When Does Derivative Supervision Help? A Study of Label Construction and Noise

Anonymous authors
Paper under double-blind review

Abstract

Derivative labels describe local structure that value observations alone may not provide. Their usefulness, however, depends on how that structure is observed. We study label construction and loss balancing under a shared neural training procedure, then connect the results to controlled noise experiments. Across discontinuous-payoff problems, construction contributes more to value and derivative-error rankings than balancing, while the preferred construction reflects the information retained by the label. Varying value and derivative labels separately shows that the advantage of smoothed finite-difference (fuzzy) labels comes from their derivatives rather than from smoothing the value labels. Matched noise experiments reveal where useful derivative supervision becomes harmful to value prediction. A finite-design risk decomposition explains the trade-off between value-noise reduction, derivative-noise cost and approximation-bias change; its Fourier specialization shows how representation and sampling shape this balance. Combining a measured clean-label benefit with a last-layer estimate of the noise cost predicts whether derivative supervision still helps on new configurations for similar targets. Scientific extensions demonstrate value-prediction gains in a stochastic-volatility barrier problem and Burgers learning with a Fourier neural operator. Together with comparisons of higher-order labels, derivatives outside the value training region, and molecular objectives, these results establish a practical priority: determine what derivative information is supplied and how reliably it constrains the prediction before optimizing its weight.

1 Introduction

Scientific data can reveal both a value and how it changes: a price and its sensitivity, an energy and forces, or a field and its spatial derivatives. Sobolev training and differential machine learning fit these quantities together (Czarnecki et al., 2017; Huge & Savine, 2020). Derivative labels act as a form of regularization, constraining local target structure that value observations alone may leave unresolved.

The first choice is *which derivative information to supply*. Analytic derivatives, differentiated simulation paths, score-function estimators and differences of neighboring observations differ in bias, variance, cost and sometimes the accompanying value target. A loss balancer changes a label's influence but cannot restore information its construction omits. This creates a practical attribution problem: did a construction help because of its derivatives, its accompanying value targets, or the training procedure used to fit them? Controlled comparisons are needed to separate these effects.

We cross derivative sources with balancing rules on shared samples and models, then change the derivative source while holding value labels fixed. Much of the construction gain survives this control, separating useful derivative information from value smoothing. The expanded balancing comparison shows that the same weighting intervention can help one construction and harm another. Digital and barrier payoffs illustrate two ways a label can help: recovering missing sensitivity or retaining useful signal at low variance despite bias.

Matched noise sweeps then ask how much corruption useful derivative information tolerates. We specialize the standard bias–variance decomposition to a finite value–derivative design, retaining approximation error and sampling geometry. A measured clean-label benefit combined with a last-layer noise-cost calculation predicts the direction of the value-error change on fresh configurations.

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

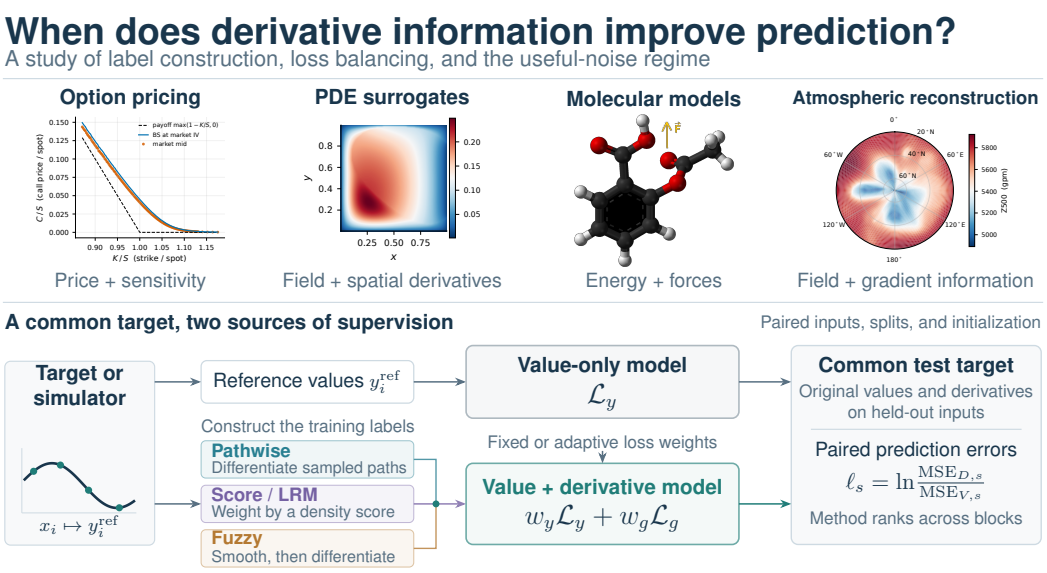

Figure 1: Where derivative information enters learning. Construction supplies the labels, weighting determines their influence, and a common prediction target measures their usefulness. The experiments separate changes to derivative information, value targets and observation access.

Scientific extensions evaluate the quantities ultimately needed: prices, reconstructed fields, energies and forces. For Burgers fields, we also give value-only training the same observations used to form derivative differences, testing what the constraints contribute beyond access to those observations.

Higher-order and support experiments complete the study by testing what additional local information supplies and when its benefit stops transferring.

2 Related Work

Work on derivative supervision spans the information supplied by labels, its use in training and its statistical consequences. **Sobolev training.** Czarnecki et al. (2017) study derivatives and random projections in approximation, identifiability, regression and distillation. Huge & Savine (2020) develop differential machine learning for simulation labels, including normalization, smoothing and higher orders. **Construction and weighting.** Kissel & Diepold (2020) compare exact and reconstructed derivatives under function-evaluation budgets; Glasserman & Karmarkar (2025) compare pathwise, smoothed and likelihood-ratio labels, including Heston digitals, barriers and weight sensitivity. GradNorm (Chen et al., 2018), ReLoBRaLo (Bischof & Kraus, 2022), conflict-averse aggregation (Liu et al., 2021) and gradient-statistic annealing (Wang et al., 2021) address how strongly to fit the supplied targets. Our crossed comparisons separate derivative construction from value smoothing and examine its interaction with weighting. **Higher orders and scientific learning.** Higher-order supervision and schedules are established (Hartmann et al., 2019; Avrutskiy, 2021). Reconstructed derivatives improve operator learning (Cho et al., 2025), with frequency-dependent and optimization effects (Liu et al., 2024; Oh et al., 2025). DINO supervises input-function sensitivities (O'Leary-Roseberry et al., 2022), while PINO constrains PDE residuals (Li et al., 2021b). Molecular benefits depend on the target quantity (Christensen & von Lilienfeld, 2020; Eberhard et al., 2026); we evaluate energy and forces using PaiNN and MACE (Schütt et al., 2021; Batatia et al., 2022). **Interpretation and statistical explanations.** Differential machine learning can be interpreted as data augmentation and regularization through constraints on local shape (Huge & Savine, 2020). Nonparametric theory explains improved statistical rates (Hall & Yatchew, 2007; Stone, 1982); proportional random-feature analysis characterizes noise, interpolation and acquisition costs, including cases where gradients do not improve a fit (Fisher et al., 2025). Our finite-design analysis retains sampling geometry, misspecification and general label covariance, and motivates a conditional prediction rule for trained

networks. Regularization, approximation and noise sensitivity can all contribute to the same observed benefit.

3 STUDY DESIGN

A shared core with focused extensions. We first compare label constructions and loss balancers using the same MLP, samples and random seeds, with smooth Softplus activations for derivative losses. Controlled functions isolate noise sensitivity by varying corruption within otherwise fixed training configurations. We then vary derivative order, observation support and model family, evaluating the relevant scientific quantities in each application. Table 1 summarizes the paired outcomes; Appendices B and C define the datasets and complete procedures.

Learning objective. For inputs x_i, a construction c supplies labels $(y_i^{(c)}, g_i^{(c)})$. The underlying target is f^*, but an individual label may be noisy or biased. Writing $\hat{f}_\theta$ for the predictor, define

$$\begin{aligned} \mathcal{L}_y &= \frac{1}{n}\sum_i (\hat{f}_\theta(x_i) - y_i^{(c)})^2, \quad \mathcal{L}_g = \frac{1}{nd}\sum_i \|\nabla_x \hat{f}_\theta(x_i) - g_i^{(c)}\|_2^2, \\ \mathcal{L} &= w_y \mathcal{L}_y + w_g \mathcal{L}_g. \end{aligned} \tag{1}$$

Derivative supervision differentiates predictions with respect to *inputs*; optimizing this objective also differentiates with respect to *parameters*. The value-only comparator sets $w_g = 0$. Application-specific derivatives, such as forces and spatial field derivatives, replace the input-gradient term with their stated physical operator. The core fits normalization statistics on training data only and uses one disjoint raw-value validation monitor for scheduling, stopping and checkpoint selection across all constructions. The appendices specify each experiment's normalization, derivative-coordinate scaling and selection rules.

Construction and balancing are different interventions. The core compares pathwise differentiation, likelihood-ratio (LRM, or score-function) estimation and fuzzy smoothing. These differentiate, respectively, a sampled payoff, its sampling law, or a continuous ramp approximation to a sharp transition. The nine task blocks comprise a Black–Scholes digital, six basket dimensions, a Black–Scholes barrier and a Heston digital. Fixed weighting, GradNorm and value-first warmup act on the resulting losses. Because fuzzy construction may change both y and g, a second crossed comparison independently varies raw versus smoothed values and likelihood-ratio versus fuzzy derivatives, using fixed weighting throughout. All methods are evaluated against the same unsmoothed reference. The synthetic and core weighting conventions, including their relationship to Huge & Savine (2020), are specified in the appendices.

Magnitude, consistency and uncertainty. For a matched experimental configuration and seed s, the paired log error ratio is

$$\ell_s = \log\left(\frac{\mathrm{MSE}_{D,s}}{\mathrm{MSE}_{V,s}}\right). \tag{2}$$

We use natural logarithms: $\ell_s < 0$ favors derivative supervision, and $\exp(\bar{\ell})$ is a geometric error ratio. Other controls name their comparison and reference explicitly. Ranks and lower-error counts summarize consistency; paired ratios retain effect size. We report value and derivative errors separately. The appendices define each comparison's averaging and uncertainty procedure, preserving pairing and shared noise trajectories when resampling.

4 LABEL CONSTRUCTION CHANGES WHAT BALANCING CAN ACHIEVE

The first question is whether optimization can compensate for a poor derivative target. Discontinuous sampled payoffs provide a revealing test: their expected values can be smooth even when direct differentiation of a sampled payoff misses the sensitivity of the event boundary. Score-function estimation and smoothing recover information about that boundary in different ways (Glasserman & Karmarkar, 2025). Evaluation targets the expected unsmoothed payoff and its derivative, using closed forms where available and precision-checked numerical references otherwise (Appendix D.1). This separates predictive error from the noise of the training labels.

Table 1: Prediction across the study. Each block names a baseline B and comparison D. Lower errors and ratios below one favor D.

A1. Full construction: pathwise B versus fuzzy D; fixed weights

	Value MSE ($\cdot 10^{-3}$) ↓			Derivative MSE ($\cdot 10^{-5}$) ↓			Lower error
Task / setting	B	D	Paired ratio	B	D	Paired ratio	pairs
BS digital	15.4	0.102	**0.0053**	2.26	0.0475	**0.02**	10/10
Basket, six dimensions	3.23	0.185	**0.052**	1.66	0.0905	**0.077**	60/60
BS barrier	557	525	1.042	463	760	**1.6**	5/10
Heston digital	12.2	0.227	**0.014**	4.83	0.136	**0.025**	10/10

A2. Derivative source: LRM B versus fuzzy D; identical raw values

BS digital	0.136	0.164	**1.4**	0.0513	0.0565	1.1	2/10
Basket, six dimensions	0.274	0.172	**0.66**	0.144	0.0878	**0.72**	47/60
BS barrier	8230	402	**0.094**	11300	716	**0.13**	9/10
Heston digital	1.16	0.266	**0.25**	4.22	0.121	**0.034**	9/10

B. Noise: value-only B versus first-order supervision D

	Value MSE ↓			Derivative MSE ↓			
Poly–trig 5D, $\rho = 0$	$1.55 \cdot 10^{-3}$	$8.79 \cdot 10^{-5}$	**0.061**	0.021	$9.91 \cdot 10^{-4}$	**0.05**	5/5
Poly–trig 5D, $\rho = 0.5$	$1.55 \cdot 10^{-3}$	$4.85 \cdot 10^{-3}$	**3.3**	0.021	0.0229	1.2	0/5
Trig 5D, $\rho = 0$	2.05	1.18	**0.16**	5.13	2.43	**0.14**	4/5
Trig 5D, $\rho = 0.5$	2.05	1.99	0.84	5.13	4.23	**0.65**	2/5
Bachelier 10D, $\rho = 0$	$3.76 \cdot 10^{-7}$	$7.97 \cdot 10^{-8}$	**0.21**	$2.83 \cdot 10^{-6}$	$5.42 \cdot 10^{-7}$	**0.19**	5/5
Bachelier 10D, $\rho = 0.5$	$3.76 \cdot 10^{-7}$	$5.72 \cdot 10^{-7}$	**1.5**	$2.83 \cdot 10^{-6}$	$2.78 \cdot 10^{-6}$	0.9746	0/5

C. Curvature: first-order B versus second-order supervision D

Poly–trig, $d = 2$	$6.1 \cdot 10^{-7}$	$3.48 \cdot 10^{-8}$	**0.064**	$4.35 \cdot 10^{-5}$	$1.67 \cdot 10^{-6}$	**0.044**	5/5
Poly–trig, $d = 10$	$5.6 \cdot 10^{-3}$	$4.53 \cdot 10^{-3}$	**0.82**	0.0185	0.014	**0.77**	5/5
Trig, $d = 2$	$1.54 \cdot 10^{-4}$	$1.38 \cdot 10^{-5}$	**0.15**	$6.87 \cdot 10^{-3}$	$4.81 \cdot 10^{-4}$	**0.11**	5/5
Trig, $d = 10$	7.99	7.99	1.000†	6.61	6.61	1.000†	2/5†

D. Support: original-region B versus added target-region derivatives D

SIREN, near region	1.18	0.759	**0.62**	164	87.2	**0.53**	5/5
SIREN, far region	1.29	1.25	0.9594	209	196	0.929	3/5

E. Heston barrier: value-only B versus fuzzy warmup D

	Value MSE ($\cdot 10^{-5}$) ↓			Derivative MSE ($\cdot 10^{-3}$) ↓			
Heston, narrow	2.1	0.227	**0.18**	4.63	0.185	**0.063**	4/5
Heston, wide	3.22	0.882	0.29	2.43	0.183	**0.1**	4/5

F1. Burgers: value-only B versus derivative supervision D

	Full-field MSE ($\cdot 10^{-3}$) ↓			Pooled relative L^2 ($\cdot 10^{-2}$) ↓			
Burgers FNO	3.31	2.22	**0.67**	9.77	8	**0.82**	20/20

F2. Darcy: value-only B versus derivative supervision D

	Full-field MSE ($\cdot 10^{-3}$) ↓			Boundary-strip MSE ($\cdot 10^{-3}$) ↓			
Darcy MLP	1.07	1.02	0.9136	0.599	0.137	**0.23**	4/5

G1. PaiNN: named baseline B versus native energy–force D

	Energy MAE (meV) ↓			Force MAE (meV/Å) ↓			
PaiNN: value-only	74.5	7.42	**0.1**	396	13	**0.036**	48/50
PaiNN: equal weight	5.78	7.42	1.3	22.8	13	**0.57**	24/50

G2. MACE: low-rate energy-only B versus energy–force D

MACE: aspirin	53	4.9	**0.093**	208	11.7	**0.057**	5/5
MACE: benzene	5.4	0.214	**0.04**	50.3	1.61	**0.032**	5/5
MACE: ethanol	13.7	1.24	**0.091**	81.4	7.35	**0.09**	5/5

H. ERA5 (four-layer MLP): value-only B versus fixed derivative weight D

Field MSE ($\cdot 10^{-1}$) ↓	B	D	Ratio D/B	Lower error pairs
ERA5, state	1.08	1.28	**1.2**	0/5

Errors are arithmetic means; ratios are paired geometric means, $\exp\{\overline{\log(E_D/E_B)}\}$. **Green** and **rust** mark ratios whose intervals lie below and above one; uncolored intervals include one. The final column counts pairs with lower error in the left-hand metric. Full grids, intervals and procedures accompany the corresponding experiments in the appendices. Curvature uses the dimension-scaled loss; †differences are smaller than 10^{-4} of the reference error. Support uses clean derivative labels. MACE compares different optimization schedules. ERA5 uses width 256.

Recovering missing sensitivity across balancing rules. Using the common MLP, we test each construction under fixed weighting, GradNorm and warmup. Expanded balancing, tuning and Transformer experiments test whether the findings depend on these initial choices (Appendix D.6). Comparing sources within the same task and balancing rule isolates the effect of changing the supplied information.

Digital and basket pathwise labels are zero almost everywhere, although the expected payoff varies with the input. Replacing them with fuzzy labels lowers mean value error in every task–balancer comparison; likelihood-ratio labels do so in all but one. Both also improve derivative prediction in most comparisons. These constructions supply boundary information that pathwise labels miss, with value gains across all three balancing rules (Appendix D).

Useful labels can be biased. The barrier call provides a complementary case. Its pathwise derivative retains the surviving call's sensitivity while omitting the barrier-boundary term. Here pathwise training improves derivative prediction under every tested recipe, whereas fuzzy and likelihood-ratio training do not improve on value-only training. The strongest derivative fits use pathwise labels, whose variance is much lower than that of the likelihood-ratio estimator. Thus a formally incomplete derivative can still be useful when it retains the relevant signal at a favorable noise level. The digital and barrier cases explain why construction matters without prescribing one estimator for every payoff.

Separating derivative information from value smoothing. Fuzzy construction changes values and derivatives. We cross raw versus smoothed values with likelihood-ratio versus fuzzy derivatives, fixing weights, scaler and validation monitor. Fuzzy derivatives lower value error in eight of nine settings under either value target (Figure 3B). At fixed raw values, the median reduction is 48%; smoothing values alone has little median effect and no consistent direction. Gains are smaller for baskets and larger for barrier and Heston digitals. This isolates the derivative-source contribution from value smoothing (Appendix D.4).

Balancing can reverse its effect. Gradient-statistic annealing (Wang et al., 2021) reverses its effect across sources. With the derivative term anchored, adapting the value coefficient improves pathwise prediction in nearly every pair but harms fuzzy prediction in most. Anchoring the value term reverses the pathwise effect and gives mixed fuzzy results (Appendix D.6).

These comparisons require joint assessment of construction and balancing. Appendix D.12 reports ranks and family-weighted summaries. Warmup controls separate its value-first stage, learning-rate change and optimizer reset; both the initial objective and later learning rate contribute.

A stochastic-volatility barrier extension. We next combine stochastic volatility with path dependence, extending the Heston digital setting of Glasserman & Karmarkar (2025). Across both narrow and wide spot ranges, fuzzy warmup lowers price error in most pairs and delta error in every pair. It also has lower mean price and delta errors than a fixed polynomial fitted to raw payoffs (Appendix D.10). Appendix D.9 gives the full neural comparison, and Table 1 summarizes the paired value outcomes.

5 The useful-noise regime and its explanation

An unbiased derivative label can still be too variable to improve a finite fit. We isolate this effect by adding centered Gaussian perturbations to analytic derivative labels, while retaining a value-only reference on the same configuration and seed. We denote relative derivative noise by ρ: the perturbation standard deviation is ρ times the generator's gradient standard deviation. Thus ρ measures corruption relative to derivative magnitude.

Matched curves reveal a measurable transition. For poly-trig, trig and the Bachelier basket, each noise trajectory keeps the target, inputs, split, initialization and standardized corruption draw fixed. Only the corruption scale changes. Derivative supervision substantially improves value prediction at low noise in all three families (Figure 2). Poly-trig and Bachelier cross to worse prediction as noise grows, while trig remains beneficial at the largest tested level. Using the combined validation objective moves the poly-trig crossing earlier without changing the other two qualitative outcomes.

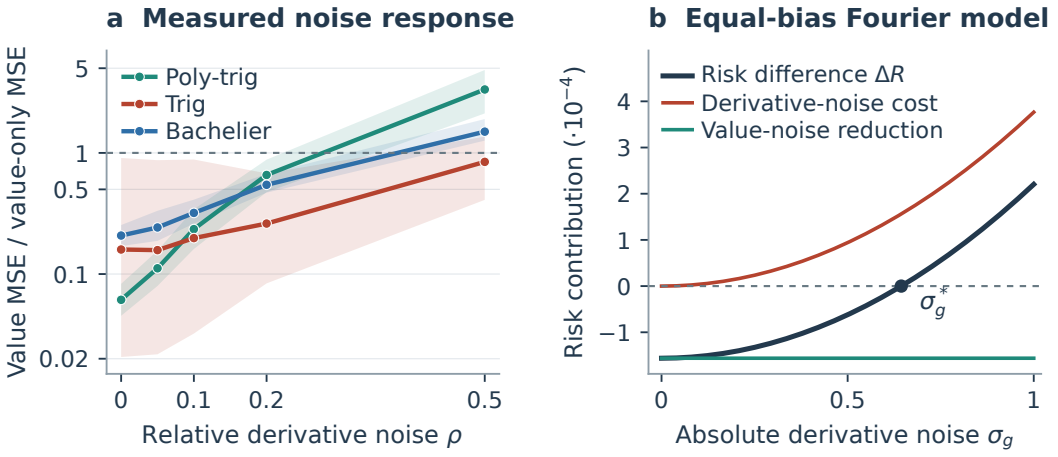

Figure 2: Comparing noise responses. Left: paired geometric error ratios show the prediction advantage disappearing for two families; parity is one and shading shows uncertainty. Right: the Fourier model separates value-noise reduction from derivative-noise cost under equal approximation bias. It uses noisy values; the neural fits use clean values. Appendices E and F give the full comparisons and derivation.

A useful-noise threshold is therefore measurable, but specific to the target and training procedure. Appendix E gives the configurations, crossings and uncertainty, including trig seed resamples that cross within the tested range.

Construction under the same added noise. At fixed raw value targets, adding the same perturbation to fuzzy and likelihood-ratio derivatives preserves the large fuzzy-over-LRM advantage for barrier prediction. This advantage holds at every tested noise level and in every paired fit. The digital contrast is smaller and does not resolve a clear preference between the sources. Appendix E.1 reports both prediction and derivative errors.

What changes at a crossover? At fixed training inputs, write $f^*(x) = \phi(x)^\top \theta_0 + r(x)$, allowing a nonzero approximation residual. Compare least-squares fits to values alone and values plus derivatives with fixed weight $\lambda > 0$. Assume full-rank value design and independent, centered channel noises with covariances $\sigma_y^2 I$ and $\sigma_g^2 I$. At a test input, denote the biases by δ_V, δ_D, value-noise propagation by A_V, A_D, and derivative-noise propagation by B_D (Appendix F). The risk difference is

$$\Delta R := R_D - R_V = \underbrace{\delta_D^2 - \delta_V^2}_{\text{bias change}} \underbrace{- \sigma_y^2 (A_V - A_D)}_{\text{value-noise reduction}} + \underbrace{\sigma_g^2 B_D}_{\text{derivative-noise cost}}, \tag{3}$$

where $A_V \geq A_D$ and $B_D \geq 0$. Derivative supervision stabilizes estimation from noisy values while importing derivative noise. Under misspecification it also changes the preferred approximation, which can favor either estimator.

The available risk margin, $\Gamma = \sigma_y^2 (A_V - A_D) - (\delta_D^2 - \delta_V^2)$, gives a useful interval $\sigma_g < \sqrt{\Gamma / B_D}$ when $\Gamma > 0$ and $B_D > 0$. A nonpositive margin gives no strictly beneficial interval. Although an error difference and log ratio agree in sign for a positive pair, their averages need not share a crossing.

Fourier structure makes the mechanism explicit. For integer-frequency Fourier features and a band-limited residual, a sufficiently fine uniform grid removes leakage by discrete orthogonality. The two estimators then have equal approximation bias, leaving a variance trade-off (Appendix F). For one sine–cosine pair of angular frequency 2π, the crossing is

$$\sigma_g^* = \sigma_y \sqrt{(2\pi)^2 + 2/\lambda}. \tag{4}$$

The synthetic sweeps keep value labels clean, so their low-noise benefit cannot arise from reducing value-label variance. In the equal-bias Fourier specialization, setting $\sigma_y = 0$ leaves no such benefit;

the unresolved residual is shared by both estimators. More generally, derivative constraints can change the approximation bias. Aliasing of unresolved frequencies makes that effect explicit even in a fixed representation, while neural feature learning and optimization provide additional mechanisms to test. Appendix F gives the multi-frequency expression and distinguishes integer-period continuation from arbitrary-phase invariance.

Predicting usefulness from a clean reference and a noise cost. The fixed-design identity explains how noise consumes a benefit; it does not determine the clean-label benefit of training an entire network. We measure that benefit once for each design, then compute the additional noise cost from the clean network's last hidden features. The rule, chosen using separate development data, predicts the sign of the value-MSE change on fresh configurations from the same three families with 88.3% accuracy, compared with 60.0% for the clean measurement alone and 76.7% for a noise-level threshold. The calculated cost therefore adds information beyond the clean reference. Error magnitudes are less well predicted, especially for Bachelier baskets, and crossing locations remain imprecise.

For a frozen last-layer response to centered derivative perturbations, the measured intercept plus the quadratic noise cost is an exact expected-risk formula. Retraining the entire network tests whether that response remains informative as features and optimization change. Appendix G gives the rule, the averaging over independent seed groups, paired uncertainty and complete curves, including the accuracy of predicted crossing locations.

6 Which local information serves the prediction task?

Higher orders constrain additional local structure, while observations in one region need not determine behavior elsewhere. Sample-budget curves show increasing benefits for some smooth targets under a common epoch cap, whereas difficult high-dimensional trigonometric fits remain near parity (Appendix H.1).

Higher orders target additional structure. Higher-order Sobolev losses are well established (Czarnecki et al., 2017; Hartmann et al., 2019; Avrutskiy, 2021). We ask what Hessian labels add to first-order supervision under the same target, dimension and sampling procedure, with independent model selection and test evaluation. They provide substantial further value-prediction gains for low-dimensional poly-trig and trig, including interacting targets with nonzero mixed derivatives. The effective weight matters: removing extra dimension-dependent downweighting of derivative terms strengthens gains in higher dimensions. Ten-dimensional trig remains near parity across orders and selects checkpoints very early. Curvature supplies useful information, but increasing its order does not by itself resolve this training behavior. Appendix I gives the full order comparisons, scaling controls and training behavior.

Higher-order directions should serve the downstream quantity: Eberhard et al. (2026), for example, improve excitation prediction by supervising physically admissible Hessian directions.

Local supervision does not specify every continuation. The support experiment fits a periodic target in one region, then evaluates interpolation and increasingly distant continuation. A sinusoidal-activation (SIREN) model benefits from clean derivatives near the training boundary, but the effect changes direction more than once as noise increases. Farther away, continuation depends strongly on the representation: a Fourier model repeats its integer-period structure by construction, whereas the neural models need not recover that continuation. Radial and half-space training regions further change the outcome (Appendix J). Local derivative information constrains behavior where it is observed without uniquely specifying behavior elsewhere.

Supplying derivatives in the target region instead tests derivative-assisted adaptation. Support and representation change both the remaining approximation bias and noise cost.

7 Beyond the reference backbone

Paired scientific applications test whether useful derivative information improves the quantity ultimately needed in task-specific models (Table 1).

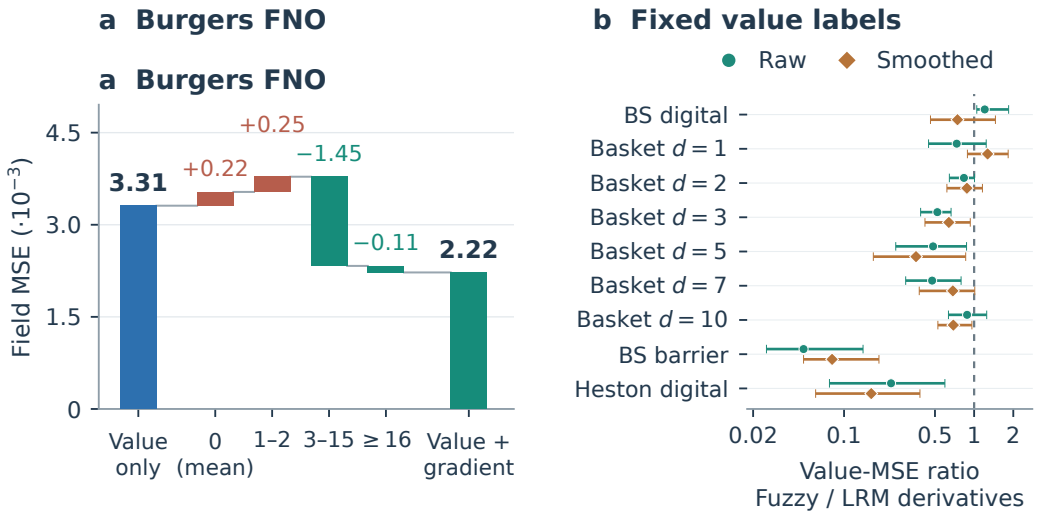

Figure 3: Where useful derivative information improves prediction. Left: intermediate spatial scales account for most of the Burgers field-error reduction. Right: changing the derivative source while holding value labels fixed separates its contribution from smoothing; ratios below one favor fuzzy derivatives. Appendices K.2 and D.4 give the detailed comparisons.

Burgers: field prediction with a neural operator. A space–time FNO learns from sampled field values and finite-difference spatial and temporal derivatives. Under a common value-validation monitor, derivative supervision lowers both sampled-point and full-field MSE in every paired run, reducing full-field error by about one third. The full-field gain reaches beyond the locations used in the supervised loss and persists under combined-loss monitoring. Initial-condition MLP controls show less consistent gains across capacities and validation monitors (Appendix K.5).

Pooled relative L^2 error also improves in every pair, while separately normalized trajectory errors are inconclusive because small-amplitude fields receive greater influence (Appendix K.1). A validation-selected PDE-residual constraint does not improve on the residual-free model here (Appendix K.3).

Equal access to neighboring observations. We give a value-only learner the same centers and stencil neighbors used by finite differences, with shared scheduling and value-validation selection. Matching derivatives still lowers full-field error in every pair (Appendix K.2). Together with the crossed label control, this shows that both the supplied information and the way it constrains the fit matter.

Where does the field improvement occur? Parseval's identity partitions squared reconstruction error into spatial components (Figure 3). In Burgers, intermediate-band error improves in every paired run and accounts for most of the total reduction, outweighing increased mean and long-wave errors. Training trajectories reveal that these scales are also learned earlier than with matched neighboring values. The lag behind long-wave learning narrows without generally reversing. Derivative supervision therefore improves both the speed of acquiring intermediate structure and its final accuracy (Appendix K.2).

In Darcy, the intermediate and finer bands improve in every pair, but the mean error rises in only three of five; two fits improve in every band. The full-field reduction is concentrated near the boundary and uncertain across seeds. A mean-error penalty is therefore not a universal requirement for improved reconstruction.

The Burgers controls retain the derivative benefit after imposing the known initial mean. Correcting the mean after training also gives lower error than imposing it during training in every paired comparison with the full value weight, while preserving the learned nonconstant modes. A long-wave penalty remains when the value coefficient is held fixed. These interventions distinguish useful local structure from removable mean error and the effects of constrained optimization (Appendix K.2).

Molecular learning and native loss weights. PaiNN's native energy–force objective improves energy MAE over value-only training in nearly every batch-matched pair. It beats equal weighting and each adaptive recipe on force MAE in every pair, while its energy comparison with equal weighting is almost evenly split (Appendix K.8). MACE energy–force training improves both quantities over common-schedule and stronger energy-only comparators (Appendix K.9). Informative force labels remain sensitive to how the objective weights them.

The derivative coordinates also matter: a distance Jacobian maps energy gradients to Cartesian forces, so auxiliary-gradient and force errors measure distinct quantities (Appendix K.7).

In ERA5, spatial derivative supervision gives near-parity field errors with coordinates alone and increases error with state-informed inputs across both MLP capacities and the Transformer (Appendix L.3). Supplying state information substantially improves reconstruction; spatial constraints do not provide the same benefit.

8 Discussion and conclusion

Derivative supervision is useful when the supplied constraints improve the target prediction. The crossed controls separate this contribution from value smoothing, and the balancing experiments show that the same weighting intervention can help one construction and harm another. Neither formal unbiasedness nor a balancing rule alone determines success: a derivative can omit essential sensitivity, while a biased label can retain valuable signal.

Noise tolerance can be tested rather than inferred from clean performance. The finite-design decomposition separates approximation bias, value-noise reduction and derivative-noise cost. Combining a clean-reference experiment with a computed cost predicts usefulness better than the clean measurement alone, conditional on that measured benefit.

