z) = 0$ with initial or boundary conditions $\mathcal{B}[u] = g$. Here d counts all input coordinates, including time when present, and $r_\Theta = \mathcal{N}[u_\Theta]$ is the physics residual. For FM3D, $z = (t, x, y)$ and $r_\Theta = \partial_t u_\Theta + a\partial_x u_\Theta + b\partial_y u_\Theta$. The other benchmarks use their respective differential operators, given in Appendix C.

Each component $G_m = (\mu_m, \sigma_m, g_m, c_m, p_m)$ has a center, positive diagonal scale, gate logit, polynomial coefficients, and active degree. Fixed input normalization is absorbed into z in the representation formulas; residual evaluation must retain the corresponding chain-rule factors. Define the complete total-degree index set $\mathcal{A}_p = \{\alpha \in \mathbb{N}_0^d : |\alpha| \le p\}$ and

$$\xi_m = (z - \mu_m) \oslash \sigma_m, \qquad \ell_m(z) = \sum_{\alpha \in \mathcal{A}_{p_m}} c_{m,\alpha} \xi_m^\alpha, \qquad |\mathcal{A}_p| = \binom{d+p}{p}. \tag{1}$$

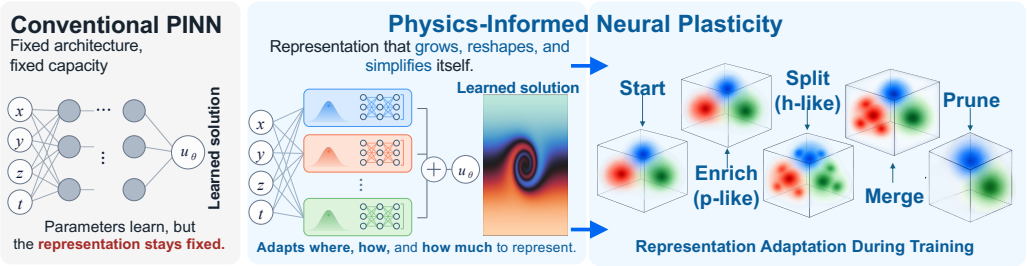


Figure 1: Unlike conventional PINNs with fixed representational capacity, ReCAP reshapes its representation during training through local enrichment, splitting, merging, and pruning, allocating capacity where the PDE requires it.

Here $\xi^\alpha = \prod_q \xi_q^{\alpha_q}$ and $\oslash$ denotes element-wise division. A local degree change therefore changes the number of active basis terms, independently of the number of Gaussians. The described three-dimensional implementation preallocates the complete basis through total degree three, including cross terms, and masks inactive degrees.

Let $\pi_m(z) \geq 0$ denote the effective responsibility used to assign evaluation points to component m, and write the prediction as $u_\Theta = \mathcal{A}_\Theta(\{\pi_m, \ell_m, g_m\}_{m=1}^M)$. The update rules use the full prediction and component responsibilities, together with local values or derivatives where required. When internal branches or channels are present, the adaptation statistics average the corresponding quantities across them. The analysis in Section 3.4 states separately when normalized responsibilities or a normalized gated-mixture output are required.

For residual points $\mathcal{C} = \{z_i\}_{i=1}^{N_c}$ and value observations $\mathcal{D}_\Gamma = \{(z_i^\Gamma, u_i^\Gamma)\}_{i=1}^{N_\Gamma}$, define

$$\eta_m^{\mathrm{res}} = \frac{\sum_i \pi_m(z_i)|r_\Theta(z_i)|^2}{\sum_i \pi_m(z_i) + \varepsilon}, \qquad \eta_m^{\mathrm{bdry}} = \frac{\sum_i \pi_m(z_i^\Gamma)|u_\Theta(z_i^\Gamma) - u_i^\Gamma|^2}{\sum_i \pi_m(z_i^\Gamma) + \varepsilon}. \tag{2}$$

The normalization separates the *intensity* of local PDE violation from the amount of responsibility mass assigned to a component: a small unresolved region need not be overshadowed by a broad component. The stabilizer $\varepsilon > 0$ attenuates scores with negligible support. These are empirical averages over the evaluation points, so their interpretation also depends on how those points are sampled. Initial and boundary observations are combined when initial conditions are imposed softly; exactly enforced conditions contribute no mismatch. The displayed data term applies to prescribed values; derivative conditions require their corresponding operator mismatch. To compare overlapping local approximations, form a k-nearest-neighbor graph of centers and evaluate

$$\begin{aligned} J_{mn} &= \frac{\sum_{i=1}^{N_j} \pi_m(z_i)\pi_n(z_i)\|\nabla \ell_m(z_i) - \nabla \ell_n(z_i)\|_2^2}{\sum_{i=1}^{N_j} \pi_m(z_i)\pi_n(z_i) + \varepsilon}, \\ \eta_m^{\mathrm{jump}} &= \frac{\sum_{n \in \mathcal{N}_k(m)} J_{mn}}{|\mathcal{N}_k(m)|}. \end{aligned} \tag{3}$$

The gradients are taken with respect to the same global input coordinates, so discrepancies compare the local functions in a common coordinate system. The responsibility product weights the comparison toward overlap, with ε attenuating pairs of negligible overlap mass. Despite the notation "jump," this is a disagreement score between overlapping models, rather than an interface jump of the global predictor. The adaptation score is $\eta_m = \eta_m^{\mathrm{res}} + \lambda_{\mathrm{bdry}}\eta_m^{\mathrm{bdry}} + \lambda_{\mathrm{jump}}\eta_m^{\mathrm{jump}}$; the weights are nonnegative and either optional term can be disabled. These scores rank candidates; the reliability result below instead uses mass-weighted residual and condition indicators.

3.2 Polynomial Enrichment, Pruning, and Functional Merging

Let $\widetilde{\eta} = \mathrm{median}_m\, \eta_m$. Relative thresholding identifies components whose scores are large compared with the current representation, without prescribing a fixed absolute residual tolerance. A component satisfying $\eta_m > \tau_{\mathrm{split}}\widetilde{\eta}$ is eligible for p-enrichment, $p_m \leftarrow \min(p_m + 1, p_{\max})$. When

the pre-update degree satisfies $p_m < p_{\max}$, this activates preallocated coefficients while preserving its center, scale, and component count. If the newly activated coefficients are zero, enrichment preserves the predictor exactly at the update and makes additional polynomial directions available to subsequent optimization.

With $\gamma_m = \mathrm{sigmoid}(g_m)$, pruning targets $\gamma_m < \tau_{\mathrm{gate}}$, optionally also requiring $\eta_m < \tau_{\mathrm{merge}}\widetilde{\eta}$. Candidates are ordered by gate and then error, while retaining at least $M_{\min}$ components. This is a gate-based selection heuristic: in a normalized mixture, a small absolute gate alone does not bound a component's relative influence. Merging instead screens low-error components for both effective overlap and agreement of local predictions. At evaluation points $\{z_q\}_{q=1}^{N_e}$, use

$$O_{ij} = \frac{1}{N_e}\sum_q \pi_i(z_q)\pi_j(z_q), \qquad D_{ij} = \frac{\sum_q \pi_i(z_q)\pi_j(z_q)|\ell_i(z_q) - \ell_j(z_q)|^2}{\sum_q \pi_i(z_q)\pi_j(z_q) + \varepsilon}. \tag{4}$$

Accept only pairs with $O_{ij} > \tau_{\mathrm{overlap}}$ and $D_{ij} < \tau_{\mathrm{sim}}$, rank them by decreasing overlap, increasing discrepancy, and increasing combined error, then select disjoint pairs. Define the average responsibilities $q_m = N_e^{-1}\sum_q \pi_m(z_q)$ and weights $w_i = q_i/(q_i + q_j + \varepsilon)$ and $w_j = q_j/(q_i + q_j + \varepsilon)$. The merged geometry uses the following regularized moment-based update:

$$\begin{aligned} \mu_{ij} &= w_i\mu_i + w_j\mu_j, \\ \sigma_{ij}^{\odot 2} &= w_i[\sigma_i^{\odot 2} + (\mu_i - \mu_{ij})^{\odot 2}] \\ &\quad + w_j[\sigma_j^{\odot 2} + (\mu_j - \mu_{ij})^{\odot 2}]. \end{aligned} \tag{5}$$

Here $\odot 2$ denotes element-wise squaring. With $\varepsilon = 0$ and positive total mass, these formulas match the weighted mean and diagonal variance; for $\varepsilon > 0$, the weights sum to less than one. The same weights combine coefficients, $c_{ij} = w_i c_i + w_j c_j$, and gate probabilities, $\gamma_{ij} = w_i\gamma_i + w_j\gamma_j$, before $g_{ij} = \mathrm{logit}(\gamma_{ij})$. Any latent code is averaged likewise, and $p_{ij} = \max(p_i, p_j)$. This merge initialization is heuristic: functional agreement before merging does not ensure an unchanged predictor, because the polynomials use different local coordinates and the responsibilities also change. Appendix A gives the full rules.

3.3 Residual-directed splitting and coordinate-aware initialization

The remaining capacity $M_{\max} - M$, a fractional budget, and a hard per-event cap limit splitting. The geometric baseline selects the parent's largest-scale axis. The residual-directed alternative uses η_m^{res} for candidate selection and constructs

$$\begin{aligned} \omega_{im} &= \pi_m(z_i)|r_\Theta(z_i)|^2, \qquad \bar{z}_m = \frac{\sum_i \omega_{im} z_i}{\sum_i \omega_{im} + \varepsilon}, \\ C_m &= \frac{\sum_i \omega_{im}(z_i - \bar{z}_m)(z_i - \bar{z}_m)^\top}{\sum_i \omega_{im} + \varepsilon}. \end{aligned} \tag{6}$$

Choose a unit leading eigenvector $v_m \in \arg\max_{\|v\|_2=1} v^\top C_m v$. Responsibilities emphasize the region associated with the parent, while squared residual emphasizes unresolved physics within that region. The resulting direction tracks the dominant spread of residual energy in the normalized input coordinates; it need not align with a coordinate axis or the parent's largest scale. Negligible residual mass or an invalid covariance triggers the geometric fallback. A deterministic subset may be selected using residual energy and responsibility. With exact normalization by $W = \sum_i \omega_{im} > 0$, this direction maximizes weighted projected variance (Proposition B.2); this characterizes the geometry of the refinement, without asserting that it maximizes subsequent error reduction.

Set $s_m = (v_m^\top \mathrm{diag}(\sigma_m^2) v_m)^{1/2}$ and clamp the tentative displacement $\alpha_{\mathrm{split}} s_m v_m$ in parent-normalized coordinates. The default anchor is the trained parent center; an optional residual-weighted anchor is also clamped. The child's scale is $\mathrm{clip}(\beta_\sigma\sigma_m, \sigma_{\min}, \sigma_{\max})$, with $0 < \beta_\sigma < 1$ and $0 < \sigma_{\min} \le \sigma_{\max}$. The full displacement and fallback rules appear in Appendix A.

