SHRINKAGE-WEIGHTED BELLMAN REGRESSION FOR NEURAL PDE SOLVERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Random-walk PDE solvers fit neural networks to sampled Bellman targets whose noise varies across the domain. Weighting by return spread can control noisy updates, but the spread estimates are themselves uncertain and downweighting can weaken learning near steep solution features. We propose bounded shrinkage weighting for Bellman regression. The method accounts for sampling uncertainty in the estimated spread and shrinks weights toward uniformity when spatial differences are poorly resolved. Clipping and a uniform weight floor limit concentration and preserve a minimum contribution from every point. Our analysis distinguishes minimax noise-amplitude weighting from minimum-variance weighting, bounds the bias caused by dependence between weights and targets, and characterizes how reweighting changes a local least-squares fit. In a baseline study covering seven Burgers settings with 40 paired trials per setting, plain inverse-spread weighting lowers mean sampled hard-region error by approximately 42–71% relative to uniform regression, while individual runs can be less accurate. These results identify the practical challenge addressed by the safeguards: controlling noisy updates without losing useful corrections near sharp layers. The bounded-shrinkage comparisons and component ablations remain pending.

1 INTRODUCTION

Neural PDE solvers must turn the equation and its initial or boundary data into a useful training signal. Derivative-free loss methods (DFLMs) do this through stochastic consistency relations: a prediction at one point is regressed toward the average of values reached by short random walks (Han et al., 2020; Park & Stinchcombe, 2022; Cai et al., 2023). This avoids evaluating a strong-form PDE residual during regression, but replaces it with a sampled target. The quality of the update then depends on both the Bellman correction and the noise in that target.

The difficulty is especially clear near a viscous layer. Neighboring walkers can land on opposite sides of a sharp transition and return quite different values. Their mean is informative, yet its sampling error can exceed the correction the network should make. Decreasing the time step improves local discretization while also reducing the useful correction; increasing the walker count costs additional network evaluations. These interactions make time-step and sampling choices central to DFLM training (Han & Lee, 2025).

We study a complementary question: how should a fixed set of noisy targets contribute to a shared network update? Uniform regression gives every point the same weight. Inverse-spread regression reduces the influence of points with variable returns, but it can also weaken the learning signal precisely where the solution changes most rapidly. Moreover, spread is itself estimated from a finite group. A large observed difference between two points may reflect sampling variation rather than a stable difference in target uncertainty.

Our method, *bounded shrinkage Bellman regression*, addresses both effects. It estimates the noise contribution to cross-point log-spread variation and shrinks the weighting exponent accordingly. It then clips extreme relative weights and mixes them with a uniform component. The gate controls how much heterogeneity to trust; the floor controls how much signal to retain. The resulting update uses the same walkers and a detached regression objective, with independent target and weight groups available when their dependence is important.

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.

The paper makes three contributions. We formulate a weighting rule that accounts for finite-sample uncertainty while retaining a minimum contribution from every point. We derive its noise, dependence-bias, and local fitting tradeoffs. We then design an evaluation that separates accuracy, fixed-time efficiency, and reliability, using completed inverse-spread baseline measurements to motivate the tests. The central question is whether less concentrated weights preserve useful corrections at points with high return spread.

2 Problem and estimator definitions

We consider a scalar or componentwise vector field $u(x,t)$ satisfying

$$u_t + b[u]\cdot\nabla u = \nu\Delta u + f, \qquad u|_{t=0} = u_0, \qquad u|_{\partial\Omega} = g, \tag{1}$$

with periodic conditions for the periodic tasks. The drift $b[u]$ is prescribed or a specified function of the field, $\nu > 0$ is diffusivity, and f is a known source. A neural network u_θ represents the solution. During a regression step, a frozen copy u_{old} supplies the drift and target values.

At each of B interior collocation points, draw M independent standard Gaussian vectors $\xi_i^{(m)}$ and form the reverse-time Euler step

$$h_i = \min(\Delta t, t_i), \qquad X_i^{(m)} = x_i - b[u_{\mathrm{old}}](x_i,t_i)h_i + \sqrt{2\nu h_i}\,\xi_i^{(m)}, \tag{2}$$

$$G_i^{(m)} = u_{\mathrm{old}}(X_i^{(m)}, t_i - h_i) + f(x_i,t_i)h_i, \qquad E_i = M^{-1}\sum_m G_i^{(m)}. \tag{3}$$

For steady problems $h_i = \Delta t$. The displayed return describes an interior step. Initial-time hits use u_0, periodic endpoints are wrapped, and spatial exits use the prescribed boundary value at an approximate segment intersection, with the source duration shortened accordingly. Appendix D.2 details the boundary and discretization treatment.

The training objective is

$$L_w(\theta) = B^{-1}\sum_i w_i \|u_\theta(x_i,t_i) - \mathrm{sg}(E_i)\|^2, \qquad \mathrm{sg}(w_i) = w_i, \tag{4}$$

where sg denotes stop-gradient. Targets and weights remain fixed while differentiating with respect to θ. Throughout, an *exact regression gradient* means the derivative of this frozen-target loss. Refreshing the target network is a separate operation.

Let $J_i = \partial u_\theta(x_i,t_i)/\partial\theta$ have one row per output component. The batch gradient is a sum of $2w_iJ_i^\top(u_\theta - E_i)/B$. This expression shows why a scalar weight changes both the useful residual and its sampling noise, and why the same output uncertainty can have different effects at points with different Jacobians. We use spread as an inexpensive proxy and test this approximation through repeated-group measurements.

For the estimator comparison, let $a = \mu_\theta + \sigma z$, with $z \sim N(0,I)$ independent of the target and σ fixed with respect to the mean parameters. The raw score-function estimator of the quadratic loss is $\|a-E\|^2\nabla_\theta \log \pi_\theta(a)$. Group normalization and PPO clipping define separate comparators because they change this objective.

3 Noise, weighting, and local fitting

3.1 The scale of the Bellman correction

Lemma 1 (Interior near-identity expansion)**.** *For a sufficiently smooth u with locally bounded third derivatives and polynomial growth, bounded Lipschitz frozen drift, and an interior point away from initial/boundary exits, the population Euler target satisfies*

$$\mathcal{B}_h(u) - u = h(-u_t - b\cdot\nabla u + \nu\Delta u + f) + o(h). \tag{5}$$

The remainder need not be uniform over vanishing-viscosity families.

Proof. Expand $u(x - bh + \sqrt{2\nu h}\xi, t - h)$ in space and time. The increment has mean $-bh$ and second moment $2\nu hI + O(h^2)$. Gaussian moment bounds and the derivative/growth assumptions control the remainder. Taking expectations and adding fh yields (5). □

Actual training compares two networks and a sampled target:

$$u_\theta - E = (u_\theta - u_{\text{old}}) + (u_{\text{old}} - \mathcal{B}_h(u_{\text{old}})) + (\mathcal{B}_h(u_{\text{old}}) - E). \tag{6}$$

The first term is target lag, the second is the population Bellman correction, and the third is sampling error. Target refresh controls the first term; walker count controls the third. Locally, the population variance of a smooth scalar return is

$$s^2 = 2\nu h\|\nabla u_{\text{old}}\|^2 + o(h). \tag{7}$$

Spread is largest where the frozen field has a large spatial gradient. This explains why weighting can help control noisy updates near a layer while also reducing the signal needed to sharpen that layer.

3.2 Separating action noise from target noise

Theorem 1 (Conditional covariance decomposition)**.** *Fix parameters, collocation points, the frozen target construction, and a Gaussian variance $\sigma^2 I$ independent of the mean parameters, with action noise independent of the target. Let J be the Jacobian of μ_θ. For an unnormalized, unclipped score-function loss gradient g_{sf} with finite second moment,*

$$\mathbb{E}[g_{\text{sf}} \mid E] = g_{\text{ex}} = 2J^\top(\mu_\theta - E), \tag{8}$$

$$\text{Cov}(g_{\text{sf}}) = \text{Cov}(g_{\text{ex}}) + \mathbb{E}[\text{Cov}(g_{\text{sf}} \mid E)]. \tag{9}$$

Consequently the covariance difference is positive semidefinite.

Proof. With $\epsilon = a - \mu \sim N(0, \sigma^2 I)$, the score is $J^\top\epsilon/\sigma^2$. Expanding $\|\mu - E + \epsilon\|^2\epsilon$, the constant and cubic odd moments vanish and the cross term gives $2\sigma^2(\mu - E)$. This proves (8). The law of total covariance gives (9). □

Analytic regression removes the conditional action-variance term. Averaging A independent actions divides that term by A, while target variance remains. The theorem applies to the raw quadratic score estimator under its stated fixed-variance assumptions.

Proposition 1 (Scalar score variance)**.** *For $\delta = \mu - E$ and $z \sim N(0, 1)$, the raw loss-gradient estimator has*

$$\mathbb{E}[g] = 2\delta, \qquad \text{Var}(g) = \delta^4/\sigma^2 + 14\delta^2 + 15\sigma^2. \tag{10}$$

Using the two actions $\mu + \sigma z$ and $\mu - \sigma z$ and averaging their raw score estimates instead gives

$$\bar{g} = 2\delta z^2, \qquad \text{Var}(\bar{g}) = 8\delta^2. \tag{11}$$

Proof. The raw estimator equals $\delta^2 z/\sigma + 2\delta z^2 + \sigma z^3$. Use $\mathbb{E}z^2 = 1$, $\mathbb{E}z^4 = 3$, and $\mathbb{E}z^6 = 15$ and subtract the squared mean. Replacing z by $-z$ cancels both odd terms, leaving $2\delta z^2$; $\text{Var}(z^2) = 2$ yields (11). □

Antithetic sampling cancels the odd terms and removes the fixed-σ variance floor as $\delta \to 0$. It is therefore a useful control for the matched estimator study.