Scientific comparisons connect these choices to the quantity ultimately needed. Burgers reconstruction improves beyond access to the same observations, with earlier learning and lower final error at intermediate scales. Molecular objectives, atmospheric reconstruction and distant continuation show why the target and representation remain consequential. The practical sequence is to identify what the labels supply, test how the objective uses that information, and evaluate prediction together with its tolerance to corruption.

USE OF GENERATIVE AI

Generative AI tools assisted with experiment code and debugging, and with organizing parts of the appendices and manuscripts editing in the appendices. The authors are responsible for the accuracy, attribution and final verification of the code and artifacts.

REPRODUCIBILITY STATEMENT

Appendix C specifies the study's training and pairing conventions, and Appendix B describes the data and derivative targets. Code for data generation, training and analysis will be released with the camera-ready version.

REFERENCES

V. I. Avrutskiy. Enhancing function approximation abilities of neural networks by training derivatives. *IEEE Transactions on Neural Networks and Learning Systems*, 32(2):916–924, 2021. doi: 10.1109/TNNLS.2020.2979706. URL https://doi.org/10.1109/TNNLS.2020.2979706.

Ilyes Batatia, Dávid Péter Kovács, Gregor N. C. Simm, Christoph Ortner, and Gábor Csányi. MACE: Higher order equivariant message passing neural networks for fast and accurate force fields. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Rafael Bischof and Michael A. Kraus. Multi-objective loss balancing for physics-informed deep learning. *arXiv preprint arXiv:2110.09813*, 2022.

Fischer Black and Myron Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81(3):637–654, 1973. doi: 10.1086/260062.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018.

Stefan Chmiela, Alexandre Tkatchenko, Huziel E. Sauceda, Igor Poltavsky, Kristof T. Schütt, and Klaus-Robert Müller. Machine learning of accurate energy-conserving molecular force fields. *Science Advances*, 3(5):e1603015, 2017.

Namkyeong Cho, Junseung Ryu, and Hyung Ju Hwang. Sobolev training for operator learning. *Journal of Computational Physics*, 543:114408, 2025. doi: 10.1016/j.jcp.2025.114408. URL https://doi.org/10.1016/j.jcp.2025.114408.

Anders S. Christensen and O. Anatole von Lilienfeld. On the role of gradients for machine learning of molecular energies and forces. *Machine Learning: Science and Technology*, 1(4):045018, 2020. doi: 10.1088/2632-2153/abba6f. Revised MD17 dataset, Figshare DOI 10.6084/m9.figshare.12672038.

Wojciech Marian Czarnecki, Simon Osindero, Max Jaderberg, Grzegorz Świrszcz, and Razvan Pascanu. Sobolev training for neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Eike S. Eberhard, Luca A. Thiede, Abdul Aldossary, Andreas Burger, Nicholas Gao, Vignesh Bhethanabotla, Alán Aspuru-Guzik, and Stephan Günnemann. Derivative informed learning of exchange-correlation functionals. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2606.04279.

Jianqing Fan and Irène Gijbels. *Local Polynomial Modelling and Its Applications*. Monographs on Statistics and Applied Probability 66. Chapman & Hall, 1996.

Katharine E. Fisher, Matthew T. C. Li, Youssef Marzouk, and Timo Schorlepp. Precise asymptotic analysis of Sobolev training for random feature models, 2025. URL https://arxiv.org/abs/2511.03050.

Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. Datasheets for datasets. *Communications of the ACM*, 64(12): 86–92, 2021. doi: 10.1145/3458723.

Paul Glasserman. *Monte Carlo Methods in Financial Engineering*. Springer, 2004.

Paul Glasserman and Siddharth Hemant Karmarkar. Differential ML with a difference. arXiv preprint arXiv:2512.05301, 2025. URL https://arxiv.org/abs/2512.05301v2. Version 2, revised 22 April 2026.

Peter Hall and Adonis Yatchew. Nonparametric estimation when data on derivatives are available. *The Annals of Statistics*, 35(1):300–323, 2007. doi: 10.1214/009053606000001127.

Thomas Hartmann, Matthias Kissel, and Klaus Diepold. Sobolev Training With Higher Order Derivatives. Technical report, 2019. URL https://collab.dvb.bayern/download/attachments/75112352/fp_hartmann.pdf. Undated manuscript; circa 2019 from institutional version history.

Hans Hersbach, Bill Bell, Paul Berrisford, Shoji Hirahara, András Horányi, Joaquín Muñoz-Sabater, Julien Nicolas, Carole Peubey, Raluca Radu, Dinand Schepers, Adrian Simmons, Cornel Soci, Saleh Abdalla, Xavier Abellan, Gianpaolo Balsamo, Peter Bechtold, Gionata Biavati, Jean Bidlot, Massimo Bonavita, Giovanna De Chiara, Per Dahlgren, Dick Dee, Michail Diamantakis, Rossana Dragani, Johannes Flemming, Richard Forbes, Manuel Fuentes, Alan Geer, Leo Haimberger, Sean Healy, Robin J. Hogan, Elías Hólm, Marta Janisková, Sarah Keeley, Patrick Laloyaux, Philippe Lopez, Cristina Lupu, Gabor Radnoti, Patricia de Rosnay, Iryna Rozum, Freja Vamborg, Sebastien Villaume, and Jean-Noël Thépaut. The ERA5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, 146(730):1999–2049, 2020. doi: 10.1002/qj.3803.

Steven L. Heston. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *The Review of Financial Studies*, 6(2):327–343, 1993. doi: 10.1093/rfs/6.2.327.

Brian Huge and Antoine Savine. Differential machine learning. *arXiv preprint arXiv:2005.02347*, 2020.

Matthias Kissel and Klaus Diepold. Sobolev training with approximated derivatives for black-box function regression with neural networks. In *Machine Learning and Knowledge Discovery in Databases: ECML PKDD 2019*, volume 11907 of *Lecture Notes in Computer Science*, pp. 399–414. Springer, 2020. doi: 10.1007/978-3-030-46147-8_24. URL https://doi.org/10.1007/978-3-030-46147-8_24.

kylegraupe. $SPY option chains – Q1 2020 – Q4 2022. Kaggle dataset kylegraupe/spy-daily-eod-options-quotes-2020-2022, version of 14 March 2023, 2023. CC0 1.0. Accessed 18 September 2026.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations (ICLR)*, 2021a.

Zongyi Li, Hongkai Zheng, Nikola Kovachki, David Jin, Haoxuan Chen, Burigede Liu, Kamyar Azizzadenesheli, and Anima Anandkumar. Physics-informed neural operator for learning partial differential equations. *arXiv preprint arXiv:2111.03794*, 2021b. URL https://arxiv.org/abs/2111.03794.

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

Xinliang Liu, Bo Xu, Shuhao Cao, and Lei Zhang. Mitigating spectral bias for the multiscale operator learning. *arXiv preprint arXiv:2210.10890*, 2024. URL https://arxiv.org/abs/2210.10890.

Marcos Lopez de Prado. *Advances in Financial Machine Learning*. Wiley, 2018.

ML Commons. Croissant: A metadata format for ML-ready datasets, specification 1.0. https://github.com/mlcommons/croissant, 2024.

Jong Kwon Oh, Hanbaek Lyu, and Hwijae Son. Sobolev acceleration for neural networks, 2025. URL https://arxiv.org/abs/2509.19773.

Thomas O'Leary-Roseberry, Peng Chen, Umberto Villa, and Omar Ghattas. Derivative-informed neural operator: An efficient framework for high-dimensional parametric derivative learning. *arXiv preprint arXiv:2206.10745*, 2022. URL https://arxiv.org/abs/2206.10745.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Dräxler, Min Lin, Fred A. Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International Conference on Machine Learning (ICML)*, 2019.

Antoine Savine. *Modern Computational Finance: AAD and Parallel Simulations*. Wiley, 2018.

Kristof T. Schütt, Pan Kessel, Michael Gastegger, Kim A. Nicoli, Alexandre Tkatchenko, and Klaus-Robert Müller. SchNetPack: A deep learning toolbox for atomistic systems. *Journal of Chemical Theory and Computation*, 15(1):448–455, 2018. doi: 10.1021/acs.jctc.8b00908.

Kristof T. Schütt, Oliver T. Unke, and Michael Gastegger. Equivariant message passing for the prediction of tensorial properties and molecular spectra. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021.

Vincent Sitzmann, Julien N. P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Charles J. Stone. Optimal global rates of convergence for nonparametric regression. *The Annals of Statistics*, 10(4):1040–1053, 1982. doi: 10.1214/aos/1176345969.

Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, Dan MacKinlay, Francesco Alesiani, Dirk Pflüger, and Mathias Niepert. PDEBench: An extensive benchmark for scientific machine learning. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2022.

Alexandre B. Tsybakov. *Introduction to Nonparametric Estimation*. Springer Series in Statistics. Springer, 2009. doi: 10.1007/b13794.

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.

Liu Ziyin, Tilman Hartwig, and Masahito Ueda. Neural networks fail to learn periodic functions and how to fix it. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

A BACKGROUND AND DEFINITIONS

Derivative supervision appears in several domains under different names. This appendix establishes the common mathematical objects and the meaning of the domain-specific labels used in the study. Appendix B describes the data sources and distinguishes experiments that use different label-generation procedures.

A.1 THREE DERIVATIVES WITH DIFFERENT ROLES

Let $x \in \mathbb{R}^d$ denote an input, $f^*(x)$ a target function, and $f_\theta(x)$ its learned approximation. Three derivatives must be distinguished.

1. A *target derivative* $\nabla_x f^*(x)$ is information about the function being learned. It may be available analytically, computed by automatic differentiation (AD) of a known function, or estimated from a simulator or sampled field. The resulting training label $\widetilde{g}$ need not be exact.
2. A *predicted input derivative* $\nabla_x f_\theta(x)$ is a derivative of the learned function with respect to its input. For a coordinate network this is obtained by differentiating the network. It is the prediction compared with $\widetilde{g}$.
3. A *parameter gradient* $\nabla_\theta \mathcal{L}$ is the quantity used by the optimizer to update the network. Differentiating a derivative-matching loss involves mixed derivatives such as $\partial_\theta \partial_x f_\theta$. This computation does not create target derivative labels.

For scalar targets, a typical objective is

$$\mathcal{L}(\theta) = w_y \mathcal{L}_y(\theta) + w_g \mathcal{L}_g(\theta), \qquad \mathcal{L}_g(\theta) = \frac{1}{nd} \sum_{i=1}^{n} \|\nabla_x f_\theta(x_i) - \widetilde{g}_i\|_2^2. \tag{5}$$

The value loss averages squared scalar residuals. Weights, normalization, and any restriction to observed derivative coordinates are part of the training recipe. Additional dimension-dependent weights are distinct from the component averaging already present in $\mathcal{L}_g$. Sobolev training (Czarnecki et al., 2017), differential machine learning (Huge & Savine, 2020), and energy–force training share this use of value and derivative information, although their observations and objectives differ.

AD evaluates derivatives of the implemented computational map, up to floating-point error and its differentiation rules. It does not remove simulation discretization error, Monte-Carlo error, or an invalid exchange of expectation and differentiation. This distinction is central when a smooth expected target is represented by nonsmooth sample payoffs.

A.2 EXPECTED TARGETS, PAYOFFS, AND LABEL CONSTRUCTION

A simulation-based regression target often has the form

$$f^*(x) = \mathbb{E}_Z[Y(x, Z)], \tag{6}$$

where Y is one simulated outcome and the law of Z is independent of x after reparameterization. The regularity of f^* and that of $Y(\cdot, Z)$ are different properties. Integration can produce a smooth price from a kinked or discontinuous payoff.

Pathwise construction. When differentiation and expectation can be interchanged, averaging $\nabla_x Y(x, Z)$ gives an unbiased derivative estimator of the implemented expectation (Glasserman, 2004). Regularity and integrability conditions are needed; almost-everywhere differentiability alone does not suffice. For a digital payoff, naively differentiating its indicator gives zero almost everywhere and misses the moving-boundary contribution to the expected price. For a barrier call, differentiating the surviving call payoff can give a nonzero term while still missing a barrier contribution. The information retained by a pathwise label therefore depends on the payoff.

Likelihood-ratio or score construction. For $f^*(x) = \int h(z)p(z; x)\, dz$, a suitable differentiable density and differentiation-under-the-integral conditions give

$$\nabla_x f^*(x) = \mathbb{E}_{p(\cdot;x)}[h(Z)\nabla_x \log p(Z; x)]. \tag{7}$$

If h also depends explicitly on x, its derivative contributes an additional term. Parameter-dependent support requires corresponding care at the boundary. The score construction can therefore handle payoff discontinuities without differentiating the indicator, but its validity depends on the density and score actually used. In a time-discretized simulator, the score must match that discrete law. Variance may be substantial and depends on the payoff, parameter, sampling scheme, and time step (Glasserman, 2004).

Fuzzy or smoothed-payoff construction. A smoothed payoff Y_ε replaces a sharp transition by a bandwidth-dependent approximation. Its value and derivative labels approximate those of $f_\varepsilon^*(x) = \mathbb{E}[Y_\varepsilon(x, Z)]$. The bandwidth controls a bias–variance trade-off relative to the unsmoothed target (Savine, 2018). An $O(\varepsilon^2)$ bias expansion requires suitable smoothness, symmetry and integrability assumptions; it is not a universal property of smoothing every payoff. In the construction panel, fuzzy construction changes both value and derivative training labels. Thus it is a comparison of complete label-generation procedures, not an intervention on derivative labels alone.

Noise and empirical crossovers. We write ρ for the dimensionless empirical corruption level. In the synthetic noise helper, independent Gaussian perturbations have standard deviation ρs_g, where s_g is the standard deviation of the supplied derivative array; when $s_g < 10^{-10}$, the helper uses $s_g = 1$. The absolute noise scale in the theoretical model is denoted σ_g. These quantities are not interchangeable without the reference scale. An empirical crossover is a zero crossing of a specified paired benefit statistic under a specified configuration or aggregation, rather than a universal threshold for all neural networks.

A.3 Finance: prices, Greeks, and simulated labels

Black–Scholes. Under the risk-neutral Black–Scholes model (Black & Scholes, 1973),

$$dS_t = rS_t\,dt + \sigma S_t\,dW_t.$$

A European call has payoff $(S_T - K)^+$ and price $P(S_0) = e^{-rT}\mathbb{E}[(S_T - K)^+]$. For $S_0, K, T, \sigma > 0$, the analytical price is smooth in the nondegenerate parameter interior although the terminal payoff has a kink. The synthetic call-price family varies spot with the other parameters fixed; other pricing and market-surface experiments use their own input coordinates.

Bachelier. For a normally distributed terminal basket with current mean $b = d^{-1}\sum_j S_j$ and standard deviation $s > 0$, the undiscounted call price is

$$P(b) = (b - K)\Phi(a) + s\phi(a), \qquad a = (b - K)/s, \tag{8}$$

with standard-normal distribution and density Φ and ϕ. Its spot partials are $\Phi(a)/d$. This price is smooth; the kink belongs to the payoff $(b_T - K)^+$. The smooth Bachelier call-price family and the Bachelier *digital-basket* construction panel use different payoffs and labels.

Heston. The stochastic-volatility model (Heston, 1993) uses

$$\begin{aligned} dS_t &= rS_t\,dt + \sqrt{V_t}S_t\,dW_t^{(1)}, \\ dV_t &= \kappa(\bar{v} - V_t)\,dt + \xi\sqrt{V_t}\,dW_t^{(2)}, \qquad d\langle W^{(1)}, W^{(2)}\rangle_t = c\,dt. \end{aligned}$$

Here κ controls mean reversion, $\bar{v}$ the long-run variance, ξ volatility of variance, and c correlation. European prices admit a characteristic-function representation requiring numerical integration. Simulation-based labels also depend on the discretization and sampling budget. The smooth synthetic Heston family uses Monte-Carlo bump-and-reprice labels; the separate barrier extension compares pathwise, score and fuzzy constructions.

Barriers and digital payoffs. A digital pays an indicator such as $\mathbf{1}\{S_T > K\}$. A discretely monitored down-and-out call pays only if the path satisfies its barrier condition. The dedicated single-monitor extension uses $e^{-rT_2}(S_{T_2} - K)^+\mathbf{1}\{S_{T_1} > B\}$, with $T_1 < T_2$. The monitoring convention is part of the payoff definition; it should not be conflated with continuous monitoring.

Greeks and implied-volatility surfaces. Delta is $\partial P/\partial S_0$, gamma is $\partial^2 P/\partial S_0^2$, and vega is $\partial P/\partial \sigma$. A derivative with respect to moneyness S_0/K differs from spot delta by the chain-rule factor K. Market-implied volatility is the volatility parameter reproducing an observed price under a pricing model. An SVI fit smooths the implied-volatility smile within a date–maturity slice. Supervision constructed from such a fit concerns a fitted pricing surface; it is not a direct observation of an exact market Greek.

A.4 Molecular energies and Cartesian forces

For positions $\mathbf{R} \in \mathbb{R}^{3N}$, a conservative molecular model predicts a scalar energy $E_\theta(\mathbf{R})$ and forces $\mathbf{F}_\theta = -\nabla_{\mathbf{R}} E_\theta$. The energy and force labels in rMD17 come from electronic-structure calculations (Christensen & von Lilienfeld, 2020). Energy–force supervision is established practice (Schütt et al., 2018; 2021). PaiNN uses equivariant scalar/vector representations. GATv2 and the pairwise-distance MLP test different molecular representations.

The native energy–force comparator uses weights $(0.01, 0.99)$ in the PaiNN comparison. These weights define a fixed derivative-supervised objective together with the stated units, normalization and loss reduction.

If the input is a vector of pairwise distances $q(\mathbf{R})$, Cartesian forces are obtained by the chain rule, $\mathbf{F}_\theta = -J_q(\mathbf{R})^\top \nabla_q E_\theta$. A force vector need not uniquely determine $\nabla_q E$ because the transpose Jacobian can have a null space. Force MAE or MSE measures the physical Cartesian prediction; a coordinate-gradient error can also depend on the chosen representation of that nonunique label.

A.5 PDE fields and operator predictions

Burgers' equation, $\partial_t u + u\,\partial_x u = \nu_{\text{eff}} \partial_{xx} u$, describes nonlinear transport with diffusion. Darcy flow in this study is the elliptic problem $-\nabla \cdot (a(x)\nabla u) = \beta$, where the coefficient field a varies between simulations. PDEBench supplies sampled numerical solutions (Takamoto et al., 2022). For the Burgers archive labelled $\nu = 0.01$, the generation code uses $\nu_{\text{eff}} = \nu/\pi$.

A coordinate network predicts a field value from space–time or spatial coordinates, optionally augmented by information identifying the initial condition or coefficient field. Coordinates alone do not identify a solution when several simulations have different initial conditions or coefficients. The Burgers augmented input contains Fourier features of the initial field; derivative supervision is restricted to physical coordinates rather than assigning observed derivatives to the conditioning features.

In the FNO experiment the model predicts a solution field. Central finite differences of the predicted field are compared with finite-difference labels at the sampled query points. These are output-field derivatives, not the Jacobian of the operator output with respect to its initial-condition coefficients. The reported sampled-point value MSE is also distinct from a full-field relative L^2 error.

A.6 ERA5, geopotential, and physical derivatives

ERA5 is an atmospheric reanalysis combining observations with a numerical model through data assimilation (Hersbach et al., 2020). The 500-hPa geopotential field is denoted Φ, in $\mathrm{m^2 s^{-2}}$. Geopotential height is $Z = \Phi/g_0$, in metres, where g_0 is standard gravity.

The reconstruction input contains location and seasonal coordinates, with an optional empirical-orthogonal-function (EOF) representation of the same day's anomaly. EOF features identify aspects of the state being reconstructed. This is field reconstruction, not forecasting an unseen future state.

For longitude λ and latitude φ in radians, physical zonal and meridional derivatives are $(a\cos\varphi)^{-1}\partial_\lambda \Phi$ and $a^{-1}\partial_\varphi \Phi$, where a is Earth's radius. The geostrophic diagnostic is

$$u_g = -\frac{1}{f_c a}\partial_\varphi \Phi, \qquad v_g = \frac{1}{f_c a \cos\varphi}\partial_\lambda \Phi, \qquad f_c = 2\Omega \sin\varphi.$$

If height is used instead, both right-hand sides acquire g_0. These relations connect spatial derivatives to a physical diagnostic. Measuring its error requires colocated predictions and references in the same coordinates. Our atmospheric comparison reports field-value error; its finite-difference labels contain both discretization and reanalysis uncertainty.

A.7 READING THE REPORTED METRICS

For m scalar predictions, value MSE is $m^{-1}\sum_i(\hat{y}_i - y_i)^2$. A derivative MSE additionally averages over the observed components, or uses the explicitly stated mask. Physical MAE averages absolute errors after the specified unit conversion. Comparison with a Monte-Carlo reference measures error against that finite reference and includes its uncertainty.

For paired errors $E_{A,s}, E_{B,s} > 0$, the natural-log ratio $r_s = \log(E_{A,s}/E_{B,s})$ is negative when A is better. The geometric error ratio $\exp(\overline{r})$, the ratio of mean errors $\overline{E}_A/\overline{E}_B$, and the mean paired relative reduction $\overline{1 - E_{A,s}/E_{B,s}}$ answer different questions. Method ranks describe ordering across blocks but do not measure error magnitude. In particular, a ratio of sums of squares of ranks is not an error-reduction factor. Block definitions, matching, aggregation and validation policies are specified in Appendix C.

B DATA SOURCES AND SUPERVISORY INFORMATION

The study combines generated functions and simulated payoffs with four external sources: rMD17, PDEBench, SPY end-of-day option quotes and ERA5 reanalysis. Table 2 summarizes the supervisory information available in each setting. Following Gebru et al. (2021), we describe composition, provenance, preprocessing and intended use below. Training and evaluation procedures appear in Appendix C and the experiment-specific appendices.

B.1 GENERATED SMOOTH AND DISCONTINUOUS FUNCTIONS

Purpose and composition. Six generator families vary target structure, dimension, sampling density, support, and derivative-label quality. Each configuration and seed determine its inputs, values and derivative labels; the noise experiments subsequently corrupt those labels.

Polynomial plus trigonometric target. The `poly_trig` family is additive:

$$f^*(x) = \sum_{j=1}^{d}\sum_{k=0}^{3} c_{jk}x_j^k + \alpha\sum_{j=1}^{d}\sin(\omega x_j), \quad x_j \sim \mathcal{U}[-1,1]. \tag{9}$$

Generator defaults are $\alpha = 0.5$, $\omega = 2$, and $c_{jk} = 0.9^k U_{jk}$ with $U_{jk} \sim \mathcal{U}[-1,1]$. Gradients, and Hessians in the order study, are obtained by AD of this known smooth function.

Trigonometric target. The `trig` family is $f^*(x) = \sum_j a_j \sin(\omega_j x_j)$, with $x_j \sim \mathcal{U}[-\pi,\pi]$, $\omega_j \sim \mathcal{U}[0.5,5]$ and $a_j \sim \mathcal{U}[0.5,2]$ by default. The gradient is $a_j\omega_j\cos(\omega_j x_j)$ in coordinate j. Both smooth function families support dimension and spectral variation; changing the input support is a separate experiment.

Step target. The `step` generator samples inputs in $[0,1]^d$ and five ordered breakpoints per coordinate. The value in each interval is sampled in $[0,1]$, and the target sums the coordinate-wise piecewise-constant values. Its supplied classical derivative is zero almost everywhere. This label contains no information about jump location or jump size through a nonzero local slope.

Smooth Bachelier call-price family. For spot inputs sampled uniformly in $[0.5K, 1.5K]^d$, the `bachelier` generator applies the analytical call-price formula in Equation 8 to their mean. The default strike is $K = 1$, maturity $T = 1$, and effective basket standard deviation is $s = 0.2\sqrt{T}$. The supplied spot derivatives are analytical. The regression target is smooth for $s > 0$; the underlying call payoff is kinked. This family is separate from the digital-basket construction study.

Black–Scholes call-price family. The `black_scholes` generator is single-asset, with $S_0 \sim \mathcal{U}[0.5K, 1.5K]$ and fixed pricing parameters. Defaults are $K = 100$, $T = 1$, $r = 0.05$ and $\sigma = 0.2$. Prices and spot deltas are analytical. The input is spot, rather than a four-dimensional vector containing every model parameter.

Table 2: Value and derivative supervision in the study's experimental settings.

Setting	Value label	Derivative label
Smooth synthetic functions	Analytical function or call price	Analytical derivative or AD of the known function; optional corruption
Step functions	Piecewise-constant function	Zero classical derivative almost everywhere
Synthetic Heston call	Finite Monte-Carlo price	Bump-and-reprice estimate with independent path sets
Digital construction panel	Monte-Carlo payoff mean, or fuzzy-smoothed counterpart	Pathwise, likelihood-ratio, or fuzzy construction
Dedicated Heston barrier	Discretely monitored call payoff estimate	Surviving-payoff pathwise, score, multistep score, or fuzzy construction
rMD17	Electronic-structure energy	Cartesian force, with negative-gradient convention
PDEBench	Stored numerical solution	Finite differences of the solution field
SPY surface tasks	Black–Scholes price at quote or SVI-fitted volatility	Pricing-map partial derivatives at the supplied volatility
ERA5 reconstruction	Reanalysis geopotential anomaly	Spatial finite differences, converted to physical coordinates

Synthetic Heston call-price family. The `heston` generator takes (S_0, V_0) as input, with $S_0 \sim \mathcal{U}[0.7K, 1.3K]$ and initial variance sampled in $[0.5v_0, 1.5v_0]$. Defaults are $K = 100$, $v_0 = 0.04$, $\kappa = 2$, long-run variance 0.04, volatility of variance 0.3, correlation -0.7, $r = 0.05$, and $T = 1$. The numerical generator uses 1,000 paths and 50 Euler steps per price estimate. Spot and initial-variance derivatives use symmetric relative bumps of 1% and fresh path sets for the bumped prices. Thus the labels include both discretization and finite-sample error; this suite does not use variance-reduced pathwise Greeks. Experiment configurations specify changes to these defaults and the sample budgets.

Sampling and corruption. The generator uses seeded JAX or NumPy random streams according to the function implementation. Synthetic Gaussian derivative corruption has standard deviation ρs_g, with s_g and its near-zero fallback defined in Appendix A.2. Analytical or AD labels are exact derivatives of their implemented analytic targets before this corruption; Monte-Carlo derivatives remain estimates even at $\rho = 0$. These functions provide controlled regression targets.

B.2 Construction core and Heston-barrier extension

Matched digital panel. The digital-payoff subset of the construction core contains a Black–Scholes digital and equal-weight Bachelier digital baskets at $d \in \{1, 2, 3, 5, 7, 10\}$. These are seven blocks, six of which belong to the same basket family. The full nine-block core also includes a Black–Scholes barrier and a Heston digital, specified in Appendix D.1. Digital Black–Scholes uses $K = 100$, $\sigma = 0.2$, $r = 0.05$ and $T = 1$. The baskets use payoff $\mathbf{1}\{d^{-1}\sum_j S_{T,j} > K\}$, $K = 100$, $T = 1$, base normal volatility 20, and pairwise Gaussian correlation 0.5. Initial spots are sampled as independent $\mathcal{N}(100, 10^2)$ inputs. Per-asset volatilities have seeded 10% Gaussian variation around the base magnitude, with absolute values used to keep them nonnegative.

Each input uses ten paths. The generated input budget N is 2,048 for digital Black–Scholes and baskets of dimensions 1, 2 and 3, and 4,096 for baskets of dimensions 5, 7 and 10. A seeded permutation allocates $\lfloor 0.2N \rfloor$ points to the evaluation partition and the remainder to the fitting partition, with the training procedure detailed in Appendix D. The balanced comparison crosses three complete label constructions with three balancers under ten seeds; the same value-only reference is shared within a block and seed.

Pathwise and score procedures use the unsmoothed payoff means for value supervision. Fuzzy construction supplies its smoothed value and derivative labels. Value and derivative errors in this panel are evaluated against the expected unsmoothed payoff and its derivative, using closed forms or precision-checked numerical references appropriate to each contract (Appendix D). Evaluation

does not reuse the noisy training payoff labels. Digital pathwise labels are zero almost everywhere; call-payoff derivatives generally are not.

Dedicated stochastic-volatility barrier extension. This suite uses the single-monitor Heston down-and-out payoff specified in Appendix A.3, with $K = 1$, $B = 0.85$, $V_0 = 0.04$, $\kappa = 1$, long-run variance 0.04, volatility of variance 0.15, correlation -0.7 and $r = 0$. Monitoring and maturity are $T_1 = 1/3$ and $T_2 = 2/3$, with 84 simulation substeps in each interval. The spot ranges are $[0.7K, 1.3K]$ and $[0.5K, 1.5K]$. There are 1,024 generated input points and ten paths per training input. The method-independent evaluation reference uses 200 spots and 100,000 paths with a separate seed. The neural panel compares twelve recipes at five seeds per range. Appendix D.9 specifies the estimator–balancer combinations. This barrier contract differs from the smooth Heston call-price family defined above.

B.3 rMD17 MOLECULAR ENERGIES AND FORCES

Origin and composition. rMD17 (Christensen & von Lilienfeld, 2020) recomputes configurations from MD17 (Chmiela et al., 2017) with tighter electronic-structure settings at the PBE/def2-SVP level. The ten molecules are aspirin, azobenzene, benzene, ethanol, malonaldehyde, naphthalene, paracetamol, salicylic acid, toluene and uracil. Each has approximately 10^5 molecular configurations, each comprising Cartesian positions, an energy and a $3N$-component force vector.

Collection and splits. The configurations originate in molecular-dynamics trajectories; energies and forces are obtained from electronic-structure single-point calculations. The study uses all ten molecules and five supplied split indices per molecule, with a 1,000-frame fitting pool per split. PaiNN uses 950 of these frames for optimization, 50 for validation, and 1,000 separate evaluation configurations; other backbones retain their stated suite-specific partitions. The source is the authors' Figshare archive associated with Christensen & von Lilienfeld (2020).

Use and measurement. The same Cartesian force labels support the PaiNN, GATv2 and pairwise-distance representations. The domain-native energy–force weighting is an explicit PaiNN comparator. Physical force MAE is reported in $\mathrm{meV}/\text{Å}$ after unit conversion. Representation and batch-matching details belong to Appendix K. The data cover small organic molecules at one electronic-structure level; they do not establish transfer to arbitrary chemical species or reaction pathways.

B.4 PDEBENCH BURGERS AND DARCY FIELDS

Origin and composition. PDEBench (Takamoto et al., 2022) provides numerical solution archives for learning PDE surrogates. The study uses the published HDF5 outputs for one-dimensional Burgers labelled $\nu = 0.01$ (the generator uses diffusion coefficient ν/π) and two-dimensional Darcy with forcing $\beta = 1$. Burgers archives contain initial conditions, evolving fields and space–time coordinates. Darcy is a steady elliptic problem; its instances contain coefficient fields and solution fields, rather than a trajectory of time-evolved states. The documented Darcy resolution is 128×128 with log-normal coefficient fields.

Sampling. The coordinate-regression protocol uses 1,800 training, 100 validation and 100 test simulations per PDE. It samples 300 query points per training simulation and 100 per validation or test simulation. These budgets correspond to 540,000 training and 10,000 test queries. Burgers queries are physical (x, t) points; Darcy queries are spatial points. In the Burgers implementation, queries exclude a four-grid-point margin in both space and time. Derivative arrays use central finite differences in the interior and one-sided stencils at array boundaries. The four-cell query margin excludes those boundary stencils from both training and sampled-point evaluation. Whole-field relative error additionally covers the boundary values.

Conditioning and operator extension. The limited-input comparison omits the full simulation state: Burgers uses (x, t), while Darcy uses $(x, y, a(x, y))$, including local permeability but not the entire coefficient field. Burgers initial-condition augmentation adds the real and imaginary parts of the first 16 Fourier coefficients of the initial field to (x, t); only physical coordinate derivatives are supervised. The FNO extension uses the same Burgers simulation partitions and sampled query

locations, predicting fields and applying the same central-difference stencils to its outputs. Its sampled-point metrics have the same query reference as the coordinate model; they are not full-field relative L^2 errors.

B.5 SPY END-OF-DAY OPTION SURFACES

Origin and composition. The option panel is the public Kaggle dataset of SPY end-of-day option chains from January 2020 through December 2022 (kylegraupe, 2023), released under CC0 1.0 in a single version dated 14 March 2023 as one file of 3,589,079 chain rows; the archive checksum is recorded with the repository's data documentation. Each row lists the quote date, strike, expiration, underlying close, and the call and put bid, ask, implied volatility, volume and vendor Greeks at the 16:00 close. The study uses the call side, using a fixed panel of 1,576,419 rows over 758 trading days. The panel contains quotes with spot-to-strike ratio in $[0.85, 1.15]$, between 8 and 1,096 calendar days to expiration (time to expiry T is days over 365.25), implied volatility of at least 0.01, positive bid and ask, and mid price of at least \$0.05. These ranges describe the panel; its complete membership and ordering are defined by the supplied zero-based source-row index list. Applying that list and the specified transformations to the raw CSV reproduces the input, quote-date, market-price and analytical Black–Scholes reference arrays exactly. The data manifest records the raw archive, row-list and array checksums, including an explicit rule for matching duplicate rows.