Because coefficients refer to local coordinates, simply copying them changes the polynomial. Parent and child coordinates instead satisfy

$$\xi_p = d + b \odot \xi_c, \qquad d = (\mu_c - \mu_p) \oslash \sigma_p, \qquad b = \sigma_c \oslash \sigma_p. \tag{7}$$

The multivariate binomial expansion gives

$$c_\beta^c = \sum_{\substack{\alpha \in \mathcal{A}_{p_p} \\ \alpha \geq \beta}} c_\alpha^p \binom{\alpha}{\beta} d^{\alpha-\beta} b^\beta, \qquad \binom{\alpha}{\beta} = \prod_q \binom{\alpha_q}{\beta_q}. \tag{8}$$

Here $\alpha \geq \beta$ is component-wise, and the child retains the parent's active degree. The complete total-degree basis is closed under this shift and diagonal rescaling, giving $\ell_c(z) = \ell_p(z)$ (Lemma B.3). For example, shifting and rescaling a linear polynomial generally changes both its slope in child coordinates and its constant coefficient; copying the coefficient vector misses this correction. Inactive higher-order coefficients must be zero or excluded, since they can translate into active lower-order terms. Translation therefore preserves the *local function*; controlling the *global predictor* additionally requires controlling the inserted component's weight.

The default additive split retains the parent and initializes a quiet child:

$$\gamma_c = \mathrm{clip}(\min\{\rho_g \gamma_p, \gamma_0, \gamma_{\max}\}, \gamma_{\min}, \gamma_{\max}), \qquad g_c = \mathrm{logit}(\gamma_c), \qquad \rho_g \ll 1. \tag{9}$$

The bounds $0 < \gamma_{\min} \leq \gamma_{\max} < 1$ make the logit finite. The gate is intended to limit the child's initial contribution while leaving its geometry and polynomial available for subsequent specialization. Together, translation and gating address different sources of disturbance: translation removes the coordinate-change error, while gating controls the change in aggregation. Section 3.4 quantifies this effect for a normalized mixture. Optional symmetric replacement shifts both offspring oppositely, shrinks and translates both, and divides the gate; it changes the parent as well and is outside that bound.

3.4 Conditional reliability and controlled adaptation

Residual accuracy implies solution accuracy only under stability in the chosen norms (Hu et al., 2022; Jiang et al., 2026), not for arbitrary PDEs or losses (Wang et al., 2022). For this analysis, let Q be the PDE domain and Γ the condition domain, equipped with probability measures μ_Q, μ_Γ; let $\kappa_\Gamma \geq 0$. Suppose $Lu = f$, $Bu = g$, and

$$\|v - u\|_X^2 \leq C_{\mathrm{stab}}^2 \big(\|Lv - f\|_{L^2(\mu_Q)}^2 + \kappa_\Gamma \|Bv - g\|_{L^2(\mu_\Gamma)}^2 \big). \tag{10}$$

Theorem 3.1 (Conditional ReCAP reliability). *Under equation 10, assume $0 \leq \pi_m \leq 1$, $\sum_m \pi_m = 1$, and $\varepsilon > 0$. For masses $s_m = \int_Q \pi_m \, d\mu_Q$ and $s_m^\Gamma = \int_\Gamma \pi_m \, d\mu_\Gamma$, define continuum indicators by the corresponding ε-regularized weighted averages of $|r|^2$ and $|b|^2$, where $r = Lu_\Theta - f$ and $b = Bu_\Theta - g$. Then*

$$\|u_\Theta - u\|_X^2 \leq C_{\mathrm{stab}}^2 \left[\sum_m (s_m + \varepsilon) \eta_m^{\mathrm{res}} + \kappa_\Gamma \sum_m (s_m^\Gamma + \varepsilon) \eta_m^{\mathrm{bdry}} \right]. \tag{11}$$

The independent-sample certificate, including its finite-sample correction terms and assumptions, is stated and proved in Theorem B.1 in Appendix B.

The key identity is $\sum_m (s_m + \varepsilon) \eta_m^{\mathrm{res}} = \|r\|_{L^2(\mu_Q)}^2$: mass weighting reconstructs the global residual norm, whereas unweighted scores identify where to adapt. The theorem transfers an assumed PDE stability estimate to these localized quantities; it does not establish that estimate for every benchmark or make the jump score an error bound. The general condition residual $b = Bu_\Theta - g$ reduces to the value mismatch in equation 2 when B is a value trace. Independent certification samples are essential: reusing training points requires uniform generalization control (Hu et al., 2022) or continuous-domain verification (Eiras et al., 2024). The proof (Theorem B.1) and an explicit transport estimate appear in Appendix B.

Controlling the effect of a new child. For one output channel, assume a normalized gated mixture with Gaussian weights ϕ_m:

$$u^- = \frac{\sum_m q_m \ell_m}{D}, \qquad q_m = \gamma_m \phi_m, \qquad D = \sum_m q_m. \tag{12}$$

Additive insertion of a translated child with $\ell_c = \ell_p$ and $q_c = \gamma_c \phi_c$ gives $u^+ - u^- = \frac{q_c}{D + q_c}(\ell_p - u^-)$. The change depends on the child's relative weight and parent–mixture disagreement. Other aggregation rules require a separate perturbation analysis.

Theorem 3.2 (Additive quiet-child stability). *Let $L = \partial_t + \beta \cdot \nabla$ and use equation 12. Retain the parent unchanged and translate the child so that $\ell_c = \ell_p$. Assume the gate is constant in z, $0 \le \gamma_c \le \epsilon_g$, and $0 \le \phi_c \le 1$, $D \ge d_0 > 0$ on $Q \cup \Gamma$, with $\|L\phi_c\|_{L^\infty(Q)} \le K_\phi$ and $\|LD\|_{L^\infty(Q)} \le K_D$. For $J(v) = \|Lv - f\|^2_{L^2(Q)} + \kappa_\Gamma \|v - g\|^2_{L^2(\Gamma)}$, put $h = \ell_p - u^-$ and $K_\alpha = K_\phi/d_0 + K_D/d_0^2$. Define the finite constant*

$$C_{\text{split}} = \left[\left(K_\alpha \|h\|_{L^2(Q)} + d_0^{-1} \|Lh\|_{L^2(Q)} \right)^2 + \kappa_\Gamma d_0^{-2} \|h\|^2_{L^2(\Gamma)} \right]^{1/2}. \tag{13}$$

The post-split predictor satisfies

$$\begin{aligned} \sqrt{J(u^+)} &\le \sqrt{J(u^-)} + \epsilon_g C_{\text{split}}, \\ J(u^+) - J(u^-) &\le 2\epsilon_g C_{\text{split}} \sqrt{J(u^-)} + \epsilon_g^2 C^2_{\text{split}}. \end{aligned} \tag{14}$$

If admissible child gates can approach zero, the refined class contains the old class in its closure, so its optimal achievable loss cannot increase.

The proof appears with Theorem B.4 in Appendix B. For controlled C_{split}, a smaller gate limits the immediate loss increase. The constant also exposes the relevant conditions: adequate mixture coverage through d_0, controlled child derivatives through K_ϕ, and agreement between the parent and current mixture through h. A small absolute gate alone is therefore insufficient when the denominator is very small or the child is excessively narrow. A fixed positive gate floor also precludes the zero-gate argument for class inclusion. This result concerns additive insertion for the stated transport objective, rather than monotonic training loss or a guarantee for nonlinear and higher-order benchmark operators. Zero-initialized p-enrichment preserves the current function exactly; symmetric replacement, pruning, and merging require separate perturbation analysis.

4 Experimental Results

Benchmarks, baselines and implementation details. We evaluate ReCAP on five PDE benchmarks: flow mixing (FM3D), Helmholtz (H3D), nonlinear diffusion (ND3D), and Klein–Gordon equations (KG3D and KG4D). The dimension counts all input coordinates, including time: H3D has three spatial coordinates, FM3D, ND3D, and KG3D have two spatial coordinates and time, and KG4D has three spatial coordinates and time. Appendix C provides the problem definitions. All experiments run on a single NVIDIA RTX 4090, with baselines reimplemented in the same hardware and software environment. We compare against eleven methods: PINN (Raissi et al., 2019), ActNet (Guilhoto & Perdikaris, 2025), PirateNets (Wang et al., 2024a), SPINN (Cho et al., 2023), JAXPI (Wang et al., 2023), CoupledNet (Meng et al., 2026), CausalPINN (Wang et al., 2024b), DB-PINN (Zhou et al., 2025), SINN (Yu & Oseledets, 2026), Scale-PINN (Chiu et al., 2026), and PIG (Kang et al., 2025). ReCAP hyperparameters and computational costs appear in Appendices D and F, respectively. We report relative L^2 error to compare solution accuracy across these benchmarks.

Main Results. ReCAP achieves the lowest reported relative L^2 error on each of the five benchmarks in Table 1. Compared with the strongest baseline on each problem, ReCAP reduces error by 27.5% on FM3D (PirateNets), 10.7% on H3D (SPINN), 12.1% on ND3D (PIG), 15.7% on KG3D (SPINN), and 12.2% on KG4D (JAXPI). These comparisons establish the improvement on the tested three- and four-dimensional problems; the ablations below examine how the representation and its adaptation contribute to the observed behavior.

The strongest baseline varies across the benchmarks: PirateNets, SPINN, PIG, and JAXPI each provide the best competing result on at least one problem. ReCAP improves over these problem-specific reference points across transport, elliptic, nonlinear diffusion, and wave equations. This breadth supports its effectiveness on the tested problems; the ablations below examine whether the gains depend on adaptive allocation beyond component-count scaling.

Figure 2 provides spatial evidence complementary to the aggregate error metrics. On ND3D, ReCAP closely reproduces the reference solution, including the localized peaks and surrounding low-amplitude structure. For KG3D, the predicted field preserves the pronounced spatiotemporal variations while the corresponding pointwise error remains small over most of the domain. The FM3D

Table 1: Reported relative L^2 errors on five PDE benchmarks (lower is better). The lowest error in each column is highlighted in green, and the second-lowest is underlined. The final row gives ReCAP's percentage error reduction relative to the strongest baseline, computed from the displayed values. A dash indicates an unreported result.