3.3 Two distinct objectives for weighting

Write $E_i = \mathcal{B}_i + \eta_i$, with centered noise of covariance Σ_i/M, and define

$$v_i = 2\sqrt{\text{tr}(J_i^\top \Sigma_i J_i)/M}. \tag{12}$$

Here Σ_i is the single-return covariance, and independent returns give the target covariance Σ_i/M. For deterministic weights, $w_i v_i$ is the RMS norm of the pointwise noise gradient. The trace retains output covariance directions. If the population spread is $s_i^2 = \text{tr}\,\Sigma_i$, then $v_i \le 2s_i\|J_i\|_{\text{op}}/\sqrt{M}$.

Theorem 2 (Minimax amplitude versus minimum total variance)**.** *For $v_i > 0$ and weights $w_i \geq 0$ satisfying $\sum_i w_i = B$, the minimizers of $\max_i w_i v_i$ and $\sum_i w_i^2 v_i^2$, respectively, are*

$$w_i^{\mathrm{mm}} = \frac{B/v_i}{\sum_j 1/v_j}, \tag{13}$$

$$w_i^{\mathrm{tv}} = \frac{B/v_i^2}{\sum_j 1/v_j^2}. \tag{14}$$

The second objective represents total batch noise only when cross-point noise covariances vanish.

Proof. If $c = \max_i w_i v_i$, then $B = \sum_i w_i \leq c \sum_i 1/v_i$. Equality is achieved by (13). Cauchy–Schwarz gives $B^2 \leq (\sum_i w_i^2 v_i^2)(\sum_i 1/v_i^2)$, with equality precisely at (14). □

When local Jacobian scales and output covariance shapes are comparable, $v_i \propto s_i$ motivates $1/s_i$ for minimax amplitude and $1/s_i^2$ for total variance. These are noise objectives: the resulting regression error also depends on the signal retained by the weights and the network's capacity.

3.4 Finite-group bias and finite-network projections

Theorem 3 (Plug-in weighting bias)**.** *Condition on parameters and collocation points, and let J_i be fixed, $\mathbb{E}\eta_i = 0$, and w_i be any square-integrable random normalized weight. Then*

$$\|\mathbb{E}[2w_i J_i^\top \eta_i]\| \leq 2\|J_i\|_{\mathrm{op}} \sqrt{\mathrm{Var}(w_i)\mathbb{E}\|\eta_i\|^2}. \tag{15}$$

If the entire vector used to construct and normalize weights is independent of the target samples, this term vanishes.

Proof. Because $\mathbb{E}\eta_i = 0$, replace w_i by $w_i - \mathbb{E}w_i$ in the expectation. Apply the operator-norm inequality and Cauchy–Schwarz. Independence factors the expectation into $\mathbb{E}w_i \mathbb{E}\eta_i = 0$. □

Independent groups remove this covariance term at a sampling cost. The split-budget and double-cost constructions in Section 4 expose that tradeoff.

Proposition 2 (Local curvature and projection shift)**.** *For a fixed linearized least-squares problem, let $H_0 = B^{-1} \sum_i J_i^\top J_i \succ 0$ and $H_w = B^{-1} \sum_i w_i J_i^\top J_i$. If $w_i \geq m > 0$, then $H_w \succeq m H_0$. For unweighted and weighted solutions θ_0, θ_w, with $r_i = b_i - J_i \theta_0$,*

$$\theta_w - \theta_0 = H_w^{-1} B^{-1} \sum_i (w_i - 1) J_i^\top r_i. \tag{16}$$

Proof. The curvature inequality follows termwise. Subtract the two normal equations and use $\sum_i J_i^\top r_i = 0$ to obtain (16). □

The inverse has norm at most $1/(m\lambda_{\min}(H_0))$. Thus a positive floor limits local sensitivity, while the residual term specifies how reweighting moves the fit. For a realizable pointwise target all residuals vanish; with limited capacity they generally do not. The conclusion is local to an identifiable linearization with $H_0 \succ 0$.

4 Bounded shrinkage Bellman regression

The method adapts the strength of weighting to the information in each batch. It uses the return samples already needed for the regression target and adds only moment estimates and scalar operations per collocation point.

4.1 ESTIMATING SPATIAL HETEROGENEITY

Let the empirical return spread and a plain normalized weight be

$$s_i^2 = M^{-1} \sum_m \|G_i^{(m)} - E_i\|^2, \qquad q_i = (s_i + \epsilon)^{-\alpha}, \qquad \widehat{w}_i = q_i / (B^{-1} \sum_j q_j). \tag{17}$$

The exponent $\alpha = 1$ gives inverse spread and $\alpha = 2$ gives inverse variance. The stabilizer $\epsilon > 0$ limits sensitivity to near-zero spread. The empirical second moment uses divisor M, matching the implementation.

To decide whether spread differences are well resolved, compute $\ell_i = \log(s_i + \epsilon)$ and their sample variance $V = (B-1)^{-1} \sum_i (\ell_i - \bar{\ell})^2$. Under scalar Gaussian returns without the stabilizer, the sampling variance of log spread is $\frac{1}{4}\psi_1((M-1)/2)$, where ψ_1 is the trigamma function. We supplement this reference floor with an empirical fourth-moment estimate:

$$F = \max \left\{ \tfrac{1}{4}\psi_1((M-1)/2), \frac{(Q_{.75}(\widehat{\kappa}_i) - 1)_+}{4M} \right\}, \qquad \lambda = \frac{(V - F)_+}{\max(V, 10^{-12})}, \tag{18}$$

$$\widehat{\kappa}_i = \frac{M^{-1} \sum_m \|G_i^{(m)} - E_i\|^4}{\max(s_i^4, 10^{-20})}. \tag{19}$$

The upper quartile $Q_{.75}$ makes the batch floor sensitive to points with heavy-tailed returns. This extension is a moment-based approximation for non-Gaussian and vector returns; its derivation is given in Appendix D.3. If $V \le F$, the gate is zero and the update uses uniform weights. As the resolved heterogeneity increases, λ approaches one and permits stronger spatial weighting.

4.2 RETAINING SIGNAL AT NOISY POINTS

We form the final weights as

$$q_i = \exp[-\alpha\lambda(\ell_i - \bar{\ell})], \qquad \widetilde{w}_i = \mathrm{clip}\left(q_i/\bar{q}, .1, 10\right),$$
$$w_i = m + (1-m)\frac{\widetilde{w}_i}{\bar{\widetilde{w}}}, \qquad 0 < m < 1. \tag{20}$$

The gate, clipping, and floor have distinct roles. The gate responds to uncertainty in the estimated spread; clipping limits relative concentration; the floor keeps every point active in regression. The weights have mean one and satisfy $w_i \ge m$, although the weighted gradient need not have the same magnitude as the unweighted one. The floor is especially useful near steep features, where high return variation and large learning corrections can occur together.

Figure 1 shows the intended tradeoff. Inverse weighting strongly suppresses the high-spread front. Uniform mixing restores part of its contribution, accepting more local noise in exchange for more learning signal. The figure isolates the weighting operation using an exact derivative; training uses empirical returns from the frozen network. The final upper weight can exceed the raw clipping cap after renormalization. Appendix C.1 gives the resulting bounds and describes the role of ϵ.

4.3 SAMPLING AND IMPLEMENTATION

Targets and weights normally share one group of M returns. An independent split-group option allocates $M/2$ returns to the target and $M/2$ to the weights. A double-cost control uses M for each. Independence removes the covariance term in Theorem 3, but splitting increases target variance and makes the spread estimate noisier. We therefore compare both allocations at their recorded costs.

For a shared group, moment estimation costs $O(BMd_u)$ scalar operations for d_u outputs and adds no network evaluations beyond target construction. Independent groups incur the declared return cost. The gate is recomputed for each batch and is detached with the weights. This prevents the network from reducing the loss by differentiating through its uncertainty estimate. The default exponent is one. Development selection compares floors .2 and .5 and the split-group .2 variant, then fixes one configuration across test tasks.

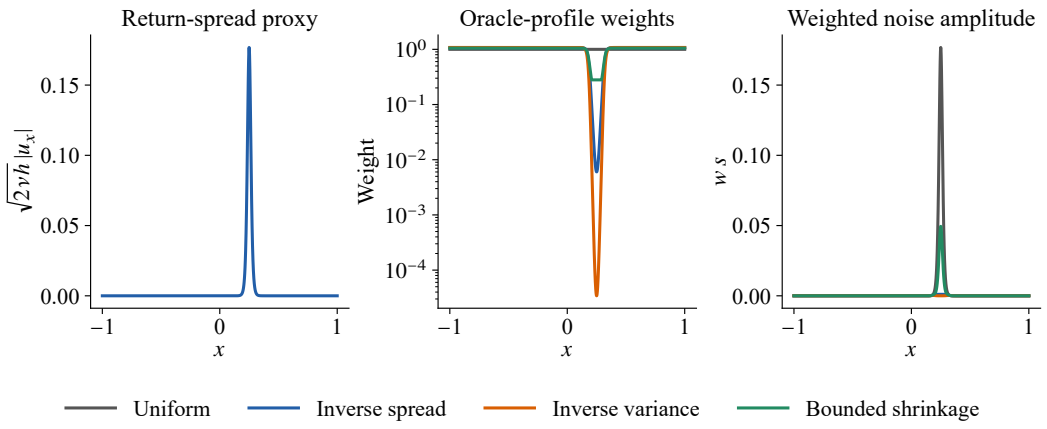

Figure 1: Effect of spatial weighting on an exact Burgers profile. The left panel uses the local spread proxy $\sqrt{2\nu h}|u_x|$; the middle and right panels show normalized weights and weighted noise amplitudes. Bounded shrinkage retains more weight near the layer than plain inverse spread. This analytic illustration uses $\nu = h = .005$, $M = 64$, $\epsilon = .001$, and $m = .2$.