Input coordinates and rates. The four inputs are $(S/K, T, r, \sigma_{\mathrm{IV}})$: spot-to-strike ratio, time to expiry in years, annual continuously compounded risk-free rate, and annual implied volatility. The processed rate column is constant at $r = 0.02$. The pricing map sets dividend yield to zero, and the target is the strike-normalized call price C/K. The optional fifth input, log-volume, is omitted in the four-input surface comparison. In the SVI configuration only the volatility coordinate is replaced by the fitted value. Affine feature normalization is applied by the training procedure; moneyness itself is not log-transformed. These conventions define the pricing surface used as the regression target.

Constructed targets. Both configurations sample the Black–Scholes pricing map and its analytical partial derivatives in independent input coordinates. The Black–Scholes-target configuration uses each row's quoted implied volatility. The SVI-supervised configuration supplies a volatility fitted within the corresponding date–maturity slice. The price and Greeks are evaluated at that supplied volatility; sensitivities to spot, strike or other coordinates hold the volatility input fixed. These labels therefore omit derivatives through the fitted smile. They are constructed pricing-map sensitivities rather than independently observed market Greeks. Results are reported separately for these two target constructions.

Splits and evaluation. The temporal protocol places observations through 30 June 2021 on the training side and observations from 1 July 2021 on the test side, applying a five-trading-day embargo. A purged five-fold walk-forward protocol with a five-trading-day embargo supplies a separate temporal robustness comparison (Lopez de Prado, 2018). The evaluation measures temporal transfer of the constructed pricing surface over this asset and period.

B.6 ERA5 500-HPA GEOPOTENTIAL RECONSTRUCTION

Origin and composition. ERA5 (Hersbach et al., 2020) is ECMWF's atmospheric reanalysis. The study uses 12 UTC daily 500-hPa geopotential fields from 2014–2020, covering 20–70°N at 1° resolution. The archive contains 2,557 daily fields on a 51×360 grid. Geopotential is measured in $\mathrm{m^2 s^{-2}}$; height in metres is obtained by dividing by standard gravity. The two directional labels are physical spatial derivatives, with longitude wrap-around.

Collection and representation. The source is the Copernicus Climate Data Store. Preprocessing converts geopotential to height, normalizes it with training-set statistics, forms anomalies and fits the first 16 EOF components on training snapshots for state conditioning. The coordinate-only input has five components: latitude, sine/cosine longitude, and sine/cosine day of year. Adding the same day's 16 anomaly coefficients produces a 21-dimensional input. The supervised derivatives concern physical space, not derivatives with respect to the seasonal or EOF coordinates.

Splits and use. The chronological partition contains 1,731 training, 383 validation and 383 test days, with two 30-day embargo intervals. The sampling budget is 10^5 query points per epoch. This suite compares fixed-weight methods in pointwise field reconstruction. The atmospheric comparisons use normalized field-value MSE. Physical derivative coordinates and their possible geostrophic interpretation are defined in Appendix A.6. Same-day state augmentation makes this reconstruction rather than weather forecasting.

B.7 Licences and data availability

Table 3 lists the source licences and attributions for the distributions used in this study.

Table 3: Source licences and distributions.

Source	Licence	Distribution
Synthetic generators	MIT	Maintained with the study's generator code.
rMD17	Creative Commons Attribution 4.0	Original authors' Figshare distribution and dataset DOI.
PDEBench	Apache 2.0	Original authors' published simulation archive and release.
SPY quote panel	CC0 1.0	Kaggle dataset `kylegraupe/spy-daily-eod-options-quotes-2020-2022`, single version of 14 March 2023; archive checksum recorded with the data documentation.
ERA5	Copernicus Climate Change Service licence	Copernicus Climate Data Store; attribution to Copernicus Climate Change Service information.

Generated observations are reproducible from their configuration and seed; external archives are available from the providers listed above. Data documentation includes Croissant metadata in schema version 1.0 (ML Commons, 2024). No personal identifiers are part of the scientific inputs: the molecules and PDEs are simulated, the financial observations are publicly traded option quotes, and the atmospheric inputs are gridded reanalysis fields.

Data coverage. The analytic targets exercise selected smooth, oscillatory and piecewise-constant structures. The digital panel contains several dimensions of one payoff family rather than independent domains. rMD17 covers small molecules at one electronic-structure level; PDEBench uses selected coefficients and parameter ranges; SPY covers one asset and period; ERA5 covers one variable, region, resolution and reconstruction task. These properties define the questions each suite can answer.

C Training, pairing and reproducibility

C.1 Experimental unit and population

The construction core consists of nine blocks: a Black–Scholes digital, digital baskets of dimensions $1, 2, 3, 5, 7, 10$, a Black–Scholes barrier and a Heston digital. Three constructions and three balancers give nine methods in every block, each evaluated at ten seeds. The 810 derivative-supervised fits use 90 shared value-only fits as references. The seven digital and basket blocks form the mechanistic subset with almost-everywhere-zero pathwise labels: 630 derivative-supervised fits and 70 value-only references. Sharing a reference does not create additional independent value-only runs. The dedicated Heston barrier, synthetic noise, order, support and architecture experiments have separate populations, described in their respective appendices. Each population is analyzed separately.

C.2 Core data and model defaults

The core uses a four-hidden-layer, 256-unit softplus MLP with Kaiming-normal initialization. Softplus provides smooth input derivatives for the supervised gradient loss; Appendix J compares continuation with this architecture and a sinusoidal representation. The Black–Scholes digital and barrier, and basket dimensions $1, 2, 3$, use 2,048 generated inputs, split into 1,311 training, 328 validation and 409 test points. Basket dimensions $5, 7, 10$ use 4,096 inputs, split into 2,621, 656 and 819 points. The Heston digital uses 1,024 inputs, with 819 in the outer training pool and 205 in the test set; the inner split reserves 20% of the outer pool for validation. Each input uses ten simulation paths. Normalization statistics are fitted to the training inputs and raw value labels only. Every construction in a block and seed shares these statistics.

For coordinate scales $s_{x,j}$ and output scale s_y, the transformed derivative follows the chain rule:

$$\tilde{x}_j = (x_j - \mu_{x,j})/s_{x,j}, \quad \tilde{y} = (y - \mu_y)/s_y, \quad \frac{\partial \tilde{y}}{\partial \tilde{x}_j} = \frac{s_{x,j}}{s_y}\frac{\partial y}{\partial x_j}.$$

The core uses this chain-rule transformation without the additional coordinatewise derivative-RMS normalization of Huge & Savine (2020). Its dimension-aware and equal-weight objectives are compared in Appendix D.6.

C.3 Objectives and stopping

With $\mathcal{L}_g$ defined as the mean over derivative coordinates, fixed weighting uses

$$w_y = \frac{1}{1 + \lambda d}, \qquad w_g = \frac{\lambda d}{1 + \lambda d}, \qquad \lambda = 1.$$

Fixed-half instead sets both weights to $1/2$ in the suites where it is included. GradNorm (Chen et al., 2018) adapts weights using task-gradient magnitudes and relative training rates. ReLoBRaLo (Bischof & Kraus, 2022) uses loss-ratio balancing with a stochastic historical reference; it is included in the expanded balancing panel (Appendix D.6). We use temperature 0.1, exponential averaging factor 0.999 and probability 0.999 of retaining the preceding historical weight rather than looking back to initialization. The two weights sum to two.

The core uses AdamW with learning rate 0.005, zero weight decay, batch size 256 and gradient-norm clipping at one. The maximum duration is 500 epochs, with ReduceLROnPlateau patience 20 and factor 0.5, and early-stopping patience 50. The raw-value MSE on the disjoint validation partition controls the scheduler, stopping and checkpoint selection for every method, whatever its training objective. The complete training procedures share one selection rule; their realized optimization trajectories can differ.

Warmup allocates at most 250 epochs to value-only training and 250 to GradNorm derivative training. It restores the selected first-stage model, starts a fresh optimizer at learning rate 0.0005, and does not reallocate unused first-stage epochs. Both stages use the same raw-value validation monitor. Equal epoch caps do not imply equal measured compute, because derivative objectives require additional differentiation and stopping can occur at different times.

C.4 Statistics and interpretation

Main and core paired log ratios use the natural logarithm of the two errors; supplementary summaries using another base identify it explicitly. For a complete block, compute the arithmetic seed-mean error of each method and rank those nine means. Average ranks across blocks for the rank summary. These ranks and the pair-level log ratios answer different questions: ranking describes relative order in a block, while the log ratio measures multiplicative change against a reference. Formal omnibus rank tests treat blocks as the repeated comparison units; their applicability to a broader population is limited by the related basket blocks.

Within a seed, multiple methods and noise levels can share data and references. Noise bootstrap resamples a seed's entire panel together. A method-level bootstrap or test must preserve the pairing rather than independently resample the two methods. Seed intervals condition on the chosen tasks, data generation and configuration grid; they do not quantify variation across all possible datasets.

C.5 Computation and reproducibility materials

Experiments use a workstation with two RTX A6000 GPUs, an Intel Xeon CPU and approximately 251 GB RAM. Software versions are specified per experiment; these include Python 3.13.3 and PyTorch 2.8 with CUDA 12.8, with SchNetPack 2.2.0 for PaiNN. Reported training times measure process duration on shared hardware, rather than exclusive GPU occupancy.

The data generators, training configurations and analysis scripts will be released with the camera-ready version. The implementation links each comparison to its seeds, split definitions, software versions and per-run metrics; saved prediction arrays and checksums support independent recomputation of the reported errors. Training and aggregation are separate entry points, so tables can be regenerated without repeating optimization. Floating-point results of fresh GPU training can vary across software and hardware environments.

D Joint label construction and loss balancing

D.1 Nine-block core and complementary contrasts

The core contains nine task blocks: a Black–Scholes digital, six Bachelier basket dimensions, a Black–Scholes down-and-out call, and a Heston digital. The balanced three-construction, three-balancer panel contains 810 neural fits and 90 shared value-only references. The expanded panel in Section D.6 adds five balancing rules on the same data. We also examine the seven-block digital and basket subset, whose almost-everywhere-zero pathwise labels expose the effect of missing boundary sensitivity.

The Black–Scholes barrier uses $K = 100$, $B = 80$, volatility 0.2, rate 0.05, maturity 1, and 252 monitoring steps. Its 2048 inputs have ten paths each. The pathwise label is the derivative of the surviving call, $e^{-rT}\mathbf{1}\{\text{alive}\}\mathbf{1}\{S_T > K\}S_T/S_0$; the omitted barrier-boundary term makes it biased but not identically zero. The Heston digital uses $K = 100$, $v_0 = \bar{v} = 0.04$, $\kappa = 2$, volatility of variance 0.3, $c = -0.7$, $r = 0.05$, maturity 1, and 50 full-truncation Euler steps. Its 1024 inputs have ten paths each; the outer 819/205 split is followed by an inner 80/20 training/validation split. Its likelihood-ratio score includes both correlated Gaussian increments. This digital is distinct from the single-monitor Heston barrier in Section D.9.

Evaluation targets. Every fit is evaluated on the held-out inputs against the expected value of the contract that generated its training labels, and against the derivative of that expectation, rather than against a noisy label. The Black–Scholes digital uses the closed-form discounted expectation; each Bachelier basket uses the exact Gaussian expectation with the per-asset volatilities drawn for that seed; the barrier uses the discretely monitored quadrature of Section D.5; the Heston digital uses a conditional Monte Carlo reference for the same full-truncation Euler scheme, in which the orthogonal spot increments are integrated analytically after conditioning on the variance path, with $262{,}144$ paths in each of eight replications and a replication standard error below 3.1×10^{-4} for values and 1.6×10^{-5} for derivatives in every seed. Evaluation against these expectations separates prediction error from conditional training-label noise: for a ten-path Bernoulli average, the conditional variance before discounting is $p(x)(1 - p(x))/10$.

D.2 Digital and basket subset: target construction

The digital and basket subset crosses three joint label constructions with three balancing recipes on seven blocks: a Black–Scholes digital and Bachelier basket digitals in dimensions $d \in \{1, 2, 3, 5, 7, 10\}$. Each cell has ten seeds, giving $7 \times 3 \times 3 \times 10 = 630$ derivative-supervised fits and 70 shared value-only fits. The seed list is $42, 123, 456, 789, 1337, 2024, 3141, 5926, 8008, 9999$.

For each input, ten simulated payoffs are averaged to construct the value label. *Pathwise* (PW) differentiation of an unsmoothed digital payoff produces a zero derivative almost everywhere. The corresponding derivative loss consequently penalizes the model's input sensitivity even though the conditional expected payoff is nonconstant. *Likelihood-ratio* (LRM) labels instead multiply the unsmoothed payoff by the score of the sampling density. *Fuzzy* labels replace the payoff indicator by

the call-spread ramp

$$S_\epsilon(c) = \frac{(c+\epsilon/2)_+ - (c-\epsilon/2)_+}{\epsilon}, \qquad S'_\epsilon(c) = \frac{\mathbf{1}\{|c| < \epsilon/2\}}{\epsilon} \quad \text{almost everywhere.} \tag{10}$$

Here c is terminal spot minus strike for the Black–Scholes digital, or terminal basket value minus strike for the Bachelier digital. This is a piecewise-linear payoff approximation. Fuzzy value labels average the ramp payoff, and derivative labels average its pathwise derivative. This panel compares joint value/derivative constructions; the crossed control in Section D.4 separates the two channels.

For the Black–Scholes problem, $K = 100$, volatility 0.2, rate 0.05 and maturity 1, with initial spots uniform on $[50, 150]$. The ramp width is $\max\{0.5\ \mathrm{sd}(S_T - K), 10^{-6}\}$, estimated on an independent pilot simulation with its own random stream, so that the training inputs and paths are the same for every construction. For each basket, $K = 100$, maturity 1, equal weights $1/d$, and correlation 0.5 between distinct assets. Initial spots are independently distributed as $100 + 10Z$, $Z \sim \mathcal{N}(0, 1)$; per-asset normal-model volatilities are $|20(1 + 0.1Z_j)|$, drawn once per seed and used unchanged by the exact evaluation reference. Basket ramp widths use the same $0.5\ \mathrm{sd}(c)$ rule on a separate calibration simulation of $\min(N, 1000)$ inputs, with a random stream offset from the training seed.

The total generated sample size is $N = 2048$ for the Black–Scholes problem and the $d = 1, 2, 3$ baskets, and $N = 4096$ for $d = 5, 7, 10$. A seeded permutation reserves $\lfloor 0.2N \rfloor$ inputs for evaluation; an inner 80/20 split of the remainder provides training and validation. The resulting (training, validation, evaluation) counts are $(1311, 328, 409)$ and $(2621, 656, 819)$, respectively. Input and value normalization statistics are fitted to the training partition only, using the raw value labels, and the same scaler is shared by every construction in a block and seed. For coordinate j, the normalized derivative is $g_j s_{x,j}/s_y$, consistent with the chain rule. There is no additional derivative-RMS normalization.

Every method is evaluated against the exact expected value of the unsmoothed contract and its exact derivative: the closed-form discounted expectation for the Black–Scholes digital and the exact Gaussian expectation for every basket dimension. Value MSE and derivative MSE are on different scales; absolute differences between these metrics are not directly comparable.

D.3 Balancing and optimization

All core fits use four hidden layers of width 256, softplus activations, Kaiming-normal initialization, a scalar linear output, batch size 256, AdamW with zero weight decay and initial learning rate 0.005, and gradient clipping at norm 1. The maximum is 500 epochs. A plateau scheduler halves the learning rate after 20 unimproved validation epochs; early stopping has patience 50. With component-averaged normalized errors L_y and L_g, the fixed recipe uses

$$L_{\text{fixed}} = \frac{L_y + dL_g}{1 + d}. \tag{11}$$

Equal convex weights are therefore obtained only when $d = 1$ in this panel. GradNorm adapts the two task weights using relative training rates and shared parameter-gradient norms (Chen et al., 2018); its exponent is 1.5, weight-optimizer learning rate 0.025, and task weights are normalized to sum to two. The shared parameters are in the last hidden layer.

The warmup recipe allocates at most 250 epochs to value-only training and 250 to GradNorm training. It restores the best first-stage model, creates a fresh optimizer, and lowers the initial learning rate of the second stage by ten. Early stopping can shorten either stage; unused first-stage epochs are not reassigned. Thus its total epoch cap is 500. Every fit, whatever its training objective, monitors the raw-value MSE on the disjoint validation partition for the plateau scheduler, early stopping and checkpoint selection; warmup uses this monitor in both stages. Equal epoch caps need not yield equal realized update counts or wall-clock costs.

D.4 Separating the value target and derivative source

The crossed control uses four fixed-weight training configurations: $(y_{\text{raw}}, g_{\text{LRM}})$, $(y_{\text{raw}}, g_{\text{fuzzy}})$, $(y_{\text{smooth}}, g_{\text{LRM}})$ and $(y_{\text{smooth}}, g_{\text{fuzzy}})$. The two mixed configurations comprise 180 fits, with the other two supplied by the ten-seed core panel on the same nine blocks. Every configuration uses

Table 4: Crossed label control: median ratios of block-mean errors. Counts report blocks with a strictly lower mean under the replacement. Each block has ten seeds in every configuration. Section D.5 specifies the barrier reference.

Replacement	Held fixed	Value ratio	Lower	Gradient ratio	Lower
LRM → fuzzy gradients	Raw values	0.5215	8/9	0.5234	6/9
LRM → fuzzy gradients	Smoothed values	0.6864	8/9	0.6113	8/9
Raw → smoothed values	Fuzzy gradients	0.9291	5/9	1.0205	4/9
Raw → smoothed values	LRM gradients	1.0107	4/9	1.0112	4/9

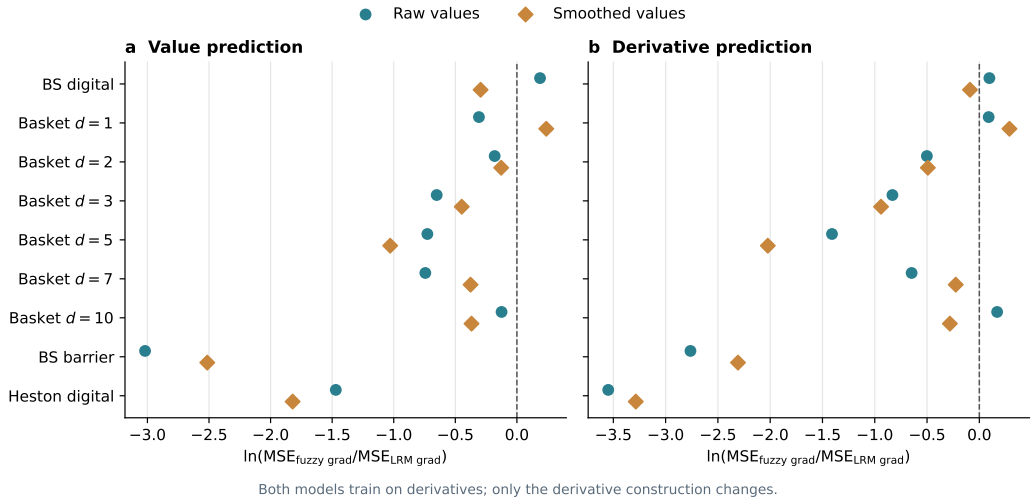

Figure 4: Changing the derivative source while holding value targets fixed. Each point compares ten-seed mean errors using fuzzy versus likelihood-ratio gradients under fixed weighting; negative log ratios favor fuzzy gradients. Colors distinguish raw and smoothed value targets. All fits share the unsmoothed contract's evaluation reference, using closed forms or precision-checked numerical estimates as specified in Appendix D.1.

the same input split, the same scaler fitted on the raw training labels, the same raw-value validation monitor and the common unsmoothed evaluation reference. Changing the derivative source therefore isolates that channel's effect within one training procedure; changing the value target changes only the value labels seen by the loss.

For each block we compare the ten-seed arithmetic mean MSEs, then report the median of their ratios over the nine blocks. A ratio below one favors the replacement. This differs from the mean of seed-paired log ratios.

Replacing likelihood-ratio gradients by fuzzy gradients lowers the value error in eight of nine blocks at each value target; the exceptions are the Black–Scholes digital with raw values and the one-dimensional basket with smoothed values, where the two derivative sources are close. The median value reductions are 48% with raw values and 31% with smoothed values, with the largest changes in the barrier and Heston digital blocks. The derivative error falls by a median 48% and 39%, in six and eight blocks respectively; the one-dimensional basket favors the likelihood-ratio gradient for derivative error at both value targets. The value-smoothing contrasts have median value ratios of 0.929 and 1.011 with mixed directions (five and four blocks lower) and no consistent derivative effect. The gradient-source contrast therefore persists with either value target; smoothing the value labels alone does not reproduce its pattern.

D.5 Barrier evaluation reference and complete comparison

The barrier training labels average a down-and-out call whose knock-out is checked at the 252 monitoring times $\Delta t, 2\Delta t, \ldots, T$ and not at time zero. The evaluation reference is the expected

Table 5: Value MSE in the four-way label control: ten-seed means against the expected unsmoothed payoff reference. Columns cross the value target with the derivative source.

	Raw values		Smoothed values	
Block	LRM gradients	Fuzzy gradients	LRM gradients	Fuzzy gradients
BS digital	$1.360 \cdot 10^{-4}$	$1.641 \cdot 10^{-4}$	$1.375 \cdot 10^{-4}$	$1.024 \cdot 10^{-4}$
Basket $d = 1$	$1.671 \cdot 10^{-4}$	$1.228 \cdot 10^{-4}$	$1.201 \cdot 10^{-4}$	$1.524 \cdot 10^{-4}$
Basket $d = 2$	$2.575 \cdot 10^{-4}$	$2.150 \cdot 10^{-4}$	$2.269 \cdot 10^{-4}$	$1.997 \cdot 10^{-4}$
Basket $d = 3$	$3.403 \cdot 10^{-4}$	$1.775 \cdot 10^{-4}$	$3.589 \cdot 10^{-4}$	$2.294 \cdot 10^{-4}$
Basket $d = 5$	$2.568 \cdot 10^{-4}$	$1.241 \cdot 10^{-4}$	$2.925 \cdot 10^{-4}$	$1.046 \cdot 10^{-4}$
Basket $d = 7$	$3.843 \cdot 10^{-4}$	$1.827 \cdot 10^{-4}$	$3.330 \cdot 10^{-4}$	$2.286 \cdot 10^{-4}$
Basket $d = 10$	$2.381 \cdot 10^{-4}$	$2.102 \cdot 10^{-4}$	$2.804 \cdot 10^{-4}$	$1.940 \cdot 10^{-4}$
BS barrier	$8.227 \cdot 10^{0}$	$4.018 \cdot 10^{-1}$	$6.484 \cdot 10^{0}$	$5.248 \cdot 10^{-1}$
Heston digital	$1.159 \cdot 10^{-3}$	$2.664 \cdot 10^{-4}$	$1.402 \cdot 10^{-3}$	$2.270 \cdot 10^{-4}$

Table 6: Derivative MSE in the four-way label control: ten-seed means against the derivative reference. Columns cross the value target with the derivative source.

	Raw values		Smoothed values	
Block	LRM gradients	Fuzzy gradients	LRM gradients	Fuzzy gradients
BS digital	$5.130 \cdot 10^{-7}$	$5.648 \cdot 10^{-7}$	$5.192 \cdot 10^{-7}$	$4.748 \cdot 10^{-7}$
Basket $d = 1$	$1.731 \cdot 10^{-6}$	$1.893 \cdot 10^{-6}$	$1.704 \cdot 10^{-6}$	$2.272 \cdot 10^{-6}$
Basket $d = 2$	$1.902 \cdot 10^{-6}$	$1.152 \cdot 10^{-6}$	$1.923 \cdot 10^{-6}$	$1.175 \cdot 10^{-6}$
Basket $d = 3$	$1.949 \cdot 10^{-6}$	$8.488 \cdot 10^{-7}$	$2.052 \cdot 10^{-6}$	$8.015 \cdot 10^{-7}$
Basket $d = 5$	$1.760 \cdot 10^{-6}$	$4.301 \cdot 10^{-7}$	$2.769 \cdot 10^{-6}$	$3.662 \cdot 10^{-7}$
Basket $d = 7$	$9.275 \cdot 10^{-7}$	$4.854 \cdot 10^{-7}$	$6.320 \cdot 10^{-7}$	$5.043 \cdot 10^{-7}$
Basket $d = 10$	$3.853 \cdot 10^{-7}$	$4.568 \cdot 10^{-7}$	$4.126 \cdot 10^{-7}$	$3.114 \cdot 10^{-7}$
BS barrier	$1.134 \cdot 10^{-1}$	$7.161 \cdot 10^{-3}$	$7.642 \cdot 10^{-2}$	$7.601 \cdot 10^{-3}$
Heston digital	$4.217 \cdot 10^{-5}$	$1.213 \cdot 10^{-6}$	$3.631 \cdot 10^{-5}$	$1.359 \cdot 10^{-6}$

Table 7: Which label construction improves prediction? The first comparison changes the full construction; the second fixes the raw value target and changes only the derivative source. Training weights are fixed in both comparisons.

	Value MSE ($\cdot 10^{-3}$) ↓			Derivative MSE ($\cdot 10^{-5}$) ↓			Lower error
Task / setting	B	D	Ratio [95% CI]	B	D	Ratio [95% CI]	pairs
Full label construction: pathwise B versus fuzzy D; fixed weights							
BS digital	15.4	0.102	**0.0053** [0.0036, 0.0081]	2.26	0.0475	**0.02** [0.014, 0.028]	10/10
Basket, six dimensions	3.23	0.185	**0.052** [0.043, 0.062]	1.66	0.0905	**0.077** [0.066, 0.089]	60/60
BS barrier	557	525	1.042 [0.77, 1.4]	463	760	**1.6** [1.4, 1.9]	5/10
Heston digital	12.2	0.227	**0.014** [0.0078, 0.025]	4.83	0.136	**0.025** [0.018, 0.035]	10/10
Derivative source only: LRM B versus fuzzy D; identical raw values							
BS digital	0.136	0.164	**1.4** [1.002, 2]	0.0513	0.0565	1.1 [0.65, 1.9]	2/10
Basket, six dimensions	0.274	0.172	**0.66** [0.58, 0.75]	0.144	0.0878	**0.72** [0.55, 0.9]	47/60
BS barrier	8230	402	**0.094** [0.045, 0.21]	11300	716	**0.13** [0.065, 0.3]	9/10
Heston digital	1.16	0.266	**0.25** [0.096, 0.56]	4.22	0.121	**0.034** [0.017, 0.074]	9/10

Errors are arithmetic means. Ratios are $\exp\{\overline{\log(E_D/E_B)}\}$, calculated separately for each metric; values below one favor D. Brackets give 95% paired-cluster bootstrap intervals. **Green** / **rust** bold ratios have intervals below / above one; uncolored ratios include one. The final column counts pairs with lower error in the left-hand metric. Basket dimensions: $\{1, 2, 3, 5, 7, 10\}$. Basket dimensions remain together within each of ten seed clusters.

Table 8: Complete Black–Scholes barrier comparison. Entries are ten-seed means and sample standard deviations against the discretely monitored reference. Derivative MSE is in units of 10^{-3}. Both endpoints use the same fitted model in every cell.

Method	Value MSE	Derivative MSE ($\cdot 10^{-3}$)
Value only	0.416 ± 0.265	5.867 ± 2.012
PW + fixed	0.557 ± 0.380	4.632 ± 1.250
PW + GradNorm	0.489 ± 0.187	4.421 ± 0.989
PW + warmup	0.412 ± 0.193	4.160 ± 0.799
PW + ReLoBRaLo	0.540 ± 0.296	4.334 ± 0.850
PW + uncertainty	0.602 ± 0.244	4.705 ± 0.962
PW + SoftAdapt	0.355 ± 0.140	4.133 ± 0.745
PW + PCGrad	0.547 ± 0.245	4.136 ± 1.134
PW + ConFIG	0.566 ± 0.275	4.301 ± 1.070
LRM + fixed	8.227 ± 12.757	113.417 ± 201.256
LRM + GradNorm	9.549 ± 11.430	132.089 ± 179.999
LRM + warmup	0.691 ± 0.463	14.366 ± 10.604
LRM + ReLoBRaLo	9.788 ± 16.126	112.085 ± 148.286
LRM + uncertainty	2.355 ± 1.746	52.786 ± 31.046
LRM + SoftAdapt	6.766 ± 9.147	80.618 ± 63.419
LRM + PCGrad	3.867 ± 3.676	75.747 ± 108.904
LRM + ConFIG	1.363 ± 0.772	72.527 ± 77.305
Fuzzy + fixed	0.525 ± 0.168	7.601 ± 2.300
Fuzzy + GradNorm	0.426 ± 0.177	7.713 ± 3.176
Fuzzy + warmup	0.430 ± 0.118	7.336 ± 1.102
Fuzzy + ReLoBRaLo	0.405 ± 0.149	7.769 ± 2.903
Fuzzy + uncertainty	0.551 ± 0.192	7.672 ± 1.692
Fuzzy + SoftAdapt	0.469 ± 0.339	6.331 ± 1.313
Fuzzy + PCGrad	0.503 ± 0.206	7.607 ± 1.799
Fuzzy + ConFIG	0.428 ± 0.265	6.939 ± 3.418

discounted payoff of exactly this discretely monitored contract. Writing $z = \ln(S/B)$ for the log-price above the barrier, one-step drift $\mu = (r - \sigma^2/2)\Delta t$ and one-step standard deviation $s = \sigma\sqrt{\Delta t}$, the undiscounted surviving value satisfies

$$V_n(z) = \int_0^\infty V_{n+1}(z')\,\frac{1}{s}\,\varphi\!\left(\frac{z' - z - \mu}{s}\right) dz', \qquad V_{252}(z) = \left(Be^z - K\right)_+, \tag{12}$$

where φ is the standard normal density and the integral is restricted to the surviving region $z' > 0$. The price reference is $e^{-rT}V_0(\ln(S_0/B))$, with the same discount applied to its spot derivative. The recursion is evaluated on a midpoint grid with 64 nodes per one-step log standard deviation by FFT convolution, and the reference delta differentiates the first transition density analytically. For every seed, halving the grid resolution changes values and deltas by less than 3×10^{-4}, and extending the truncation tail from 12 to 14 standard deviations changes them by less than 10^{-7}. The continuous-monitoring reflection formula describes a different contract and is not used as a reference.

Both value and derivative errors in Table 8 come from the same ten-seed fits, evaluated against this reference; the table includes the five balancing rules of the expanded panel on the same data. Every pathwise recipe has lower derivative error than value-only training (4.1–4.7×10^{-3} against 5.9×10^{-3}); SoftAdapt, PCGrad and warmup are within one percent of each other at the low end. The fuzzy recipes lie between 6.3 and 7.8×10^{-3}. The likelihood-ratio recipes have far larger and highly dispersed errors, with the warmup recipe the only one that keeps the value error within a factor of two of the value-only fit; its high-variance label is the mechanism discussed in the main text.

D.6 Expanded balancing, tuning and architecture checks

The extensions ask whether the ordering depends on a narrow choice of balancing rules, their defaults, or the reference MLP. The expanded panel adds ReLoBRaLo, uncertainty weighting, SoftAdapt, PCGrad and ConFIG to fixed weighting, GradNorm and warmup on the same nine blocks and ten seeds (1,350 additional fits). With eight balancing rules the construction-to-balancing rank ratio is 5.586 for value error and 13.463 for derivative error (3.011 and 9.417 under equal family weighting).

Table 9: One-factor changes within the fixed-budget stage experiment. The other two stage choices, block, construction and five seeds are held fixed. Each row contains 48 block–construction–setting comparisons; ratios are medians of ratios of five-seed arithmetic mean errors.

Change	Value ratio	Lower	Derivative ratio	Lower
Divide stage-two LR by ten	0.783	36/48	0.834	33/48
Use values alone in stage one	0.867	29/48	0.931	30/48
Reset optimizer at transition	0.981	26/48	0.988	26/48

A conflict-averse aggregation rule (CAGrad with $c = 0.5$; Liu et al., 2021) trained as a fourth rule beside fixed weighting, GradNorm and warmup gives ratios of 4.308 and 19.254 (1.882 and 10.605 under equal family weighting); its mean ranks track those of fixed weighting.

The gradient-statistic annealing rule of Wang et al. (2021) scales one loss by the ratio of the largest gradient magnitude of an anchored loss to the mean gradient magnitude of the other, with the retention and update interval of the released implementation. Its physics-informed formulation anchors the residual term, which has no unique counterpart among a value and a derivative loss. We compare both mappings on the same nine blocks, constructions and seeds (540 fits). The two mappings behave oppositely, and each mapping's effect reverses with the construction. Anchoring the derivative loss (the value loss is rescaled) lowers value error under pathwise construction in 87 of 90 seed pairs against fixed weighting (mean log ratio -2.45), gives mixed results for likelihood-ratio construction (41 of 90 improve), and raises the error of fuzzy construction (26 of 90, $+0.50$). Anchoring the value loss (the derivative loss is rescaled) does the reverse: pathwise error rises in 86 of 90 pairs ($+1.48$) while fuzzy error falls modestly (52 of 90, -0.17). Over five rules the construction-to-balancing ratios are 2.538 for value error and 6.237 for derivative error (0.988 and 3.651 under equal family weighting); the mapping choice is itself a large balancing effect, but its effect depends on the derivative construction. The comparison identifies the effect of the complete weighting intervention; its absolute loss scale and resulting optimization trajectory also change. Fuzzy warmup remains the best-ranked value-error cell.