Method	FM3D	H3D	ND3D	KG3D	KG4D
PINN (Raissi et al., 2019)	0.001968	0.527911	0.007788	0.042415	0.009683
SPINN (Cho et al., 2023)	0.007995	<u>0.066970</u>	0.005182	<u>0.003773</u>	0.004786
JAXPI (Wang et al., 2023)	0.000328	0.110786	0.002229	0.007136	<u>0.003942</u>
PirateNets (Wang et al., 2024a)	<u>0.000291</u>	0.077632	0.002505	0.003860	0.007206
CausalPINN (Wang et al., 2024b)	0.000383	-	0.006638	0.021933	0.008271
DB-PINN (Zhou et al., 2025)	0.002053	0.587973	0.005622	0.007994	0.017544
PIG (Kang et al., 2025)	0.000305	0.184739	<u>0.001791</u>	0.004206	0.005286
ActNet (Guilhoto & Perdikaris, 2025)	0.001787	0.093405	0.004842	0.006199	0.005991
CoupledNet (Meng et al., 2026)	0.000856	0.158846	0.006488	0.070929	0.009005
SINN (Yu & Oseledets, 2026)	0.003955	0.700347	0.003598	0.011043	0.059185
Scale-PINN (Chiu et al., 2026)	0.034121	0.565737	0.082126	0.092426	0.093574
ReCAP (Ours)	0.000211	0.059810	0.001574	0.003181	0.003463
Improvement	27.5%	10.7%	12.1%	15.7%	12.2%

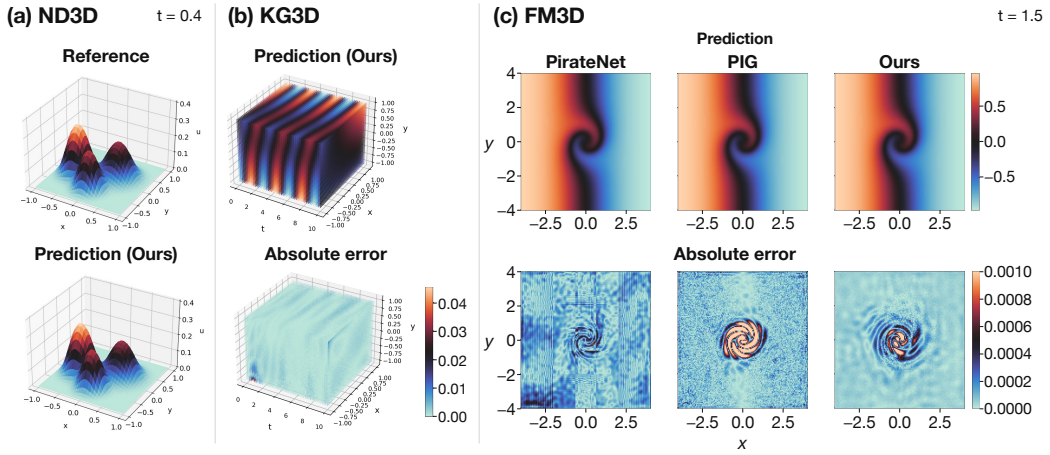


Figure 2: **Qualitative solution and error comparisons.** (a) ND3D at $t = 0.4$: reference solution and ReCAP prediction. (b) KG3D: ReCAP prediction and pointwise absolute error. (c) FM3D at $t = 1.5$: predictions and corresponding absolute-error maps for PirateNets, PIG, and ReCAP. ReCAP accurately resolves the dominant solution structures while maintaining small and spatially localized errors, including around rapidly varying regions.

comparison further highlights the difference between methods: although all three models recover the overall spiral structure, ReCAP exhibits substantially less background error and concentrates the remaining error near the most rapidly varying region around the spiral core and fronts. These visualizations complement the aggregate errors by showing how ReCAP resolves localized solution structures and where prediction errors remain. Additional solution visualizations are provided in Appendix E.

Ablation study. Figure 3 examines ReCAP's key design choices on KG3D, including (i) the cumulative contributions of its adaptation components, (ii) the benefits of adaptive capacity allocation beyond particle-count scaling, and (iii) sensitivity to the splitting threshold.

Cumulative component ablation. We first evaluate the incremental benefits of the adaptation mechanisms on KG3D (Figure 3, left). Starting from M1, a dynamic Gaussian-particle representation with hard constraints, we successively introduce particle splitting for local refinement (M2),

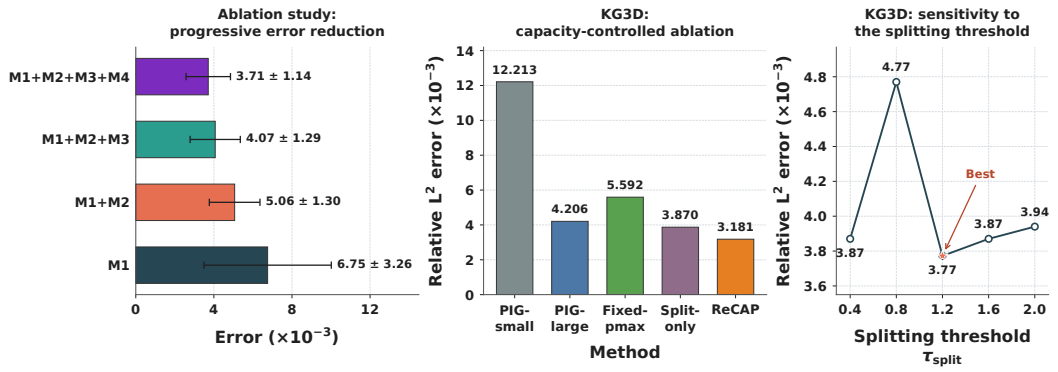


Figure 3: KG3D ablations. Left: cumulative error reduction from adding M2–M4 to the base representation M1. Middle: capacity-controlled comparisons. Right: splitting-threshold sensitivity, with $\tau_{\text{split}} = 1.2$ giving the lowest error among the tested values.

pruning and merging for redundancy control (M3), and residual-based adaptive collocation sampling (M4). The relative L^2 error decreases at every stage, from 6.75×10^{-3} to 5.06×10^{-3}, 4.07×10^{-3}, and finally 3.71×10^{-3}, corresponding to a 45.0% reduction over M1. The reductions continue after splitting, supporting refinement together with redundancy control and adaptive sampling. Since the components are introduced cumulatively, the observed reductions quantify incremental gains conditional on preceding components, rather than their independent effects.

Adaptive allocation versus particle-count scaling. We next examine whether increasing the particle count is sufficient to recover ReCAP's accuracy (Figure 3, middle). PIG-small uses the initial Gaussian count, whereas PIG-large matches the final count reached by ReCAP. Two additional controls restrict how capacity is allocated: Fixed-$p_{\max}$ assigns the maximum polynomial order uniformly, while Split-only uses adaptive splitting without polynomial-order refinement or coarsening. ReCAP achieves a relative L^2 error of 3.181×10^{-3}, reducing the error by 24.4% relative to PIG-large, 43.1% relative to Fixed-$p_{\max}$, and 17.8% relative to Split-only. Thus, matching the final particle count does not close the accuracy gap, and neither uniformly maximal polynomial order nor splitting alone matches the complete method. Together, these comparisons support ReCAP's combined strategy of local spatial refinement, polynomial-order adaptation, and redundancy control over the tested alternatives.

Splitting-threshold sensitivity. We evaluate sensitivity to the splitting threshold over $\tau_{\text{split}} \in \{0.4, 0.8, 1.2, 1.6, 2.0\}$ on KG3D (Figure 3, right). The best tested setting, $\tau_{\text{split}} = 1.2$, achieves a relative L^2 error of 3.772×10^{-3}, a 20.9% reduction relative to $\tau_{\text{split}} = 0.8$. The response is non-monotonic: although 0.8 produces the largest error, the remaining thresholds, 0.4, 1.6, and 2.0, yield errors between 3.870×10^{-3} and 3.940×10^{-3}, all within 4.5% of the minimum. The sweep therefore favors $\tau_{\text{split}} = 1.2$, with several other thresholds giving similar accuracy. These values correspond to the sensitivity sweep; the main-benchmark results are reported separately in Table 1.

5 Conclusion

ReCAP treats representational capacity as an adaptive resource for physics-informed PDE solving. It couples residual-guided polynomial enrichment and splitting with function-aware coarsening, using coordinate-aware quiet-child initialization to introduce new capacity. Across five three- and four-dimensional PDE benchmarks, ReCAP achieves the lowest reported relative L^2 error; capacity-controlled comparisons further support the benefit of targeted allocation beyond increasing component count alone. Under the stated assumptions, the analysis connects mass-weighted physics indicators to solution error and bounds the perturbation from additive refinement. Together, these results support physics-informed neural plasticity: jointly learning the solution and adapting where, how, and how much capacity is used to represent it.

AI USE STATEMENT

In this work, we used generative AI tools to assist in developing and writing proofs of mathematical claims, provide feedback on methodological exposition, and refine the interpretation and presentation of results. Additionally, we used these tools to polish the writing, improve clarity and organization, We have reviewed all AI-assisted work, including checking the assumptions, mathematical statements, and individual steps of the proofs, and verifying that the methodological descriptions and interpretations are consistent with our implementation and experimental evidence. We take responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Ivo Babuška and Manil Suri. The p and h-p versions of the finite element method, basic principles and properties. *SIAM Review*, 36(4):578–632, 1994. doi: 10.1137/1036141. 3

Jonah Botvinick-Greenhouse, Wael H Ali, Mouhacine Benosman, and Saviz Mowlavi. AB-PINNs: Adaptive-basis physics-informed neural networks for residual-driven domain decomposition. *Machine Learning: Science and Technology*, 7:045025, 2026. doi: 10.1088/2632-2153/ae8638. 2, 3

David S. Broomhead and David Lowe. Multivariable functional interpolation and adaptive networks. *Complex Systems*, 2:321–355, 1988. 3

Pao-Hsiung Chiu, Jian Cheng Wong, Chin Chun Ooi, Chang Wei, Yuchen Fan, and Yew-Soon Ong. Scale-PINN: Learning efficient physics-informed neural networks through sequential correction, 2026. URL https://arxiv.org/abs/2602.19475. 7, 8

Junwoo Cho, Seungtae Nam, Hyunmo Yang, Seok-Bae Yun, Youngjoon Hong, and Eunbyung Park. Separable physics-informed neural networks. *Advances in Neural Information Processing Systems*, 36:23761–23788, 2023. 3, 7, 8

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 *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 7264–7302. PMLR, 2023. URL https://proceedings.mlr.press/v202/daw23a.html. 1, 3

Francisco Eiras, Adel Bibi, Rudy R Bunel, Krishnamurthy Dj Dvijotham, Philip Torr, and M. Pawan Kumar. Efficient error certification for physics-informed neural networks. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12318–12347. PMLR, 2024. URL https://proceedings.mlr.press/v235/eiras24a.html. 6, 19

Leonardo Ferreira Guilhoto and Paris Perdikaris. Deep learning alternatives of the Kolmogorov superposition theorem. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/c2469e35d469e3c0eca09dbe484eb474-Abstract-Conference.html. 7, 8

Zheyuan Hu, Ameya D Jagtap, George Em Karniadakis, and Kenji Kawaguchi. When do extended physics-informed neural networks (XPINNs) improve generalization? *SIAM Journal on Scientific Computing*, 44(5):A3158–A3182, 2022. doi: 10.1137/21M1447039. 6, 18, 19

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. *Communications in Computational Physics*, 28(5):2002–2041, 2020. doi: 10.4208/cicp.OA-2020-0164. 3

Deqing Jiang, Justin Sirignano, and Samuel N Cohen. Global convergence of deep galerkin and PINN methods for solving partial differential equations. *SIAM Journal on Financial Mathematics*, 17(2):620–645, 2026. doi: 10.1137/24M1701502. 6, 18

Namgyu Kang, Jaemin Oh, Youngjoon Hong, and Eunbyung Park. PIG: Physics-Informed Gaussians as Adaptive Parametric Mesh Representations. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=y5B0ca4mjt. 2, 3, 7, 8

George E 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. doi: 10.1038/s42254-021-00314-5. 1, 2

Ehsan Kharazmi, Zhongqiang Zhang, and George Em Karniadakis. hp-VPINNs: Variational physics-informed neural networks with domain decomposition. *Computer Methods in Applied Mechanics and Engineering*, 374:113547, 2021. doi: 10.1016/j.cma.2020.113547. 3

Aditi S. Krishnapriyan, Amir Gholami, Shandian Zhe, Robert Kirby, and Michael W Mahoney. Characterizing possible failure modes in physics-informed neural networks. *Advances in Neural Information Processing Systems*, 34, 2021. 1, 3