Algorithm 1 Bounded shrinkage Bellman update

1: Sample B collocation points and refresh u_{old} when scheduled.
2: Draw returns using Eqs. (2)–(3) and construct the target mean E_i.
3: Estimate spread and radial shape from the shared or independent weight group.
4: Compute the gate in Eq. (18) using the weight-group size.
5: Form and detach the weights in Eq. (20).
6: Take an optimizer step on Eq. (4).

5 Empirical motivation: spatial weighting near sharp layers

5.1 Aggregate accuracy and run-to-run variation

The completed baseline study compares uniform and inverse-spread regression on four one-dimensional and three oblique Burgers settings, with 40 paired trials per setting. The metric is the maximum error over sampled points in the hard region. Table 1 reports the mean and standard deviation across runs. These experiments evaluate plain inverse spread; the bounded-shrinkage evaluation is specified in Section 6.

Inverse spread reduces the mean in all seven settings. The unrounded aggregates give ratios of 1.74–3.45, or roughly 42–71% lower error, with 234 wins among 280 paired trials. The effect is strongest in the sharpest one-dimensional case: 35 of 40 uniform runs exceed error .5, compared with 2 of 40 inverse-spread runs. These results point to a substantial reliability problem for uniform regression in that setting. The related viscosities and planar oblique solutions share structure, so they characterize behavior within a layer family.

The mean improvement is accompanied by important variation. In a quadratic-flux case at $\nu = .02$, inverse spread reduces the mean from .134 to .088 while increasing the standard deviation from .097 to .193. A weighting rule can therefore improve average accuracy and still worsen some runs. This motivates the uniform floor and the use of upper-tail errors as a separate evaluation target. PINN reference solutions are more accurate on the low-dimensional analytic examples; the contribution sought here is a better use of noisy Bellman targets at a specified cost.

5.2 What the solution profiles reveal

Figure 2 separates front position from transition width and amplitude. Both networks capture the broad traveling-wave shape, but the inverse-spread profile deviates more around the transition in this

Table 1: Uniform and inverse-spread baseline regression: sampled hard-region maximum error, mean ± standard deviation over 40 paired trials. Ratio is uniform mean divided by inverse-spread mean, computed from displayed values.

Case	ν	Uniform	Inverse spread	Ratio
1D	.0200	$.108 \pm .083$	$.049 \pm .040$	2.20
1D	.0100	$.191 \pm .118$	$.091 \pm .029$	2.10
1D	.0050	$.333 \pm .220$	$.171 \pm .106$	1.95
1D	.0025	$.844 \pm .229$	$.244 \pm .104$	3.46
2D	.0100	$.419 \pm .240$	$.169 \pm .053$	2.48
2D	.0050	$.455 \pm .278$	$.262 \pm .078$	1.74
2D	.0025	$1.067 \pm .203$	$.398 \pm .100$	2.68

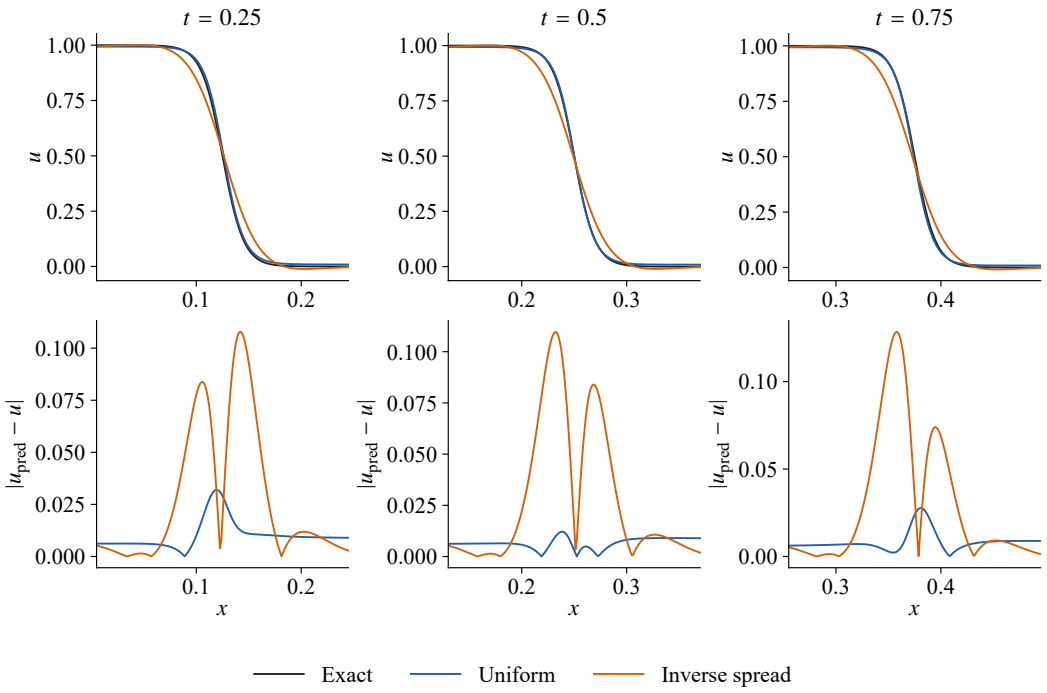

Figure 2: Burgers baseline solutions at $\nu = .005$, seed 0. Top: exact and predicted profiles near the front at three times. Bottom: absolute pointwise errors on the same slices. Uniform regression is more accurate in this fixed example, despite the aggregate inverse-spread advantage in Table 1. Evaluation and checkpoint details are in Appendix A.3.

seed. On full 1,601-point spatial grids, the uniform maxima at $t = .25, .5, .75$ are .03198, .01212, and .02776; the corresponding inverse-spread values are .10781, .10968, and .12844. These fixed-time profiles complement the aggregate space–time samples and expose local errors that a global norm can hide.

The example illustrates the method's central design problem. A point with variable returns may be difficult because it straddles a physically important layer. Suppressing it too strongly can produce a smooth but misplaced fit. The figures alone do not identify the cause of this particular error, but they specify what the component ablations should measure: whether the gate and floor retain useful corrections while reducing unstable updates. The oblique-field plots in Appendix A.4 provide the corresponding spatial view, including a common color scale for the two error fields.

6 EVALUATION OF BOUNDED SHRINKAGE

Evaluation status. The following pre-specified study covers 2,524 runs. Its bounded-shrinkage comparisons and ablations are pending in the available experiment record. The completed measurements in Section 5 are kept separate because their target construction and training budgets differ. Appendix A.1 records the outstanding main-result cells.

6.1 TASKS AND COMPETING METHODS

The eight tasks comprise Burgers at $\nu = .0025$ and $.0042$, an oblique layer at $.003$, a curved layer at $(\nu, \kappa) = (.006, .18)$, a quadratic-flux layer, Taylor–Green at $Re = 60$, periodic two-dimensional transport, and three-dimensional heat flow. The set progresses from narrow-layer geometry to variable drift and coupled-coordinate diffusion. Analytic definitions and reference plots appear in Appendices A and B.

The periodic transport task on $[-\pi, \pi]^2$ uses $\nu = .03$, $T = .5$, and

$$b(x, y) = (\sin y, .7 \sin x), \qquad f = 0, \tag{21}$$

$$u_0(x, y) = e^{8(\cos(x-.6)+\cos(y+.4)-2)} + .6e^{5(\cos(x+1)+\cos(y-.8)-2)}. \tag{22}$$

Its evaluation reference is computed by a dealiased Fourier/RK4 solver with space and time refinement. The heat task uses three Fourier modes spanning three spatial directions and has an exact solution. Both problems use known initial data and zero forcing. Taylor–Green uses a prescribed pressure-gradient source in the velocity equation.

Eight baselines isolate different choices. Uniform, inverse-spread, and inverse-variance regression test the value of the weighting safeguards. Residual weighting and residual-adaptive collocation direct computation toward large Bellman discrepancies. Antithetic walkers reduce sampling variation without changing spatial weights. Standard and residual-adaptive PINNs provide strong-form solver references. The adaptive regression baseline resamples from a $4B$ candidate pool every 100 updates and includes its extra walker calls in the cost. Its implementation is residual-based sampling, distinct from the trajectory-sampling algorithm of Chen et al. (2023).

6.2 BUDGETS, SELECTION, AND METRICS

All regression methods use three 128-unit tanh hidden layers and Adam. Scalar tasks use 4,800 updates with batch 1,024; the velocity task uses 9,600 updates with batch 512. PINNs receive 15,000 updates. The default walker count is 64, time step .005, and stabilizer .001. The principal methods receive the same development search over learning rates $.0003, .001, .003$. The main comparison reports accuracy together with measured cost; a separate 900-second study compares equal-time endpoints.

Configuration selection uses the mean log error ratio across three development tasks and one global learning rate per principal method. A bounded-shrinkage configuration is eligible when its mean error ratio to uniform is at most 1.1 on every development task and its geometric aggregate is below one. If none passes, the lowest-scoring configuration is evaluated with that outcome recorded. Development and confirmation use disjoint seeds. Appendix E.2 gives the pairing and inference rules.