The tuning panel selects among GradNorm exponents $\{0.5, 1, 1.5\}$, ReLoBRaLo temperatures $\{0.1, 0.25, 0.5\}$ and warmup fractions $\{0.05, 0.1, 0.5\}$, chosen for each block and construction by the mean raw-value validation MSE over the first five seeds, with ties resolved toward the default and then the smaller value. Test errors of the chosen settings on those five seeds give ratios of 3.620 and 8.801 (2.918 and 3.774 under equal family weighting). This hyperparameter-selection rule is distinct from the checkpoint rule, which is the same raw-value validation monitor for every fit (Section D.3).

Warmup as an optimization recipe. The warmup rule bundles three choices: a value-only first stage, a tenfold learning-rate reduction at the transition, and a fresh optimizer. A $2 \times 2 \times 2$ control separates them on four blocks (the Black–Scholes digital, the seven-dimensional basket, the barrier and the Heston digital), three constructions and five seeds (480 fits): the first stage trains on values alone or on the joint GradNorm objective, the second-stage learning rate is divided by one or by ten, and the optimizer state is either kept or reset; both stages take 250 epochs without early stopping, the best-monitor checkpoint of the first stage is restored at the transition, and the scheduler and GradNorm state restart identically. Table 10 compares each variant with the core warmup recipe restricted to the same five seeds. Because the control uses fixed stage lengths while the core allows early stopping, the direct factorial contrasts in Table 9 isolate the three choices within the common 500-epoch budget. Holding the other two choices fixed, reducing the second-stage learning rate lowers mean value error in 36 of 48 block–construction–setting comparisons, with a median ratio of 0.783. A value-only first stage improves 29 of 48 comparisons (median 0.867), whereas optimizer reset has a smaller median effect (0.981). Both the initial objective and the refinement learning rate therefore contribute to the warmup recipe; their effects depend on the construction and the other stage choices. These controls distinguish the information supplied by the labels from the optimization used to exploit it.

A coordinate-weighting control replaces the dimension-aware fixed weights of Eq. (11) by equal weights $1/2$ in every block and construction (270 fits). The two rules coincide and give identical results in the four one-dimensional blocks. In the five multidimensional baskets, equal weights increase both fuzzy errors in every block (median block-mean ratios 1.92 for values and 1.90 for

Table 10: Warmup stage controls: median over twelve block–construction cells of the ratio of five-seed mean errors to the core warmup recipe on the same cells; counts give cells with a lower mean.

First stage	LR divisor	Reset	Value ratio	Lower	Derivative ratio	Lower
value	1	off	1.529	1/12	1.179	3/12
value	1	on	1.476	1/12	1.442	3/12
value	10	off	1.063	5/12	1.024	5/12
value	10	on	0.933	8/12	1.038	5/12
joint (GradNorm)	1	off	1.653	2/12	1.190	2/12
joint (GradNorm)	1	on	1.879	3/12	1.148	2/12
joint (GradNorm)	10	off	1.171	4/12	1.042	5/12
joint (GradNorm)	10	on	1.278	3/12	1.055	5/12

Table 11: Mean within-block ranks of the nine factorial cells on the Transformer backbone (nine blocks, five seeds). Each rank is based on the five-seed mean MSE; blocks contribute equally.

	Value MSE			Derivative MSE		
Balancer	PW	LRM	Fuzzy	PW	LRM	Fuzzy
Fixed	8.333	5.000	2.889	7.778	4.111	2.778
GradNorm	8.000	5.889	4.000	7.000	5.111	2.778
Warmup	5.444	3.111	2.333	6.111	4.333	5.000

derivatives), while reducing both pathwise errors in every block (ratios 0.264 and 0.354). Likelihood-ratio effects are mixed, with increases in three of five blocks. Reducing derivative influence helps the almost-everywhere-zero pathwise target but harms the informative fuzzy target. Coordinate weighting therefore changes how effectively a construction serves the prediction task.

Transformer backbone. A four-block Transformer with 64-dimensional tokens replaces the MLP on the same nine blocks, three constructions and three balancing rules for the first five seeds (450 fits, with value-only references). The construction ordering is unchanged (Table 11): fuzzy construction beats pathwise in 26 of 27 block–balancer contrasts on value error and likelihood-ratio construction in 23 of 27, and Fuzzy + warmup is again the best-ranked value-error cell. The construction-to-balancing ratios are 3.004 for value error and 176.077 for derivative error (0.961 and 5.217 under equal family weighting); the derivative ratio is large because the three balancing rules barely change derivative ranks on this backbone. Against the MLP on the same block and recipe, the Transformer's block-mean value error is close to parity (median ratio 0.99, lower in 41 of 81 recipe–block cells) and its derivative error somewhat higher (median 1.21), so the backbone changes the error level far less than the label construction does.

The additional balancing rules, tuning, backbone and stage controls thus test distinct ways of using the supplied labels. Their effects depend on the construction, as the matched comparisons above make explicit.

D.7 Sensitivity of the balancing rules

We also test balancing sensitivity beyond the construction core. Five-seed sweeps vary the GradNorm rate exponent $\alpha \in \{0.5, 1, 1.5, 2\}$ and ReLoBRaLo temperature $\tau \in \{0.1, 0.25, 0.5, 1\}$. The synthetic aggregate combines poly-trig, trig and step cells with $d = 5, n = 1{,}024$; Burgers is reported separately. Error magnitudes are compared within columns.

The synthetic aggregate varies little relative to its dispersion, whereas Burgers shows occasional unstable settings. Balancer sensitivity is therefore application dependent. The table evaluates each parameter setting separately; it does not evaluate a validation-selected tuning procedure.

Table 12: Mean value MSE and sample standard deviation under balancing-parameter changes. Synthetic entries pool 15 family–seed cells (three families, five seeds); Burgers entries use five seed cells.

Rule	Parameter	Synthetic	Burgers
ReLoBRaLo	$\tau = 0.10$	2.97 ± 2.33	0.065 ± 0.014
	$\tau = 0.25$	3.00 ± 2.36	0.219 ± 0.361
	$\tau = 0.50$	3.27 ± 2.66	0.067 ± 0.018
	$\tau = 1.00$	3.30 ± 2.71	0.086 ± 0.075
GradNorm	$\alpha = 0.5$	2.80 ± 2.25	0.241 ± 0.138
	$\alpha = 1.0$	2.77 ± 2.24	0.287 ± 0.125
	$\alpha = 1.5$	2.77 ± 2.25	0.228 ± 0.155
	$\alpha = 2.0$	2.77 ± 2.25	0.518 ± 0.744

D.8 A SIMPLIFIED RELATIVE-LOSS BALANCING ALTERNATIVE

We also replace ReLoBRaLo's lookback mechanism by softmax weighting of recent relative losses. The comparison contains 2,670 synthetic pairs, 50 molecular MLP pairs and 50 molecular GAT pairs. Table 13 reports median base-ten log error ratios.

Table 13: Simplified relative-loss rule divided by ReLoBRaLo. Negative median $\log_{10}$ error ratio favors the simplified rule. Lower-error counts refer to paired comparisons within the tested task set.

Panel	Value log ratio	Gradient log ratio	Lower value error	Lower gradient error
Synthetic	$+0.009$	$+0.014$	1119/2670	1031/2670
Molecular MLP	-0.003	$+0.015$	26/50	18/50
Molecular GAT	-0.605	-0.542	50/50	50/50

The simplified rule improves both errors in every GAT pair, while the MLP and synthetic comparisons are mixed. These comparisons concern the complete MLP and GAT training procedures. Both molecular panels evaluate Cartesian-force MSE; their ratios are distinct from the force-MAE comparison in Appendix K. A 10^{-12} floor is applied before forming log ratios.

D.9 STOCHASTIC-VOLATILITY BARRIER EXTENSION

The barrier experiment adds a path-dependent payoff and stochastic volatility. Its discounted payoff is

$$P = e^{-rT_2}(S_{T_2} - K)_+ \mathbf{1}\{S_{T_1} > B\}, \qquad T_1 = 1/3, \quad T_2 = 2/3. \tag{13}$$

The barrier is checked once, at T_1. Under Heston dynamics we use $K = 1$, $B = 0.85$, $v_0 = \bar{v} = 0.04$, mean reversion $\kappa = 1$, volatility-of-variance $\xi = 0.15$, correlation $c = -0.7$, and $r = 0$. Full-truncation Euler simulation uses 84 substeps in each of the two time intervals. Initial spots are uniform on either $[0.7K, 1.3K]$ or $[0.5K, 1.5K]$. Each seed generates 1024 input spots with ten paths per spot, then uses an 819/205 training/validation split. An independent set of 200 uniformly sampled spots with 100,000 paths per spot supplies a common unsmoothed price and delta reference. It is separate from both optimization and selection.

The PW label retains the call's pathwise sensitivity while omitting the barrier-boundary contribution. In particular, it is not an all-zero label:

$$g_{\mathrm{PW}} = e^{-rT_2}\mathbf{1}\{S_{T_1} > B\}\mathbf{1}\{S_{T_2} > K\}\frac{S_{T_2}}{S_0}. \tag{14}$$

Fuzzy construction replaces the barrier indicator in Equation 13 by $S_{\epsilon_B}(S_{T_1} - B)$ and differentiates the entire product:

$$g_{\mathrm{fuzzy}} = e^{-rT_2}\Bigg[\mathbf{1}\{S_{T_2} > K\}\frac{S_{T_2}}{S_0}S_{\epsilon_B}(S_{T_1} - B) \tag{15}$$

$$+ (S_{T_2} - K)_+ S'_{\epsilon_B}(S_{T_1} - B)\frac{S_{T_1}}{S_0}\Bigg]. \tag{16}$$

Table 14: Heston barrier neural comparison: five-seed means and population standard deviations. Price MSE is in units of 10^{-5} and delta MSE in units of 10^{-3}. All methods use the same independent evaluation reference in each range.

Method	$S_0/K \in [0.7, 1.3]$ Price	$S_0/K \in [0.7, 1.3]$ Delta	$S_0/K \in [0.5, 1.5]$ Price	$S_0/K \in [0.5, 1.5]$ Delta
Value only	2.101 ± 1.599	4.632 ± 3.198	3.223 ± 1.927	2.428 ± 1.634
PW + fixed	0.754 ± 0.452	0.387 ± 0.206	1.110 ± 1.343	0.197 ± 0.124
PW + GradNorm	1.123 ± 0.964	0.440 ± 0.245	1.204 ± 1.046	0.229 ± 0.093
PW + ReLoBRaLo	1.325 ± 1.404	0.311 ± 0.221	2.247 ± 1.868	0.316 ± 0.140
PW + warmup	0.481 ± 0.410	0.209 ± 0.037	0.979 ± 0.787	0.193 ± 0.118
LRM + fixed	33.665 ± 8.932	101.712 ± 107.870	538.956 ± 449.299	84.153 ± 20.197
LRM + GradNorm	260.111 ± 207.507	32.209 ± 14.929	893.195 ± 1198.115	94.796 ± 115.496
LRM + warmup	2.832 ± 1.201	18.983 ± 12.434	7.555 ± 3.186	18.440 ± 6.285
Multistep LRM + fixed	1.727 ± 0.801	1.958 ± 0.965	9.934 ± 8.629	3.491 ± 1.307
Multistep LRM + warmup	1.391 ± 0.810	2.974 ± 1.858	2.366 ± 1.482	2.627 ± 1.814
Fuzzy + fixed	0.797 ± 0.619	0.360 ± 0.284	1.785 ± 1.504	0.192 ± 0.105
Fuzzy + warmup	0.227 ± 0.068	0.185 ± 0.059	0.882 ± 0.688	0.183 ± 0.057

Table 15: Price and delta prediction on the Heston barrier task.

Heston prices and deltas: value-only B versus fuzzy warmup D

Task / setting	Value MSE ($\cdot 10^{-5}$) ↓ B	Value MSE ($\cdot 10^{-5}$) ↓ D	Value MSE ($\cdot 10^{-5}$) ↓ Ratio [95% CI]	Derivative MSE ($\cdot 10^{-3}$) ↓ B	Derivative MSE ($\cdot 10^{-3}$) ↓ D	Derivative MSE ($\cdot 10^{-3}$) ↓ Ratio [95% CI]	Lower error pairs
Narrow range	2.1	0.227	**0.18** [0.063, 0.67]	4.63	0.185	**0.063** [0.026, 0.17]	4/5
Wide range	3.22	0.882	0.29 [0.061, 2.2]	2.43	0.183	**0.1** [0.058, 0.22]	4/5

Errors are arithmetic means. Ratios are paired geometric means, computed separately for each metric; brackets give 95% paired bootstrap intervals. Green and rust ratios have intervals below and above one; uncolored intervals include one. The final column counts lower-error pairs for the left-hand metric. Intervals resample the five paired seeds.

Here ϵ_B is half the empirical standard deviation of $S_{T_1} - B$ in an independent calibration simulation over the same spot range. Single-step and multistep likelihood-ratio constructions provide further comparators. The single-step score uses the joint spot/variance transition, including the spot–variance correlation. These estimators change the derivative labels while the balancing rule controls their contribution to training.

Table 14 contains all 120 neural fits: twelve recipes, five seeds $(42, 123, 456, 789, 1337)$ and two ranges. The network and epoch cap follow the core recipe. The non-warmup fits use early-stopping patience 200; the two-stage warmup recipe uses patience 50 within each stage. ReLoBRaLo adds a relative-loss balancing comparator (Bischof & Kraus, 2022). Fuzzy warmup has the lowest mean price and delta MSE in both ranges. Relative to value-only training it lowers price error in four of five seeds in each range and delta error in all five. The large single-step LRM errors and their reduction with multistep labels also show that the choice of score estimator remains material after adopting a valid likelihood-ratio construction.

D.10 Polynomial Heston barrier comparator

The Heston experiment varies initial spot while holding the other pricing parameters fixed. We fit a degree-five polynomial by ordinary least squares to raw payoff observations from the neural experiment's generator and seed-specific training split: 819 of the 1,024 sampled locations, with ten Monte Carlo paths per location. The remaining 205 locations are excluded from fitting. The degree is fixed across both spot ranges and all five seeds. Prices and analytical derivatives of the fitted polynomial are evaluated against the same independent 200-location, 100,000-path reference used for the neural comparison in Appendix D.9. Its delta reference uses central differences with a spot bump of 0.001 and common random numbers.

The quintic has lower mean price error than the value-only neural model in both ranges, but higher mean delta error. Fuzzy warmup has lower mean errors than the quintic for both quantities, improving price prediction in four of five paired seeds and delta prediction in all five within each range. The compact polynomial is therefore a useful value-only comparator, while the derivative-trained neural procedure achieves the stronger joint prediction result in this study.

Table 16: Heston barrier comparison with a polynomial trained on raw payoff observations. Entries are means ± population standard deviations over five seeds, conditional on the common independent Monte Carlo evaluation reference. Price MSE is in units of 10^{-5} and delta MSE in units of 10^{-3}. Fuzzy warmup changes both value and derivative training targets.

Range	Model	Price MSE	Delta MSE
Narrow	Quintic	1.144 ± 0.978	5.754 ± 6.886
	Value-only neural	2.101 ± 1.599	4.632 ± 3.198
	Fuzzy warmup	0.227 ± 0.068	0.185 ± 0.059
Wide	Quintic	1.952 ± 1.526	5.349 ± 5.160
	Value-only neural	3.223 ± 1.927	2.428 ± 1.634
	Fuzzy warmup	0.882 ± 0.688	0.183 ± 0.057

D.11 Bandwidth sensitivity of the Heston barrier construction

Smoothing introduces a second design choice within a derivative construction. Figure 5 varies the Heston barrier ramp width as $\varepsilon_B = c\,\widehat{\mathrm{sd}}(S_{T_1} - B)$, with $c \in \{0.1, 0.25, 0.5, 0.75, 1\}$; the scale is estimated by a separate calibration simulation for each seed. Five paired seeds share their input spots and simulation draws across the sweep. The training targets are the smoothed payoff and its pathwise derivative, so both change with bandwidth. This experiment measures sensitivity of the full smoothed construction; the crossed comparisons in Appendix D.4 separately hold the value labels fixed.

The model uses 1,024 generated spots, ten paths per spot, a four-layer width-256 softplus network, fixed derivative weighting and a 500-epoch limit. Its internal 80/20 split provides 819 gradient-update and 205 validation examples. Evaluation uses the same 200 spots and 100,000-path Heston reference for every bandwidth and training seed; reference deltas use central differences with common random numbers and a spot bump of 0.001. The test target remains the unsmoothed contract. The seed intervals are conditional on this shared Monte Carlo reference.

Across the tested endpoints, increasing c from 0.1 to 1 changes mean value MSE by a factor of 0.45 and mean delta MSE by a factor of 0.64. Thus a broader ramp improves both prediction quantities over these tested endpoints, despite changing the smoothed training target.

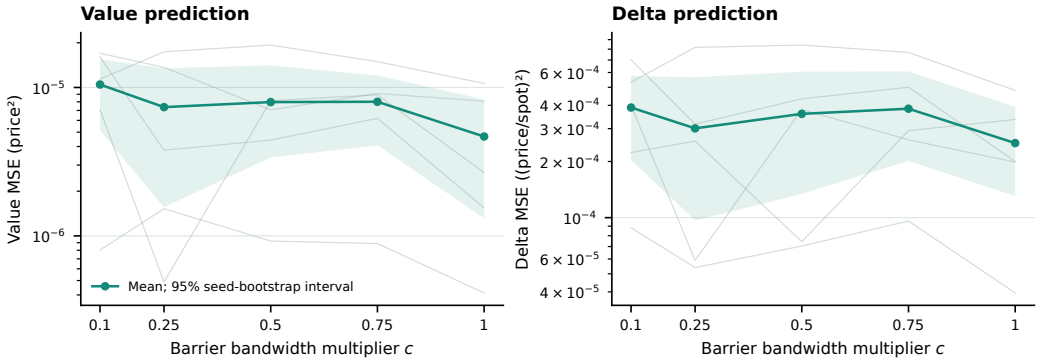

Figure 5: Heston barrier sensitivity to the bandwidth of the full smoothed label construction. Thin lines retain the five individual seeds; thick lines show arithmetic means and shading gives pointwise 95% seed-bootstrap intervals. Value and delta errors use a common, unsmoothed, independent reference for every bandwidth. Values are in price-squared units; deltas are in squared price/spot units. Both value and derivative training labels change with bandwidth.

D.12 Block ranks and paired effects

We first summarize the seven-block digital-and-basket subset, then the full nine-block core.

Table 17: Mean within-block ranks of the nine factorial cells. Each rank is based on ten-seed mean MSE; seven blocks contribute equally.

	Value MSE			Derivative MSE		
Balancer	PW	LRM	Fuzzy	PW	LRM	Fuzzy
Fixed	8.714	5.286	3.000	8.714	4.714	2.429
GradNorm	8.000	5.000	4.143	7.000	4.429	4.000
Warmup	6.429	3.000	1.429	7.000	4.286	2.429

Table 18: Descriptive decomposition of rank sum of squares; the total is 420 for each metric. The last column compares rank contributions, not prediction errors.

Metric	SS_C	SS_A	$SS_{C\times A}$	SS_R	SS_C/SS_A
Value	258.000	60.095	6.762	95.143	4.293
Derivative	232.667	6.000	19.905	161.429	38.778

Seven-block digital-and-basket subset. Let E_{jcas} be the evaluation error for block j, construction c, balancer a, and seed s. For each metric separately, first average the ten seeds, then rank the nine construction–balancer cells within a block: $r_{jca} = \mathrm{rank}_{ca}(10^{-1}\sum_s E_{jcas})$. Rank 1 is best; the value-only baseline is not ranked in this factorial decomposition. There are no ties in these block means. Writing bars for averages over the omitted indices, the rank sum of squares decomposes as

$$SS_C = 21\sum_c(\bar{r}_{\cdot c\cdot} - 5)^2, \qquad SS_A = 21\sum_a(\bar{r}_{\cdot\cdot a} - 5)^2, \tag{17}$$

$$SS_{C\times A} = 7\sum_{ca}(\bar{r}_{\cdot ca} - \bar{r}_{\cdot c\cdot} - \bar{r}_{\cdot\cdot a} + 5)^2, \qquad SS_R = \sum_{jca}(r_{jca} - \bar{r}_{\cdot ca})^2. \tag{18}$$

Their sum is $\sum_{jca}(r_{jca} - 5)^2 = 420$. The residual includes block-specific interactions. These are descriptive rank sums of squares, not fractions of neural-network prediction variance and not accuracy ratios.

Construction accounts for 61.4% of total value-error rank variation and 55.4% of derivative-error rank variation, compared with 14.3% and 1.4% for balancing. The construction-to-balancer rank-SS ratios are 4.29 and 38.8. The Friedman statistic over the nine methods and seven blocks is $Q = 43.314$ for values and $Q = 34.476$ for derivatives. The usual χ^2_8 reference gives $p = 7.7\times10^{-7}$ and $p = 3.3\times10^{-5}$, respectively. Because six blocks are dimensions of one basket family, these reference p-values do not establish significance across independent application domains. We therefore interpret the rankings alongside the matched construction contrasts.

Fuzzy construction lowers the block-mean value MSE relative to PW in all 21 block–balancer contrasts, and LRM construction in 20 of 21 (the exception is the ten-dimensional basket under GradNorm). For derivative MSE each construction improves on PW in 19 of 21 contrasts; the two exceptions are the seven- and ten-dimensional baskets under GradNorm, where all three constructions have small errors. This joint improvement is more informative than a derivative-error gain alone. It shows that avoiding a misleading zero-derivative target improves prediction of the expected values themselves. For fuzzy versus PW, the geometric means across blocks of the ratios of ten-seed mean value MSE are 0.045 with fixed weights, 0.097 with GradNorm and 0.277 with warmup. These are comparisons between constructions; improvements over value-only training are smaller.

We also report seed-paired comparisons to the shared value-only baseline. Define $\ell_{jcas} = \ln(E_{jcas}/E_{j,\mathrm{value},s})$. Table 19 gives the mean over the 70 block–seed pairs and the number with $\ell < 0$. Its exponential is a geometric error ratio. It differs from a ratio of arithmetic mean errors and from an average percentage reduction. These paired summaries give each block equal weight. The pathwise recipes with fixed weighting or GradNorm are worse than value-only training in most pairs, whereas fuzzy warmup improves on it in 65 of 70 value pairs and 68 of 70 derivative pairs.

Nine-block core and family weighting. For nine blocks, the rank decomposition in Eqs. (17)–(18) uses factors 27 instead of 21 for the two main effects and 9 instead of 7 for the interaction. The

Table 19: Paired effects relative to value-only training. Negative mean natural-log error ratios favor derivative supervision. Counts record strictly lower MSE on the same block and seed.

	Value MSE		Derivative MSE	
Method	Mean ℓ	Lower error	Mean ℓ	Lower error
PW + fixed	+2.4572	0/70	+1.4506	5/70
LRM + fixed	-0.5052	57/70	-0.9836	63/70
Fuzzy + fixed	-0.8287	58/70	-1.3131	68/70
PW + GradNorm	+1.8751	10/70	+0.9004	21/70
LRM + GradNorm	-0.4111	53/70	-0.7732	58/70
Fuzzy + GradNorm	-0.5222	53/70	-0.8929	59/70
PW + warmup	-0.0497	34/70	-0.2095	43/70
LRM + warmup	-0.7941	66/70	-0.9009	66/70
Fuzzy + warmup	-1.4846	65/70	-1.5673	68/70

Table 20: Sensitivity of the construction-to-balancing rank sum-of-squares ratio to task-family weighting. Family exclusions start from equal family weighting. Ratios describe variation in ranks, not factors of error reduction.

Panel weighting	Value MSE	Derivative MSE
Nine equally weighted blocks	3.202	15.515
Four equally weighted families	1.326	8.885
Without BS barrier	1.955	26.394
Without baskets	0.812	6.840
Without BS digital	2.730	4.127
Without Heston digital	0.864	8.407

total rank sum of squares is 540. The value-error decomposition is $SS_C = 257.852$, $SS_A = 80.519$, $SS_{C\times A} = 3.852$ and $SS_R = 197.778$, giving $SS_C/SS_A = 3.202$. The ratio ranges from 2.676 to 4.298 when one block is removed. Equal family weighting assigns each basket dimension weight $1/6$ and each of the other three blocks weight one. It averages within-block ranks, so errors on different physical scales are never averaged together. The construction-to-balancing ratios then become 1.326 for value error and 8.885 for derivative error. Table 20 also removes one family at a time. The value-rank advantage therefore depends in part on the panel's coverage of basket dimensions; the matched within-block contrasts identify the effect of changing the labels. For derivative MSE, $SS_C = 196.519$ and $SS_A = 12.667$, giving $SS_C/SS_A = 15.515$. The Friedman statistic over the nine recipes and nine blocks is $Q = 45.630$ for values and $Q = 30.163$ for derivatives; because six blocks are dimensions of one family, these are descriptive summaries rather than tests across independent domains.

D.13 Complete seven-block errors

Tables 21–24 report all nine factorial cells and the shared baseline. Each entry is a ten-seed mean with sample standard deviation; notation $(a \pm b)10^k$ means $(a \pm b) \times 10^k$. Unrounded block means determine the ranks above.

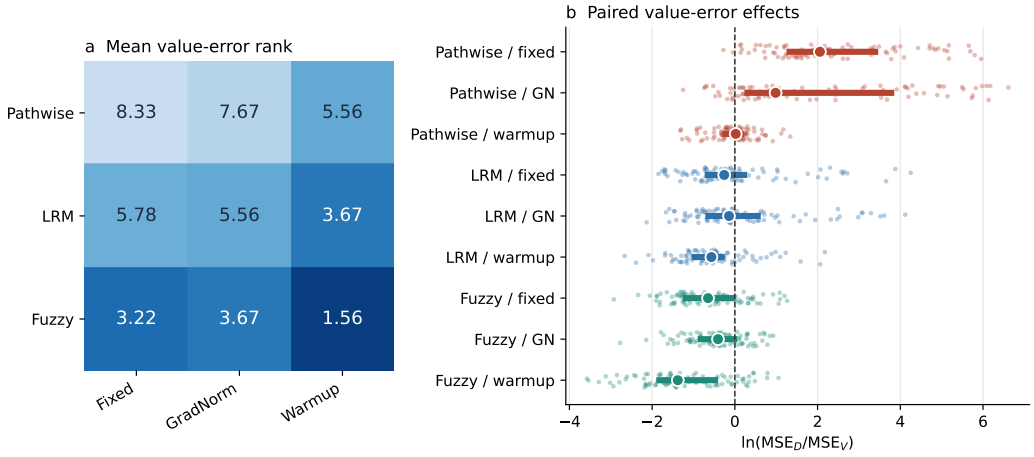

Figure 6: Value-error ranks and paired effects in the nine-block core. Ranks average the nine factorial cells within each block after averaging ten seed errors. Small points show the 90 block–seed ratios per recipe; large points and segments show medians and interquartile ranges. Negative natural-log ratios favor the derivative recipe over the same block and seed's value-only fit. Six blocks are dimensions of one family.

Table 21: Value MSE for the digital-and-basket subset, part A. All entries average ten seeds; lower is better.

Method	BS digital	Basket $d = 1$	Basket $d = 2$
Value only	$(1.57 \pm 2.50)10^{-4}$	$(1.95 \pm 1.29)10^{-4}$	$(4.39 \pm 2.20)10^{-4}$
PW + fixed	$(1.54 \pm 0.39)10^{-2}$	$(5.75 \pm 1.23)10^{-3}$	$(4.12 \pm 1.30)10^{-3}$
LRM + fixed	$(1.36 \pm 2.44)10^{-4}$	$(1.67 \pm 1.23)10^{-4}$	$(2.57 \pm 1.83)10^{-4}$
Fuzzy + fixed	$(1.02 \pm 0.86)10^{-4}$	$(1.52 \pm 0.82)10^{-4}$	$(2.00 \pm 1.55)10^{-4}$
PW + GradNorm	$(2.74 \pm 0.56)10^{-2}$	$(8.92 \pm 2.52)10^{-3}$	$(3.19 \pm 1.08)10^{-3}$
LRM + GradNorm	$(6.22 \pm 3.56)10^{-5}$	$(1.03 \pm 0.86)10^{-4}$	$(3.27 \pm 3.19)10^{-4}$
Fuzzy + GradNorm	$(1.18 \pm 0.53)10^{-4}$	$(8.41 \pm 3.03)10^{-5}$	$(2.17 \pm 1.34)10^{-4}$
PW + warmup	$(7.81 \pm 6.13)10^{-5}$	$(2.20 \pm 1.23)10^{-4}$	$(3.35 \pm 0.57)10^{-4}$
LRM + warmup	$(5.26 \pm 3.49)10^{-5}$	$(1.04 \pm 0.64)10^{-4}$	$(1.30 \pm 0.44)10^{-4}$
Fuzzy + warmup	$(6.40 \pm 2.59)10^{-5}$	$(9.52 \pm 7.68)10^{-5}$	$(9.76 \pm 6.37)10^{-5}$

Table 22: Value MSE for the digital-and-basket subset, part B. All entries average ten seeds; lower is better.

Method	Basket $d = 3$	Basket $d = 5$	Basket $d = 7$	Basket $d = 10$
Value only	$(7.18 \pm 2.68)10^{-4}$	$(3.30 \pm 1.32)10^{-4}$	$(5.21 \pm 1.78)10^{-4}$	$(4.92 \pm 3.05)10^{-4}$
PW + fixed	$(3.67 \pm 0.85)10^{-3}$	$(2.45 \pm 0.47)10^{-3}$	$(1.86 \pm 0.44)10^{-3}$	$(1.54 \pm 0.25)10^{-3}$
LRM + fixed	$(3.40 \pm 1.54)10^{-4}$	$(2.57 \pm 2.11)10^{-4}$	$(3.84 \pm 3.04)10^{-4}$	$(2.38 \pm 1.08)10^{-4}$
Fuzzy + fixed	$(2.29 \pm 1.35)10^{-4}$	$(1.05 \pm 0.69)10^{-4}$	$(2.29 \pm 2.63)10^{-4}$	$(1.94 \pm 0.82)10^{-4}$
PW + GradNorm	$(1.48 \pm 0.73)10^{-3}$	$(6.89 \pm 3.25)10^{-4}$	$(5.01 \pm 1.78)10^{-4}$	$(5.22 \pm 1.05)10^{-4}$
LRM + GradNorm	$(4.27 \pm 3.02)10^{-4}$	$(2.68 \pm 0.84)10^{-4}$	$(3.54 \pm 1.31)10^{-4}$	$(5.24 \pm 2.35)10^{-4}$
Fuzzy + GradNorm	$(2.30 \pm 1.32)10^{-4}$	$(1.86 \pm 0.74)10^{-4}$	$(3.61 \pm 2.35)10^{-4}$	$(4.90 \pm 2.19)10^{-4}$
PW + warmup	$(5.45 \pm 1.99)10^{-4}$	$(4.43 \pm 1.24)10^{-4}$	$(5.40 \pm 2.01)10^{-4}$	$(4.59 \pm 1.41)10^{-4}$
LRM + warmup	$(3.14 \pm 2.18)10^{-4}$	$(1.72 \pm 0.87)10^{-4}$	$(2.35 \pm 1.06)10^{-4}$	$(3.02 \pm 1.45)10^{-4}$
Fuzzy + warmup	$(1.16 \pm 1.35)10^{-4}$	$(6.34 \pm 4.43)10^{-5}$	$(1.13 \pm 1.11)10^{-4}$	$(8.70 \pm 4.74)10^{-5}$

Table 23: Derivative MSE for the digital-and-basket subset, part A. All entries average ten seeds; lower is better.