Lingshi Meng, Haosen Shi, and Sinno Jialin Pan. Deep coupling learning for solving PDEs. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=4pfxoYPQSY. 7, 8

Ben Moseley, Andrew Markham, and Tarje Nissen-Meyer. Finite basis physics-informed neural networks (FBPINNs): a scalable domain decomposition approach for solving differential equations. *Advances in Computational Mathematics*, 49(4):62, 2023. doi: 10.1007/s10444-023-10065-9. 2, 3

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5301–5310. PMLR, 2019. URL https://proceedings.mlr.press/v97/rahaman19a.html. 1, 3

Maziar Raissi, Paris Perdikaris, and George E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019. doi: 10.1016/j.jcp.2018.10.045. 1, 2, 7, 8

Panos Tamamidis and Dennis N. Assanis. Evaluation of various high-order-accuracy schemes with and without flux limiters. *International Journal for Numerical Methods in Fluids*, 16(10):931–948, 1993. 21

Chuwei Wang, Shanda Li, Di He, and Liwei Wang. Is L^2 physics informed loss always suitable for training physics informed neural network? *Advances in Neural Information Processing Systems*, 35:8278–8290, 2022. 6, 18

Sifan Wang, Yujun Teng, and Paris Perdikaris. Understanding and mitigating gradient flow pathologies in physics-informed neural networks. *SIAM Journal on Scientific Computing*, 43(5):A3055–A3081, 2021. doi: 10.1137/20M1318043. 1, 2

Sifan Wang, Shyam Sankaran, Hanwen Wang, and Paris Perdikaris. An expert's guide to training physics-informed neural networks. *arXiv preprint arXiv:2308.08468*, 2023. 3, 7, 8

Sifan Wang, Bowen Li, Yuhan Chen, and Paris Perdikaris. PirateNets: Physics-informed deep learning with residual adaptive networks. *Journal of Machine Learning Research*, 25(402):1–51, 2024a. URL https://jmlr.org/papers/v25/24-0313.html. 1, 3, 7, 8

Sifan Wang, Shyam Sankaran, and Paris Perdikaris. Respecting causality for training physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 421:116813, 2024b. doi: 10.1016/j.cma.2024.116813. 7, 8

Chenxi Wu, Min Zhu, Qinyang Tan, Yadhu Kartha, and Lu Lu. A comprehensive study of non-adaptive and residual-based adaptive sampling for physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 403:115671, 2023. doi: 10.1016/j.cma.2022.115671. 1, 3

Tianchi Yu and Ivan Oseledets. Spectral-informed neural networks outperform spectral methods in high-dimensional PDEs. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=KAHCMPsPeI. 7, 8

Chenhong Zhou, Jie Chen, Zaifeng Yang, and Ching Eng Png. Dual-balancing for physics-informed neural networks. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, IJCAI '25, pp. 7164–7172, 2025. doi: 10.24963/ijcai.2025/797. URL https://www.ijcai.org/proceedings/2025/797. 7, 8

Supplementary Material

Physics-Informed Neural Plasticity: PDE Solvers That Reshape Themselves

CONTENTS

A Detailed Representation and Adaptation Rules **13**

A.1 Problem Formulation 13

A.2 Adaptive Gaussian Local Representation 14

A.3 Physics-Guided Local Error Estimation 14

A.4 Adaptive Representation Refinement and Coarsening 15

B Further Theoretical Results **18**

C PDE Benchmarks **21**

C.1 (2+1)D Flow Mixing 21

C.2 3D Helmholtz Equation 22

C.3 (2+1)D Nonlinear Diffusion 22

C.4 Klein–Gordon Equations 22

D Experimental Settings **22**

E Visualization **23**

F Computational Cost **26**

A DETAILED REPRESENTATION AND ADAPTATION RULES

The following construction expands the representation and component-update rules summarized in Section 3.

A.1 PROBLEM FORMULATION

Let $\Omega \subset \mathbb{R}^d$ denote a spatial or spatiotemporal domain, and consider a differential equation of the form $\mathcal{N}[u](\mathbf{z}) = 0, \quad \mathbf{z} \in \Omega$, subject to initial or boundary conditions $\mathcal{B}[u](\mathbf{z}) = g(\mathbf{z}), \quad \mathbf{z} \in \partial\Omega$. We denote the learned solution by u_Θ, where Θ contains both the continuously optimized parameters of the base model and the parameters of the adaptive Gaussian representation. The corresponding physics residual is $r_\Theta(\mathbf{z}) = \mathcal{N}[u_\Theta](\mathbf{z})$.

The adaptation mechanism is independent of the precise composition of the base training objective. We therefore denote that objective abstractly by $\mathcal{L}_{\text{base}}(\Theta)$. Between structural adaptation events, all currently active continuous parameters are optimized by gradient-based updates to $\mathcal{L}_{\text{base}}$.

For example, the FM3D benchmark uses coordinates $\mathbf{z} = (t, x, y)$, with linear transport residual

$$r_\Theta(t, x, y) = \partial_t u_\Theta(t, x, y) + a\,\partial_x u_\Theta(t, x, y) + b\,\partial_y u_\Theta(t, x, y), \tag{15}$$

equation 15 is the residual for FM3D. For the other benchmarks, the same adaptation rules use the corresponding PDE residual specified in Appendix C.

A.2 Adaptive Gaussian Local Representation

At a given optimization stage, the representation contains M Gaussian-localized components $\mathcal{G} = \{G_m\}_{m=1}^{M}$. For concise notation, we write $G_m = (\boldsymbol{\mu}_m, \boldsymbol{\sigma}_m, g_m, \mathbf{c}_m, p_m)$, where $\boldsymbol{\mu}_m \in \mathbb{R}^d$ is the component center, $\boldsymbol{\sigma}_m \in \mathbb{R}_+^d$ is its diagonal scale, $g_m \in \mathbb{R}$ is a gate logit, $\mathbf{c}_m$ contains local polynomial coefficients, and p_m is the currently active polynomial order.

After absorbing the fixed coordinate normalization used by the model into $\mathbf{z}$, the normalized coordinate of component m is $\boldsymbol{\xi}_m(\mathbf{z}) = (\mathbf{z} - \boldsymbol{\mu}_m) \oslash \boldsymbol{\sigma}_m$, where $\oslash$ denotes element-wise division. The component represents a local polynomial

$$\ell_m(\mathbf{z}) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}_{p_m}} c_{m,\boldsymbol{\alpha}}\, \boldsymbol{\xi}_m(\mathbf{z})^{\boldsymbol{\alpha}}, \qquad \mathcal{A}_{p_m} = \left\{ \boldsymbol{\alpha} \in \mathbb{N}_0^d : |\boldsymbol{\alpha}| \le p_m \right\}. \tag{16}$$

Here, $\boldsymbol{\xi}^{\boldsymbol{\alpha}} = \prod_{q=1}^{d} \xi_q^{\alpha_q}$. In the three-dimensional setting, the basis contains all monomials up to total degree three, including quadratic and cubic cross terms, while the local order p_m determines which basis terms are active for each component.

Let $\pi_m(\mathbf{z}) \ge 0$ denote the effective responsibility of component m. Any additional internal branch or channel index is averaged out when computing the adaptation statistics, including responsibilities, centers, local values, and local gradients, and is omitted from the notation for clarity. Since the adaptation procedure is agnostic to the specific responsibility and output-aggregation rules, we express the model prediction abstractly as $u_\Theta(\mathbf{z}) = \mathcal{A}_\Theta\left(\{\pi_m(\mathbf{z}), \ell_m(\mathbf{z}), g_m\}_{m=1}^{M}\right)$. Our adaptation procedure requires only that the base model expose the prediction u_Θ, the effective responsibilities $\{\pi_m\}$, and, when necessary, the component-wise local values $\{\ell_m\}$.

A.3 Physics-Guided Local Error Estimation

The global physics loss measures aggregate equation and condition mismatch, but does not identify which components require additional capacity. We localize these quantities using the learned component responsibilities; a solution-error interpretation additionally requires the stability assumptions in Appendix B.

Let $\mathcal{C} = \{\mathbf{z}_i\}_{i=1}^{N_c}$ denote the collocation set. The component-wise residual indicator is

$$\eta_m^{\mathrm{res}} = \frac{\sum_{i=1}^{N_c} \pi_m(\mathbf{z}_i)\, |r_\Theta(\mathbf{z}_i)|^2}{\sum_{i=1}^{N_c} \pi_m(\mathbf{z}_i) + \varepsilon}. \tag{17}$$

Thus, a large value of η_m^{res} indicates a large weighted PDE residual in the region for which component m is responsible.

When value-based initial or boundary observations are available, we add a condition-error indicator. Let $\mathcal{D}_\Gamma = \{(\mathbf{z}_i^\Gamma, u_i^\Gamma)\}_{i=1}^{N_\Gamma}$ denote the condition points. Then

$$\eta_m^{\mathrm{bdry}} = \frac{\sum_{i=1}^{N_\Gamma} \pi_m(\mathbf{z}_i^\Gamma) \left| u_\Theta(\mathbf{z}_i^\Gamma) - u_i^\Gamma \right|^2}{\sum_{i=1}^{N_\Gamma} \pi_m(\mathbf{z}_i^\Gamma) + \varepsilon}. \tag{18}$$

In the implementation, initial and boundary points are combined when the initial condition is imposed softly. If the initial condition is already hard-constrained by the model, only the remaining boundary points are used in equation 18.

Residual error alone may fail to identify neighboring components whose local approximations are mutually inconsistent. We therefore construct a k-nearest-neighbor graph between Gaussian centers. Let $\mathcal{N}_k(m)$ denote the neighbors of component m. For a pair (m, n), define their overlap-weighted gradient discrepancy as

$$J_{mn} = \frac{\sum_{i=1}^{N_j} \pi_m(\mathbf{z}_i)\pi_n(\mathbf{z}_i)\, \|\nabla \ell_m(\mathbf{z}_i) - \nabla \ell_n(\mathbf{z}_i)\|_2^2}{\sum_{i=1}^{N_j} \pi_m(\mathbf{z}_i)\pi_n(\mathbf{z}_i) + \varepsilon}. \tag{19}$$

The component-wise inconsistency indicator is then

$$\eta_m^{\mathrm{jump}} = \frac{1}{|\mathcal{N}_k(m)|} \sum_{n \in \mathcal{N}_k(m)} J_{mn}. \tag{20}$$

The product $\pi_m \pi_n$ weights the comparison toward points at which both components are relevant. The denominator regularizer attenuates pairs with negligible overlap mass. Gradients are taken in a common global coordinate system, including the chain-rule factors from each component's local coordinates. This is a local disagreement score, not a literal interface jump of the global predictor.

The complete adaptation score is

$$\eta_m = \eta_m^{\text{res}} + \lambda_{\text{bdry}} \eta_m^{\text{bdry}} + \lambda_{\text{jump}} \eta_m^{\text{jump}}, \tag{21}$$

where $\lambda_{\text{bdry}}, \lambda_{\text{jump}} \geq 0$ and either optional term can be disabled. Unlike a global loss, equation 21 assigns the current physics error to individual learned components and therefore provides a criterion for targeted changes in representation complexity.