Layer tasks use the refined-grid hard-region maximum as their primary metric; other tasks use relative L^2. Grids increase from 513 to 2049 points in one dimension, 65^2 to 129^2 in two, and 17^3 to 33^3 in three, at three evaluation times. A separate 20,000-point assessment supports comparison to sample-based measurements. Every cell reports completion count, mean, standard deviation, median, 90th percentile, paired wins, elapsed time, and peak memory. Failure-threshold curves describe reliability over a range of error tolerances.

6.3 MECHANISM TESTS AND COMPONENT ABLATIONS

At fixed development checkpoints, 16,384 independent walkers define a reference Bellman target. Independent groups of 64 estimate target covariance, gradient noise, and changes in update direction under weighting.

Table 2: Pre-specified evaluation matrix. Counts are planned independent runs.

Study	Runs	Main question
Development and rate selection	324	One configuration across tasks
Main comparisons	1350	Accuracy against eight baselines
Component ablations	360	Gate, clipping, floor, sampling groups
Sensitivity	300	Walker count, step, stabilizer, refresh
Matched estimators	100	Action-noise contribution
Fixed-time comparisons	70	Error after 900 seconds
Network capacity	20	Widths 64 and 256
Total	2524	

The ablations remove the gate, clipping, or floor one at a time. Shared, split, and full independent groups distinguish dependence from sampling cost; shuffled inverse-spread weights test spatial assignment. All use matched seeds and learning rate 10^{-3} (Appendix E.1).

The estimator study fixes batches, walker count, target refresh, and updates while comparing exact regression with raw, centered, antithetic, and group-normalized score updates. A four-epoch clipped GRPO control with learned bounded action scale enters the fixed-time study, which also tests its other optimization choices.

Paired inference combines a practical effect threshold, bootstrap intervals, and multiplicity-adjusted sign tests (Appendix E.2).

7 Related work

Stochastic neural PDE solvers. DFLM learns a solution through stochastic representations rather than a strong-form residual (Han et al., 2020). Martingale and reinforcement-learning formulations extend it to boundary-value problems and viscous flow (Cai et al., 2023; Park & Stinchcombe, 2022). Time-step and walker-size analysis (Han & Lee, 2025) motivates our focus on how sampled targets enter a shared regression update.

Adaptive sampling and loss weighting. Adaptive trajectory sampling changes where stochastic PDE information is collected (Chen et al., 2023). Inverse-variance regression uses target uncertainty (Mai et al., 2021); inverse-Dirichlet weighting balances gradient scales in PINN objectives (Maddu et al., 2022). PINNs use strong-form residuals (Raissi et al., 2019), with adaptive methods for uneven training dynamics (Wang et al., 2021; McClenny & Braga-Neto, 2023). Our gate tests whether observed return heterogeneity justifies spatial reweighting, while a uniform component retains signal near high-spread solution features.

Monte Carlo gradient estimation. Conditional expectation and pathwise differentiation are standard ways to reduce Monte Carlo gradient variance (Mohamed et al., 2020). For the quadratic objective they yield the exact regression gradient. Group-relative policy optimization adds normalization and clipping (Shao et al., 2024), which require separate controls beyond the raw-score covariance theorem.

8 Discussion and conclusion

Spatial weighting reallocates noise and signal across a shared network. Bounded shrinkage combines a finite-group gate, clipping, and a uniform floor to limit concentration while retaining corrections near steep layers. Independent groups address weight–target covariance. The completed inverse-spread baselines motivate testing whether these safeguards improve accuracy and reliability at fixed cost; that comparison remains pending.

REPRODUCIBILITY STATEMENT

Algorithm 1 specifies the update. The appendices provide problem definitions, numerical-reference checks, evaluation grids, ablation configurations, and paired inference rules. Appendix E.3 describes the construction and checks of the reference and checkpoint figures.

AI USE STATEMENT

OpenAI Codex was used for coding assistance and language editing. The authors reviewed all AI-assisted content and take full responsibility for the manuscript, implementation, and reported results.

REFERENCES

Wei Cai, Andrew He, and Daniel Margolis. DeepMartNet—a martingale based deep neural network learning method for dirichlet BVPs and eigenvalue problems of elliptic PDEs in $\mathbb{R}^d$. *arXiv preprint arXiv:2311.09456*, 2023. URL https://arxiv.org/abs/2311.09456.

Xingyu Chen, Jianhuan Cen, and Qingsong Zou. Adaptive trajectories sampling for solving pdes with deep learning methods. *arXiv preprint arXiv:2303.15704*, 2023. URL https://arxiv.org/abs/2303.15704.

Jihun Han and Yoonsang Lee. An analysis of the derivative-free loss method for solving pdes. *arXiv preprint arXiv:2309.16829*, 2025. URL https://arxiv.org/abs/2309.16829.

Jihun Han, Mihai Nica, and Adam R. Stinchcombe. A derivative-free method for solving elliptic partial differential equations with deep neural networks. *Journal of Computational Physics*, 419: 109672, 2020. doi: 10.1016/j.jcp.2020.109672.

Suryanarayana Maddu, Dominik Sturm, Christian L. Müller, and Ivo F. Sbalzarini. Inverse-dirichlet weighting enables reliable training of physics informed neural networks. *Machine Learning: Science and Technology*, 3(1):015026, 2022. doi: 10.1088/2632-2153/ac3712.

Vincent Mai, Waleed Khamies, and Liam Paull. Batch inverse-variance weighting: Deep heteroscedastic regression. *arXiv preprint arXiv:2107.04497*, 2021. URL https://arxiv.org/abs/2107.04497.

Levi D. McClenny and Ulisses M. Braga-Neto. Self-adaptive physics-informed neural networks. *Journal of Computational Physics*, 474:111722, 2023. doi: 10.1016/j.jcp.2022.111722.

Shakir Mohamed, Mihaela Rosca, Michael Figurnov, and Andriy Mnih. Monte carlo gradient estimation in machine learning. *Journal of Machine Learning Research*, 21(132):1–62, 2020. URL https://jmlr.org/papers/v21/19-346.html.

Kevin Min Seong Park and Adam R. Stinchcombe. Deep reinforcement learning of viscous incompressible flow. *Journal of Computational Physics*, 467:111455, 2022. doi: 10.1016/j.jcp.2022.111455.

Maziar Raissi, Paris Perdikaris, and George Em 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.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. URL https://arxiv.org/abs/2402.03300.

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.

A BENCHMARK REFERENCES AND BASELINE EVIDENCE

A.1 MAIN-COMPARISON COMPLETION

The table records completed/expected seeds for each method–task pair in the pre-specified main comparison. Budgets are specified in Section 6.

Method	B1	B2	O	C	Q	TG	T	H
Uniform	0/20	0/20	0/20	0/20	0/20	0/10	0/20	0/20
Inverse spread	0/20	0/20	0/20	0/20	0/20	0/10	0/20	0/20
Inverse variance	0/20	0/20	0/20	0/20	0/20	0/10	0/20	0/20
Residual weighting	0/20	0/20	0/20	0/20	0/20	0/10	0/20	0/20
Adaptive collocation	0/20	0/20	0/20	0/20	0/20	0/10	0/20	0/20
Antithetic walkers	0/20	0/20	0/20	0/20	0/20	0/10	0/20	0/20
PINN	0/20	0/20	0/20	0/20	0/20	0/10	0/20	0/20
Adaptive PINN	0/20	0/20	0/20	0/20	0/20	0/10	0/20	0/20
Bounded shrinkage	0/20	0/20	0/20	0/20	0/20	0/10	0/20	0/20

B1: Burgers, $\nu = .0025$; B2: Burgers, $\nu = .0042$; O: oblique layer; C: curved layer; Q: quadratic flux; TG: Taylor–Green; T: periodic transport; H: three-dimensional heat flow.

A.2 EXACT TRAVELING-LAYER SOLUTIONS

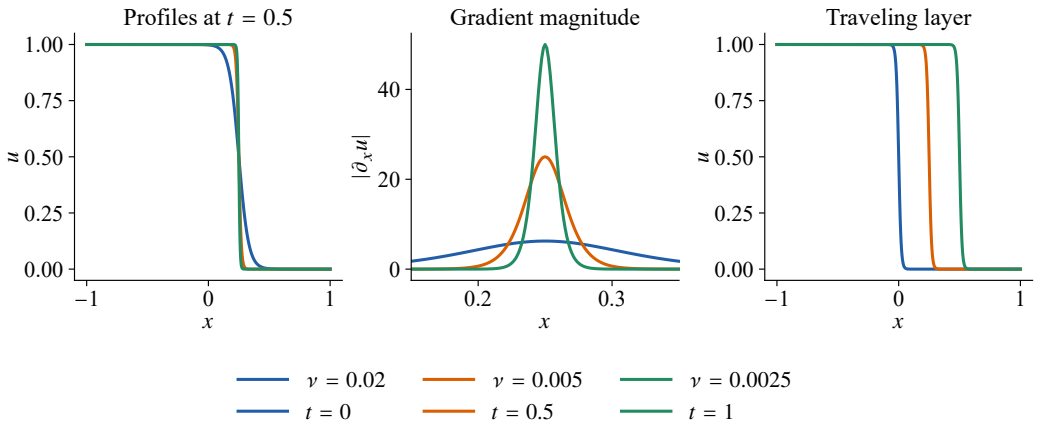

Figure 3: Analytic Burgers reference solutions. Left: viscosity controls layer width. Middle: the corresponding derivative magnitude. Right: the layer translates at speed $c = .5$. All panels use $a = .5$.