Method	BS digital	Basket $d = 1$	Basket $d = 2$
Value only	$(1.23 \pm 0.72)10^{-6}$	$(9.11 \pm 6.90)10^{-6}$	$(8.81 \pm 8.08)10^{-6}$
PW + fixed	$(2.26 \pm 0.48)10^{-5}$	$(5.90 \pm 1.55)10^{-5}$	$(2.00 \pm 0.59)10^{-5}$
LRM + fixed	$(5.13 \pm 3.84)10^{-7}$	$(1.73 \pm 1.43)10^{-6}$	$(1.90 \pm 1.00)10^{-6}$
Fuzzy + fixed	$(4.75 \pm 2.04)10^{-7}$	$(2.27 \pm 1.23)10^{-6}$	$(1.18 \pm 0.47)10^{-6}$
PW + GradNorm	$(3.54 \pm 0.68)10^{-5}$	$(8.77 \pm 2.65)10^{-5}$	$(1.41 \pm 0.56)10^{-5}$
LRM + GradNorm	$(3.42 \pm 1.48)10^{-7}$	$(1.46 \pm 1.07)10^{-6}$	$(2.01 \pm 1.90)10^{-6}$
Fuzzy + GradNorm	$(4.18 \pm 2.01)10^{-7}$	$(1.65 \pm 1.10)10^{-6}$	$(1.38 \pm 0.60)10^{-6}$
PW + warmup	$(8.02 \pm 3.34)10^{-7}$	$(4.44 \pm 2.32)10^{-6}$	$(4.34 \pm 1.81)10^{-6}$
LRM + warmup	$(5.28 \pm 3.05)10^{-7}$	$(1.94 \pm 1.76)10^{-6}$	$(1.89 \pm 1.93)10^{-6}$
Fuzzy + warmup	$(4.24 \pm 2.69)10^{-7}$	$(2.41 \pm 2.59)10^{-6}$	$(1.41 \pm 1.03)10^{-6}$

Table 24: Derivative MSE for the digital-and-basket subset, part B. All entries average ten seeds; lower is better.

Method	Basket $d = 3$	Basket $d = 5$	Basket $d = 7$	Basket $d = 10$
Value only	$(7.92 \pm 10.82)10^{-6}$	$(1.26 \pm 0.63)10^{-6}$	$(1.35 \pm 0.51)10^{-6}$	$(7.07 \pm 3.91)10^{-7}$
PW + fixed	$(1.14 \pm 0.25)10^{-5}$	$(4.95 \pm 0.83)10^{-6}$	$(2.73 \pm 0.57)10^{-6}$	$(1.67 \pm 0.34)10^{-6}$
LRM + fixed	$(1.95 \pm 1.44)10^{-6}$	$(1.76 \pm 3.32)10^{-6}$	$(9.27 \pm 10.02)10^{-7}$	$(3.85 \pm 2.67)10^{-7}$
Fuzzy + fixed	$(8.02 \pm 2.95)10^{-7}$	$(3.66 \pm 1.88)10^{-7}$	$(5.04 \pm 5.39)10^{-7}$	$(3.11 \pm 1.87)10^{-7}$
PW + GradNorm	$(4.40 \pm 1.97)10^{-6}$	$(1.42 \pm 0.55)10^{-6}$	$(9.26 \pm 3.44)10^{-7}$	$(7.77 \pm 2.61)10^{-7}$
LRM + GradNorm	$(2.98 \pm 2.29)10^{-6}$	$(1.15 \pm 0.65)10^{-6}$	$(1.06 \pm 0.69)10^{-6}$	$(9.08 \pm 5.42)10^{-7}$
Fuzzy + GradNorm	$(1.11 \pm 0.40)10^{-6}$	$(8.07 \pm 2.98)10^{-7}$	$(1.15 \pm 0.64)10^{-6}$	$(1.15 \pm 0.97)10^{-6}$
PW + warmup	$(5.78 \pm 6.31)10^{-6}$	$(1.46 \pm 0.53)10^{-6}$	$(1.24 \pm 0.49)10^{-6}$	$(7.18 \pm 3.10)10^{-7}$
LRM + warmup	$(4.08 \pm 5.74)10^{-6}$	$(6.86 \pm 4.47)10^{-7}$	$(7.25 \pm 3.15)10^{-7}$	$(5.69 \pm 3.36)10^{-7}$
Fuzzy + warmup	$(8.62 \pm 10.89)10^{-7}$	$(3.21 \pm 2.04)10^{-7}$	$(3.40 \pm 3.26)10^{-7}$	$(2.02 \pm 1.26)10^{-7}$

E NOISE SENSITIVITY AND EMPIRICAL CROSSINGS

The matched panel contains 165 fits: three families, five seeds $(42, 123, 456, 789, 1000)$, one value-only fit per family and seed, and derivative-supervised fits at $\rho \in \{0, 0.05, 0.1, 0.2, 0.5\}$ under two validation monitors. Each family is one configuration: poly-trig and trig in $d = 5$ and the smooth Bachelier basket price in $d = 10$, each with $1{,}024$ generated inputs split 655/164/205 into training, validation and evaluation. Within a family and seed, every fit shares the target function, the inputs and their split, the initialization, and one standardized Gaussian corruption draw; the noise level only rescales that draw. The derivative-supervised recipe is the dimension-aware fixed weighting of Eq. (11); the value-only reference is trained once per family and seed. All fits use the core architecture, optimizer and epoch budget (Appendix C), batch size 512, a training-only scaler, and the raw-value validation monitor unless stated otherwise.

Noise is added to analytic derivative labels with standard deviation ρs_g, where s_g is the standard deviation over the exact derivative array in the outer fitting pool (training and validation inputs), pooling examples and coordinates, with a unit fallback for degenerate scales. The resulting absolute noise level varies across families and dimensions. Value labels remain exact, and evaluation uses exact values at held-out inputs.

For level ρ_j, let m_j be the mean paired natural-log value-error ratio over the five seeds. The reported crossing linearly interpolates the first neighboring pair with $m_j < 0 \le m_{j+1}$:

$$\hat{\rho}_* = \rho_j + (\rho_{j+1} - \rho_j)\frac{-m_j}{m_{j+1} - m_j}.$$

This interpolation summarizes the first observed beneficial-to-harmful transition on the measured noise grid, including when the response is nonmonotone. All three observed family-mean curves begin below zero. A curve that remains below zero throughout the tested range has a right-censored crossing, reported as exceeding the largest level. Absence of a crossing for an already harmful curve is a different outcome.

We enumerate all $5^5 = 3{,}125$ ordered bootstrap resamples of the five seed IDs, keeping each sampled seed's complete noise trajectory together. We recompute the mean response and interpolated crossing in each resample. The resulting percentile intervals and figure bands preserve dependence

Table 25: Matched noise responses for the full tested grid. The value-only reference is fixed within each family and seed.

Poly–trig, $d = 5$: value-only B versus first-order D

	Value MSE ($\cdot 10^{-3}$) ↓			Derivative MSE ($\cdot 10^{-2}$) ↓			Lower error
Task / setting	B	D	Ratio [95% CI]	B	D	Ratio [95% CI]	pairs
$\rho = 0$	1.55	0.0879	**0.061** [0.045, 0.083]	2.1	0.0991	**0.05** [0.043, 0.065]	5/5
$\rho = 0.05$	1.55	0.155	**0.11** [0.08, 0.16]	2.1	0.147	**0.076** [0.062, 0.1]	5/5
$\rho = 0.1$	1.55	0.326	**0.23** [0.16, 0.34]	2.1	0.244	**0.13** [0.097, 0.17]	5/5
$\rho = 0.2$	1.55	0.932	**0.66** [0.47, 0.88]	2.1	0.558	**0.28** [0.21, 0.37]	4/5
$\rho = 0.5$	1.55	4.85	**3.3** [2.1, 4.8]	2.1	2.29	1.2 [0.86, 1.6]	0/5

Trig, $d = 5$: value-only B versus first-order D

	Value MSE ↓			Derivative MSE ↓			Lower error
Task / setting	B	D	Ratio [95% CI]	B	D	Ratio [95% CI]	pairs
$\rho = 0$	2.05	1.18	**0.16** [0.021, 0.9084]	5.13	2.43	**0.14** [0.023, 0.71]	4/5
$\rho = 0.05$	2.05	1.11	**0.16** [0.022, 0.87]	5.13	2.13	**0.14** [0.024, 0.66]	4/5
$\rho = 0.1$	2.05	1.16	**0.2** [0.032, 0.88]	5.13	2.41	**0.17** [0.035, 0.77]	4/5
$\rho = 0.2$	2.05	0.837	**0.26** [0.084, 0.68]	5.13	1.67	**0.2** [0.068, 0.58]	4/5
$\rho = 0.5$	2.05	1.99	0.84 [0.41, 1.4]	5.13	4.23	**0.65** [0.34, 0.9319]	2/5

Bachelier, $d = 10$: value-only B versus first-order D

	Value MSE ($\cdot 10^{-7}$) ↓			Derivative MSE ($\cdot 10^{-6}$) ↓			Lower error
Task / setting	B	D	Ratio [95% CI]	B	D	Ratio [95% CI]	pairs
$\rho = 0$	3.76	0.797	**0.21** [0.17, 0.25]	2.83	0.542	**0.19** [0.16, 0.23]	5/5
$\rho = 0.05$	3.76	0.945	**0.24** [0.19, 0.33]	2.83	0.621	**0.22** [0.17, 0.28]	5/5
$\rho = 0.1$	3.76	1.23	**0.32** [0.26, 0.41]	2.83	0.76	**0.27** [0.22, 0.33]	5/5
$\rho = 0.2$	3.76	2.08	**0.55** [0.47, 0.67]	2.83	1.16	**0.41** [0.35, 0.48]	5/5
$\rho = 0.5$	3.76	5.72	**1.5** [1.3, 1.9]	2.83	2.78	0.9746 [0.79, 1.2]	0/5

Errors are arithmetic means. Ratios are paired geometric means, computed separately for each metric; brackets give 95% paired bootstrap intervals. Green and rust ratios have intervals below and above one; uncolored intervals include one. The final column counts lower-error pairs for the left-hand metric. Intervals resample all 5^5 ordered panels of the five seeds, keeping each noise trajectory intact.

Table 26: Empirical value-error crossings from matched curves under the two validation monitors. Intervals enumerate all 5^5 ordered seed-panel resamples and condition on the stated configuration. †The trig mean trajectory does not cross within the tested range under either monitor (final mean log ratios -0.17 and -0.25); 1042 and 641 of the 3,125 resamples cross, at levels between 0.40 and 0.50 (value monitor) and 0.36 and 0.50 (combined monitor).

		Value monitor		Combined monitor	
Family	Seeds	Crossing $\hat{\rho}_*$	95% interval	Crossing $\hat{\rho}_*$	95% interval
Poly-trig	5	0.277	[0.223, 0.349]	0.223	[0.184, 0.304]
Trig	5	> 0.5	†	> 0.5	†
Bachelier	5	0.380	[0.316, 0.429]	0.379	[0.311, 0.433]

across noise levels and quantify seed variability within each configuration. All resamples cross for poly-trig and Bachelier under both monitors. For trig, 2,077 value-monitor resamples and 2,483 combined-monitor resamples remain beneficial throughout; six and one, respectively, remain harmful throughout. The latter are not counted as right-censored helpful regimes. The remaining resamples contain a beneficial-to-harmful transition as counted in Table 26. The trig band in Figure 2 is wide because two seeds gain far more from clean derivatives than the other three; one of the latter is slightly worse than value-only training at every level.

Validation-monitor sensitivity. The monitor controls the learning-rate schedule, early stopping and the selected checkpoint, so it is part of the training procedure rather than a post-training choice. Repeating every derivative-supervised fit with the recipe's combined validation objective as monitor gives the second column group of Table 26. The poly-trig crossing moves earlier, from 0.277 to 0.223, under the changed scheduling, stopping and selection rule; the Bachelier crossing is nearly

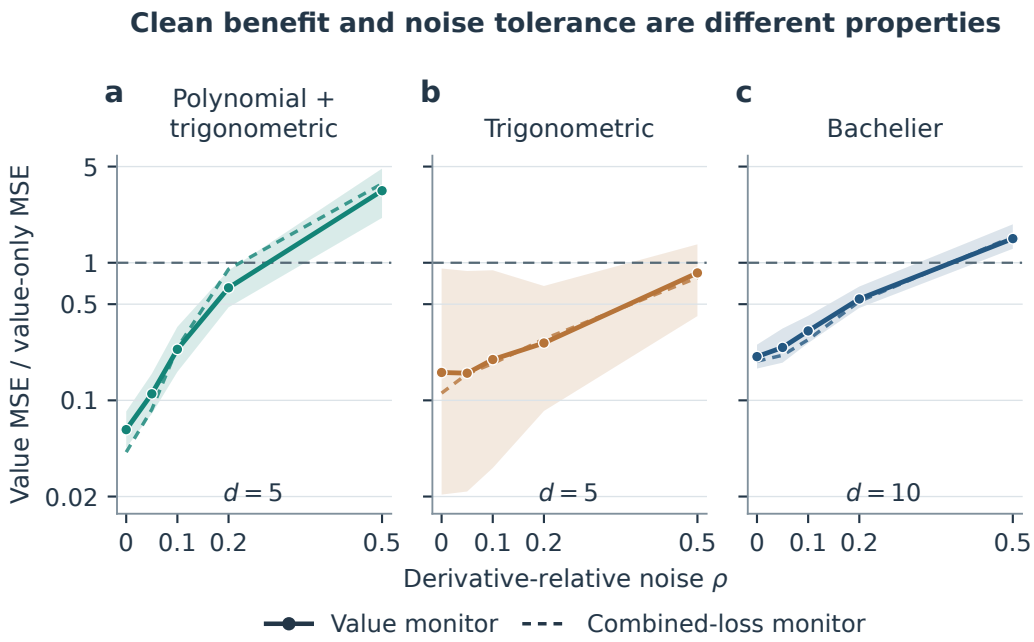

Figure 7: Matched noise responses under value and combined-loss validation monitoring. Curves are geometric paired value-MSE ratios; shading gives pointwise 95% percentile intervals over all ordered resamples of the five seeds, retaining each seed's complete noise trajectory. The monitors govern scheduling, stopping and checkpoint selection.

unchanged (0.379) and trig remains censored. The direction and order of magnitude of the transition therefore do not depend on the monitor, while its exact location does within roughly the width of the seed intervals.

The finite-design analysis in Appendix F explains the competition between useful constraints and derivative noise through an expected squared-error difference for linear estimators. Its threshold uses absolute noise variances. The neural curves here instead use derivative-relative noise and average log error ratios; their crossing locations therefore need not equal those of the linear risk difference.

E.1 Label construction under common added noise

The analytic-label curves isolate corruption of a known derivative. A complementary experiment asks whether the source comparison persists when the same perturbation is added to two constructed derivative labels. For BS digital and discretely monitored BS barrier, we hold raw value labels fixed and compare likelihood-ratio and fuzzy derivatives at $\rho \in \{0, 0.25, 1\}$. Five seeds $(42, 123, 456, 789, 1337)$ give 60 derivative-supervised fits and ten shared value-only references. Each task–seed pair uses 2,048 inputs split 1,311/328/409 into training, validation and evaluation. All seven fits share these partitions, raw value labels, initial parameters, and the scaler fitted on training inputs and raw values. They use the four-layer, width-256 Softplus network, batch size 256 and the optimization recipe in Appendix C, with a 500-epoch cap and raw-value validation for scheduling, stopping and checkpoint selection. In these one-dimensional tasks the fixed derivative objective assigns weight $1/2$ to each loss.

For construction $c \in \{L, F\}$, the perturbed training label is

$$\widetilde{g}_{c,i}(\rho) = g_{c,i} + \rho s_g z_i,$$

where s_g is the standard deviation of reference derivatives over the training inputs only. One Gaussian draw per task and seed is centered and scaled to unit empirical standard deviation, then shared across both sources and all levels. Values and validation/evaluation labels are unchanged. The reference is the exact digital expectation or the discrete barrier expectation used in Appendix D; s_g ranges from 0.00657 to 0.00676 for digital and from 0.388 to 0.398 for barrier. Thus the two sources receive the

Table 27: Constructed derivative labels under common added noise, with raw value labels held fixed. Errors are arithmetic means over five seeds. For the metric in each panel, $G_{F/L}$ is the geometric paired fuzzy/LRM error ratio; lower-error counts compare fuzzy with LRM. Pointwise percentile intervals enumerate all $5^5 = 3,125$ ordered seed resamples, keeping each seed's sources and complete noise trajectory together. The same value-only fits are repeated across levels for reference.

Task	ρ	Value-only	LRM	Fuzzy	$G_{F/L}$ [95% CI]	Lower error
Value MSE						
BS digital	0	$2.31\cdot10^{-4}$	$2.16\cdot10^{-4}$	$2.46\cdot10^{-4}$	1.33 [0.952, 2.12]	1/5
	0.25	$2.31\cdot10^{-4}$	$1.26\cdot10^{-4}$	$1.84\cdot10^{-4}$	1.34 [0.951, 1.8]	1/5
	1	$2.31\cdot10^{-4}$	$1.05\cdot10^{-4}$	$2.29\cdot10^{-4}$	1.52 [0.932, 2.49]	2/5
BS barrier	0	0.307	5.82	0.401	0.0846 [0.0425, 0.184]	5/5
	0.25	0.307	6.17	0.402	0.0691 [0.0443, 0.111]	5/5
	1	0.307	5.84	0.311	0.0595 [0.0304, 0.133]	5/5
Derivative MSE						
BS digital	0	$1.5\cdot10^{-6}$	$6.8\cdot10^{-7}$	$7.25\cdot10^{-7}$	0.956 [0.4, 2.71]	3/5
	0.25	$1.5\cdot10^{-6}$	$4.84\cdot10^{-7}$	$4.72\cdot10^{-7}$	0.864 [0.436, 1.71]	3/5
	1	$1.5\cdot10^{-6}$	$7.57\cdot10^{-7}$	$7.8\cdot10^{-7}$	0.891 [0.482, 1.65]	3/5
BS barrier	0	0.00544	0.0662	0.0068	0.103 [0.0794, 0.128]	5/5
	0.25	0.00544	0.149	0.007	0.0511 [0.0327, 0.077]	5/5
	1	0.00544	0.148	0.00712	0.0639 [0.0362, 0.122]	5/5

Table 28: Source-by-added-noise interactions from Eq. (19), in each task's original squared-error units. Intervals use the same paired seed resamples as Table 27. The value and derivative metrics answer different questions and are analyzed separately.

Task	ρ	Value-error $I(\rho)$ [95% CI]	Derivative-error $I(\rho)$ [95% CI]
BS digital	0.25	$2.8\cdot10^{-5}$ $[-5.26\cdot10^{-5}, 1.26\cdot10^{-4}]$	$-5.66\cdot10^{-8}$ $[-3.36\cdot10^{-7}, 2.23\cdot10^{-7}]$
	1	$9.34\cdot10^{-5}$ $[-3.97\cdot10^{-5}, 2.53\cdot10^{-4}]$	$-2.22\cdot10^{-8}$ $[-4.84\cdot10^{-7}, 7.92\cdot10^{-7}]$
BS barrier	0.25	-0.349 $[-1.47, 1.18]$	-0.0827 $[-0.16, -0.0216]$
	1	-0.108 $[-1.9, 2.11]$	-0.0814 $[-0.163, -0.00506]$

same added perturbation in original derivative units. Their intrinsic sampling noise and smoothing bias remain properties of the construction.

For barrier prediction, fuzzy derivatives outperform LRM derivatives in all five seeds at every level, with geometric value-error ratios 0.0846, 0.0691 and 0.0595 (Table 27). This large source contrast survives the imposed perturbations without changing the value targets. It does not imply that either derivative source must outperform value-only training: the barrier LRM mean value error is 19–20 times the value-only error, whereas fuzzy remains within 1.01–1.31 times it. For digital prediction, LRM has the lower mean value error at all three levels, but the paired source-contrast intervals include parity. The digital derivative errors are likewise similar between sources.

To separate an existing source contrast from a different response to added noise, define the interaction for either error metric E as

$$I(\rho) = \frac{1}{5}\sum_{s} \{[E_{F,s}(\rho) - E_{F,s}(0)] - [E_{L,s}(\rho) - E_{L,s}(0)]\}\,. \tag{19}$$

A negative value means that the perturbation changes fuzzy error less adversely than LRM error. The value-error interaction intervals include zero on both tasks (Table 28); their width does not establish equal noise responses. For the secondary derivative metric, barrier LRM error rises from 0.0662 to about 0.149, while fuzzy error remains near 0.007. Both barrier derivative-error interaction intervals exclude zero. Added label noise therefore affects derivative recovery differently across these sources even though the value-error interaction is unresolved. This distinction is why target prediction, rather than derivative fit alone, determines whether supervision is useful.

E.2 Beyond additive Gaussian corruption

On poly-trig and trig with $d = 5$ and 2,048 generated inputs, we compare dropout, additive Gaussian noise, multiplicative noise, sparse-coordinate corruption and mixed corruption. Five severity levels $(0.01, 0.05, 0.10, 0.30, 1.00)$, three methods (value-only, fixed-weight DML and GradNorm), and five seeds per family yield 750 combinations. Table 29 displays pooled means for dropout, additive Gaussian and multiplicative corruption at severities 0.01, 0.10 and 1.00. Aggregate counts of lower error below include all five operators and all five severity levels.

Table 29: Value MSE under derivative-label corruption, pooled over ten family–seed pairs (two families, five seeds each). Value-only training does not consume the corrupted channel. Severity is specific to each corruption operator and is not an absolute Gaussian standard deviation shared across types.

Corruption	Severity	Value-only	Fixed DML	DML lower error
Dropout	0.01	0.642	0.037	9/10
	0.10	0.642	0.222	5/10
	1.00	0.642	1.748	0/10
Additive Gaussian	0.01	0.642	0.048	10/10
	0.10	0.642	0.055	10/10
	1.00	0.642	1.063	0/10
Multiplicative	0.01	0.642	0.048	10/10
	0.10	0.642	0.049	10/10
	1.00	0.642	1.201	1/10

Across all five operators, the numbers of fixed-weight improvements among 50 type–family–seed pairs are $43, 35, 34, 21, 1$ as severity increases. Useful derivative supervision under mild corruption becomes predominantly harmful at the highest severity. The operators use different severity scales, so this aggregate pattern does not define a common numerical crossing. Dropout and mixed corruption can also change the label mean, an effect included in the general moment formula of Appendix F.

F Finite-design explanation and nonparametric context

This appendix explains how derivative observations change value-prediction risk in two statistical settings. First, a standard bias–covariance expansion gives an exact identity for least squares with a fixed representation and derivative weight. A Fourier specialization makes the effects of sampling and unresolved frequencies explicit. Second, a nonparametric estimator adapts its smoothing bandwidth to the sample size, yielding a complementary sample–noise tradeoff. Appendix G uses the fixed-representation noise cost to construct and test a predictor for trained networks.

F.1 Observation model and conditioning

Let $\phi(x) \in \mathbb{R}^p$ be a differentiable feature map, and let $\psi(x) = \phi'(x)$. The derivation is written for one input derivative; several derivative coordinates can be included by stacking their feature rows and specifying the resulting covariance. Fix training inputs $x_1, \dots, x_N$. The matrices $\Phi, \Psi \in \mathbb{R}^{N\times p}$ contain the rows $\phi(x_i)^\top$ and $\psi(x_i)^\top$. Write

$$f^*(x) = \phi(x)^\top \theta^* + r(x), \qquad y = \Phi\theta^* + r_X + \varepsilon, \qquad g = \Psi\theta^* + r'_X + \eta, \tag{20}$$

where $r_X = (r(x_i))_{i=1}^N$ and $r'_X = (r'(x_i))_{i=1}^N$. Conditional on the design, assume

$$\mathbb{E}\varepsilon = \mathbb{E}\eta = 0, \qquad \mathrm{Cov}(\varepsilon) = \sigma_y^2 I_N, \qquad \mathrm{Cov}(\eta) = \sigma_g^2 I_N, \tag{21}$$

with independent channels ε and η. Gaussianity is not required. Here σ_g is an absolute derivative-noise scale. It is distinct from the empirical multiplier ρ in Section 5, which scales a configuration-specific derivative standard deviation.

Set $H = \Phi^\top\Phi$, $G = \Psi^\top\Psi$, and $M = H + \lambda G$, where $\lambda > 0$ is fixed independently of the realized label noise. Assume $H \succ 0$, which also implies $M \succ 0$. The estimators minimize $\|\Phi\theta - y\|_2^2$ and $\|\Phi\theta - y\|_2^2 + \lambda\|\Psi\theta - g\|_2^2$, respectively. The value-only and derivative estimators are

$$\hat\theta_V = H^{-1}\Phi^\top y, \qquad \hat\theta_D = M^{-1}(\Phi^\top y + \lambda\Psi^\top g). \tag{22}$$

Their pointwise risks are $R_j(x) = \mathbb{E}[(\phi(x)^\top \hat{\theta}_j - f^*(x))^2 \mid x_1, \ldots, x_N]$. Thus the expectation is over label noise, not over the training design. For a random design, an unconditional risk requires an additional design expectation.

Weight normalization. All finite-design formulas use λ in $H + \lambda G$. Loss averaging, coordinate counts and label-variance normalization must be absorbed into this effective weight. Under the coordinate change $z = cx$, with the same physical target represented as $\tilde{f}(z) = f(z/c)$, derivatives scale as $g_z = g_x/c$; preserving the loss therefore requires $\lambda_z = c^2\lambda_x$. This translation is necessary before comparing an absolute theoretical σ_g with an empirical relative noise multiplier.

F.2 Exact risk identity and helpful-noise regime

For $v = \phi(x)$, define

$$\begin{aligned} \delta_V(x) &= v^\top H^{-1}\Phi^\top r_X - r(x), \\ \delta_D(x) &= v^\top M^{-1}(\Phi^\top r_X + \lambda\Psi^\top r'_X) - r(x), \end{aligned} \tag{23}$$

$$\begin{aligned} A_V(x) &= v^\top H^{-1} v, \qquad A_D(x) = v^\top M^{-1} H M^{-1} v, \\ B_D(x) &= \lambda^2 v^\top M^{-1} G M^{-1} v. \end{aligned} \tag{24}$$

Both biases include the unresolved target component $-r(x)$.

Proposition 1 (Finite-design value-risk decomposition)**.** *Under* (20)–(22)*,*

$$\begin{aligned} R_V(x) &= \delta_V(x)^2 + \sigma_y^2 A_V(x), \\ R_D(x) &= \delta_D(x)^2 + \sigma_y^2 A_D(x) + \sigma_g^2 B_D(x). \end{aligned} \tag{25}$$

In particular,

$$\Delta R(x) := R_D(x) - R_V(x) = \delta_D(x)^2 - \delta_V(x)^2 - \sigma_y^2\{A_V(x) - A_D(x)\} + \sigma_g^2 B_D(x), \tag{26}$$

where $A_V(x) - A_D(x) \geq 0$ *and* $B_D(x) \geq 0$*.*

Proof. The normal equations give the biases in Eq. (23); centering and channel independence remove all noise cross terms. Their covariances give Eq. (25). For $C = \lambda H^{-1/2} G H^{-1/2} \succeq 0$,

$$M^{-1} H M^{-1} = H^{-1/2}(I + C)^{-2} H^{-1/2} \preceq H^{-1}.$$

Thus $A_D \leq A_V$; positive semidefiniteness of G gives $B_D \geq 0$. Subtracting the two risks proves the identity. □

This ordering concerns the stated unpenalized estimators. The learned-feature predictor uses a small ridge, including the intercept, in a separately defined anchored response (Appendix G); its noise-cost identity does not require this ordering.

The derivative constraints therefore reduce sensitivity to value noise. The net gain also depends on the approximation-bias change and the cost of derivative noise. Define

$$\Gamma(x) = \sigma_y^2\{A_V(x) - A_D(x)\} - \{\delta_D(x)^2 - \delta_V(x)^2\}. \tag{27}$$

If $B_D(x) > 0$ and $\Gamma(x) > 0$, the fixed rule improves value prediction exactly when

$$0 \leq \sigma_g < \sigma_g^*(x), \qquad \sigma_g^*(x) = \sqrt{\Gamma(x)/B_D(x)}. \tag{28}$$

If $\Gamma(x) \leq 0$, there is no positive helpful-noise interval. Assigning zero to the threshold in that case is a convention, not an equality of risks when $\Gamma(x) < 0$. If $B_D(x) = 0$, the derivative-noise scale does not enter this pointwise risk; the sign is determined by $-\Gamma(x)$.

For a prescribed test distribution Q, integrate the individual terms: $\Delta R_Q = -\Gamma_Q + \sigma_g^2 B_{D,Q}$, with $\Gamma_Q = \mathbb{E}_Q\Gamma(X)$ and $B_{D,Q} = \mathbb{E}_Q B_D(X)$. When both are positive, the integrated-risk threshold is $\sigma_{g,Q}^* = \sqrt{\Gamma_Q/B_{D,Q}}$. It is not generally the average of the pointwise thresholds.

F.3 Label bias and general covariance

A derivative-label procedure with conditional bias b_X replaces r'_X by $r'_X + b_X$ in (23). This changes the bias term even before additional centered noise is introduced. A procedure can also alter the value targets or generate correlated value and derivative errors, as occurs with shared Monte Carlo paths. The following form keeps those effects explicit.

Proposition 2 (Risk of a linear fit under general label moments)**.** *Let the stacked labels $z = (y^\top, g^\top)^\top$ have finite conditional mean μ and covariance Σ. For any deterministic linear estimator $\hat{\theta} = Lz$, a test feature matrix F, a diagonal nonnegative weight matrix W with $\mathrm{tr}(W) = 1$, and the full test target vector f, the conditional weighted risk is*

$$R(L; \mu, \Sigma) = (FL\mu - f)^\top W(FL\mu - f) + \mathrm{tr}(WFL\Sigma L^\top F^\top). \tag{29}$$

Proof. Write $z = \mu + \xi$. Centering eliminates the cross term in the squared error, while $\mathbb{E}[\xi\xi^\top] = \Sigma$ gives the trace term. □

For the estimators above, $L_V = [H^{-1}\Phi^\top, 0]$ and $L_D = M^{-1}[\Phi^\top, \lambda\Psi^\top]$. If two label procedures have different means or covariances, evaluate (29) separately for each procedure. This retains changes in both value and derivative targets. A scalar threshold in σ_g^2 follows only when the label means and other covariance components are fixed and the varying covariance has the stated form. If λ adapts to the realized labels, L_D is itself random and the fixed-linear-map identity does not by itself describe its risk.

F.4 Exact Fourier specialization under no aliasing

Take

$$\phi_K(x) = (\sin(2\pi x), \cos(2\pi x), \ldots, \sin(2\pi Kx), \cos(2\pi Kx))^\top,$$

and let r be any linear combination of integer modes $K+1, \ldots, K'$, with $K' \geq K$. Set $\omega_k = 2\pi k$ and $D = \mathrm{diag}(\omega_1^2, \omega_1^2, \ldots, \omega_K^2, \omega_K^2)$.

Corollary 1 (Exact misspecified Fourier crossover)**.** *On the grid $x_i = -1 + 2i/N$, $i = 0, \ldots, N-1$, the sufficient condition $N > 2(K + K')$ gives $H = (N/2)I$, $G = (N/2)D$, and $\Phi^\top r_X = \Psi^\top r'_X = 0$. Thus $\delta_V(x) = \delta_D(x) = -r(x)$ at every test point. For $\sigma_y > 0$, the exact finite-design crossover is*

$$(\sigma_g^*)^2 = \sigma_y^2 \frac{S_K(\lambda)}{T_K(\lambda)}, \qquad S_K(\lambda) = \sum_{k=1}^{K} [1 - (1+\lambda\omega_k^2)^{-2}], \quad T_K(\lambda) = \lambda^2 \sum_{k=1}^{K} \omega_k^2 (1+\lambda\omega_k^2)^{-2}. \tag{30}$$

It is independent of the test phase and the residual coefficients.

Proof. The grid sum $\sum_{j=0}^{N-1} e^{4\pi \mathrm{i} mj/N}$ vanishes unless N divides $2m$. The stated condition excludes nonzero alias frequencies in products of fitted modes and fitted–residual pairs, giving the Gram identities and zero leakage. The sine/cosine squares sum to one at each frequency, so

$$A_V = \frac{2K}{N}, \quad A_D = \frac{2}{N} \sum_{k=1}^{K} (1 + \lambda\omega_k^2)^{-2}, \quad B_D = \frac{2\lambda^2}{N} \sum_{k=1}^{K} \omega_k^2 (1 + \lambda\omega_k^2)^{-2}.$$

Substitution into Eq. (28) gives the result. □

For $K = 1$, (30) becomes $\sigma_g^* = \sigma_y \sqrt{(2\pi)^2 + 2/\lambda}$. At $\lambda = 1$, the factors are 6.4403740267 for $K = 1$ and 11.8278908835 for $K = 5$. When $\sigma_y = 0$, both estimators already share the same irreducible residual risk at zero derivative noise, and any positive derivative noise increases R_D in this orthogonal case. The target need not lie in the fitted span: equality of the biases, rather than absence of misspecification, is what yields the corollary.

The condition is sufficient, not necessary. The exact leakage test is $N \nmid 2(\ell - k)$ and $N \nmid 2(\ell + k)$ for every fitted/residual pair. On the stated grid there are $N/\gcd(N, 2)$ distinct phases modulo one. If aliasing occurs, the bias terms in (26) remain essential.

Periodicity and limits. At any fixed design, both fitted and residual integer modes have period one. Consequently each risk satisfies $R_j(x+q) = R_j(x)$ for integer q. Independence of the *crossover* from arbitrary test phase is stronger and follows from Corollary 1; it does not hold for every finite design. Even in that corollary, the individual risks can vary with the common residual term $r(x)^2$. The periodic model cannot describe a nonconstant globally monotone increase of risk with extrapolation distance.