A.4 Adaptive Representation Refinement and Coarsening

The localized error indicators in Section A.3 determine how representation capacity is redistributed across the domain. At each adaptation event, we first compute the robust reference value

$$\widetilde{\eta} = \text{median}\,\{\eta_m\}_{m=1}^{M}.$$

The median limits the influence of a small number of extreme residuals and makes the adaptation criteria relative to the current distribution of component-wise errors. Based on these statistics, the representation is modified through polynomial enrichment, pruning, merging, and Gaussian splitting.

Polynomial-order refinement. Before introducing additional Gaussian components, we first attempt to increase the approximation order of high-error components. A component satisfying

$$\eta_m > \tau_{\text{split}} \widetilde{\eta} \tag{22}$$

is eligible for local polynomial enrichment, with

$$p_m \leftarrow \min\{p_m + 1, p_{\max}\}. \tag{23}$$

When the pre-update degree satisfies $p_m < p_{\max}$, this operation activates the next group of pre-allocated polynomial terms without increasing the number of Gaussian components. It therefore provides local p-adaptation: the active polynomial basis expands while the Gaussian geometry and component count remain unchanged. Setting the newly activated coefficients to zero preserves the current predictor.

Gate-based pruning. Pruning targets components with small gate probabilities. Let

$$\gamma_m = \text{sigmoid}(g_m)$$

denote the effective gate probability, averaged over internal branches when applicable. Component m becomes a pruning candidate when

$$\gamma_m < \tau_{\text{gate}}. \tag{24}$$

An optional conservative criterion additionally requires the component to have low localized error,

$$\eta_m < \tau_{\text{merge}} \widetilde{\eta}.$$

When the number of candidates exceeds the removal budget, components with the smallest gates are prioritized, with localized error used as an additional ordering criterion. Pruning is constrained so that at least $M_{\min}$ components remain, preventing excessive collapse of the representation.

Functional merging. Pruning removes weakly active components individually, whereas merging coarsens redundant components that represent similar local functions. Low-error components satisfying

$$\eta_m < \tau_{\text{merge}} \widetilde{\eta}$$

are considered for merging. Importantly, geometric proximity alone is not sufficient: two components must exhibit both non-negligible effective overlap and similar local predictions.

Using a set of evaluation points $\{\mathbf{z}_q\}_{q=1}^{N_e}$, we define the effective overlap between components i and j as

$$O_{ij} = \frac{1}{N_e} \sum_{q=1}^{N_e} \pi_i(\mathbf{z}_q)\pi_j(\mathbf{z}_q), \tag{25}$$

and their overlap-weighted functional discrepancy as

$$D_{ij} = \frac{\sum_{q=1}^{N_e} \pi_i(\mathbf{z}_q)\pi_j(\mathbf{z}_q) \left|\ell_i(\mathbf{z}_q) - \ell_j(\mathbf{z}_q)\right|^2}{\sum_{q=1}^{N_e} \pi_i(\mathbf{z}_q)\pi_j(\mathbf{z}_q) + \varepsilon}. \tag{26}$$

A pair is accepted only if

$$O_{ij} > \tau_{\text{overlap}}, \qquad D_{ij} < \tau_{\text{sim}}. \tag{27}$$

Thus, merging is restricted to components that are simultaneously relevant in the same region and already describe similar local functions. Candidate pairs are ranked according to large overlap, small functional discrepancy, and small combined error. Selected pairs are disjoint, so each component participates in at most one merge during an adaptation event.

To construct the merged component, define the average effective responsibilities and corresponding regularized weights as

$$q_m = \frac{1}{N_e} \sum_{q=1}^{N_e} \pi_m(\mathbf{z}_q), \qquad w_i = \frac{q_i}{q_i + q_j + \varepsilon}, \qquad w_j = \frac{q_j}{q_i + q_j + \varepsilon}. \tag{28}$$

The merged center and diagonal variance use the following moment-based update. It gives exact weighted moment matching when $\varepsilon = 0$ and $q_i + q_j > 0$; positive ε makes the weights sum to less than one:

$$\boldsymbol{\mu}_{ij} = w_i \boldsymbol{\mu}_i + w_j \boldsymbol{\mu}_j, \tag{29}$$

$$\boldsymbol{\sigma}_{ij}^{\odot 2} = w_i \left[\boldsymbol{\sigma}_i^{\odot 2} + (\boldsymbol{\mu}_i - \boldsymbol{\mu}_{ij})^{\odot 2}\right] + w_j \left[\boldsymbol{\sigma}_j^{\odot 2} + (\boldsymbol{\mu}_j - \boldsymbol{\mu}_{ij})^{\odot 2}\right]. \tag{30}$$

The polynomial coefficients and gate probabilities are combined using the same weights, while the larger polynomial order is retained:

$$\mathbf{c}_{ij} = w_i \mathbf{c}_i + w_j \mathbf{c}_j, \tag{31}$$

$$\gamma_{ij} = w_i \gamma_i + w_j \gamma_j, \qquad g_{ij} = \text{logit}(\gamma_{ij}), \tag{32}$$

$$p_{ij} = \max\{p_i, p_j\}. \tag{33}$$

Any associated latent code is likewise averaged. The functional-similarity test in Equation equation 27 screens candidates before this heuristic initialization. It does not imply function preservation: coefficient vectors refer to different local coordinates, and the merged geometry and responsibilities also differ. The quiet-child perturbation theorem does not cover this merge rule.

Residual-guided Gaussian splitting. Complementing polynomial enrichment, splitting introduces additional localized representation capacity in unresolved regions. The number of newly created components is limited by the remaining capacity $M_{\max} - M$, a fractional split budget, and a hard per-adaptation cap. These constraints prevent a single adaptation event from producing a large architectural change.

Largest-scale direction. A simple geometric strategy selects a high-error parent and splits it along its largest spatial or spatiotemporal scale. Specifically,

$$k_m^\star = \arg \max_{q \in \{1,\ldots,d\}} \sigma_{m,q}, \tag{34}$$

and the corresponding split direction is $\mathbf{v}_m = \mathbf{e}_{k_m^\star}$. This produces anisotropic refinement along the direction in which the parent Gaussian currently has the broadest support.

Residual-covariance direction. A more informative alternative uses the distribution of unresolved physics error to determine the split direction. In this mode, split candidates are selected using

the residual-only indicator η_m^{res}. For each candidate component m, define its residual-weighted responsibility as

$$\omega_{im} = \pi_m(\mathbf{z}_i) \left| r_\Theta(\mathbf{z}_i) \right|^2 . \tag{35}$$

The corresponding residual-weighted center and covariance are

$$\overline{\mathbf{z}}_m = \frac{\sum_i \omega_{im} \mathbf{z}_i}{\sum_i \omega_{im} + \varepsilon}, \tag{36}$$

$$\mathbf{C}_m = \frac{\sum_i \omega_{im} \left(\mathbf{z}_i - \overline{\mathbf{z}}_m\right) \left(\mathbf{z}_i - \overline{\mathbf{z}}_m\right)^\top}{\sum_i \omega_{im} + \varepsilon}. \tag{37}$$

The split direction is chosen as the principal eigenvector

$$\mathbf{v}_m \in \arg\max_{\|\mathbf{v}\|_2 = 1} \mathbf{v}^\top \mathbf{C}_m \mathbf{v}. \tag{38}$$

Hence, refinement is oriented along the direction in which the responsibility-weighted residual exhibits the largest spread, rather than along a random or purely geometric direction. If the residual mass is negligible or the covariance is numerically invalid, the implementation falls back to a geometric direction, typically the largest-scale axis. When only a subset of residual points is used, these points can additionally be selected deterministically according to their residual energy and responsibility, avoiding additional randomness from point subsampling.

For either splitting strategy, we define the directional scale of the parent by

$$s_m = \sqrt{\mathbf{v}_m^\top \operatorname{diag}\left(\boldsymbol{\sigma}_m^{\odot 2}\right) \mathbf{v}_m}. \tag{39}$$

A tentative displacement is $\alpha_{\mathrm{split}} s_m \mathbf{v}_m$. To prevent the newborn component from being placed far outside the effective support of its parent, we apply the scale-normalized clamp

$$\mathrm{Clamp}_{\boldsymbol{\sigma}}(\boldsymbol{\delta}) = \boldsymbol{\delta} \min\left\{1,\ \frac{\delta_{\max}}{\|\boldsymbol{\delta} \oslash \boldsymbol{\sigma}\|_2 + \varepsilon}\right\}.$$

The final displacement is therefore

$$\Delta\boldsymbol{\mu}_m = \mathrm{Clamp}_{\boldsymbol{\sigma}_m}\left(\alpha_{\mathrm{split}} s_m \mathbf{v}_m\right). \tag{40}$$

By default, the trained parent center is used as the split anchor, $\mathbf{a}_m = \boldsymbol{\mu}_m$. The residual-weighted center $\overline{\mathbf{z}}_m$ may alternatively be used, while constraining its displacement from the parent in normalized parent-scale units. The child geometry is then initialized as

$$\boldsymbol{\mu}_{m'} = \mathbf{a}_m + \Delta\boldsymbol{\mu}_m, \qquad \boldsymbol{\sigma}_{m'} = \mathrm{clip}\left(\beta_\sigma \boldsymbol{\sigma}_m,\ \sigma_{\min},\ \sigma_{\max}\right), \tag{41}$$

where $0 < \beta_\sigma < 1$ and $0 < \sigma_{\min} \le \sigma_{\max}$. Reducing the scale of the child enables it to specialize to a smaller unresolved region.

Near-function-preserving split initialization. A structural refinement should ideally increase future representation capacity without causing a large instantaneous perturbation to the learned solution. Simply copying the parent polynomial coefficients to a displaced and rescaled child does not achieve this, because the coefficients are defined relative to the component's normalized local coordinates.

Let the parent and child coordinates be

$$\boldsymbol{\xi}_p = (\mathbf{z} - \boldsymbol{\mu}_p) \oslash \boldsymbol{\sigma}_p, \qquad \boldsymbol{\xi}_c = (\mathbf{z} - \boldsymbol{\mu}_c) \oslash \boldsymbol{\sigma}_c.$$

These coordinates satisfy the affine relation

$$\boldsymbol{\xi}_p = \mathbf{d} + \mathbf{b} \odot \boldsymbol{\xi}_c, \qquad \mathbf{d} = (\boldsymbol{\mu}_c - \boldsymbol{\mu}_p) \oslash \boldsymbol{\sigma}_p, \qquad \mathbf{b} = \boldsymbol{\sigma}_c \oslash \boldsymbol{\sigma}_p. \tag{42}$$

Suppose the active parent polynomial is

$$\ell_p(\boldsymbol{\xi}_p) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}_p} c_{\boldsymbol{\alpha}}^p \boldsymbol{\xi}_p^{\boldsymbol{\alpha}}.$$

Substituting Equation equation 42 and applying the multi-index binomial expansion yields the child coefficients

$$c^c_{\boldsymbol{\beta}} = \sum_{\substack{\boldsymbol{\alpha}\in\mathcal{A}_p \\ \boldsymbol{\alpha}\succeq\boldsymbol{\beta}}} c^p_{\boldsymbol{\alpha}} \binom{\boldsymbol{\alpha}}{\boldsymbol{\beta}} \mathbf{d}^{\boldsymbol{\alpha}-\boldsymbol{\beta}} \mathbf{b}^{\boldsymbol{\beta}}, \tag{43}$$

where

$$\binom{\boldsymbol{\alpha}}{\boldsymbol{\beta}} = \prod_{q=1}^{d} \binom{\alpha_q}{\beta_q},$$

and $\boldsymbol{\alpha} \succeq \boldsymbol{\beta}$ denotes component-wise inequality. For a complete monomial basis through the active degree, this translation makes the child polynomial represent the same function of the global coordinate as the parent polynomial at initialization.