The steep-layer setting gives a concrete meaning to hard-region accuracy. For the one-dimensional Burgers equation, define the traveling coordinate $r = x - ct$ and the dimensionless coordinate $z = ar/(2\nu)$. The exact solution and its first two derivatives are

$$u(x,t) = c - a\tanh z, \qquad u_x = -\frac{a^2}{2\nu}\operatorname{sech}^2 z, \tag{23}$$

$$u_{xx} = \frac{a^3}{2\nu^2}\operatorname{sech}^2 z\tanh z, \qquad u_t = -cu_x. \tag{24}$$

Substitution gives $u_t + uu_x = (u-c)u_x = \nu u_{xx}$, so the plotted profile satisfies the PDE without a manufactured source. The initial condition and the Dirichlet boundary values are obtained by evaluating this formula at the initial time and the spatial endpoints. The network receives those prescribed data through the original training construction, rather than interior labels.

Figure 3 shows that decreasing viscosity does not increase the solution's range: its two limiting values remain $c - a$ and $c + a$. Instead, the transition concentrates over a distance proportional to

ν/a. The maximum derivative is $a^2/(2\nu)$, so the same displacement of a learned front can produce very different pointwise errors at different viscosities. For a small displacement ε_x, a local first-order estimate is

$$|u(x+\varepsilon_x,t)-u(x,t)| \simeq |\varepsilon_x|\frac{a^2}{2\nu}\operatorname{sech}^2 z. \tag{25}$$

This expansion describes a small displacement of the exact profile. A learned solution can also have the wrong amplitude, a broad transition, or oscillations away from the layer.

These geometric facts guide reporting. Whole-domain averages can be small even when the transition is misplaced because the steep set occupies little volume. Conversely, a layer maximum is sensitive to evaluation resolution. The evaluation therefore reports both global error and a refined-grid layer metric. The profile width should be compared with the grid spacing before interpreting an apparent improvement in a heat map. The reference profiles give the solution and error plots a common geometric interpretation.

A.3 Evaluation of the baseline checkpoints

The solution plots use a fixed seed pair for each scalar problem. Seed 0 was chosen before viewing the predictions. The uniform and inverse-spread fields are reconstructed from the saved network parameters. Figure 2 in the main text shows the one-dimensional comparison; Figure 4 gives the spatial comparison for the oblique problem.

Network and training configuration. Both problems use a tanh MLP with three hidden layers of width 128 and a scalar output. The inputs are (x,t) for Burgers and (x,y,t) for the oblique field. The baseline configurations specify 64 walkers per point, a time step of .005, batch size 2,048, learning rate .001, and 16×300 updates. Inverse-spread weights use stabilizer .001. These configurations belong to the completed baseline study and are not pooled with the budgets in Section 6.

Reconstructing the fields. For each saved affine layer, evaluation uses $h_{k+1}=\tanh(W_k h_k+b_k)$; the final layer is linear. Saved float32 tensors are converted to float64 for evaluation. The storage loader accepts only the tensor metadata and numeric storage types needed by these checkpoints. An independent scalar-loop forward calculation at fixed coordinates checks the vectorized matrix implementation.

For the one-dimensional field, 1,601 equally spaced points cover $[-1,1]$ at $t=.25,.5,.75$. A plotting window centered at ct exposes the transition; the reported maximum is evaluated on the entire grid. For the oblique field, both networks and the reference are evaluated on the same 161×161 grid at $t=.5$. All solution panels share one color range and both error panels share a second range. The $y=0$ slice is additional to the field calculation.

Relationship to the reported error metrics. The plotted slices diagnose shape at fixed times. The aggregate baseline metrics sample space–time points and summarize repeated runs. Their maxima need not agree: they cover different sets, and a maximum is sensitive to whether a narrow error peak is sampled. The fixed-grid plots are therefore reported with their own times and resolutions. They are used to inspect front location, transition width, and local error rather than to replace the paired-run distribution in Table 1.

A.4 Oblique-layer fields and error localization

For $\beta=(1/2,1/2)$ and $q=\beta\cdot x-ct$, the reference field is

$$u(x,y,t)=c-a\tanh\left(\frac{aq}{2\nu\|\beta\|^2}\right), \qquad \|\beta\|^2=\tfrac{1}{2}. \tag{26}$$

Because $\beta_x+\beta_y=1$, substitution reduces $u_t+u(u_x+u_y)=\nu(u_{xx}+u_{yy})$ to a traveling-wave equation in q. The field is evaluated in two spatial dimensions but depends on one combined coordinate. Thus it tests oblique geometry and spatial sampling, not unrestricted two-dimensional solution complexity.

For the fixed seed-0 pair, a common 161×161 grid gives maxima 0.10595 and 0.26743, respectively. Fixed color limits prevent a small error field from appearing as intense as a large one merely through

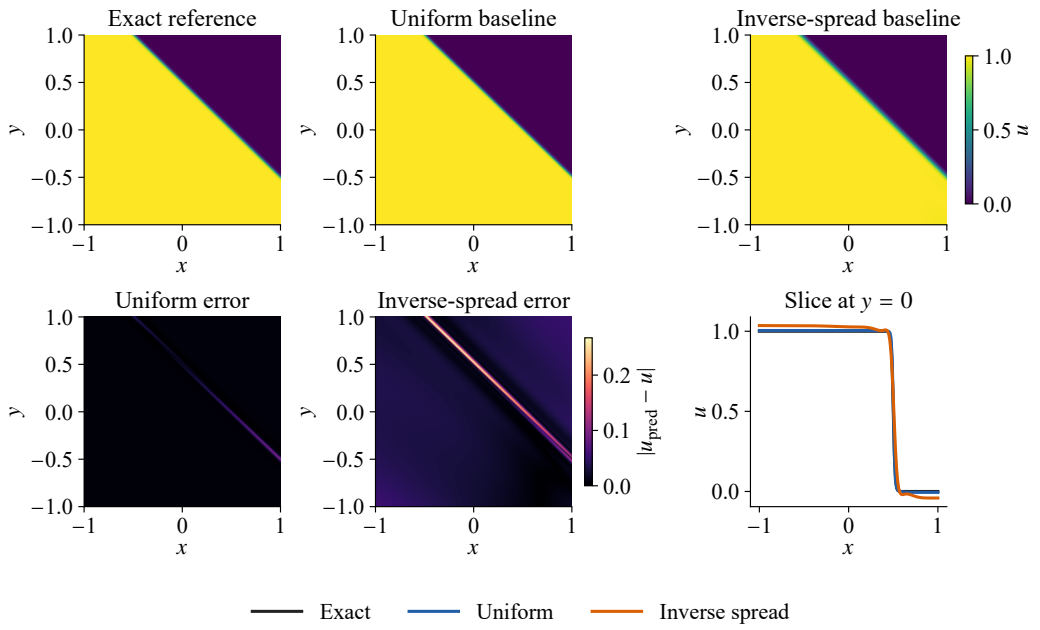

Figure 4: Oblique-layer baseline checkpoints, seed 0, $\nu = .005$, $t = .5$. Solution panels share the same value scale and error panels share the same error scale. The final panel cuts across the layer at $y = 0$. The fields are reconstructed from the baseline checkpoints.

rescaling. The chosen slice supplements the full-field plots; it cannot establish the maximum over all spatial points or times. The main evaluation uses $\nu = .003$; the plotted baseline case uses $\nu = .005$. Uniform regression has the lower grid maximum in this example.

B Reference fields and problem specifications

B.1 A curved manufactured layer and its prescribed forcing

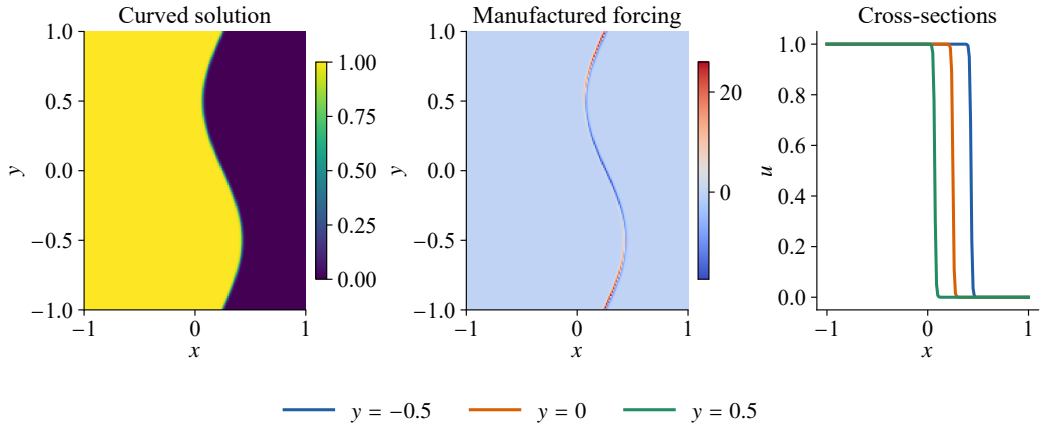

Figure 5: Analytic curved-layer reference at $t = .5$, $\nu = .006$, and $\kappa = .18$. The middle panel is the prescribed source that makes the curved profile an exact manufactured solution. The panels specify the benchmark field and source.