Normalized Gram convergence alone also does not establish the corollary for a sequence of random designs. Let a_N and b_N denote the value-only and derivative fitted residual leakages at a test point. If both are $O_p(N^{-1/2})$, the derivative-minus-value squared-bias difference is $(b_N - r(x))^2 - (a_N - r(x))^2 = (b_N - a_N)(a_N + b_N - 2r(x))$, which need not be $o_p(N^{-1})$ while the variance terms have order N^{-1}. The exact no-aliasing proof avoids this rate problem.

F.5 Nonparametric sample–noise tradeoff

The nonparametric calculation follows the derivative-data construction of Hall & Yatchew (2007), combined with local-polynomial regression rates (Stone, 1982; Fan & Gijbels, 1996; Tsybakov, 2009). Let $k \geq 2, d \geq 2$, and let f range over an isotropic C^k class on $[0,1]^d$ with all derivatives through order k bounded by M_0. Observe $X_i = (U_i, V_i)$ uniformly on the cube, values $Y_i = f(X_i) + \varepsilon_i$, and one first partial derivative $Z_i = \partial_1 f(X_i) + \eta_i$. The noises are centered, mutually independent and independent of the design, with sub-Gaussian scales σ_y, σ_g.

For an evaluation coordinate u, define $\kappa_u(t) = \mathbf{1}\{t \leq u\} - (1-t)$ and $T_i(u) = Y_i + \kappa_u(U_i) Z_i$. For fixed u, v and a function $h(\cdot, v) \in C^1([0,1])$, integration by parts on $[0,u]$ and $[u,1]$ gives

$$\int_0^1 [h(t,v) + \kappa_u(t)\partial_1 h(t,v)]\, dt = h(u,v), \qquad \mathbb{E}[T_i(u) \mid V_i = v] = f(u,v). \tag{31}$$

Indeed, the boundary terms from $t\partial_1 h$ and $(t-1)\partial_1 h$ sum to $h(u,v)$, and their integral terms cancel $\int_0^1 h(t,v)dt$. Thus estimation at fixed u requires local smoothing only in the remaining $d-1$ coordinates.

The centered signal fluctuation $f(U_i,V_i) + \kappa_u(U_i)\partial_1 f(U_i,V_i) - f(u,V_i)$ has magnitude at most M_0: for $U_i \leq u$, apply the fundamental theorem of calculus to bound it by uM_0, and for $U_i > u$ by $(1-u)M_0$. Hoeffding's lemma and the conditional noise moment bound therefore give the composite sub-Gaussian envelope $\tau^2 = \sigma_y^2 + \sigma_g^2 + M_0^2$.

For fixed nondegenerate value noise and sufficiently large n, standard boundary-corrected local-polynomial estimators, set to zero on poorly conditioned local designs and clipped to the target range, give the following bounds on expected integrated squared error over $[0,1]^d$. Use a bounded nonnegative kernel nondegenerate near zero, and choose the Gram cutoff below half the uniform population minimum eigenvalue. In effective dimension q, the stabilized estimator has risk at most $C(M_0^2 h^{2k} + \nu^2/(nh^q)) + CM_0^2 e^{-cnh^q}$; the last term controls the bad-design event and is absorbed for the optimized bandwidth at fixed positive noise scale. Thus

$$U_V(n) = C_V \sigma_y^{4k/(2k+d)} n^{-2k/(2k+d)}, \qquad U_D(n) = C_D \tau^{4k/(2k+d-1)} n^{-2k/(2k+d-1)}. \tag{32}$$

The derivative estimator uses evaluation-dependent pseudo-labels and a bandwidth of order $(\tau^2/n)^{1/(2k+d-1)}$, with constants absorbing M_0; it is not a fixed Fourier fit. For Gaussian value noise bounded away from zero, the value-only minimax lower rate follows from Stone's theorem. A matching derivative-enhanced lower rate follows by embedding $(d-1)$-dimensional functions as $f(u,v) = g(v)$. On this subclass the derivative channel is independent noise. Averaging an estimator over u and conditioning on (V_i, Y_i) cannot increase integrated squared error, reducing the lower bound to value-only regression in dimension $d-1$, the same subclass mechanism used by Hall and Yatchew. Thus, at fixed noise levels, the ratio of the two minimax risks has order $n^{-\delta}$, $\delta = 2k/[(2k+d-1)(2k+d)]$, with two-sided constants but no asserted asymptotic limiting constant.

Let $\delta = 2k/[(2k+d-1)(2k+d)]$ and let T_n denote the expression inside the square brackets below. Algebraically the positive bound crossing is

$$(\sigma^*_{g,\mathrm{bound}})^2 = \left[\left\{ \frac{C_V}{C_D} \sigma_y^{4k/(2k+d)} n^{\delta} \right\}^{(2k+d-1)/(2k)} - \sigma_y^2 - M_0^2 \right]_+ . \tag{33}$$

If $T_n > 0$, the equality occurs at $\sqrt{T_n}$ and the derivative bound is smaller below it. If $T_n = 0$, equality holds only at zero noise; if $T_n < 0$, there is no nonnegative equality point. The positive-part convention then records no beneficial interval, not a zero-noise crossing. A strictly positive bound threshold requires $n > n_0$, where

$$n_0 = \left[\frac{C_D}{C_V}\sigma_y^{-4k/(2k+d)}(\sigma_y^2 + M_0^2)^{2k/(2k+d-1)}\right]^{1/\delta}.$$

This statement also requires the sample-size range in which the displayed bounds and their bandwidth choices apply. The leading variance threshold grows as $n^{1/(2k+d)}$; its standard-deviation scale grows as $n^{1/[2(2k+d)]}$. These are asymptotic scalings of the bound comparison, unlike the sample-size-independent Fourier ratio at fixed K and λ. This is a crossing of upper bounds for two constructions, not the exact minimax-risk crossing or the numerical neural threshold. In particular, an estimator with optional derivative data can ignore them, so its minimax risk is never larger than the value-only minimax risk. The calculation describes an improved sample-size exponent with a larger noise prefactor. It does not imply one removed dimension per derivative order or determine the constants of a neural training procedure.

G Predicting the effect of derivative-label noise

G.1 Prediction rule and fixed-feature calculation

For each design and seed group r, train a value-only network and a derivative-supervised network with clean labels. Their value-MSE difference, $a_r = E_{D,r}(0) - E_{V,r}$, supplies the clean-label effect. The rule

$$\widehat{\Delta}_r(\rho) = a_r + \rho^2 b_r, \qquad b_r = s_{g,r}^2 \widehat{B}_r, \tag{34}$$

combines this measurement with a computed derivative-noise cost. Here $s_{g,r}$ is the standard deviation of the clean fitting gradients, so the injected noise has standard deviation $\rho s_{g,r}$. No slope coefficient is fitted to the evaluation noise responses.

Freeze the clean derivative-trained network's hidden layers and augment their last activations by an intercept to obtain features ϕ_r. Let Φ be their fitting matrix, F their independent quadrature matrix, and Ψ_s their input-Jacobian rows in normalized coordinates. The diagonal matrix S repeats the fitting-input standard deviation s_j in stacked input–coordinate order. The dimension-aware training loss is equivalent, up to a common scale, to physical value SSE plus $\lambda \sum_j s_j^2$ times physical derivative SSE, with $\lambda = 1$. With $W = I/4096$ and $p = 257$ including the intercept, define

$$M = \Phi^\top \Phi + \lambda \Psi_s^\top \Psi_s + \alpha I, \qquad \alpha = 10^{-10}\,\mathrm{tr}(\Phi^\top \Phi)/p. \tag{35}$$

The resulting slope is

$$\widehat{B}_r = \lambda^2\,\mathrm{tr}\left(WFM^{-1}\Psi_s^\top S^2 \Psi_s M^{-1} F^\top\right). \tag{36}$$

Linear solves evaluate the trace. The small ridge includes the intercept and stabilizes this calculation; neural training uses zero weight decay.

For an additional physical derivative perturbation η with conditional mean zero and covariance $\sigma_g^2 I$, the frozen response is $h_\eta(x) = \lambda \phi_r(x)^\top M^{-1}\Psi_s^\top S\eta$. Conditioning on the clean functions, features and fitting design, define $f_{D,\eta} = f_{D,0} + h_\eta$ and $R_Q(f) = \mathbb{E}_Q[(f(X) - f^*(X))^2]$. Centering removes the cross term, giving

$$\mathbb{E}_\eta[R_Q(f_{D,\eta}) - R_Q(f_{V,0})] = a_Q + \sigma_g^2 B_Q, \tag{37}$$

where $a_Q = R_Q(f_{D,0}) - R_Q(f_{V,0})$ and B_Q is the integrated squared sensitivity estimated by Eq. (36). This identity holds for the frozen response even when the clean readout does not minimize the linear objective. Its use for networks whose features and stopping times change with noise is tested empirically.

G.2 Evaluation design

The rule and comparison threshold were selected using a separate development corpus. Evaluation has 36 fresh seed groups, twelve per family: poly-trig, trig and the smooth Bachelier basket price. The

Table 30: Last-layer rule by family on the independent test stream. Accuracy concerns the sign of the twenty arithmetic condition means. Balanced accuracy averages recall over beneficial and harmful conditions. $R^2 = 1 - \sum(\Delta - \widehat{\Delta})^2 / \sum(\Delta - \bar{\Delta})^2$ assesses their magnitudes separately within each family; negative values indicate worse squared error than its constant mean. Pooling physical MSE across these differently scaled families would give $R^2 = 0.62$, but would obscure the basket-price magnitude error.

Family	Conditions	Accuracy (%)	Balanced (%)	Magnitude R^2
Poly-trig	20	85.0	83.3	0.91
Trig	20	90.0	75.0	0.40
Bachelier basket	20	90.0	87.5	-8.89

first two draw new target coefficients; Bachelier keeps its pricing parameters and draws new inputs, splits and training randomness. Each group supplies $d \in \{3, 6\}$, fitting pools $n \in \{512, 1024\}$, and $\rho \in \{0, 0.05, 0.1, 0.2, 0.35, 0.5\}$.

Training/validation counts are 409/103 or 819/205. All networks use four 256-unit softplus layers, AdamW at learning rate 0.005 with zero weight decay, batches of 512, norm-1 clipping and a 500-epoch cap. Validation value MSE controls scheduling, early stopping and checkpoint selection. Normalization uses fitting data. Each trajectory shares the design, initialization and standardized gradient-noise draw. Three independent streams, each with 4096 inputs, provide the clean reference, slope quadrature and noisy-model evaluation. No positive-noise outcome enters the prediction.

For each family–dimension–budget condition, average the twelve group predictions to obtain $\widehat{\Delta}(\rho) = \bar{a} + \rho^2 \bar{b}$. The 144 clean-reference pairs are included in 1008 fits, with value-only fits reused across noise levels. Accuracy excludes $\rho = 0$, leaving 60 condition means, equally weighted and with no ties. Comparators are the clean measurement alone, a development-selected threshold $\rho = 0.275$, and an always-beneficial prediction. The 10,000 bootstrap resamples draw seed groups within family, retaining related dimensions, budgets and full noise trajectories, and recompute both predicted and measured condition means before taking signs.

G.3 Sign accuracy, magnitudes and crossings

Sign accuracy is 88.3%, versus 60.0% for the clean measurement and 76.7% for the noise threshold. The paired differences are 28.3 percentage points (95% percentile interval $[23.3,\ 35.0]$) and 11.7 points ($[5.0,\ 21.7]$), respectively. Balanced accuracy is 87.5%; beneficial/harmful counts are 36/24 overall and 12/8, 16/4 and 8/12 by family. The computed cost improves sign prediction, while magnitude accuracy varies substantially across families (Table 30).

When $\bar{a} < 0$ and $\bar{b} > 0$, the predicted crossing is $\widehat{\rho}_* = \sqrt{-\bar{a}/\bar{b}}$. It falls inside the first observed crossing interval for 3 of 9 crossing trajectories. The remaining 3 trajectories stay beneficial through $\rho = 0.5$, and 3 predictions lie beyond that endpoint. These are censoring agreements. One trig trajectory changes sign more than once, which the monotone quadratic rule cannot reproduce. The rule predicts noise tolerance after a clean-reference experiment; the initial benefit of acquiring derivatives is supplied by that experiment.

Value-label noise. A separate 288-fit comparison adds value noise at 0.1 or 0.3 times the clean fitting-value standard deviation, with $d = 3$, $n = 1024$ and $\rho \in \{0, 0.1, 0.35\}$. All eighteen seed-averaged condition means favor derivative supervision. The full fixed-dictionary score predicts seventeen signs correctly; the always-helpful rule predicts all eighteen. This comparison supports benefit under value noise, but contains no harmful conditions on which to distinguish the two sign predictors.

H SAMPLE BUDGET AND FUNCTION CLASS

The number of observations and the fitted function class determine how much structure value-only regression can recover. We first vary the sample budget within a fixed neural architecture, then compare that architecture with classical value-only estimators.

H.1 SAMPLE BUDGET AND DIMENSION

Figures 8 and 9 compare value-only and fixed-weight derivative supervision across sample budgets. The input dimension, target family and initialization seed are paired within each comparison. Clean derivative targets are used throughout. The polynomial–trigonometric and trigonometric panels contain five total sample budgets, $N \in \{256, 512, 1024, 2048, 4096\}$; the Bachelier panels contain $N \in \{256, 1024, 4096\}$. The same four-layer, width-256 softplus architecture uses a 500-epoch limit, learning rate 0.005 and batch size 512. Eighty percent of generated examples form the outer training pool; its internal 80/20 split gives $n = \lfloor 0.8\lfloor 0.8N \rfloor \rfloor$ gradient-update examples. The remaining examples provide validation and held-out evaluation. Normalization uses the outer training pool. Each method selects its checkpoint using its own validation objective (value loss or the combined value–derivative loss). The epoch cap is fixed, while optimizer-update counts vary with sample budget and stopping.

Absolute MSE shows prediction accuracy at each sample budget; paired ratios show the relative benefit of derivative supervision. The full dimension grid separates changes in task difficulty from changes in that benefit.

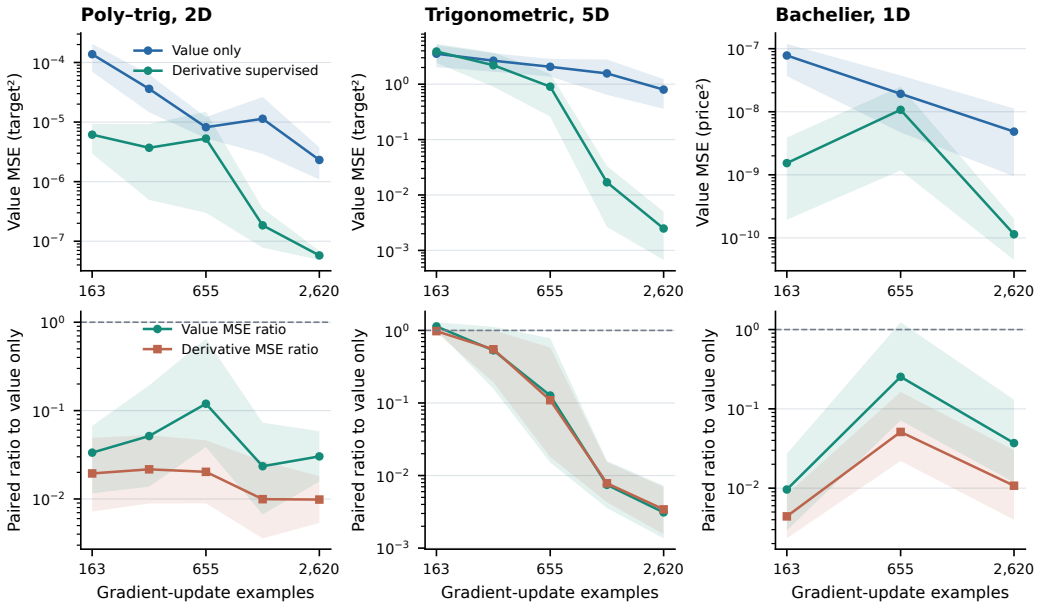

Figure 8: Sample-budget comparisons for three representative configurations: polynomial–trigonometric 2D, trigonometric 5D, and Bachelier 1D. Top: arithmetic mean value MSE in original units. Bottom: geometric mean of the five seed-paired derivative-supervised/value-only error ratios, separately for values and derivatives; values below one favor derivative supervision. Shading gives pointwise 95% percentile intervals from all 5^5 ordered bootstrap resamples of the five seeds. The horizontal axis counts examples used for gradient updates, rather than the total generated sample budget. Derivative MSE averages squared errors over test inputs and derivative coordinates.

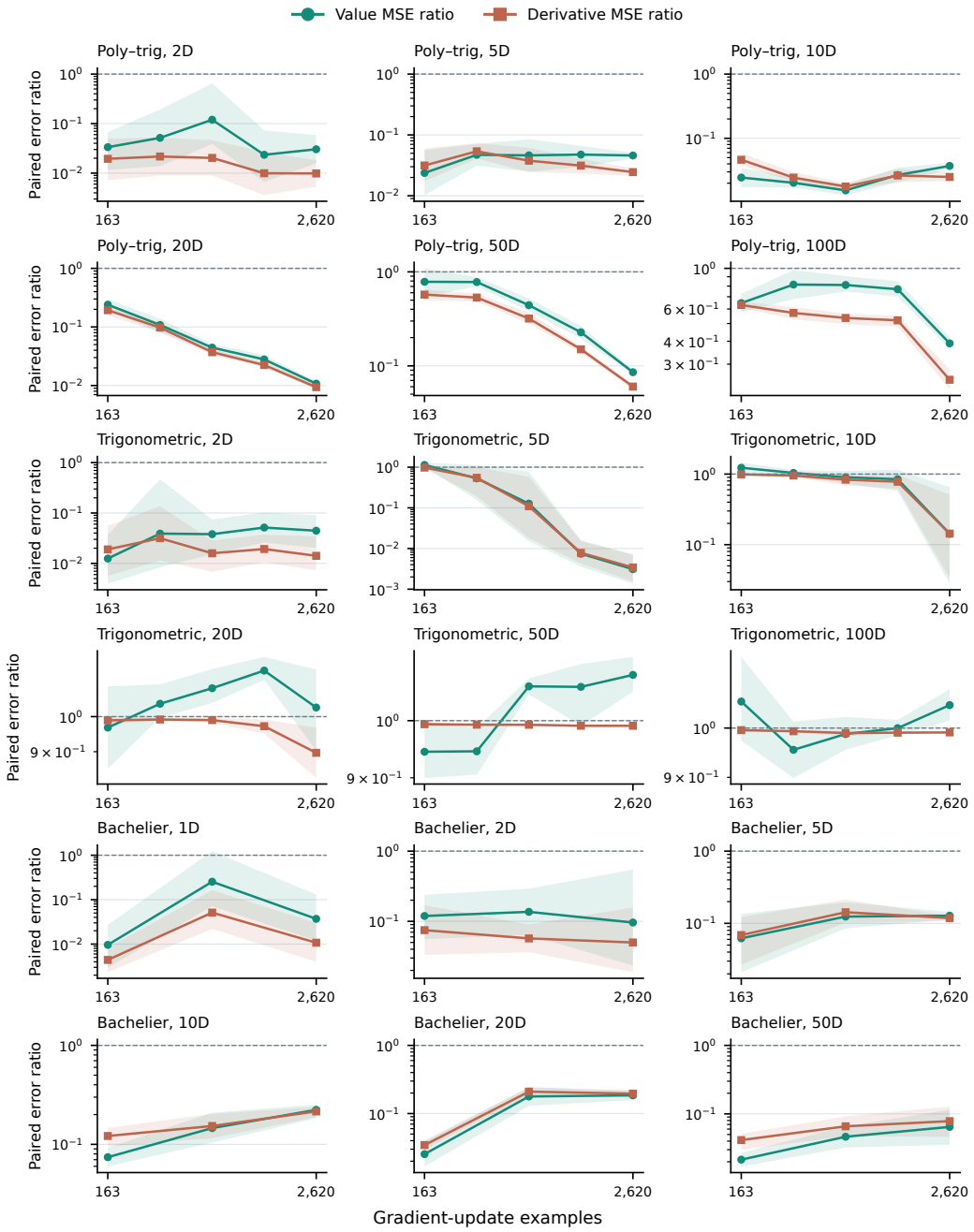

Figure 9: Complete sample-budget comparison across the six dimensions available for each family. Each point is the geometric mean of five paired error ratios; shading uses the same pointwise seed bootstrap as Figure 8. The target function is fixed by seed before inputs are sampled, but the input and evaluation sets need not be nested across budgets. Each panel uses its own vertical scale. The curves describe finite-budget performance.

H.2 Classical regression on smooth targets

The classical panel covers poly-trig, trig and Bachelier. It uses an RBF Gaussian process with likelihood-selected kernel parameters, RBF kernel ridge regression with fixed $\alpha = 1$ and the default kernel scale, and a 200-tree random forest with unrestricted tree depth and a minimum leaf size of five. Kernel-ridge parameters are fixed rather than selected by cross-validation. The GP comparison uses the feasible sample-size panel because its dense memory cost is quadratic.

Each comparator and fixed-weight DML use the same approximately 64% subset of the generated data and a common evaluation partition. Their fitting procedures differ in tuning and computational cost. The panel contains 990 evaluations, paired by family, dimension, total sample count and seed:

Value-only comparator	Matched pairs	DML lower value MSE
Gaussian process	90	50.0%
Kernel ridge	225	83.6%
Random forest	225	74.7%

Across the matched configurations, DML has lower error in half of the GP comparisons; DML has lower error more often against the fixed-parameter KRR and forest recipes. These counts summarize related configurations rather than independent draws from a task population. All three classical methods use values only, so the comparison includes both representation and supervisory information. Derivative-aware kernels provide another established way to use gradient labels, including in molecular learning (Christensen & von Lilienfeld, 2020).

I Higher-order supervision and loss scaling

I.1 Functions and derivative tensors

Higher-order supervision extends the information supplied at each observed input. We study value-only, value-plus-gradient, and value-plus-gradient-plus-Hessian objectives on two analytic families in dimensions $d \in \{2, 5, 10\}$. The trigonometric family is

$$f_{\mathrm{trig}}(x) = \sum_{j=1}^{d} a_j \sin(\omega_j x_j), \quad a_j \sim \mathcal{U}[0.5, 2], \quad \omega_j \sim \mathcal{U}[0.5, 5], \quad x \sim \mathcal{U}[-\pi, \pi]^d. \tag{38}$$

The polynomial–trigonometric family is

$$f_{\mathrm{poly}}(x) = \sum_{j=1}^{d} \sum_{k=0}^{3} c_{jk} x_j^k + \tfrac{1}{2} \sum_{j=1}^{d} \sin(2x_j), \quad c_{jk} \sim 0.9^k \mathcal{U}[-1, 1], \quad x \sim \mathcal{U}[-1, 1]^d. \tag{39}$$

Each seed fixes the coefficients and the input sample shared across orders. Exact gradients and full Hessian tensors are obtained by automatic differentiation of these analytic functions; model derivatives are obtained by differentiating the scalar network output. These families are additive, so their mixed-coordinate target Hessian entries are zero. An interacting control adds $0.25 \sum_{j<d} \sin(x_j x_{j+1})$ to each target, which makes the adjacent mixed Hessian entries nonzero while keeping every other setting fixed.

I.2 Loss conventions and evaluation protocol

For a minibatch of size B, define the component-averaged order-q error

$$M_q = \frac{1}{B d^q} \sum_{i=1}^{B} \| D^q f_\theta(x_i) - D^q f(x_i) \|_F^2, \qquad q \geq 1, \tag{40}$$

and let M_0 be the usual scalar value MSE. The reference objective is

$$L_K = M_0 + \sum_{q=1}^{K} d^{-q} M_q, \qquad K \in \{0, 1, 2\}. \tag{41}$$

Table 31: Prediction with first- and second-order derivative labels under the dimension-scaled objective.

Curvature: first-order B versus second-order D

	Value MSE ($\cdot 10^{-4}$) ↓			Derivative MSE ($\cdot 10^{-3}$) ↓			Lower error
Task / setting	B	D	Ratio [95% CI]	B	D	Ratio [95% CI]	pairs
Poly–trig, $d = 2$	0.0061	0.000348	**0.064** [0.043, 0.097]	0.0435	0.00167	**0.044** [0.032, 0.059]	5/5
Poly–trig, $d = 5$	1.09	0.582	**0.52** [0.43, 0.63]	1.4	0.557	**0.39** [0.31, 0.48]	5/5
Poly–trig, $d = 10$	56	45.3	**0.82** [0.75, 0.88]	18.5	14	**0.77** [0.7, 0.82]	5/5
Trig, $d = 2$	1.54	0.138	**0.15** [0.076, 0.28]	6.87	0.481	**0.11** [0.07, 0.2]	5/5
Trig, $d = 5$	17900	13400	**0.4** [0.14, 0.87]	4530	3220	**0.37** [0.13, 0.85]	4/5
Trig, $d = 10$	79900	79900	1.000†	6610	6610	1.000†	2/5†

Errors are arithmetic means. Ratios are paired geometric means, computed separately for each metric; brackets give 95% paired bootstrap intervals. Green and rust ratios have intervals below and above one; uncolored intervals include one. The final column counts lower-error pairs for the left-hand metric. Intervals resample the five seeds. †Ratios and intervals differ from one by less than 10^{-4}.

The additional d^{-q} multiplies an error that already averages all d^q tensor components. Equivalently, the order-q term is $B^{-1}d^{-2q}\sum_i \|D^q f_\theta(x_i) - D^q f(x_i)\|_F^2$, so the effective per-component weight shrinks with dimension. An element-mean control removes the extra factor and uses $L_K = M_0 + \sum_{q\le K} M_q$. Inputs, values and derivative targets remain in their original coordinates and units; neither convention equalizes empirical derivative magnitudes across orders or families, and both differ from the normalized fixed recipe in Equation 11.

The base grid comprises $2 \times 3 \times 3 \times 5 = 90$ fits, with seeds $42, 123, 456, 789, 1000$; the scaling and interaction controls add 210 fits on the same seeds. Each fit uses 1024 generated inputs: 819 for optimization and 205 for selection. All fits use four hidden layers of width 256 with softplus activations, Adam with initial learning rate 0.005, minibatches of 256, gradient clipping at norm 10 and a cap of 500 epochs. Training uses four independently sampled minibatches with replacement per epoch. The learning rate is halved after 20 unimproved selection epochs, with a floor of 10^{-6}, and early stopping has patience 50.

Model selection and test evaluation. The 205-input set provides the value-MSE monitor for the scheduler, early stopping and checkpoint selection; no derivative error enters the monitor. All reported errors are evaluated on 4096 independent test inputs drawn from a separate random stream after restoring the selected checkpoint. The test set is common across orders and variants within each seed.

I.3 Complete results and interpretation

Table 32 reports every family–dimension–order cell of the reference objective, and Table 33 gives the paired value-error effect of each supervised order under the two loss conventions and both target variants. Under the reference objective, adding gradient labels lowers the geometric mean test value MSE by factors of 29.20 ($d = 2$), 6.31 ($d = 5$) and 6.39 ($d = 10$) on the polynomial–trigonometric family, and adding Hessian labels lowers it by a further 15.51, 1.93 and 1.22 relative to first-order supervision (each in five of five seeds). The trigonometric family shows the same ordering in $d = 2$ (factors 19.70 and a further 6.69) and a weaker one in $d = 5$, while its $d = 10$ value errors remain close to parity across derivative orders.

The element-mean control shows that much of the flattening with dimension under the reference objective is a weighting convention rather than a property of curvature information. Without the extra d^{-q} factor, Hessian supervision lowers the polynomial–trigonometric test value MSE by factors of 109.89 in $d = 5$ and 67.30 in $d = 10$, against 12.20 and 7.77 under the reference weighting, and the trigonometric $d = 5$ factor rises from 5.19 to 97.37. The interacting targets, whose mixed Hessian entries are nonzero, give the same pattern under both conventions. The $d = 10$ trigonometric cells are the exception under every variant: their fits select very early (median selected epoch 5 for value-only training and 4 with Hessian labels, with no selected epoch later than 18), so no derivative order changes their error. By comparison, the median selected epoch for $d = 5$ trigonometric Hessian training is 391. These results distinguish the information supplied by curvature from the effective

Table 32: All 90 reference-objective fits, grouped into 18 cells of five seeds. Errors are on the 4096 independent test inputs, with population standard deviation. $K = 0, 1, 2$ denotes the highest supervised order. Time is mean training seconds; selected epochs are medians of the zero-based epoch index.

Family	d	K	Value MSE	Gradient MSE	Hessian MSE	Time (s)	Epoch
Poly–trig	2	0	$(1.77 \pm 0.95)10^{-5}$	$(1.93 \pm 1.17)10^{-3}$	$(1.37 \pm 0.96)10^{-1}$	12.25	492
Poly–trig	2	1	$(6.10 \pm 3.47)10^{-7}$	$(4.35 \pm 2.67)10^{-5}$	$(5.50 \pm 2.14)10^{-3}$	17.95	392
Poly–trig	2	2	$(3.48 \pm 0.92)10^{-8}$	$(1.67 \pm 0.51)10^{-6}$	$(1.40 \pm 0.51)10^{-4}$	35.78	327
Poly–trig	5	0	$(6.90 \pm 1.61)10^{-4}$	$(1.16 \pm 0.33)10^{-2}$	$(1.47 \pm 0.53)10^{-1}$	12.24	488
Poly–trig	5	1	$(1.09 \pm 0.27)10^{-4}$	$(1.40 \pm 0.23)10^{-3}$	$(2.40 \pm 0.50)10^{-2}$	22.27	484
Poly–trig	5	2	$(5.82 \pm 1.97)10^{-5}$	$(5.57 \pm 1.29)10^{-4}$	$(7.27 \pm 2.49)10^{-3}$	60.60	496
Poly–trig	10	0	$(3.90 \pm 2.10)10^{-2}$	$(1.16 \pm 0.57)10^{-1}$	$(2.37 \pm 0.94)10^{-1}$	13.54	453
Poly–trig	10	1	$(5.60 \pm 1.59)10^{-3}$	$(1.85 \pm 0.47)10^{-2}$	$(6.75 \pm 1.34)10^{-2}$	24.09	494
Poly–trig	10	2	$(4.53 \pm 0.96)10^{-3}$	$(1.40 \pm 0.31)10^{-2}$	$(4.47 \pm 0.87)10^{-2}$	90.01	495
Trig	2	0	$(1.95 \pm 1.52)10^{-3}$	$(8.56 \pm 7.09)10^{-2}$	$(3.28 \pm 2.51)10^{0}$	14.97	486
Trig	2	1	$(1.54 \pm 1.42)10^{-4}$	$(6.87 \pm 6.64)10^{-3}$	$(4.43 \pm 4.22)10^{-1}$	23.33	424
Trig	2	2	$(1.38 \pm 0.96)10^{-5}$	$(4.81 \pm 3.92)10^{-4}$	$(2.89 \pm 2.45)10^{-2}$	40.73	395
Trig	5	0	$(2.17 \pm 1.24)10^{0}$	$(5.12 \pm 3.30)10^{0}$	$(1.54 \pm 1.01)10^{1}$	3.19	28
Trig	5	1	$(1.79 \pm 1.50)10^{0}$	$(4.53 \pm 3.72)10^{0}$	$(1.46 \pm 1.10)10^{1}$	8.79	45
Trig	5	2	$(1.34 \pm 1.67)10^{0}$	$(3.22 \pm 4.15)10^{0}$	$(9.96 \pm 12.08)10^{0}$	38.57	391
Trig	10	0	$(8.05 \pm 2.37)10^{0}$	$(6.61 \pm 3.72)10^{0}$	$(8.85 \pm 7.00)10^{0}$	2.43	5
Trig	10	1	$(7.99 \pm 2.37)10^{0}$	$(6.61 \pm 3.73)10^{0}$	$(8.85 \pm 7.00)10^{0}$	3.39	4
Trig	10	2	$(7.99 \pm 2.37)10^{0}$	$(6.61 \pm 3.73)10^{0}$	$(8.85 \pm 7.00)10^{0}$	11.55	4

Table 33: Paired value-error effect of supervised orders under each loss convention and target variant. Entries are geometric mean factors by which the five-seed test value MSE falls relative to value-only training on the same seeds, with the number of seeds that improve.