Preserving the local polynomial does not, however, guarantee that the complete model prediction is exactly unchanged because the output aggregation also depends on the Gaussian responsibilities and gates. To further suppress the perturbation caused by inserting the child, the default refinement leaves the trained parent unchanged and initializes the child with a small gate probability. Let $\gamma_p = \mathrm{sigmoid}(g_p)$. The child probability is initialized schematically as

$$\gamma_c = \mathrm{clip}\left(\min\left\{\rho_g \gamma_p,\, \gamma_0,\, \gamma_{\max}\right\}, \gamma_{\min},\, \gamma_{\max}\right), \qquad g_c = \mathrm{logit}(\gamma_c), \tag{44}$$

where $\rho_g \ll 1$ and $0 < \gamma_{\min} \le \gamma_{\max} < 1$. Under the normalized-mixture assumptions of Theorem B.4, the perturbation is bounded in terms of the child gate, mixture coverage, and component derivatives. The update aims for

$$u_{\Theta^+}(\mathbf{z}) \approx u_{\Theta^-}(\mathbf{z}) \tag{45}$$

at the instant of insertion. Subsequent gradient updates can then gradually activate and specialize the newborn component in the unresolved region.

The implementation additionally supports a symmetric replacement strategy in which the parent and child are placed on opposite sides of the split anchor, both scales are reduced, both local polynomials are translated into their new coordinate systems, and the original gate probability is divided between them. The additive quiet-child strategy is the stated default: it preserves the trained parent parameters, and its instantaneous perturbation can be bounded under the assumptions in Theorem B.4.

B Further Theoretical Results

We provide an a posteriori interpretation of the localized ReCAP indicators and show that the default additive Gaussian split produces a controlled instantaneous perturbation. The analysis separates three questions: whether the local scores control solution error, whether the residual-covariance direction maximizes residual-weighted projected coordinate variance, and how inserting a new component perturbs the current physics objective. The stability premise below follows the standard principle that residual accuracy implies solution accuracy only when the underlying PDE is stable in the chosen residual norm (Hu et al., 2022; Jiang et al., 2026). This premise is not automatic for arbitrary PDEs or losses (Wang et al., 2022). The full statements and proofs are given below.

Let u solve $\mathcal{L}u = f$ in Q and $\mathcal{B}u = g$ on Γ. Let μ_Q, μ_Γ be probability measures, $\kappa_\Gamma \ge 0$, and $\varepsilon > 0$. For a candidate v, assume

$$\|v - u\|_X^2 \le C_{\mathrm{stab}}^2 \left(\|\mathcal{L}v - f\|^2_{L^2(\mu_Q)} + \kappa_\Gamma \|\mathcal{B}v - g\|^2_{L^2(\mu_\Gamma)} \right). \tag{46}$$

Let the effective Gaussian responsibilities satisfy $0 \le \pi_m \le 1$ and $\sum_{m=1}^{M} \pi_m = 1$. Define

$$s_m = \int_Q \pi_m \, d\mu_Q, \qquad \eta_m^{\mathrm{res}} = \frac{\int_Q \pi_m |r_\Theta|^2 \, d\mu_Q}{s_m + \varepsilon}, \tag{47}$$

$$s_m^\Gamma = \int_\Gamma \pi_m \, d\mu_\Gamma, \qquad \eta_m^{\mathrm{bdry}} = \frac{\int_\Gamma \pi_m |b_\Theta|^2 \, d\mu_\Gamma}{s_m^\Gamma + \varepsilon}, \tag{48}$$

where $r_\Theta = \mathcal{L}u_\Theta - f$ and $b_\Theta = \mathcal{B}u_\Theta - g$.

Theorem B.1 (Mass-weighted ReCAP reliability certificate). *Under equation 46 and the partition-of-unity condition,*

$$\|u_\Theta - u\|_X^2 \le C_{\mathrm{stab}}^2 \left[\sum_{m=1}^{M} (s_m + \varepsilon)\eta_m^{\mathrm{res}} + \kappa_\Gamma \sum_{m=1}^{M} (s_m^\Gamma + \varepsilon)\eta_m^{\mathrm{bdry}} \right]. \tag{49}$$

Moreover, freeze u_Θ and draw independent certification samples $Z_i \sim \mu_Q$, $i = 1, \dots, N_c$, and $Y_j \sim \mu_\Gamma$, $j = 1, \dots, N_\Gamma$. For $N_c, N_\Gamma \ge 1$, let the empirical masses and indicators be defined by replacing the integrals above with sample sums, and set

$$\widehat{\mathcal{E}}_{\mathrm{ReCAP}}^2 = \frac{1}{N_c} \sum_m (\widehat{s}_m + \varepsilon)\widehat{\eta}_m^{\mathrm{res}} + \frac{\kappa_\Gamma}{N_\Gamma} \sum_m (\widehat{s}_m^\Gamma + \varepsilon)\widehat{\eta}_m^{\mathrm{bdry}}. \tag{50}$$

If $|r_\Theta| \le R_\infty$ on Q and $|b_\Theta| \le G_\infty$ on Γ, then, for $0 < \delta < 1$, with probability at least $1 - \delta$,

$$\|u_\Theta - u\|_X \le C_{\mathrm{stab}} \left[\widehat{\mathcal{E}}_{\mathrm{ReCAP}}^2 + R_\infty^2 \sqrt{\frac{\log(4/\delta)}{2N_c}} + \kappa_\Gamma G_\infty^2 \sqrt{\frac{\log(4/\delta)}{2N_\Gamma}} \right]^{1/2}. \tag{51}$$

Proof. By definition and the partition-of-unity identity,

$$\sum_m (s_m + \varepsilon)\eta_m^{\mathrm{res}} = \sum_m \int_Q \pi_m |r_\Theta|^2 \, d\mu_Q = \int_Q \left(\sum_m \pi_m \right) |r_\Theta|^2 \, d\mu_Q = \|r_\Theta\|_{L^2(\mu_Q)}^2. \tag{52}$$

The same calculation gives $\sum_m (s_m^\Gamma + \varepsilon)\eta_m^{\mathrm{bdry}} = \|b_\Theta\|_{L^2(\mu_\Gamma)}^2$. Substitution into equation 46 proves equation 49.

For every realization of the certification samples, the same algebra gives

$$\frac{1}{N_c} \sum_m (\widehat{s}_m + \varepsilon)\widehat{\eta}_m^{\mathrm{res}} = \frac{1}{N_c} \sum_{i=1}^{N_c} |r_\Theta(Z_i)|^2, \tag{53}$$

and likewise on Γ. Since the predictor is frozen before the certification samples are drawn, Hoeffding's inequality yields, with probability at least $1 - \delta/2$,

$$\|r_\Theta\|_{L^2(\mu_Q)}^2 \le \frac{1}{N_c} \sum_i |r_\Theta(Z_i)|^2 + R_\infty^2 \sqrt{\frac{\log(4/\delta)}{2N_c}}, \tag{54}$$

and, with probability at least $1 - \delta/2$,

$$\|b_\Theta\|_{L^2(\mu_\Gamma)}^2 \le \frac{1}{N_\Gamma} \sum_j |b_\Theta(Y_j)|^2 + G_\infty^2 \sqrt{\frac{\log(4/\delta)}{2N_\Gamma}}. \tag{55}$$

A union bound and equation 46 prove equation 51. □

If $\kappa_\Gamma = 0$, the condition terms and their sample-count requirements may be omitted.

Theorem B.1 shows that mass-weighted local indicators reconstruct the global physics loss. The unweighted component score remains useful for ranking refinements, but it is not itself a global certificate. The independent-sample requirement is essential: reusing training points requires a uniform generalization argument such as those developed for PINNs and XPINNs (Hu et al., 2022), or a continuous-domain verification method (Eiras et al., 2024).

Explicit transport stability. For the linear transport operator $\mathcal{L} = \partial_t + \beta \cdot \nabla$ on a space–time cylinder $Q = (0, T) \times \Omega_x$, let $\Gamma_- = \{x \in \partial\Omega_x : \beta(x) \cdot n(x) < 0\}$, $\Sigma_- = (0, T) \times \Gamma_-$, where n is the outward unit normal, and $c_\beta = 1 + \|\nabla \cdot \beta\|_{L^\infty(\Omega_x)}$. If $e = u_\Theta - u$, $r = \mathcal{L}e$, $e_0 = e(0, \cdot)$, and $e_- = e|_{\Sigma_-}$, then

$$\|e\|_{L^2(Q)}^2 \le T e^{c_\beta T} \left(\|e_0\|_{L^2(\Omega_x)}^2 + \|r\|_{L^2(Q)}^2 + \|e_-\|_{L^2(\Sigma_-; |\beta \cdot n|)}^2 \right). \tag{56}$$

Indeed, multiplying $\partial_t e + \beta \cdot \nabla e = r$ by $2e$, integrating by parts, discarding the nonnegative outflow term, and applying Young's inequality gives

$$\frac{d}{dt}\|e(t)\|^2_{L^2(\Omega_x)} \le c_\beta \|e(t)\|^2_{L^2(\Omega_x)} + \|r(t)\|^2_{L^2(\Omega_x)} + \int_{\Gamma_-} |\beta \cdot n| e_-^2 \, dS. \tag{57}$$

Gronwall's inequality followed by integration over $t \in (0, T)$ proves equation 56. The calculation assumes sufficient regularity for these derivatives and traces. The displayed estimate uses volume and flux-weighted boundary measures; normalizing these to probability measures rescales the stability constants.

Proposition B.2 (Optimal residual-covariance direction)**.** *For a selected component, define* $\omega_i = \pi_m(z_i)|r_\Theta(z_i)|^2$, $W = \sum_i \omega_i > 0$, $\bar{z} = W^{-1}\sum_i \omega_i z_i$, *and* $C = W^{-1}\sum_i \omega_i (z_i - \bar{z})(z_i - \bar{z})^\top$. *If* v_1 *is a unit eigenvector associated with* $\lambda_{\max}(C)$, *then*

$$v_1 \in \arg\max_{\|v\|_2 = 1} \frac{1}{W} \sum_i \omega_i \big(v^\top (z_i - \bar{z})\big)^2. \tag{58}$$

Proof. The objective equals $v^\top C v$. Writing $C = V \Lambda V^\top$ and $a = V^\top v$ gives $v^\top C v = \sum_k \lambda_k a_k^2 \le \lambda_{\max}(C) \sum_k a_k^2 = \lambda_{\max}(C)$, with equality for a leading eigenvector. □

We next quantify the stability of the default additive split. Assume one output channel has the normalized gated mixture form

$$u^-(z) = \frac{\sum_{m=1}^M q_m(z)\ell_m(z)}{D(z)}, \qquad q_m = \gamma_m \phi_m, \qquad D = \sum_{m=1}^M q_m. \tag{59}$$

This is the aggregation assumption for the perturbation result below. A different decoder, branch combination, or hard-constraint transformation requires bounds for the corresponding output map.