Curvature changes the layer's orientation across the domain and separates geometric difficulty from simply rotating a one-dimensional front. Define $q = x + \kappa \sin(\pi y) - ct$, $\alpha = a/\nu$, and $U(q) = c - a\tanh(\alpha q)$. The chain rule gives

$$q_y = \kappa\pi\cos(\pi y), \qquad q_{yy} = -\kappa\pi^2 \sin(\pi y), \tag{27}$$

$$u_x = U', \quad u_y = q_y U', \qquad \Delta u = (1 + q_y^2)U'' + q_{yy}U'. \tag{28}$$

Consequently the source used with scalar drift (u, u) is

$$f = \big[-c + U(1 + q_y)\big]U' - \nu\big[(1 + q_y^2)U'' + q_{yy}U'\big]. \tag{29}$$

Here $U' = -a\alpha\,\mathrm{sech}^2(\alpha q)$ and $U'' = 2a\alpha^2\,\mathrm{sech}^2(\alpha q)\tanh(\alpha q)$. The same exact field supplies the initial and boundary data. These identities explain every quantity in Figure 5; the source is part of the specified PDE, not an output inferred by the solver.

In the curved case, the transition shifts with y and its derivative geometry changes across the domain. A shared network must fit that geometry, while residual-adaptive sampling and return-noise weighting may allocate points differently. The prescribed source makes this a manufactured-solution test; periodic transport adds an evolution determined from initial data.

The large source variations in the middle panel are concentrated near the layer. They show why reporting only a visually plausible solution is inadequate: small geometric errors can interact with strong local derivatives. Competing methods receive the same source, initial and boundary data, evaluation grid, and model capacity. The field and cross-sections expose the geometric challenge.

B.2 Three-dimensional heat flow with coupled Fourier modes

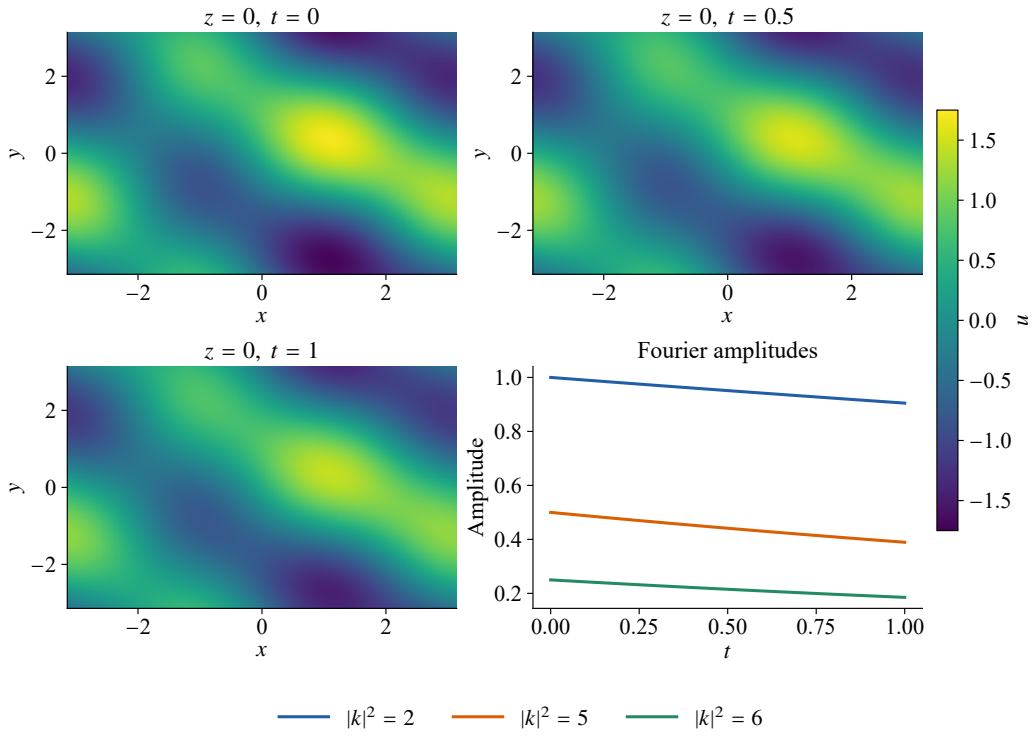

Figure 6: Exact heat-equation solution at $z = 0$ for $\nu = .05$ and three times. The last panel displays the exact amplitude decay of the three Fourier modes. The snapshots are two-dimensional slices of a three-dimensional field.

The zero-forcing periodic heat task uses the explicit solution

$$u(x, y, z, t) = e^{-2\nu t} \sin(x + y) + \tfrac{1}{2} e^{-5\nu t} \cos(y + 2z) + \tfrac{1}{4} e^{-6\nu t} \sin(2x - y + z). \tag{30}$$

Each mode has Laplacian eigenvalue $-\|k\|^2$, with $k = (1, 1, 0), (0, 1, 2), (2, -1, 1)$, respectively. Differentiating the exponential factor therefore verifies $u_t = \nu \Delta u$ term by term. The initial value is obtained at $t = 0$, and periodicity follows from integer wave numbers.

The three wave vectors are linearly independent. Higher-frequency modes decay more quickly, changing the shape as well as the amplitude. This provides a smooth comparison to the steep Burgers cases without a manufactured interior source. Figure 6 shows a $z = 0$ slice; the error metric uses the full volume grids and time slices specified in the main evaluation.

B.3 Taylor–Green velocity with the pressure source made explicit

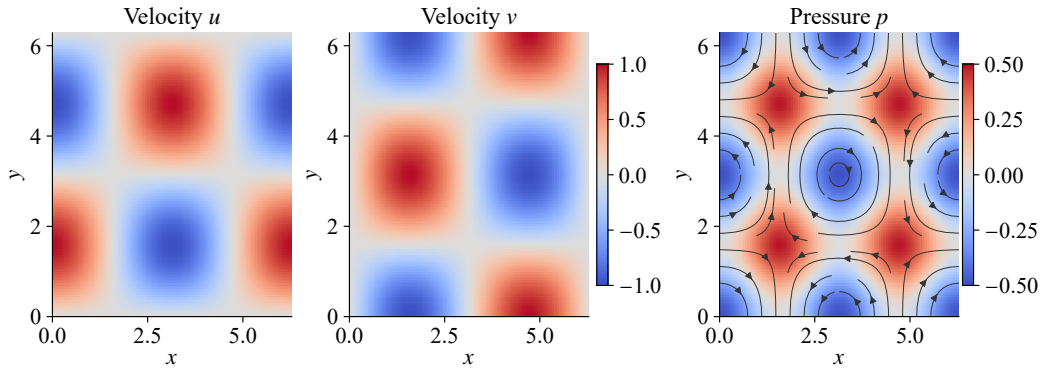

Figure 7: Exact Taylor–Green reference at $Re = 60$ and $t = .5$. The velocity components share one color scale; pressure has a separate scale. Streamlines in the last panel indicate velocity direction. This is a known-pressure test.

For $F(t) = e^{-2\nu t}$ on a periodic square, the reference is

$$u = \cos x \sin y \, F, \qquad v = -\sin x \cos y \, F, \tag{31}$$

$$p = -\tfrac{1}{4}(\cos 2x + \cos 2y) F^2, \qquad \psi = -\cos x \cos y \, F. \tag{32}$$

The streamfunction convention $u = \partial_y \psi$, $v = -\partial_x \psi$ enforces zero divergence. The velocity solves the momentum equation only when the pressure contribution is included. In the componentwise Bellman formulation, it enters as the prescribed source

$$f = -\nabla p = -\tfrac{1}{2} F^2 (\sin 2x, \sin 2y). \tag{33}$$

Indeed $u_t = \nu \Delta u$ and $v_t = \nu \Delta v$, while the convective terms equal the two components of $-\nabla p$. The Bellman return therefore includes Eq. (33). Pressure is prescribed in this velocity benchmark.

The rotating velocity directions make this a useful test of the scalar spread approximation. Two points with equal total spread can have different covariance directions relative to the network Jacobian. Equation (12) retains this directional information. Comparing that expression with the scalar proxy quantifies what is lost when a vector return is summarized by one number.

B.4 PERIODIC TRANSPORT: AN INDEPENDENTLY COMPUTED NUMERICAL REFERENCE

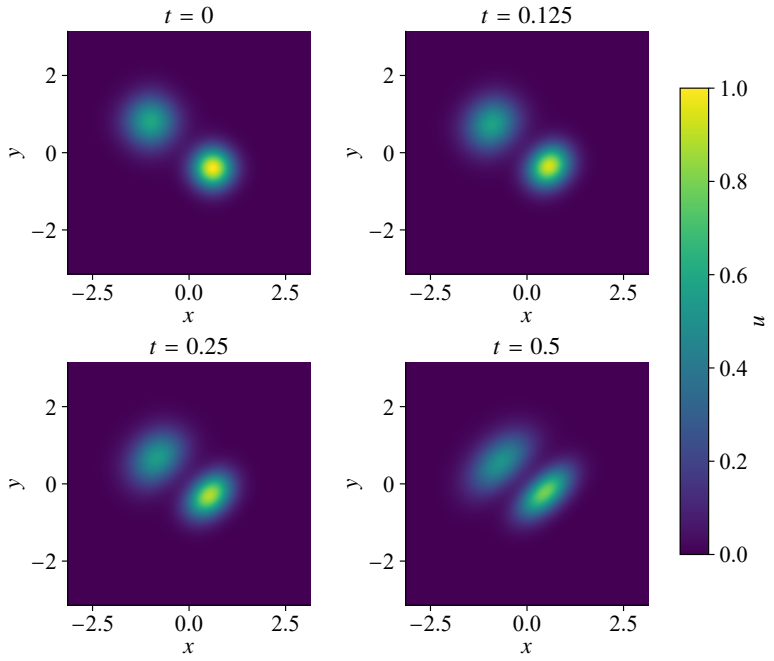

Figure 8: Initial condition and numerical reference evolution for periodic transport, $\nu = .03$. A dealiased Fourier discretization and RK4 generate the snapshots. All panels are outputs of the reference solver.