		$K = 1$ vs $K = 0$			$K = 2$ vs $K = 0$		
Variant	Family	$d = 2$	$d = 5$	$d = 10$	$d = 2$	$d = 5$	$d = 10$
Dimension-scaled d^{-q}, additive targets	Poly–trig	29.20 (5/5)	6.31 (5/5)	6.39 (5/5)	452.93 (5/5)	12.20 (5/5)	7.77 (5/5)
Dimension-scaled d^{-q}, additive targets	Trig	19.70 (5/5)	2.07 (4/5)	1.01 (4/5)	131.73 (5/5)	5.19 (4/5)	1.01 (4/5)
Element-mean, additive targets	Poly–trig	29.44 (5/5)	11.87 (5/5)	37.24 (5/5)	901.07 (5/5)	109.89 (5/5)	67.30 (5/5)
Element-mean, additive targets	Trig	21.49 (5/5)	10.66 (5/5)	1.01 (3/5)	259.04 (5/5)	97.37 (5/5)	1.01 (3/5)
Dimension-scaled d^{-q}, interacting targets	Poly–trig	20.17 (5/5)	5.07 (5/5)	6.67 (5/5)	241.16 (5/5)	8.95 (5/5)	7.99 (5/5)
Dimension-scaled d^{-q}, interacting targets	Trig	27.01 (5/5)	1.79 (4/5)	1.00 (3/5)	176.46 (5/5)	2.73 (5/5)	1.00 (3/5)
Element-mean, interacting targets	Poly–trig	23.30 (5/5)	9.81 (5/5)	40.48 (5/5)	537.89 (5/5)	71.76 (5/5)	81.48 (5/5)
Element-mean, interacting targets	Trig	26.64 (5/5)	5.66 (5/5)	0.97 (0/5)	276.68 (5/5)	31.21 (5/5)	0.98 (0/5)

weight assigned to it: Hessian labels improve prediction in several tested settings, but increasing derivative order does not resolve the early-stopping behavior of the $d = 10$ trigonometric fits.

Training times measure end-to-end optimization, including the different epoch counts induced by early stopping. They exclude a uniform accounting of label-generation cost and do not measure per-update Hessian complexity.

J Derivative support, representation and continuation

Derivative labels constrain local behavior at the inputs where they are observed. These experiments distinguish extending a learned function beyond that support from acquiring derivative labels in the target region. They also test whether changing the geometry of the support changes the value-prediction outcome.

J.1 Periodic target and model comparison

The target is

$$f^*(x) = \sin(2\pi x) + 0.5\sin(6\pi x + 0.4) + 0.25\cos(10\pi x - 0.2).$$

We train on 24 inputs in $[-1, 1]$: 22 uniform interior draws and both endpoints. Evaluation uses a 2,400-point grid in $[-3, 3]$. Each of five seeds covers 18 derivative-noise levels from zero to three.

We add $\rho[\mathrm{sd}(g) + 10^{-12}]\epsilon$ to the clean derivative labels, with standard Gaussian ϵ shared across the noise trajectory of a seed. Thus ρ measures gradient-relative corruption; Appendix F instead uses absolute noise units. Values are centered by their training sample mean. Because the Fourier model has no intercept, this centering can leave a constant residual outside its feature span.

The models are a 3×64 softplus MLP with $\beta = 2$, a SIREN with $\omega_0 = 20$ and its prescribed initialization (Sitzmann et al., 2020), a Snake network with learned per-neuron frequency parameters initialized at $\log \alpha = 0$ (Ziyin et al., 2020), and a ten-feature Fourier-linear model with no intercept and maximum frequency five. The neural comparisons change activation and associated initialization; the Fourier comparison also changes the representation and parameterization.

All models use full-batch AdamW for 1,600 updates, weight decay 10^{-7} and gradient clipping at ten, with no learning-rate scheduler, early stopping or checkpoint selection. Learning rates are 10^{-3} for softplus and Snake, 5×10^{-4} for SIREN and 5×10^{-3} for Fourier-linear. Inputs remain in raw coordinates; values are centered and divided by their training standard deviation plus 10^{-12}, and derivatives are divided by the same output scale. For these normalized targets let $s_g = \mathrm{sd}(g) + 10^{-12}$. The derivative objective is $\mathrm{MSE}(\hat{y}, y) + \mathrm{mean}[(\hat{g} - g_{\mathrm{noisy}})^2 / s_g^2]$; value-only training uses the first term. This additional derivative-scale normalization differs from the construction core and changes the effective derivative weight.

Evaluation is separated into interpolation ($|x| \leq 1$), near continuation ($1 < |x| \leq 1.5$), and far continuation ($1.5 < |x| \leq 3$). At each operating point we compute the mean seed-paired natural-log ratio of derivative-trained to value-only MSE. The reported crossover is the first zero of its LOWESS-smoothed curve, using smoothing fraction 0.20. A left-censored entry means the mean is already positive at zero corruption; a right-censored entry means no crossing is found in the tested range. This smoothing convention differs from the piecewise-linear matched-core interpolation in Appendix E.

Table 34: Periodic-target comparison. The two right columns give mean paired natural-log MSE ratio and seed standard deviation. Negative favors derivative training. A zero marked † denotes left censoring, not an estimated zero-noise crossing.

Model	Region	First crossing	$\rho = 0$	$\rho = 3$
SIREN	Near	1.157	-0.20 ± 0.16	-0.01 ± 0.13
	Interpolation	0.709	-0.35 ± 0.26	$+0.72 \pm 0.39$
	Far	0.007	-0.03 ± 0.11	$+0.05 \pm 0.13$
Softplus	Near	$0^{\dagger}$	$+0.17 \pm 1.75$	$+2.28 \pm 2.80$
	Interpolation	0.975	-0.34 ± 0.50	$+1.06 \pm 0.40$
	Far	$0^{\dagger}$	$+0.36 \pm 1.67$	$+2.43 \pm 2.59$
Snake	Near	$0^{\dagger}$	$+0.86 \pm 1.55$	$+3.36 \pm 1.58$
	Interpolation	0.760	-1.19 ± 0.98	$+2.01 \pm 0.31$
	Far	$0^{\dagger}$	$+1.06 \pm 1.01$	$+2.93 \pm 0.89$
Fourier	Near	0.518	-0.99 ± 0.69	$+2.90 \pm 1.35$
	Interpolation	0.518	-0.99 ± 0.69	$+2.91 \pm 1.34$
	Far	0.518	-0.98 ± 0.69	$+2.91 \pm 1.34$

SIREN's mean log ratio at zero corruption corresponds to an approximately 18.4% geometric near-region value improvement. Its first crossing at 1.157 is followed by further sign changes: the mean is positive at 1.2, 1.6 and 2.5, and negative at 2 and 3. The crossing marks the end of the initial beneficial interval; the subsequent response is nonmonotone. The far-region effect is smaller relative to seed variation, with an initial mean of -0.03 and standard deviation 0.11. Softplus and Snake also exhibit substantial seed dispersion, especially outside the training interval.

The Fourier model has almost identical regional crossings (0.5176–0.5183). Its period-one feature map fixes continuation over integer periods, whereas the neural models learn their continuation from the local observations. This property concerns integer-period shifts; arbitrary phase shifts also change a finite sampling design. The finite-step Adam fit differs from the exact least-squares estimator in Appendix F, whose zero-noise crossing assumes clean values and equal approximation bias. Optimization and sampling therefore remain relevant even when the target frequencies belong to the model's span.

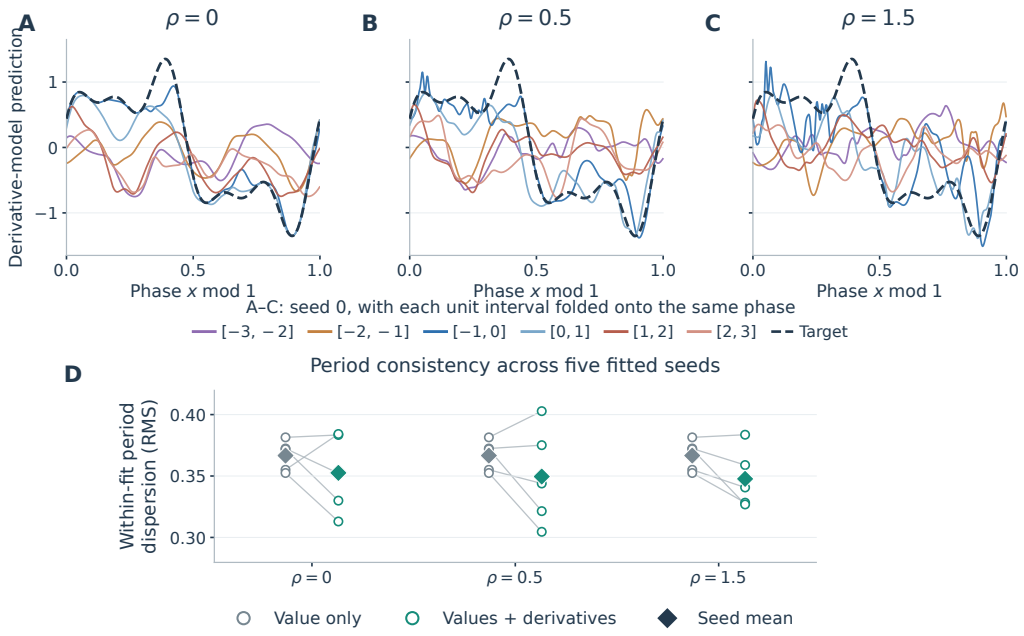

Figure 10: Period consistency of SIREN predictions. A–C: derivative-trained predictions for seed 0 at three derivative-relative noise levels. Each coloured curve is one unit interval folded onto $x \bmod 1$; the dashed curve is the target. Linear interpolation aligns the evaluation grid to 800 phase locations. D: dispersion across the six intervals, computed within each fitted model before summarizing seeds. Connected points pair value-only and derivative fits; diamonds are five-seed means. Values are in the original target scale. The figure assesses periodic continuation, while Table 34 evaluates prediction accuracy.

Period consistency within a fitted model. To test whether each SIREN fit learns a common periodic continuation, we compare its predictions at the same phase in six unit intervals. For a fixed seed and noise level, let

$$\mu(\varphi) = \frac{1}{6}\sum_{j=-3}^{2} \hat{f}(j+\varphi), \qquad D_{\text{period}} = \left[\frac{1}{6}\sum_{j=-3}^{2}\int_0^1 \left(\hat{f}(j+\varphi) - \mu(\varphi)\right)^2 d\varphi\right]^{1/2}.$$

We linearly interpolate the evaluation-grid predictions onto 800 equally spaced phases and calculate this quantity separately for every fit. The same calculation on the period-one target gives 5.08×10^{-5}, quantifying the interpolation error. Mean derivative-model dispersion over the five seeds is 0.353, 0.350 and 0.348 at $\rho = 0$, 0.5 and 1.5, compared with 0.367 for value-only training. Figure 10 shows the interval-specific curves for seed 0 and the paired dispersion across all five. The regional prediction benefit therefore coexists with substantial inconsistency across integer periods. This diagnostic measures period consistency; accuracy still requires comparison with the target, since even a constant prediction has zero period dispersion.

J.2 Acquiring derivative information outside the value support

We add 24 inputs drawn from $1 < |x| \leq 3$ and expose their analytic derivative labels to SIREN. No value labels at those inputs enter training. The model, seed schedule and noise grid match the preceding comparison. This is derivative-assisted adaptation to the target region: the training information now reaches beyond the original value support.

At zero corruption, added derivative support reduces near-region value MSE by a paired geometric factor of 1.60. The far-region factor is 1.04, with two of five seeds worsening slightly. Extending

Table 35: First smoothed crossings with and without target-region derivative labels. The > 3 entry is right-censored at the largest tested noise level.

Region	Original support	Added derivative support
Interpolation	0.71	0.78
Near	1.16	> 3
Far	0.007	0.98
All	1.01	1.14

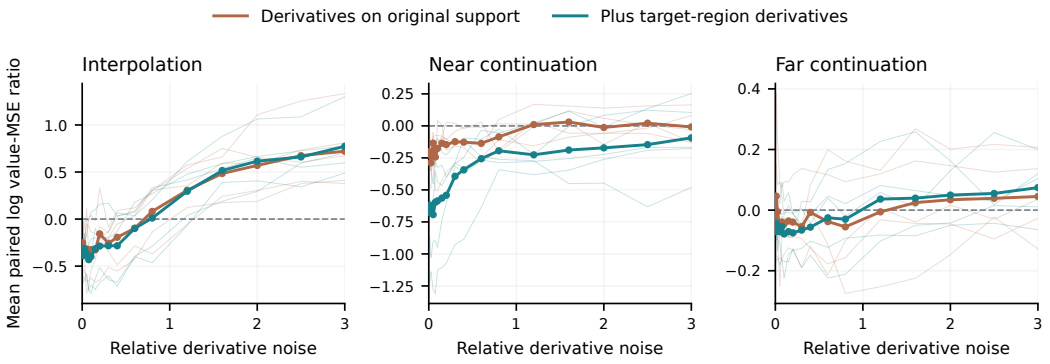

Figure 11: Acquiring derivative labels in the target region. Solid curves show five-seed mean paired log value-MSE ratios for SIREN; faint lines show individual seeds. Both comparisons share the value-only reference. Adding target-region derivatives changes the training information and sustains the near-region benefit over the tested noise range. The plotted points are unsmoothed; Table 35 uses its stated LOWESS crossing convention.

Table 36: Acquiring clean derivative labels ($\rho = 0$) in the target region while holding value-label support fixed.

SIREN support: original region B versus added target-region derivatives D

	Value MSE ↓			Derivative MSE ↓			Lower error
Task / setting	B	D	Ratio [95% CI]	B	D	Ratio [95% CI]	pairs
Near region	1.18	0.759	**0.62** [0.49, 0.77]	164	87.2	**0.53** [0.42, 0.64]	5/5
Far region	1.29	1.25	0.9594 [0.87, 1.045]	209	196	0.929 [0.86, 1.009]	3/5

Errors are arithmetic means. Ratios are paired geometric means, computed separately for each metric; brackets give 95% paired bootstrap intervals. Green and rust ratios have intervals below and above one; uncolored intervals include one. The final column counts lower-error pairs for the left-hand metric. Intervals resample the five paired seeds.

derivative support therefore has its clearest effect near the original boundary, where it both improves clean-label prediction and broadens the useful-noise interval.

J.3 Halfspace and radial support geometry

A complementary study uses the 4×256 softplus backbone on poly-trig and trig, with $d \in \{2, 5, 10\}$, training counts 512 and 2,048, two support splits, three methods (value-only, fixed derivative weight and GradNorm), and five seeds: 360 fits. No derivative corruption is added. The halfspace split uses one side of $x_1 = 0$ for training and the other for evaluation. The radial split uses an inner ℓ_∞ cube for training and the outer part for evaluation, choosing its radius to divide the cube's volume equally.

Derivative training improves 11 of 12 radial cells, with at least 50% reduction in ten. All six poly-trig halfspace cells improve, whereas all six trig halfspace cells worsen. Thus the same derivative objective responds differently to the target and support geometry: radial support gives broad gains, while the halfspace comparison separates the two target families. Together with the one-dimensional

Table 37: Relative change in mean value MSE, $100(\overline{E}_D/\overline{E}_V - 1)$, for fixed-weight derivative training. Negative is better. Each row summarizes five seeds; these are ratios of arithmetic means, not mean log ratios.

Family	d	Training count	Halfspace (%)	Radial (%)
Poly-trig	2	512	−33.2	−88.1
	2	2048	−9.9	−87.7
	5	512	−36.4	−97.0
	5	2048	−43.2	−97.2
	10	512	−68.0	−98.2
	10	2048	−57.9	−98.4
Trig	2	512	+16.9	−79.4
	2	2048	+14.1	−60.4
	5	512	+541.8	−51.0
	5	2048	+271.2	−95.0
	10	512	+41.1	+12.7
	10	2048	+688.4	−5.2

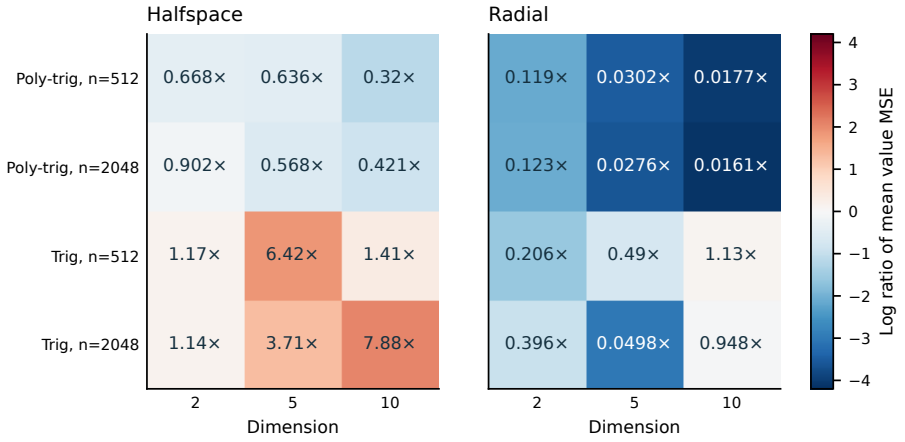

Figure 12: Support geometry and target structure jointly change derivative usefulness. Entries are ratios of five-seed mean value MSE, derivative training over value-only training, for the fixed-weight recipe. Colors use the natural logarithm on one symmetric scale across both panels; blue favors derivative training. This visualizes the complete comparison in Table 37, including cells where the direction changes between geometries.

support experiment, this shows why both the location of derivative observations and the chosen representation matter for continuation.

K Architecture, selection and physical evaluation

K.1 Burgers Fourier neural operator

We use the one-dimensional Burgers archive labelled $\nu = 0.01$ from PDEBench, whose solver uses diffusion coefficient ν/π (Takamoto et al., 2022) to test whether derivative supervision can improve value prediction with a Fourier neural operator (FNO) (Li et al., 2021a). The operator receives a band-limited initial-condition channel and the space/time coordinate channels and predicts a space–time field. The initial-condition channel is reconstructed from the same 16 complex Fourier coefficients used by the pointwise MLP. Training uses 1800 initial conditions and 300 sampled interior space–time locations per condition, totaling 540,000 supervised values. The FNO and pointwise

Table 38: Burgers FNO field quantities from the same predicted grids. Positive mean paired reduction favors derivative supervision. Every row contains twenty paired seeds under each monitor. Intervals resample training seeds, conditional on the simulation split.

	Common value monitor			Each recipe's combined monitor		
Metric	Reduction (%)	95% interval	Lower error	Reduction (%)	95% interval	Lower error
Sampled-point value MSE	+25.41	[22.33, 28.57]	20/20	+23.17	[19.14, 27.04]	20/20
Full-field MSE	+32.73	[30.11, 35.18]	20/20	+31.02	[27.40, 34.42]	20/20
Interior-field MSE	+21.59	[18.20, 24.81]	20/20	+18.71	[14.22, 23.06]	20/20
Boundary-band MSE	+63.19	[60.92, 65.29]	20/20	+65.15	[62.10, 67.25]	20/20
Mean trajectory relative L^2	-2.35	[-11.32, 5.90]	10/20	+1.20	[-5.72, 7.64]	12/20
Pooled relative L^2	+18.05	[16.48, 19.56]	20/20	+17.09	[14.93, 19.15]	20/20

datasets share their sampled coordinates, values and initial-condition representation. Evaluation uses held-out initial conditions.

The FNO has four Fourier layers, width 20, twelve retained modes in each coordinate, a temporal padding of eight cells, and 926,177 trainable parameters. Adam uses learning rate 10^{-3}, weight decay 10^{-4}, batches of 16 simulations and norm-1 gradient clipping. The epoch cap is 150, early-stopping patience is 25, and the plateau scheduler has patience 10 and factor 0.5. Seeds $1, \ldots, 20$ pair the value-only and derivative-supervised recipes on the same cached data.

What is differentiated. Let $\hat{u}(t, x)$ be the predicted field. The derivative objective compares centered differences

$$D_x\hat{u}(t,x) = \frac{\hat{u}(t, x + h_x) - \hat{u}(t, x - h_x)}{2h_x}, \qquad D_t\hat{u}(t,x) = \frac{\hat{u}(t + h_t, x) - \hat{u}(t - h_t, x)}{2h_t} \tag{42}$$

at the sampled locations. These are spatial and temporal derivatives of the output field, not derivatives of the operator with respect to its initial-condition input. Training normalizes values by their sampled training standard deviation and scales derivative components by the corresponding coordinate standard deviations divided by the value standard deviation. The derivative loss averages the two component MSEs. The derivative-supervised objective is $\frac{1}{2}L_y + \frac{1}{2}L_g$; value-only training uses L_y.

Monitoring and metrics. The primary panel uses a common validation value error for scheduling, early stopping and checkpoint selection, averaged over validation examples. A second panel uses each recipe's weighted validation objective and the trainer's mean over validation batches. Each panel contains twenty paired seeds; changing the monitor changes the training procedure, not merely which checkpoint is selected afterwards. Both methods use the same cache within each pair.

For trajectory j, let $S_j = \sum_{t,x}(\hat{u}_j - u_j)^2$ and $Q_j = \sum_{t,x} u_j^2$. Full-field MSE averages squared errors over all trajectories and grid points. The pooled relative error is $\sqrt{\sum_j S_j / \sum_j Q_j}$; the mean trajectory-relative error is $J^{-1}\sum_j \sqrt{S_j/Q_j}$. These are different scientific summaries: the latter normalizes each trajectory before averaging. Boundary-band MSE uses the four outer grid rows or columns in the space–time array; interior MSE uses the complementary rectangle. The two-component derivative MSE averages the spatial and temporal derivative errors. It is not identical to the pointwise MLP's derivative metric, which also includes initial-condition feature slots.

Paired outcomes. Table 38 reports mean paired reductions, $100(1 - E_{D,s}/E_{V,s})$, with 95% percentile intervals from 20,000 resamples of training seeds, conditional on the held-out simulations. Sampled-point and full-field errors improve in every pair under both monitors. Under value monitoring, the derivative-error reduction is 65.78% ([62.82,68.37]%), and the boundary band's share of total squared error falls from 26% to 14%.

Pooled relative L^2 improves in every pair, whereas the mean trajectory-relative change is inconclusive. The ten lowest-reference-norm trajectories have mean relative errors 0.56 and 0.79, compared with 0.110 and 0.088 for the remaining ninety. Normalizing each trajectory separately gives low-amplitude fields greater influence.

K.2 BURGERS: SPATIAL ERROR AND TRAINING CONTROLS

Spatial contributions to the error change. For periodic spatial errors $e_j(t, x)$, define the normalized discrete Fourier transform on grid indices $x = 0, \ldots, N_x - 1$ by

$$\hat{e}_j(t, k) = N_x^{-1} \sum_x e_j(t, x) \exp(-2\pi i k x / N_x).$$

One-sided powers count conjugate pairs twice, except at zero and the Nyquist frequency. Parseval's identity then gives the exact accounting

$$E_D - E_V = \sum_B \left\langle \sum_{k \in B} w_k \big(|\hat{e}_D(t, k)|^2 - |\hat{e}_V(t, k)|^2 \big) \right\rangle_{j,t}, \tag{43}$$

where B partitions the spatial modes and angle brackets average trajectories and times. The components locate prediction error by spatial scale; they are error powers rather than physical energy fluxes.

Figure 3A groups the endpoint spectra into the spatial mean, modes 1–2, modes 3–15 and the complete higher-frequency tail. In all twenty seed pairs, the intermediate band improves and the spatial mean worsens; the reduction in intermediate-band error outweighs the penalties at the mean and longest wavelengths. The same pattern occurs under combined-loss monitoring. The bands describe these observed spectra and are not architecture cutoffs: the spectral blocks retain twelve modes per coordinate, whereas the initial-condition encoding uses sixteen complex coefficients. Nonlinear layers can generate higher output frequencies.

Sobolev objectives motivate frequency-dependent weighting (Czarnecki et al., 2017; Liu et al., 2024), and training trajectories resolve the acquisition of these components (Rahaman et al., 2019). For the objective used here, centered differences have multiplier $i \sin(2\pi k / N_x)/h_x$, rather than ik over the entire resolved spectrum. Sparse observations and a shared nonlinear model can couple the frequency components. The following training controls examine observation access, learning dynamics and objective weights. The Darcy comparison in Appendix K.4 then tests which aspects of the error distribution recur in a different field.

A spectral compromise is conditional on the representation. A simple controlled example separates a possible compromise from an unavoidable one. On a complete, uniformly weighted periodic grid, take orthonormal modes ϕ_1, ϕ_8 and target $a_1\phi_1 + a_8\phi_8$, with consistent noiseless value and derivative targets. Independent fitted coefficients recover both modes exactly, with or without derivative supervision. If the model instead ties the coefficients, $h(\phi_1 + \phi_8)$, minimizing squared value error plus $\lambda \geq 0$ times squared derivative-matching error gives

$$h_\lambda = \frac{a_1(1 + \lambda q_1) + a_8(1 + \lambda q_8)}{2 + \lambda(q_1 + q_8)},$$

where q_k is the squared derivative multiplier for mode k. For centered differences on a sufficiently fine grid, $q_8 > q_1$: increasing λ moves the fitted coefficient toward a_8. When $a_1 \neq a_8$, the intermediate-mode error falls while the low-mode error rises. Total value error increases relative to the exact value-only optimum in this noiseless example. On the full grid, a separately fitted constant remains unchanged: orthogonality decouples it from the zero-mean modes, and its derivative is zero. Thus coupling can produce a spectral compromise, whereas an independently represented mean need not deteriorate. Exact least-squares controls check these separable and coupled cases under full-grid and sparse observations; their code will be released with the camera-ready version. They establish possible mechanisms; the FNO attribution requires interventions on its own training procedure.

Observation access versus derivative use. Finite differences use values at neighboring locations that do not enter the center-value objective. We therefore give a value-only FNO the deduplicated union of the centers and their four spatial/temporal stencil neighbors. Its objective N averages value errors over that union within each simulation; normalization remains based on the original training centers. The derivative objective uses $0.5V + 0.5G$, where V is center-value error and G averages the normalized spatial and temporal derivative errors. A $0.5V$ control separately holds the center-value coefficient fixed while removing derivative matching.

Table 39: Matched-observation and value-coefficient controls for Burgers FNO. Entries are arithmetic means over five paired seeds. Full-field and sampled value errors use the same held-out simulations; derivative error averages the spatial and temporal components. Checkpoints minimize validation value error.

Objective	Full-field MSE $\cdot 10^3$	Sampled MSE $\cdot 10^3$	Derivative MSE
Center values ($0.5V$)	4.449	3.707	6.322
Neighbor values (N)	3.299	2.668	5.021
Values and differences ($0.5V + 0.5G$)	2.228	1.919	1.635

All three comparisons use five paired seeds, the same training data, architecture, batches and 150-epoch budget, with no early stopping. Learning rates decrease after epochs 50 and 100; validation value error selects checkpoints independently of the learning trajectories. Thus these controls compare matched procedures, while their schedule differs from the plateau schedule of the twenty-seed application panel. Table 39 reports the common test metrics. Derivative supervision reduces full-field MSE relative to neighboring-value supervision in all five pairs: the mean paired reduction is 32.28% (95% seed-bootstrap interval [26.85,37.71]%). Against $0.5V$, the reduction is 49.86% ([46.73,53.00]%), again with lower error in all five pairs. Intervals enumerate all 5^5 ordered seed resamples, conditional on the common simulation split.

The two controls separate access to observations from their use in the objective. Union averaging gives the value learner the same unique stencil observations, while changing the weight of each center. The $0.5V$ control matches the derivative recipe's center-value coefficient. Derivative matching improves prediction against both controls. At epoch 150, the neighboring-value control reduces mean full-field MSE from 3.397 to 3.290 in units of 10^{-3}, whereas derivative matching reduces it to 2.194. The much larger improvement with derivative matching shows that observation access alone does not recover its benefit at this optimization budget.

Learning dynamics at matched observation access. Figure 13 distinguishes improved final accuracy from faster acquisition of spatial structure. For each seed and band, we divide prediction-error power by reference-field power and record the first of two consecutive observations below 0.25. Snapshots are dense during the first five epochs and then occur every five epochs. A crossing therefore identifies an observation interval, rather than an exact continuous training time. Thresholds 0.1, 0.5 and 1 provide sensitivity checks; an unobserved crossing remains right-censored.

On the held-out fields, derivative matching reaches the intermediate-band threshold 24 epochs earlier than center-value training (paired-seed 95% interval $[21, 27]$) and 23 epochs earlier than neighboring-value training ($[21, 25]$). The latter comparison also reduces integrated intermediate-band log error by 0.790 ($[0.737, 0.841]$). The same direction appears on the fixed subset of 32 training trajectories: intermediate-band acquisition is 15.47 observed epochs earlier than with neighboring values. At the primary threshold, the test-set intermediate-versus-long-wave acquisition lag falls from 28.47 to 5.47 epochs; it is reduced, rather than reversed. Thus the intermediate-scale benefit reflects changed learning dynamics as well as lower final error, and survives matched access to the observations used by the derivative stencil. This is a controlled instance of frequency-dependent acceleration, whose broader phenomenon is established in Sobolev and operator learning (Rahaman et al., 2019; Liu et al., 2024).

Loss weights and an available physical constraint. Table 40 adds full-value-weight and conserved-mean controls. Restoring the center-value coefficient to one preserves the intermediate-band gain, but does not remove the long-wave cost: relative to V, the $V + 0.5G$ objective increases $k = 1$–2 MSE by $0.182 \cdot 10^{-3}$, with paired interval $[0.127, 0.233] \cdot 10^{-3}$. The cost is therefore not explained solely by the halved value coefficient in the equal-weight recipe. The $0.5V$ comparison separately shows a strong sensitivity to objective scaling. Because this trainer uses Adam with coupled weight decay and gradient clipping, scaling the loss also changes its interaction with optimization and regularization.

Periodic Burgers dynamics conserve the initial spatial mean. The FNO input contains this quantity because its initial-condition encoding retains the zero Fourier mode. We can impose it on both value and derivative models by replacing every predicted time slice with

$$\mathcal{P}\hat{u}_j(t,x) = \hat{u}_j(t,x) - \overline{\hat{u}}_j(t) + m_{0,j}, \qquad m_{0,j} = \overline{u_j(0,\cdot)}.$$

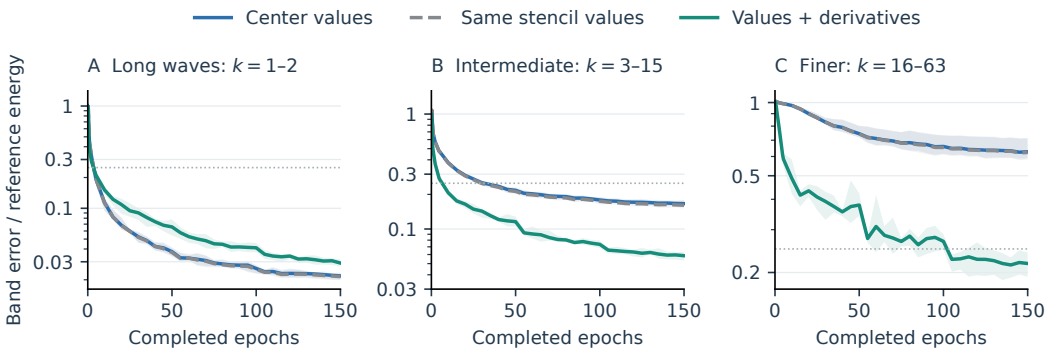

Figure 13: **Using differences changes learning dynamics beyond observation access.** All three FNOs receive the same initial-condition inputs. The neighboring-value control observes the complete deduplicated stencil used to construct derivative labels. Curves show arithmetic means over five paired seeds; shading spans the five seed outcomes. Error is normalized by the common reference energy within each band; vertical axes use logarithmic scales. The dotted line marks the 0.25 acquisition threshold, requiring two consecutive observations. Derivative matching reaches the intermediate-band threshold earlier than either value objective, while retaining greater long-wavelength error. All models share a 150-epoch budget and learning-rate schedule; the training-subset analysis gives the same acceleration conclusion.

Table 40: Burgers FNO controls over five paired seeds. All entries are arithmetic mean MSEs multiplied by 10^3. "Selected" minimizes the common validation value error; the remaining columns evaluate epoch 150. The four spectral components sum to the fixed-epoch full-field error. V uses sampled center values, N the unique stencil-value union, and G the normalized spatial/temporal derivative loss. The final row is exploratory.

Objective	Selected	Epoch 150	Mean	$k = 1$–2	$k = 3$–15	$k \geq 16$
V	3.385	3.397	0.145	0.717	2.370	0.166
$0.5V$	4.449	4.469	0.176	1.005	3.081	0.207
N	3.299	3.290	0.123	0.705	2.297	0.165
$0.5N$	4.354	4.403	0.181	1.000	3.015	0.207
$0.5V + 0.5G$	2.228	2.194	0.343	0.957	0.835	0.059
$V + 0.5G$	2.037	2.139	0.353	0.899	0.829	0.059
V + mean constraint	3.121	3.090	< 0.001	0.649	2.251	0.190
$0.5V + 0.5G$ + mean constraint	2.026	2.051	< 0.001	1.076	0.913	0.062
$V + 0.5G$ + mean constraint	1.944	1.953	< 0.001	0.998	0.893	0.062

Training with this projection nearly removes the spatial-mean error for both objectives. Derivative matching still lowers intermediate-band error by $1.339 \cdot 10^{-3}$ relative to the projected value model. The exploratory full-value-weight, projected derivative recipe has the lowest mean error among the nine trained recipes: $1.953 \cdot 10^{-3}$ at epoch 150 and $1.944 \cdot 10^{-3}$ under validation selection. It is the best trained recipe in four of five fixed-epoch comparisons and two of five validation-selected comparisons. Useful derivative constraints therefore coexist with an explicitly enforced physical quantity, although the best recipe varies across seeds and checkpoint rules.