Lemma B.3 (Exact polynomial translation)**.** *Let* $\xi_p = (z - \mu_p) \oslash \sigma_p$ *and* $\xi_c = (z - \mu_c) \oslash \sigma_c$. *Set* $d = (\mu_c - \mu_p) \oslash \sigma_p$ *and* $b = \sigma_c \oslash \sigma_p$. *If* $\ell_p = \sum_{|\alpha| \le p} c_\alpha^p \xi_p^\alpha$ *and the complete monomial basis through degree* p *is used, then the child coefficients*

$$c_\beta^c = \sum_{\substack{|\alpha| \le p \\ \alpha \succeq \beta}} c_\alpha^p \binom{\alpha}{\beta} d^{\alpha - \beta} b^\beta \tag{60}$$

satisfy $\ell_c(z) = \ell_p(z)$ *for every* z.

Proof. The coordinates satisfy $\xi_p = d + b \odot \xi_c$. Substituting this identity into the parent polynomial and applying the multi-index binomial theorem gives

$$\ell_p = \sum_{|\alpha| \le p} c_\alpha^p \sum_{\beta \preceq \alpha} \binom{\alpha}{\beta} d^{\alpha - \beta} b^\beta \xi_c^\beta. \tag{61}$$

Collecting the coefficient of each ξ_c^β gives equation 60. Completeness of the basis ensures that every resulting monomial is present. □

When component-wise local orders are used, the translation must be applied only to source coefficients whose degree is active for the parent, or all inactive stored coefficients must be zero. Otherwise an inactive higher-order coefficient can translate into an active lower-order coefficient, and exact preservation need not hold.

Theorem B.4 (Shock-controlled quiet-child refinement)**.** *Translate a new child from parent* p *according to Lemma B.3, leave the parent unchanged, and let* $q_c = \gamma_c \phi_c$ *with* $0 \le \gamma_c \le \epsilon_g$ *and* γ_c *constant in* z. *Assume* $\mathcal{L} = \partial_t + \beta \cdot \nabla$, $0 \le \phi_c \le 1$ *on* $Q \cup \Gamma$, *and that the following lower bound on* D *holds on* $Q \cup \Gamma$:

$$D \ge d_0 > 0, \qquad \|\mathcal{L}\phi_c\|_{L^\infty(Q)} \le K_\phi, \qquad \|\mathcal{L}D\|_{L^\infty(Q)} \le K_D. \tag{62}$$

Set $h = \ell_p - u^-$, $K_\alpha = K_\phi/d_0 + K_D/d_0^2$, *and*

$$C_{\text{split}} = \left[\left(K_\alpha \|h\|_{L^2(Q)} + d_0^{-1} \|\mathcal{L}h\|_{L^2(Q)} \right)^2 + \kappa_\Gamma d_0^{-2} \|h\|^2_{L^2(\Gamma)} \right]^{1/2}. \tag{63}$$

For $J(v) = \|\mathcal{L}v - f\|^2_{L^2(Q)} + \kappa_\Gamma \|v - g\|^2_{L^2(\Gamma)}$, *the post-split predictor satisfies*

$$\sqrt{J(u^+)} \le \sqrt{J(u^-)} + \epsilon_g C_{\text{split}}, \tag{64}$$

$$J(u^+) - J(u^-) \le 2\epsilon_g C_{\text{split}} \sqrt{J(u^-)} + \epsilon_g^2 C^2_{\text{split}}. \tag{65}$$

If the child gate can approach zero, then the closure of the refined hypothesis class contains the pre-split class and its optimal achievable loss cannot increase.

Proof. By Lemma B.3, the new child represents ℓ_p in global coordinates. Directly from equation 59,

$$u^+ - u^- = \alpha h, \qquad \alpha = \frac{q_c}{D + q_c}. \tag{66}$$

Since $q_c \le \epsilon_g$ and $D \ge d_0$, $0 \le \alpha \le \epsilon_g/d_0$. Moreover,

$$\mathcal{L}\alpha = \frac{D\,\mathcal{L}q_c - q_c\,\mathcal{L}D}{(D + q_c)^2}. \tag{67}$$

The gate is constant with respect to z at insertion, so $\|\mathcal{L}q_c\|_\infty \le \epsilon_g K_\phi$. Therefore

$$\|\mathcal{L}\alpha\|_\infty \le \epsilon_g \left(\frac{K_\phi}{d_0} + \frac{K_D}{d_0^2} \right) = \epsilon_g K_\alpha. \tag{68}$$

Applying the product rule to equation 66 gives

$$\|\mathcal{L}(u^+ - u^-)\|_{L^2(Q)} \le \epsilon_g \left(K_\alpha \|h\|_{L^2(Q)} + d_0^{-1} \|\mathcal{L}h\|_{L^2(Q)} \right), \tag{69}$$

$$\|u^+ - u^-\|_{L^2(\Gamma)} \le \epsilon_g d_0^{-1} \|h\|_{L^2(\Gamma)}. \tag{70}$$

Thus the product norm of the residual and boundary perturbations is at most $\epsilon_g C_{\text{split}}$. Minkowski's inequality proves equation 64; squaring it proves equation 65. Finally, retaining every old parameter and sending the child gate logit to $-\infty$ makes $q_c \to 0$, so every pre-split predictor lies in the closure of the refined class. □

Theorem B.4 applies to the additive, parent-preserving split. It does not directly cover the optional symmetric replacement strategy. Likewise, pruning and merging require separate perturbation guards. For local order enrichment, setting every newly activated polynomial coefficient to zero makes the insertion exactly function-preserving and enlarges the hypothesis class.

C PDE Benchmarks

We specify the governing equations, domains, and reference solutions or initial data for the five benchmarks. The suffix 3D or 4D counts all input coordinates, including time where applicable.

C.1 (2+1)D Flow Mixing

FM3D is a linear transport benchmark for the advection and rotational mixing of a passive scalar in a two-dimensional velocity field:

$$\begin{aligned} &u_t + a(x,y)u_x + b(x,y)u_y = 0, \quad (x,y,t) \in [-4,4]^2 \times [0,4], \\ &a(x,y) = -\frac{v_{\tan}(r)}{v_{\max}} \frac{y}{r}, \qquad b(x,y) = \frac{v_{\tan}(r)}{v_{\max}} \frac{x}{r}, \\ &v_{\tan}(r) = \operatorname{sech}^2(r) \tanh(r), \qquad r = \sqrt{x^2 + y^2}, \qquad v_{\max} = 0.385. \end{aligned} \tag{71}$$

The analytic solution is

$$u^\star(x,y,t) = -\tanh\left[\frac{y}{2} \cos(\omega(r)t) - \frac{x}{2} \sin(\omega(r)t) \right], \qquad \omega(r) = \frac{v_{\tan}(r)}{r v_{\max}}, \tag{72}$$

with initial condition $u(x,y,0) = -\tanh(y/2)$ (Tamamidis & Assanis, 1993). The apparent singularities at $r = 0$ are removable: $a(0,0) = b(0,0) = 0$ and $\omega(0) = 1/v_{\max}$. These values follow from $v_{\tan}(r)/r \to 1$ as $r \to 0$.

C.2 3D Helmholtz Equation

H3D is a stationary elliptic benchmark on $\Omega_x = [-1, 1]^3$:

$$\Delta_{\mathbf{x}} u + k^2 u = q \quad \text{in } \Omega_x, \qquad u = 0 \quad \text{on } \partial\Omega_x. \tag{73}$$

The manufactured solution and source are

$$u^\star(\mathbf{x}) = \prod_{i=1}^{3} \sin(a_i \pi x_i), \qquad q(\mathbf{x}) = \left[k^2 - \pi^2 \sum_{i=1}^{3} a_i^2\right] u^\star(\mathbf{x}), \tag{74}$$

where $k = 1$ and $(a_1, a_2, a_3) = (4, 4, 3)$. Thus $q = (1 - 41\pi^2)u^\star$, and the prescribed homogeneous boundary condition is satisfied exactly by $u^\star$.

C.3 (2+1)D Nonlinear Diffusion

ND3D describes solution-dependent diffusion in two spatial dimensions and time:

$$\begin{aligned} u_t &= 0.05\left(\|\nabla_{\mathbf{x}} u\|_2^2 + u\Delta_{\mathbf{x}} u\right), \quad (x, y, t) \in [-1, 1]^2 \times [0, 1], \\ u(\mathbf{x}, 0) &= 0.25 g\left(\mathbf{x}; 0.2, 0.3, \frac{1}{\sqrt{10}}\right) + 0.4 g\left(\mathbf{x}; -0.1, -0.5, \frac{1}{\sqrt{15}}\right) \\ &\quad + 0.3 g\left(\mathbf{x}; -0.5, 0, \frac{1}{\sqrt{20}}\right), \end{aligned} \tag{75}$$

where $\mathbf{x} = (x, y)$ and $g(\mathbf{x}; a, b, \sigma) = \exp\left(-[(x - a)^2 + (y - b)^2]/\sigma^2\right)$. The gradient and Laplacian act only on the spatial coordinates. The initial condition contains three localized peaks with different amplitudes, widths, and locations.

C.4 Klein–Gordon Equations

For KG3D and KG4D, let $d_s \in \{2, 3\}$ be the number of spatial coordinates, $\Omega_x = [-1, 1]^{d_s}$, and $t \in [0, 10]$. The total input dimension used in Section 3 is $d = d_s + 1$. Define

$$S(\mathbf{x}) = \sum_{i=1}^{d_s} x_i, \qquad P(\mathbf{x}) = \prod_{i=1}^{d_s} x_i, \qquad u^\star(\mathbf{x}, t) = S(\mathbf{x}) \cos(2t) + P(\mathbf{x}) \sin(2t). \tag{76}$$

The corresponding manufactured problem is

$$\begin{aligned} &u_{tt} - \Delta_{\mathbf{x}} u + u^2 = f, && (\mathbf{x}, t) \in \Omega_x \times (0, 10], \\ &u(\mathbf{x}, 0) = S(\mathbf{x}), \quad u_t(\mathbf{x}, 0) = 2P(\mathbf{x}), && \mathbf{x} \in \Omega_x, \\ &u(\mathbf{x}, t) = u^\star(\mathbf{x}, t), && (\mathbf{x}, t) \in \partial\Omega_x \times [0, 10], \end{aligned} \tag{77}$$

where $f = (u^\star)^2 - 4u^\star$, since $u^\star_{tt} = -4u^\star$ and $\Delta_{\mathbf{x}} u^\star = 0$. Thus $d_s = 2$ gives the stated KG3D solution and $d_s = 3$ gives KG4D. The initial-velocity condition follows by differentiating the manufactured solution; it is part of the complete second-order-in-time problem specification.

D Experimental Settings

In this section, we summarize the experimental configurations used for all benchmarks. The key hyperparameters, including the number of training epochs and other model-specific settings, are reported in Table 2.

Table 2: Key hyperparameters for the benchmarks.