The task uses the drift and initial condition of Eqs. (21)–(22), with no interior forcing. For a periodic grid, the Fourier-space evolution is

$$\partial_t \widehat{u}_k = -\widehat{b \cdot \nabla u}_k - \nu \|k\|^2 \widehat{u}_k. \tag{34}$$

The plotted reference retains advective modes below the two-thirds cutoff and uses time steps $\min(.001, .25/(.03n^2))$ on n^2 grids. Both space and time are refined. Maximum discrepancies on nested grid points, over all three positive time snapshots, are 5.272×10^{-11} for $64 \to 128$ and 4.330×10^{-14} for $128 \to 256$. The latter meets the declared 2×10^{-5} reference tolerance. This check measures successive-grid consistency; it is not a rigorous bound on the exact PDE error.

Transport deforms the two initial peaks while diffusion smooths them. Unlike a prescribed traveling profile, this field is computed from its initial data. The reference field is used for visualization and error evaluation, separately from training.

B.5 QUADRATIC-FLUX LAYER

The remaining layer benchmark uses $q = \beta_x x + \beta_y y - ct$, $U(q) = .5 - a \tanh(q/d)$, and drift (u^2, u^2). For the evaluation configuration, $(\beta_x, \beta_y) = (.4, .6)$, $d = .045$, $c = .4$, $a = .5$, and $\nu = .014$. The prescribed source is

$$f = \big[-c + U^2(\beta_x + \beta_y) \big] U' - \nu(\beta_x^2 + \beta_y^2) U''. \tag{35}$$

This isolates the effect of a nonlinear, value-dependent drift on a manufactured layer. Its source and initial and boundary values are problem data. The baseline study showed an increase in standard deviation, motivating the paired reliability analysis.

C WEIGHTING AND ESTIMATOR ANALYSIS

C.1 WEIGHT BOUNDS, SCALE DEPENDENCE, AND EFFECTIVE CONCENTRATION

The following bounds complement the update equations in the main text. Let r_i denote the clipped weights, with $l \le r_i \le c$, and let $\bar{r}$ be their batch mean. The final weights satisfy

$$m + (1-m)\frac{l}{c} \le w_i = m + (1-m)\frac{r_i}{\bar{r}} \le m + (1-m)\frac{c}{l}. \tag{36}$$

This follows from $l \le \bar{r} \le c$. The bounds are conservative, but they make clear that the final upper weight is not the clipping cap c. With $l = .1$, $c = 10$, and $m = .2$, the bound is $w_i \le 80.2$. Repeated-group measurements will show how loose this bound is in practice.

Concentration of the weighted batch. Because $\sum_i w_i = B$, a descriptive measure of concentration is

$$B_{\text{eff}} = \frac{(\sum_i w_i)^2}{\sum_i w_i^2} = \frac{B^2}{\sum_i w_i^2}. \tag{37}$$

It equals B for uniform weights and decreases as the weights concentrate. It is not an exact number of independent observations for a neural gradient, whose residuals and Jacobians differ across points. Writing $v_i = r_i/\bar{r}$ and using $\bar{v} = 1$ gives

$$\frac{1}{B}\sum_i w_i^2 = 1 + (1-m)^2\left(\frac{1}{B}\sum_i v_i^2 - 1\right). \tag{38}$$

Uniform mixing therefore contracts weight dispersion by $(1-m)^2$. This identity quantifies the floor's effect on concentration independently of any assumption about the PDE error.

The stabilizer and the units of the output. Multiplying every spread by a constant leaves normalized inverse-spread weights unchanged when $\epsilon = 0$. With a fixed nonzero stabilizer, this invariance holds only if ϵ is scaled by the same constant. In implementation, the dimensionless ratio s_i/ϵ determines whether a point lies in the stabilized regime. When $s_i \ll \epsilon$, the raw weight is nearly constant; when $s_i \gg \epsilon$, it approaches the power law. The relative-stabilizer sensitivity setting scales ϵ by the median spread to test dependence on output units.

Reading the mechanism figure. Figure 1 evaluates weights on an exact reference profile and uses the Gaussian floor because no empirical fourth moments are supplied. The illustration uses a population variance over its grid, whereas training uses a sample variance across collocation points. It isolates normalization and mixing; gate values, weight extrema, and gradient directions will be measured in the planned comparison.

C.2 CONDITIONAL GRADIENT VARIANCE AND THE LIMITS OF ESTIMATOR ARGUMENTS

The scalar identities in Proposition 1 determine the curves in Figure 9. This section examines the action-scale dependence and the distinction between conditional and total variance.

The raw variance contains opposing effects of the action scale. With a fixed nonzero mismatch, the δ^4/σ^2 term grows as exploration shrinks; large exploration increases $15\sigma^2$. Minimizing over $\sigma > 0$ gives

$$\sigma^2_{\text{opt}} = \frac{\delta^2}{\sqrt{15}}, \qquad \min_{\sigma>0} \text{Var}(g) = (14 + 2\sqrt{15})\delta^2, \quad \delta \neq 0. \tag{39}$$

This is a conditional identity for the raw scalar estimator. It does not prescribe a globally optimal learned policy variance or compare clipped and group-normalized objectives. At $\delta = 0$, the infimum is approached as $\sigma \to 0$ rather than attained by a positive scale.

Computing 2δ directly removes the action draw, while the target E remains random across walker groups. The matched estimator study fixes target generation, collocation points, refresh intervals, and update budgets to isolate this distinction during optimization. The independent-target measurement measures the estimators' conditional distributions.

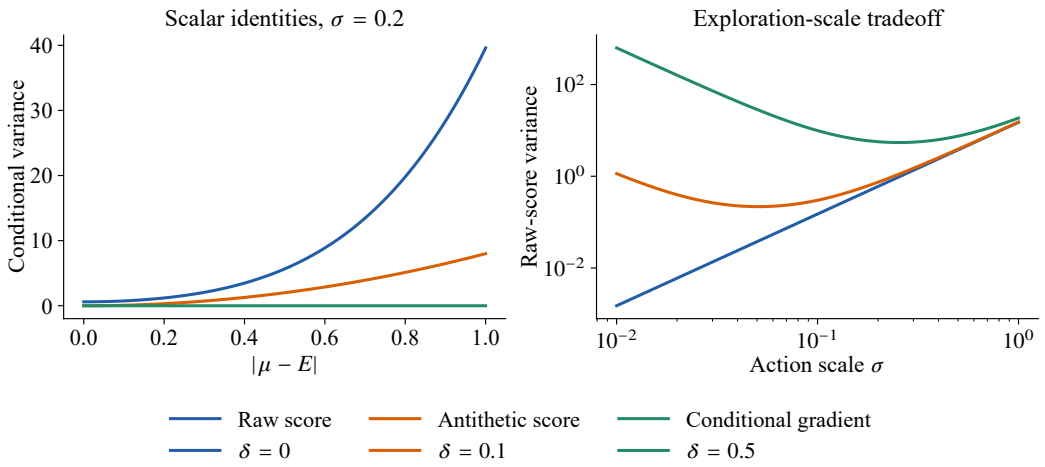

Figure 9: Exact scalar conditional variance identities. Left: target mismatch varies at fixed action scale. Right: action scale varies at fixed mismatch. The zero line is variance over action draws at a fixed target, not total training variance. The curves are computed from the variance identities.

D Error accounting and sensitivity

D.1 Sensitivity settings and measurement priorities

Sensitivity is evaluated on the Burgers task at $\nu = .0042$ and the curved task at $(\nu, \kappa) = (.006, .18)$. Uniform, inverse-spread, and the selected bounded-shrinkage method share five seeds per setting. Ten settings give $2 \times 3 \times 5 \times 10 = 300$ runs. The central configuration is the one used for the main comparison.

Factor	Values or change from the central setting
Walker count M	16, 32, or 128
Time step h	.00125 or .0025
Stabilizer ϵ	10^{-4} or 10^{-2}
Relative stabilizer	ϵ scaled by median return spread
Target refresh	Every ten updates
Coupled stress test	$M = 16$ and $h = .00125$ together

Sampling resolution. Changing M affects two estimates: the target mean and the uncertainty used to weight it. More walkers reduce target variance, but also change the log-spread noise floor and hence the gate. The analysis should therefore report gate values and empirical spread dispersion alongside solution error. A gain that appears only at large M may depend on accurate uncertainty estimates; a gain at small M may indicate that shrinkage successfully avoids overreacting to noisy spread differences.

Time step and target refresh. The time step changes the size of both the Bellman correction and the random displacement. Target refresh changes how stale the source network becomes during optimization. They are separate controls: reducing one does not substitute for changing the other. The coupled small-step, small-group setting tests a demanding combination suggested by the local signal-to-noise scaling in Appendix D.2.

Stabilization and model capacity. The absolute and relative stabilizer settings test whether the weights depend strongly on a particular return scale. A separate capacity comparison uses widths 64 and 256 for uniform and bounded shrinkage on the Burgers task, with five seeds per cell. The purpose is to distinguish a sampling benefit from a change in the finite-network projection. Increasing

width changes the Jacobian and the representable solution class, so its results are reported separately from the one-factor sampling sensitivities.

For each setting, the same solution metrics, completion accounting, and time measurements are retained. The settings diagnose mechanisms after configuration selection; they are not an additional search over test performance.

D.2 Target lag, discretization, sampling, and fitting

Equation (6) separates target lag, the population Bellman correction, and sampling error. This section describes the numerical approximations inside the population target and the measurements used to distinguish them. The continuous stochastic representation and its Euler approximation differ through drift freezing, time discretization, and boundary handling. Their errors remain even when the return mean is evaluated exactly.

Why a smaller step can be harder to learn. Away from boundary and initial-time interactions, a smooth local expansion gives a useful Bellman correction of order h, while the standard deviation of one scalar return is approximately $\sqrt{2\nu h}|\nabla u|$. For M independent walkers, the target standard error is approximately

$$\mathrm{SE}(E) \simeq \sqrt{\frac{2\nu h}{M}}|\nabla u|. \tag{40}$$

Writing the local residual magnitude as $|r|$, the associated signal-to-noise ratio scales as

$$\frac{h|r|}{\mathrm{SE}(E)} \simeq \sqrt{\frac{Mh}{2\nu}}\frac{|r|}{|\nabla u|}. \tag{41}$$

This expression assumes a smooth nonzero gradient and a fixed residual scale. It explains why reducing discretization bias by decreasing h can also make a correction harder to distinguish from noise. It does not imply that training always follows this scaling: target lag can dominate, and steep layers can invalidate a uniform local expansion. The sensitivity matrix varies M and h separately and includes a coupled small-M, small-h setting.

Initial-time and boundary handling. The implemented transient step uses $h_i = \min(h, t_i)$ before drawing the diffusive increment. Otherwise, drawing a full-step displacement and only clipping the final time would mismatch the time and spatial distributions. For periodic domains, wrapping maps endpoints into the same physical cell. For nonperiodic domains, the straight segment to the sampled endpoint provides an approximate first-exit location and a shortened source duration. A Brownian path can cross and return within one step, so this segment rule is not an exact first-passage treatment. Spatial weighting cannot correct that approximation.

What each measurement can resolve. Repeated independent return groups at fixed points estimate sampling variance. Fresh versus frozen target predictions diagnose lag. Refining h while recording the walker cost probes time-step effects. Comparing global and hard-region errors detects fitting tradeoffs. None of these measurements alone isolates every source of error in Eq. (6); their role is to prevent a favorable training-loss curve from being mistaken for verified PDE accuracy.

D.3 Sampling uncertainty of log spread

The Gaussian reference calculation. Consider M independent scalar Gaussian returns with variance σ_G^2. For the empirical second moment $s^2 = M^{-1}\sum_m (G_m - \bar{G})^2$,

$$\frac{Ms^2}{\sigma_G^2} \sim \chi^2_{M-1}. \tag{42}$$

If $Z \sim \chi^2_k$, its Mellin transform is $\mathbb{E}Z^t = 2^t\Gamma(k/2+t)/\Gamma(k/2)$. Differentiating its log twice at $t=0$ gives $\mathrm{Var}(\log Z) = \psi_1(k/2)$. Since $\log s = \log\sigma_G - \frac{1}{2}\log M + \frac{1}{2}\log Z$,

$$\mathrm{Var}(\log s) = \tfrac{1}{4}\psi_1((M-1)/2). \tag{43}$$

Using $M-1$ rather than M in the sample variance only adds a constant to $\log s$ and leaves this variance unchanged. Adding ϵ inside the logarithm changes the distribution; the formula then serves as a reference floor rather than an exact sampling variance.

A fourth-moment approximation. For centered scalar returns, let $\kappa = \mathbb{E}(G-\mu)^4/\sigma_G^4$ be kurtosis. At leading order in M^{-1}, the variance of the sample second moment is $(\kappa - 1)\sigma_G^4/M$. Applying the delta method to $\log s = \frac{1}{2}\log s^2$ gives

$$\mathrm{Var}(\log s) \simeq \frac{\kappa - 1}{4M}. \tag{44}$$

For vector returns, the implemented radial ratio replaces scalar squared deviations by squared Euclidean norms. This keeps the same computational form but does not recover the full covariance geometry. The maximum of the Gaussian floor and the upper-quartile moment estimate favors stronger shrinkage when some points have heavy-tailed or boundary-mixture returns. The quality of this approximation will be assessed by the gate ablation and repeated-group tests.

When same-group weighting has no dependence bias. For independent Gaussian groups at each point, a sample mean is independent of its centered residuals. Any weight constructed solely from those residuals, including their spread and fourth moment, is consequently independent of the target mean. If the point groups are also mutually independent, this remains true after batch normalization of the weight vector. Thus same-group weighting need not introduce a covariance bias in the Gaussian case. Non-Gaussian returns, boundary mixtures, or other dependence can break this property, which is why the general bound and independent-group controls are useful.

Splitting a total budget M into equal target and weight groups removes dependence without assuming Gaussian returns. It also changes the target variance from σ_G^2/M to $2\sigma_G^2/M$ and increases the uncertainty of the spread estimate. The double-cost control keeps M samples in both groups. Comparing these constructions separates independence from sampling precision.

E Evaluation details and statistical analysis

E.1 Component ablations and the questions they can answer

The ablation stage comprises 12 configurations on three tasks and 10 seeds per configuration, giving $12 \times 3 \times 10 = 360$ runs. The tasks are Burgers at $\nu = .0042$, the curved layer, and Taylor–Green; seeds are 1000–1009. The learning rate is fixed to 10^{-3} for this stage. The table below specifies each intervention.

Configuration	Intervention	Question
Uniform control	Exponent $\alpha = 0$	Does spatial weighting help relative to uniform?
Half-power weighting	Exponent $\alpha = .5$	Is weaker weighting sufficient?
Shuffled weights	Shuffle inverse-spread weights	Does spatial association matter?
Bounded shrinkage	Gate, clipping, floor .2	Reference safeguard combination
Higher floor	Raise floor to .5	Is stronger signal retention useful?
No gate	Disable shrinkage gate	Does heterogeneity shrinkage help?
No clipping	Disable clipping	Do extreme weights cause harm?
No floor	Remove uniform floor	Does downweighting suppress learning?
Split groups	Independent target and weight groups	Does dependence reduction repay variance cost?
Full independent groups	Full-sized separate groups	What changes with twice the returns?
Inverse spread, split	Split groups without shrinkage	Is the effect specific to the gate?
Inverse spread, full groups	Full groups without shrinkage	Separate independence from cost

Interpreting a component effect. The matched comparisons of bounded shrinkage with the no-gate, no-clipping, and no-floor variants each remove one safeguard. The paired solution errors and reliability metrics measure each component's effect. Weight histograms, gate values, and gradient norms help explain it: for example, the floor may reduce weight concentration while increasing the size of the correction near a front.

The shuffled control preserves the distribution of plain inverse-spread weights while changing their assignment to points. It tests the spatial association of that rule; it is not a complete shuffled version of every shrinkage candidate. The cross-fitting controls have explicitly different sampling costs, so they require paired accuracy and cost reporting.

Component interactions. Removing one component estimates its effect in the presence of the others. For example, clipping may matter less when the gate already produces nearly uniform weights. The recorded gate values and weight extrema help identify such interactions. The configuration remains fixed during confirmation; the ablation results explain its behavior rather than reselecting it on test seeds.

E.2 Paired inference and reliability metrics

The main study includes nine methods on eight tasks. Seven tasks use 20 seeds, and the smooth-flow task uses 10, producing $7 \times 9 \times 20 + 9 \times 10 = 1{,}350$ runs. The selected candidate is compared to eight baselines on each task, forming the frozen family of 64 comparisons. Missing measurements are represented by missing entries rather than zeros.

Paired errors and practical effect size. Let $e_{s,c}$ and $e_{s,b}$ be candidate and baseline errors for the same seed s and evaluation definition. The ratio-of-means statistic is

$$\widehat{R} = \frac{n^{-1} \sum_s e_{s,b}}{n^{-1} \sum_s e_{s,c}}. \tag{45}$$

Resampling seed pairs preserves their matching when constructing a bootstrap interval. It is different from averaging seedwise ratios, particularly when candidate errors are small or heavy-tailed. The frozen practical threshold is $\widehat{R} \geq 1.1$, corresponding to a reduction of at least $1 - 1/1.1 \simeq 9.1\%$ relative to the baseline mean.

For W strict wins among n complete pairs, the one-sided sign-test tail under a half-win null is

$$p = \sum_{j=W}^{n} \binom{n}{j} 2^{-n}, \tag{46}$$

Seed pairs are assumed independent, and ties count as non-wins. Holm adjustment orders the 64 p-values and compares them sequentially to $.05/(64-j+1)$; incomplete comparisons do not shrink the declared family. A strict advantage also requires a paired interval entirely above one and complete pairs. These conditions define the reported advantage flag.

Reliability and fixed-time efficiency. For a threshold τ, the empirical failure probability is

$$\widehat{P}_\tau = \frac{1}{n} \sum_{s=1}^{n} \mathbf{1}\{e_s > \tau\}. \tag{47}$$

Threshold curves, the median, and the 90th percentile describe complementary parts of the run distribution. Completion and failure counts accompany every cell. The 900-second study compares errors after equal elapsed training budgets and records actual updates and walker calls.

Task-level interpretation. Each rank is tied to a task, metric, budget, comparator set, and completed seed count. Mean orderings are reported descriptively unless the paired decision rule supports an advantage. Aggregate summaries retain all task-level outcomes; related Burgers viscosities describe sensitivity within one solution family.

E.3 Figure construction and reference checks

Analytic panels follow the explicit solutions and sources specified above. The baseline fields use four fixed seed-0 checkpoints, evaluated as saved tanh networks in float64 arithmetic. A scalar-loop calculation at fixed inputs checks the vectorized evaluation. No model fitting is performed when drawing these panels. The transport fields come from the independent, refined numerical reference in Appendix B.4; the baseline panels use $\nu = .005$ and the grids specified in Appendix A.3.