Hyperparameter	FM3D	H3D	ND3D	KG3D	KG4D
Training epochs	80,000	50,000	100,000	50,000	100,000
Learning rate	5×10^{-3}	1×10^{-3}	2×10^{-3}	5×10^{-3}	1×10^{-3}
Batch size	1,024	1,024	1,024	1,024	512
Hidden features	48	16	32	48	128
Initial number of Gaussians	2,000	3,000	600	3,000	2,000
Minimum/maximum Gaussians	600/4,000	600/5,000	300/1,000	600/4,000	200/2,000
Polynomial degree	2	2	2	2	2
Gaussian scale parameter	0.15	0.15	0.20	0.15	0.20
Adaptation start	3,000	1,000	3,000	2,000	1,000
Adaptation interval	2,000	2,000	2,000	3,000	2,000
Split fraction	0.02	0.03	0.03	0.015	0.03
Adaptive sampling interval	3,000	3,000	3,000	4,000	3,000
τ_{split}	1.3	1.3	1.3	1.5	1.3
τ_{merge}	0.7	0.5	0.5	0.4	0.5

E VISUALIZATION

In this section, we provide qualitative visualizations of the predicted solutions and corresponding absolute-error distributions on representative PDE benchmarks, with comparisons against reference solutions and selected baselines in Figures 4, 5, and 6.

For FM3D, ReCAP accurately captures the evolving rotational mixing structures over time and closely follows the reference solution. The corresponding error maps show that the remaining errors are mainly localized around the increasingly complex spiral interfaces, while ReCAP maintains lower and more concentrated errors than the representative baselines.

For H3D and KG3D, ReCAP also reproduces the dominant spatial patterns across different slices and time instances. In both cases, the predicted solutions remain close to the reference fields, and the absolute-error maps indicate reduced deviations compared with SPINN and PirateNets, supporting its accuracy on the tested equations.

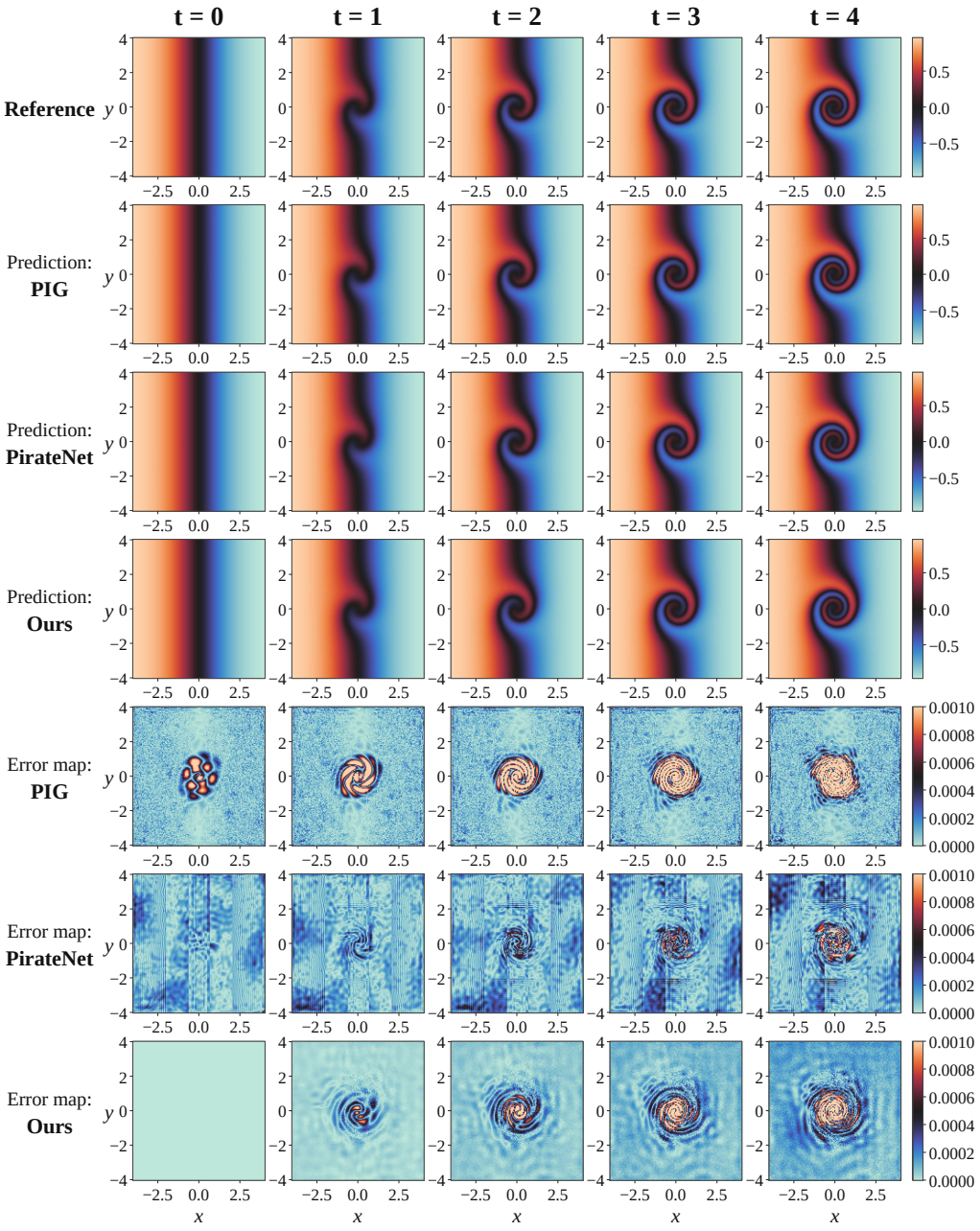


Figure 4: FM3D solution and absolute-error snapshots over the evaluated time points, comparing the reference field with ReCAP and representative baselines.

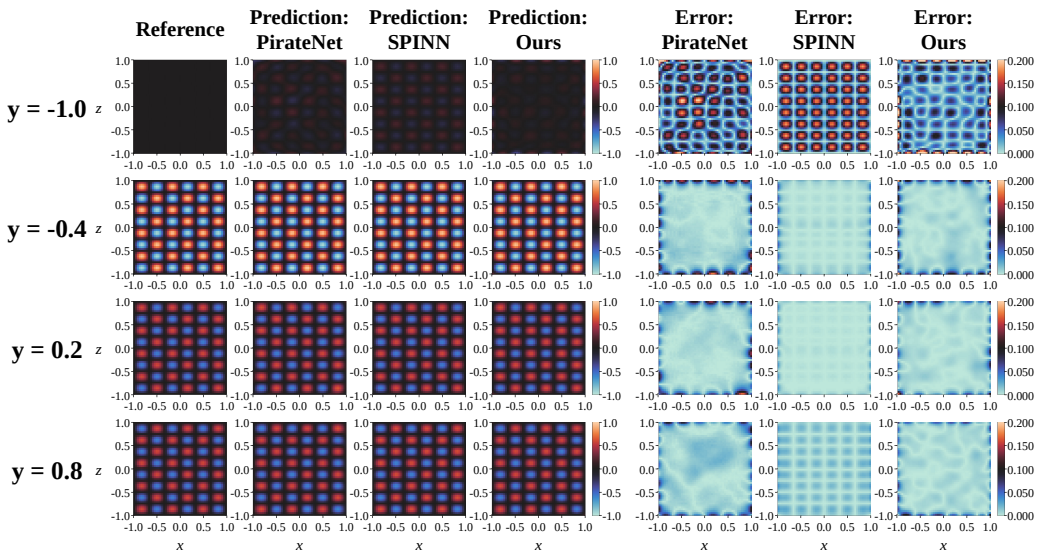


Figure 5: H3D reference solution, model predictions, and absolute-error maps on representative two-dimensional slices.

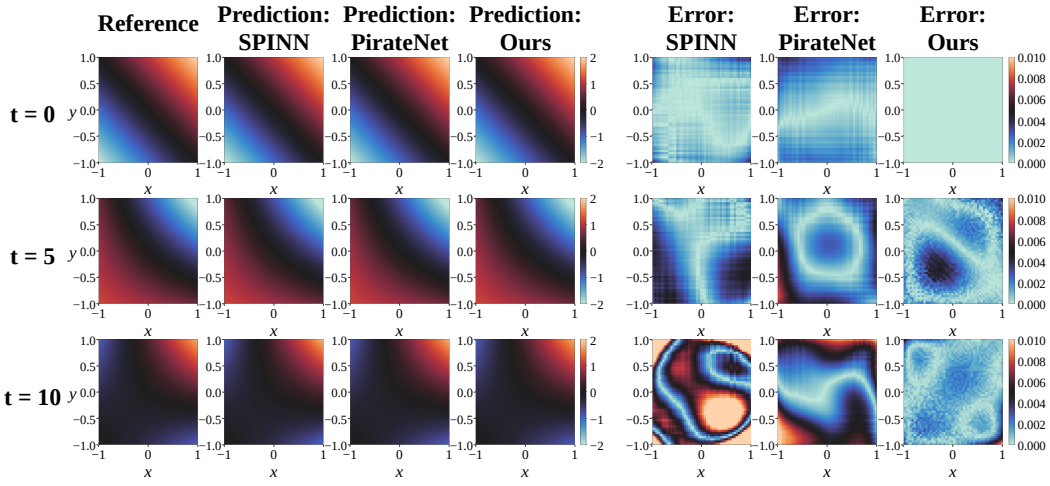


Figure 6: KG3D reference solution, model predictions, and absolute-error maps on representative two-dimensional slices.

F COMPUTATIONAL COST

Figure 7 compares reported floating-point operation counts on KG3D. ReCAP uses 0.53 GFLOPs in this comparison, matching SINN and falling below ActNet (1.64), PirateNets (2.59), and CoupledNet (4.38), while Scale-PINN uses 0.27 GFLOPs. These operation counts do not establish wall-clock training-time savings.

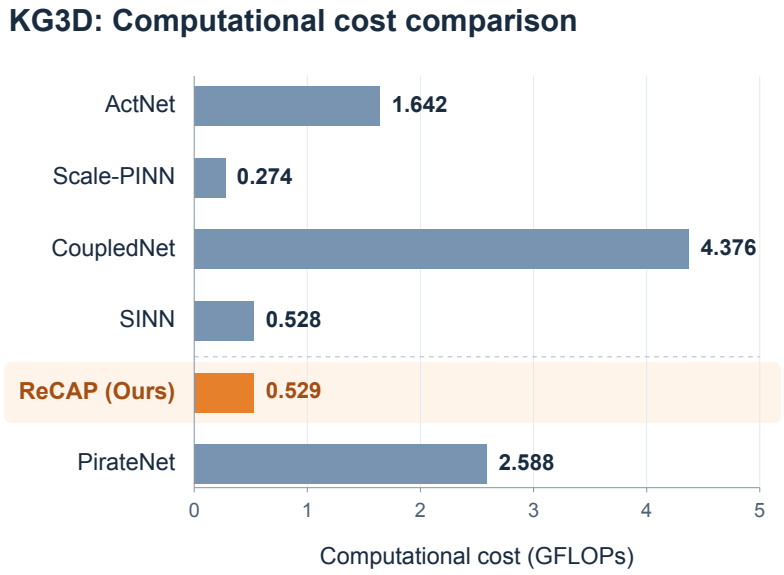


Figure 7: Reported computational cost on KG3D in GFLOPs (lower is better). ReCAP has the same reported operation count as SINN; Scale-PINN has the lowest count among the displayed methods.