Mean correction and retraining answer different questions. A spatial derivative does not identify a spatially constant offset, but the temporal derivative used here does constrain time-varying spatial means. The joint space–time constant is unconstrained by either derivative; this fact alone does not require its prediction error to increase. Moreover, training with a mean projection changes the optimization trajectory and can alter every nonconstant mode. For a frozen prediction, in contrast, $\mathcal{P}$ leaves every nonzero spatial Fourier coefficient unchanged, and its effect is exactly

$$E(\mathcal{P}\hat{u}) - E(\hat{u}) = \left\langle (m_0 - \bar{u}(t))^2 - (\overline{\hat{u}}(t) - \bar{u}(t))^2 \right\rangle_{j,t}. \tag{44}$$

The first term retains any numerical drift of the reference away from its initial mean. This identity separates removable mean error from changes in the learned nonconstant field components, allowing the frozen correction to be compared with training under the projection. Darcy has no corresponding

Table 41: Using the known initial mean after training or throughout training. Entries are mean full-field MSEs multiplied by 10^3 over five paired seeds. "Original" is the unprojected model, "Frozen" applies Eq. (44) to that same prediction, and "Trained" imposes the projection during optimization. Each trained model selects its own checkpoint by validation value error. The frozen correction uses the unprojected model's selected checkpoint without reselection.

	Epoch 150			Validation-selected		
Objective	Original	Frozen	Trained	Original	Frozen	Trained
V	3.397	3.252	3.090	3.385	3.263	3.121
$0.5V + 0.5G$	2.194	1.851	2.051	2.228	1.863	2.026
$V + 0.5G$	2.139	1.786	1.953	2.037	1.768	1.944

spatial-mean conservation law, so this particular intervention does not transfer to its reconstruction experiment. Table 41 evaluates both interventions using the initial mean encoded in the model's input. The reference's mean-drift contribution is $2.74 \cdot 10^{-9}$ MSE. Correcting the frozen $V + 0.5G$ model reduces fixed-epoch MSE from $2.139 \cdot 10^{-3}$ to $1.786 \cdot 10^{-3}$, below the $1.953 \cdot 10^{-3}$ attained by training the corresponding projected model. Frozen correction has lower error in all five pairs, with a mean difference of $-0.167 \cdot 10^{-3}$ and a paired 95% interval $[-0.231, -0.090] \cdot 10^{-3}$. The same comparison holds under validation selection: $1.768 \cdot 10^{-3}$ versus $1.944 \cdot 10^{-3}$, again in all five pairs, with difference interval $[-0.271, -0.081] \cdot 10^{-3}$.

The frozen correction changes only the spatial mean and retains the learned nonconstant modes. Its advantage over retraining therefore shows that enforcing conservation during optimization can change those modes unfavorably, even when the constraint itself is correct. This comparison identifies a useful intervention for this Burgers operator; it does not attribute every low-frequency error to the derivative null space.

K.3 Equation constraints and derivative labels

Spatial derivative matching and a PDE residual supply different information. The former supervises derivatives of simulated output; the PINO-style control adds the residual of $u_t + uu_x - (0.01/\pi)u_{xx} = 0$ on the stored grid (Li et al., 2021b). The operator, simulation split and value-evaluation locations match the FNO panel. The residual follows the periodic spatial Fourier and centered temporal discretization of the Burgers implementation; its normalization uses the mean squared temporal derivative of training fields only. The objective is $L_y + \gamma L_{\mathrm{PDE}}$, with the common raw-value validation monitor.

A five-seed validation search compares $\gamma \in \{0.1, 1, 10\}$ and an extension to 0.01 when the minimum lies at the lower boundary. All four positive weights have larger validation error than value-only training. The smallest positive weight gives 0.0361 compared with 0.0060 for value only. Thus this search provides no validation improvement from activating the equation penalty. Table 42 reports test errors for these five search seeds as a residual-weight sensitivity analysis. Derivative-label supervision improves all five of these pairs, while none of the positive residual weights does so.

The squared normalized residual evaluated on 200 reference training fields is 0.080. It is a discretization-consistency diagnostic, not a lower bound on a model's residual: a prediction can reduce the residual by departing from the simulated field. Accordingly, residual reduction and improved prediction must be evaluated separately. This experiment compares derivative labels and equation constraints within a shared sparse-supervision FNO procedure.

K.4 Darcy: spatial location and spectral scale

A complementary reconstruction uses the two-dimensional Darcy fields from PDEBench and a four-layer, width-256 softplus MLP. Inputs contain spatial coordinates, local permeability and the real and imaginary parts of a 4×4 low-frequency block of the permeability field's Fourier transform. The five paired seeds use a common simulation split, batches of 256, initial learning rate 0.005 and a 500-epoch cap. Each method's validation objective controls its plateau scheduler and selected checkpoint. Thus this extension compares the value-only and equal-weight derivative recipes, including their

Table 42: PINO residual-weight search on five paired seeds. Validation monitor and test value MSE are means. Every paired ratio is the geometric mean test-MSE ratio to value only; counts give pairs with lower test MSE.

Residual weight	Validation monitor	Test value MSE $\cdot 10^3$	Geom. paired ratio	Lower error
none (value only)	0.0060	2.563	1.00	–
none, derivative labels	0.0049	1.920	0.74	5/5
0.01	0.0361	13.842	5.38	0/5
0.1	0.0534	20.607	8.02	0/5
1	0.0834	37.430	14.59	0/5
10	0.1014	46.848	18.28	0/5

Table 43: Darcy reconstruction by evaluation region, over five paired training seeds. Value-only and derivative columns are arithmetic mean MSEs multiplied by 10^3. G is the geometric paired derivative/value-only ratio; percentile intervals enumerate all 5^5 ordered seed resamples. The four-cell boundary strip contains 1,984 of 16,384 pixels; the interior contains the remainder. All rows evaluate the same selected models on the same 100 test fields.

Region / metric	Value only	Derivative	Paired G [95% CI]	Lower error
Full grid	1.072	1.019	0.914 [0.733, 1.217]	4/5
Interior grid	1.138	1.140	0.968 [0.757, 1.303]	3/5
Four-cell boundary strip	0.599	0.137	0.234 [0.140, 0.413]	5/5
Sampled interior points	1.108	1.138	0.985 [0.748, 1.384]	3/5

monitoring procedure. Test evaluation covers 100 complete 128×128 fields, including a four-cell boundary strip excluded from sampled training locations.

For these nonperiodic square fields, an orthonormal two-dimensional DCT-II gives an exact analogue of Eq. (43): sum squared coefficients over disjoint square shells indexed by $q = \max(k_x, k_y)$ and divide by the number of grid points. These are coordinates for accounting for reconstruction error, not eigenmodes of the heterogeneous Darcy operator. We group indices into $q = 0$, $q = 1$–2, $q = 3$–15 and $q \geq 16$; these cutoffs do not equate physical scales with the Burgers Fourier bands. The $q = 3$–15 and $q \geq 16$ bands improve in all five pairs. Mean-mode error increases on average but in only three pairs, unlike its uniform increase in the Burgers panel. A mean penalty is therefore not a common requirement of the gain.

Table 43 separates location from scale. Arithmetic mean full-field MSE decreases by 5.01%, with appreciable seed variation. Interior MSE is almost unchanged, whereas the boundary-strip MSE decreases by 77.16% in the ratio of arithmetic means and improves in every pair. Weighting these two regions by their pixel counts accounts exactly for the full-field change: the boundary improvement outweighs the small interior increase. The scale decomposition and spatial partition describe complementary aspects of the same predictions. Figure 14 illustrates this distinction on the first held-out field: reduced boundary error coexists with increased interior error.

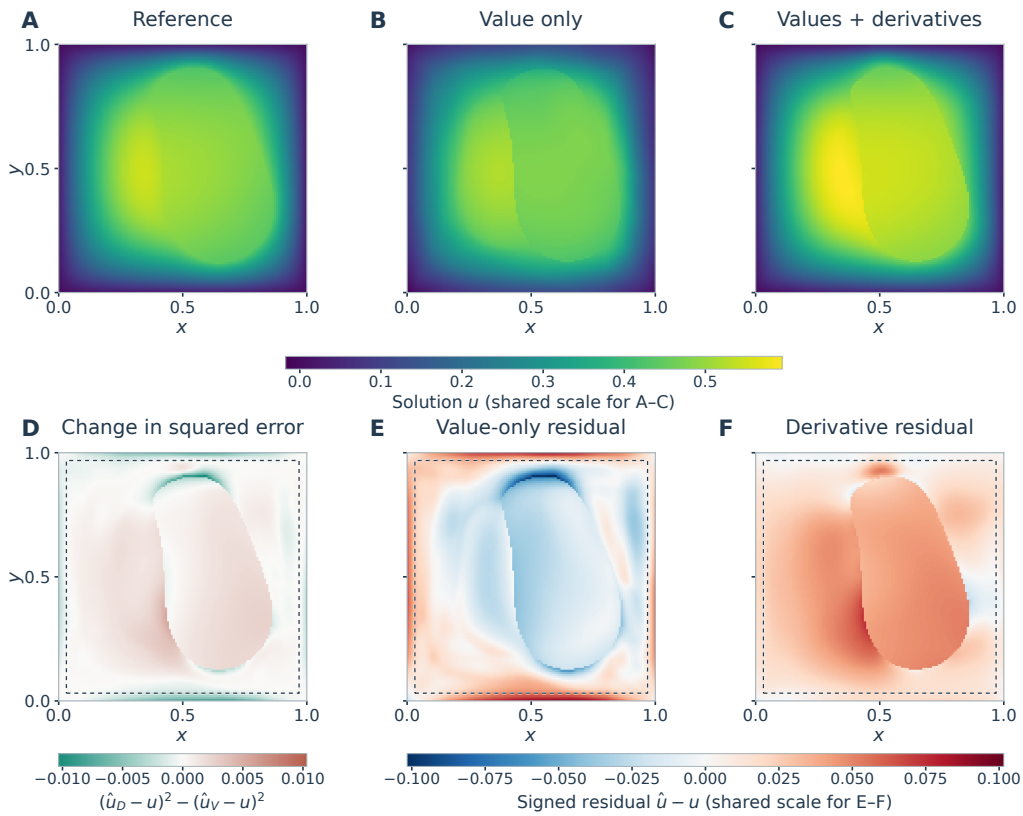

Figure 14: **An individual Darcy reconstruction and its spatial errors.** Illustration for seed 1 and the first held-out field (simulation 166). A–C show the common reference and the two selected predictions on one colour scale. D shows the local change in squared error (derivative minus value-only; teal indicates improvement). E–F show signed residuals on a shared symmetric scale. All scales span the full displayed extrema without clipping. The dashed frames mark the four-cell boundary margin excluded from training locations. Coordinates are the original unit-square coordinates; values are in the original PDEBench solution scale. Checkpoints follow each method's validation objective. For this field, boundary error decreases while interior error increases, and total MSE rises from 0.633×10^{-3} to 1.056×10^{-3}. The aggregate five-seed comparison is reported separately in Figure 3.

K.5 MLP CAPACITY AND THE VALIDATION MONITOR

Increasing width and depth changes approximation capacity and optimization behavior simultaneously. Table 45 reports Burgers initial-condition MLP comparisons with their monitoring rule explicitly identified. The "combined" rule monitors the training recipe's weighted validation objective; for value-only training this is just value MSE. The "value" rule monitors validation value MSE for both methods. In the trainer, this rule controls the plateau scheduler and early stopping as well as the checkpoint. It therefore defines a training-procedure intervention, rather than a post-training choice among a fixed collection of checkpoints.

The two 4×256 groups are separate training samples; their value-only outputs differ at every seed despite using the same nominal value monitor. Their difference therefore includes training variation as well as the monitor intervention. Seed-level summaries are particularly informative for the combined-monitor group: its ratio of mean value-only to DML error is 5.91, while DML improves only five of twenty seed pairs and has the larger median error. Four value-only errors exceed 0.1, including two exceeding 1, and the maximum is 6.546. Under the value-monitored procedure, the value-only mean is 0.01364 and DML has lower error in only one of twenty pairs. The wider value-monitored groups have mean ratios close to one. These observations show that the large ratio in the combined-monitor

Table 44: Field reconstruction: full-field error and a complementary field metric.

Burgers fields: value-only B versus derivative supervision D

	Full-field MSE ($\cdot 10^{-3}$) ↓			Pooled relative L^2 ($\cdot 10^{-2}$) ↓			Lower error
Task / setting	B	D	Ratio [95% CI]	B	D	Ratio [95% CI]	pairs
Value monitor	3.31	2.22	**0.67** [0.65, 0.7]	9.77	8	**0.82** [0.8, 0.83]	20/20
Combined monitor	3.3	2.27	**0.69** [0.65, 0.72]	9.76	8.09	**0.83** [0.81, 0.85]	20/20

Darcy fields: value-only B versus derivative supervision D

	Full-field MSE ($\cdot 10^{-3}$) ↓			Boundary-strip MSE ($\cdot 10^{-3}$) ↓			Lower error
Task / setting	B	D	Ratio [95% CI]	B	D	Ratio [95% CI]	pairs
Darcy MLP	1.07	1.02	0.9136 [0.73, 1.2]	0.599	0.137	**0.23** [0.14, 0.41]	4/5

Errors are arithmetic means. Ratios are paired geometric means, computed separately for each metric; brackets give 95% paired bootstrap intervals. Green and rust ratios have intervals below and above one; uncolored intervals include one. The final column counts lower-error pairs for the left-hand metric. Burgers uses twenty seed pairs per monitor; Darcy uses five. Pooled relative L^2 divides summed squared error by summed squared reference values before taking the square root. Darcy boundary-strip MSE uses the four-grid-cell outer margin.

Table 45: Burgers initial-condition MLP value MSE by capacity and monitoring procedure. V denotes value-only and D equal-weight derivative supervision. The ratio is a ratio of arithmetic means, not a paired geometric ratio.

Network	Monitor	Pairs	Mean V	Mean D	Median V	Median D	$\bar{E}_V / \bar{E}_D$	D lower error
4×256	Combined	20	0.48566	0.08221	0.01211	0.05101	5.907	5/20
4×256	Value	20	0.01364	0.09928	0.01164	0.04197	0.137	1/20
6×512	Value	20	0.04346	0.04433	0.01082	0.04408	0.980	2/20
8×1024	Value	10	0.27598	0.28268	0.34149	0.34146	0.976	2/10

group is driven by a few large value-only errors, while derivative training has lower error in few of the paired comparisons.

K.6 Limited-input PDE controls

Burgers uses (x, t), while Darcy uses $(x, y, a(x, y))$, adding local permeability. These inputs do not identify the full initial condition or coefficient field. A predictor with limited inputs estimates a conditional mean across unresolved states. Its squared-error risk contains the conditional variance of the field given those inputs; derivative labels cannot supply omitted state variables at evaluation time. Table 46 complements the informative-input comparisons in Appendices K.1–K.5.

Table 46: Limited-input PDE controls. Mean and sample standard deviation over 20 seeds, without injected label corruption.

Problem	Recipe	Value MSE	Derivative MSE
Burgers	Value-only	0.3418 ± 0.00029	13.59 ± 0.0000028
	Fixed	0.3417 ± 0.00031	13.59 ± 0.000033
	Half	0.3418 ± 0.00032	13.59 ± 0.0000049
	GradNorm	0.3421 ± 0.0054	13.59 ± 0.00011
	ReLoBRaLo	0.3409 ± 0.00035	13.59 ± 0.000037
	Warmup	0.3409 ± 0.00011	13.59 ± 0.0013
Darcy	Value-only	0.01317 ± 0.000083	0.1463 ± 0.0060
	Fixed	0.01459 ± 0.000095	0.1475 ± 0.011
	Half	0.01417 ± 0.000056	0.1391 ± 0.014
	GradNorm	0.01459 ± 0.00062	0.1666 ± 0.050
	ReLoBRaLo	0.01434 ± 0.00051	0.1442 ± 0.020
	Warmup	0.01355 ± 0.00018	0.1193 ± 0.0016

Burgers mean value errors are close in magnitude across recipes. Darcy shows clearer recipe differences: fixed-weight value MSE is about 10.8% above the value-only mean. The input information

defines which field can be predicted, while the training objective still affects the approximation within that information limit.

K.7 Derivative coordinates and the physical target

Physical derivative evaluation requires the coordinate map used by the model. Consider an energy network whose input is the vector of all pairwise distances $d(r) \in \mathbb{R}^p$, where $r \in \mathbb{R}^{3N}$ and $p = N(N-1)/2$. Let

$$J(r) = \frac{\partial d}{\partial r} \in \mathbb{R}^{p \times 3N}, \qquad A(r) = J(r)^\top. \tag{45}$$

The chain rule gives the physically predicted force

$$\hat{F}(r) = -A(r)\nabla_d \hat{E}(d(r)). \tag{46}$$

Translations and rotations leave distances invariant; at a nondegenerate three-dimensional geometry, $\mathrm{rank}(A) \leq 3N - 6$. For a redundant distance representation with $p > \mathrm{rank}(A)$, the force labels cannot uniquely determine a distance-gradient vector. If g_d satisfies $Ag_d = -F$, then so does $g_d + z$ for any $z \in \ker A$. A least-squares pseudoinverse selects a particular minimum-norm representative; its unidentifiable components are a choice of representation, rather than additional observed physical information.

Accordingly, the Cartesian residual

$$\|F + A\nabla_d \hat{E}\|_2^2 \tag{47}$$

is invariant to adding a null-space component to the distance gradient, whereas an unprojected Euclidean loss on a chosen distance-gradient label is not. Distance-gradient MSE and Cartesian force MAE therefore measure different quantities. This coordinate argument identifies which metric evaluates physical forces. An architecture comparison additionally depends on the data, units, selection procedures and training budgets; the coordinate map specifies the common physical quantity to compare.

K.8 PaiNN: comparisons with matched batch sizes

The molecular experiment uses PaiNN (Schütt et al., 2021) in SchNetPack (Schütt et al., 2018) on ten rMD17 molecules, with five provided splits per molecule. Each split has 950 training, 50 validation and 1000 evaluation configurations. The architecture has 128 atom-basis features, three interaction layers, twenty radial-basis features and a 5 Å cutoff. The scalar energy prediction yields Cartesian forces by $\hat{F} = -\nabla_r \hat{E}$. The optimizer is AdamW with zero weight decay and initial learning rate 10^{-3}. The cap is 1000 epochs, early-stopping patience is 200, and the plateau scheduler uses factor 0.5, patience 75 and cooldown 10. Force MAE is evaluated in meV/Å.

The native energy–force recipe weights energy and force MSE by 0.01 and 0.99, respectively. The equal-weight comparator uses $0.5/0.5$ weights on this backbone. The remaining comparators are GradNorm, value-pretrained warmup, and value-only training. These objectives act on the energy and force targets in their physical units, with the associated recipe's validation loss used for selection. Their numerical weights cannot be interpreted independently of those units or directly equated to the normalized synthetic objectives.

Matched comparisons. The five distinct recipes provide 250 fits. Within every molecule–split block all five recipes share one batch size: 128 in 42 blocks and 10 in 8 blocks (aspirin splits 1–3 and the five salicylic-acid splits). Every native-versus-competitor comparison therefore retains all 50 pairs. Table 47 reports force outcomes on these matched populations. Native energy–force training has lower force MAE in every pair against every competitor.

Batch sizes 10 and 128 imply approximately 95 and 8 updates per epoch with 950 training examples. The configured epoch cap alone therefore corresponds to different update budgets across blocks, and early stopping further changes realized budgets. The comparisons pair recipes within each block at the same batch size. They assess the complete objectives and validation procedures; the cross-block variation is not a batch-size ablation because the molecular split changes as well.

Table 47: PaiNN force-error comparisons on fifty molecule–split pairs with matched batch sizes. The geometric ratio is $\exp\{n^{-1}\sum_i \ln(\mathrm{MAE}_{i,\mathrm{competitor}}/\mathrm{MAE}_{i,\mathrm{native}})\}$; larger than one favors native energy–force training.

Competitor	Matched pairs	Native lower-error pairs	Geometric MAE ratio
Equal weight	50	50/50	1.759
GradNorm	50	50/50	2.619
Warmup	50	50/50	22.340
Value only	50	50/50	27.991

Energy and force outcomes together. Native energy–force training lowers energy MAE relative to value-only training on 48 of 50 batch-matched splits. Against the equal-weight objective it lowers energy error in 24 of 50 splits, while lowering force error in all 50. The physical force objective therefore does not yield a uniform ranking for scalar energy accuracy. Table 48 reports both quantities on each comparator's identical matched population.

Table 48: PaiNN energy and force MAEs, averaged over five matched splits per molecule. All five recipes use the same batch size within each molecule–split block; equal weight assigns coefficient 0.5 to each loss.

	Energy MAE (meV)					Force MAE (meV/Å)				
Molecule	Native	Equal weight	GradNorm	Warmup	Value only	Native	Equal weight	GradNorm	Warmup	Value only
aspirin	6.68	8.57	15.17	163.64	158.19	17.90	28.80	39.37	729.29	725.27
azobenzene	6.38	10.63	15.73	108.03	87.09	12.80	32.75	36.61	448.18	379.57
benzene	7.23	0.84	2.42	11.84	5.99	3.32	6.29	12.17	102.86	64.91
ethanol	4.03	3.04	6.66	35.94	34.69	14.93	17.86	32.15	237.34	231.25
malonaldehyde	10.34	4.30	8.86	49.34	62.16	21.18	24.19	39.13	336.03	434.03
naphthalene	7.64	8.70	12.73	29.77	31.74	8.49	27.54	37.16	149.23	167.12
paracetamol	13.95	11.22	19.48	120.92	159.08	20.81	36.54	52.25	554.20	716.29
salicylic	3.73	2.42	5.49	92.67	92.26	7.27	12.44	21.94	495.44	494.81
toluene	7.51	4.34	9.15	49.56	52.15	9.61	20.78	33.28	266.57	294.88
uracil	6.67	3.72	10.35	32.97	61.97	13.94	20.44	34.63	227.86	450.81

When force supervision begins. On rMD17 aspirin, we schedule the transition from energy-only training to GradNorm on energy and forces at fraction $\tau \in \{0.10, 0.25, 0.50\}$ of a 1000-epoch budget. The five provided splits and batch size ten are shared across all three schedules, with 950 training, 50 validation and 1000 test structures per split. Both phases use the validation sum of energy and force MSE for checkpoint selection. At the transition the model restores the best first-phase state, resets the optimizer moments and learning-rate scheduler, and uses learning rate 10^{-4} instead of the initial 10^{-3}. Early stopping can end a fit before its scheduled transition. Scheduling force supervision earlier lowers both mean errors within this procedure (Table 49). This comparison varies the allocation of training between two phases, including the time of the optimizer and learning-rate transition.

Table 49: Warmup-fraction sensitivity on rMD17 aspirin with PaiNN, batch size ten and five provided splits. Entries are arithmetic means and sample standard deviations. The model, data partitions and epoch cap are fixed.

Transition fraction τ	Energy MAE (meV)	Force MAE (meV/Å)
0.10	24.3 ± 9.4	74.1 ± 21.1
0.25	107.9 ± 71.3	468.0 ± 346.3
0.50	157.8 ± 9.3	713.2 ± 46.1

Table 50: MACE on three rMD17 molecules. Entries are mean ± sample standard deviation over five splits. The common energy-only and energy–force schedules match except for force weighting; the low-rate energy-only recipe changes the first-stage energy weight and learning rate.

Molecule	Training recipe	Energy MAE (meV)	Force MAE (meV/Å)
Aspirin	Energy only, common schedule	215.98 ± 6.41	947.88 ± 5.74
	Energy only, low-rate schedule	53.00 ± 4.98	207.81 ± 18.97
	Energy + force	4.90 ± 0.42	11.71 ± 0.38
Benzene	Energy only, common schedule	56.96 ± 17.22	422.34 ± 174.61
	Energy only, low-rate schedule	5.40 ± 0.47	50.27 ± 5.15
	Energy + force	0.21 ± 0.01	1.61 ± 0.07
Ethanol	Energy only, common schedule	49.99 ± 4.79	282.12 ± 20.65
	Energy only, low-rate schedule	13.72 ± 0.59	81.40 ± 4.00
	Energy + force	1.24 ± 0.10	7.35 ± 0.20

Table 51: Energy and force prediction under the named molecular training recipes.

	Energy MAE (meV) ↓			Force MAE (meV/Å) ↓			Lower error
Task / setting	B	D	Ratio [95% CI]	B	D	Ratio [95% CI]	pairs
PaiNN molecules: named baseline B versus native energy–force D							
Value-only	74.5	7.42	**0.1** [0.078, 0.13]	396	13	**0.036** [0.032, 0.04]	48/50
Equal weight	5.78	7.42	1.3 [0.9761, 1.6]	22.8	13	**0.57** [0.54, 0.61]	24/50
MACE molecules: low-rate energy-only B versus energy–force D							
Aspirin	53	4.9	**0.093** [0.084, 0.1]	208	11.7	**0.057** [0.052, 0.06]	5/5
Benzene	5.4	0.214	**0.04** [0.038, 0.042]	50.3	1.61	**0.032** [0.031, 0.034]	5/5
Ethanol	13.7	1.24	**0.091** [0.085, 0.096]	81.4	7.35	**0.09** [0.088, 0.093]	5/5

Errors are arithmetic means. Ratios are paired geometric means, computed separately for each metric; brackets give 95% paired bootstrap intervals. Green and rust ratios have intervals below and above one; uncolored intervals include one. The final column counts lower-error pairs for the left-hand metric. PaiNN contains ten molecules and five splits; bootstrap resampling keeps the ten molecules together within each split. MACE uses five seeds per molecule; its training recipes have different optimization schedules.

K.9 MACE ARCHITECTURE COMPARISON

MACE provides an equivariant architecture check on aspirin, benzene and ethanol, with five provided rMD17 splits per molecule. Each split contains 950 training, 50 validation and 1000 test structures. All recipes use the same molecular structures, energy reference and physical units. We report energy MAE in meV and Cartesian force MAE in meV/Å.

The common-schedule comparison uses the weighted MACE loss, initial energy weight 1, Adam/AMSGrad learning rate 0.01, batch size 5, and a cap of 150 epochs. Its second stage begins at epoch 100 with energy weight 1000 and learning rate 0.001. The energy–force recipe adds the force term; each recipe's validation objective controls its training and selection. The second energy-only recipe uses energy weight 1000 and learning rate 0.001 from the beginning, retaining the second-stage convention. This comparison changes the initial energy weight and learning rate as well as the use of forces.

Table 50 reports both energy-only recipes alongside energy–force training. The energy–force recipe has lower energy and force MAE in all fifteen paired splits against either comparator. The lower-rate, higher-energy-weight schedule substantially improves the energy-only fit without closing this gap. Together with the matched PaiNN panel, this supports testing derivative usefulness against a task-appropriate physical objective and a competent value-only comparator, while keeping architecture and optimization choices explicit.

L OPTION-SURFACE AND ATMOSPHERIC RECONSTRUCTION

Financial and field reconstructions test how the value of derivative supervision depends on the target and the information available to the predictor. SPY measures reconstruction of specified

option-pricing functions, while ERA5 measures reconstruction of contemporaneous atmospheric fields.

L.1 SPY TARGETS AND TEMPORAL EVALUATION

The options panel supplies market-observed input locations. Two supervisors define the learning target: Black–Scholes values and partial derivatives at quoted implied volatility, and values and partial derivatives at an SVI-fitted volatility. In both, volatility is treated as an independent pricer input when computing the other partial derivatives. Appendix B gives the panel membership and variable definitions. Table 52 compares two training-sample budgets under a temporal split.

SVI construction. The fixed panel contains 1,576,419 option rows across 758 dates. Each date–maturity slice is fitted separately using the raw-SVI total-variance form

$$w(k) = a + b\left[\rho(k-m) + \sqrt{(k-m)^2+\sigma^2}\right], \qquad k = \log(K/F).$$

The fitting procedure accepts 23,335 of 23,336 slices; one lacks the required five strikes. The median and 95th-percentile implied-volatility residuals are 0.016 and 0.099. Box-constrained fitting smooths the volatility observations without imposing the conditions for an arbitrage-free surface. The fitted volatility supplies the pricer's volatility coordinate. Derivative labels hold that coordinate fixed, rather than differentiating through smile calibration.

Table 52: SPY temporal evaluation: mean value and derivative MSE across ten seeds. Columns separate the supervisor and training count. Errors concern their specified pricer targets. Fixed denotes the dimension-aware fixed-weight recipe; half uses equal aggregate value and derivative weights.

Supervisor	Method	10,000 training examples		50,000 training examples	
		Value MSE	Derivative MSE	Value MSE	Derivative MSE
BS	Value-only	$1.7 \cdot 10^{-8}$	$1.78 \cdot 10^{-2}$	$9.9 \cdot 10^{-10}$	$1.76 \cdot 10^{-2}$
	Fixed	$6.8 \cdot 10^{-9}$	$2.4 \cdot 10^{-6}$	$5.9 \cdot 10^{-10}$	$2.2 \cdot 10^{-7}$
	Half	$7.6 \cdot 10^{-9}$	$2.5 \cdot 10^{-6}$	$5.5 \cdot 10^{-9}$	$6.4 \cdot 10^{-7}$
	GradNorm	$3.0 \cdot 10^{-8}$	$6.3 \cdot 10^{-6}$	$6.3 \cdot 10^{-9}$	$7.8 \cdot 10^{-7}$
	ReLoBRaLo	$8.6 \cdot 10^{-9}$	$2.6 \cdot 10^{-6}$	$2.2 \cdot 10^{-10}$	$2.2 \cdot 10^{-7}$
	Warmup	$2.8 \cdot 10^{-8}$	$5.0 \cdot 10^{-5}$	$1.8 \cdot 10^{-9}$	$3.3 \cdot 10^{-6}$
SVI	Value-only	$1.2 \cdot 10^{-8}$	$1.79 \cdot 10^{-2}$	$6.4 \cdot 10^{-10}$	$1.77 \cdot 10^{-2}$
	Fixed	$1.2 \cdot 10^{-8}$	$2.4 \cdot 10^{-6}$	$2.7 \cdot 10^{-10}$	$1.7 \cdot 10^{-7}$
	Half	$7.1 \cdot 10^{-9}$	$2.6 \cdot 10^{-6}$	$2.4 \cdot 10^{-10}$	$1.8 \cdot 10^{-7}$
	GradNorm	$1.7 \cdot 10^{-8}$	$3.3 \cdot 10^{-6}$	$1.5 \cdot 10^{-9}$	$5.2 \cdot 10^{-7}$
	ReLoBRaLo	$8.4 \cdot 10^{-9}$	$2.2 \cdot 10^{-6}$	$1.5 \cdot 10^{-10}$	$1.8 \cdot 10^{-7}$
	Warmup	$2.2 \cdot 10^{-8}$	$3.5 \cdot 10^{-5}$	$1.3 \cdot 10^{-9}$	$2.9 \cdot 10^{-6}$

These results show why value and derivative metrics should remain separate. Derivative training sharply reduces the supervised derivative error, while its value effect varies by recipe and budget. Under the SVI supervisor at 50,000 examples, for example, the fixed, half and ReLoBRaLo recipes improve value MSE, whereas GradNorm and warmup have larger value MSE than the value-only reference.

Purged temporal folds. A complementary five-fold walk-forward evaluation uses a five-trading-day embargo (Lopez de Prado, 2018). Table 53 reports each fold separately at 50,000 training examples, with ten seeds per method and fold.

The half-weight recipe is more stable across folds, particularly on the last Black–Scholes fold. Weighting therefore remains consequential within a fixed label construction. The polynomial comparison below tests a different representation on the same task.

Table 53: Relative change in fold-mean value MSE against value-only training, in percent. Negative favors DML. These folds share a financial time series and are not independent market histories.

Supervisor	Recipe	Fold 0	Fold 1	Fold 2	Fold 3	Fold 4
BS	Fixed	-69	-44	-48	-39	$+2879$
BS	Half	-70	-72	-56	-59	$+40$
SVI	Fixed	-70	-37	-53	-22	$+47$
SVI	Half	-78	-68	-54	-66	-65

L.2 Polynomial control on SPY-derived targets

The SPY control uses total-degree-five polynomials in four normalized inputs: 126 monomials, fitted by ridge regression with $\alpha = 10^{-8}$ and five seeds. The two supervisors and two temporal evaluation protocols are defined in Appendices B and L.1. Table 54 compares their value-prediction errors.

Table 54: SPY polynomial control. The last column is the lowest mean test error among the five DML recipes, not performance after independent model selection. Temporal cells use five polynomial and ten neural seeds. CV averages all five folds per seed, giving 25 polynomial and 50 neural evaluations per cell. These replication sets differ.

Supervisor	Split	Training count	Polynomial MSE	Lowest DML MSE
BS	Temporal	10,000	$1.31 \cdot 10^{-6}$	$6.83 \cdot 10^{-9}$
BS	Temporal	50,000	$1.31 \cdot 10^{-6}$	$2.16 \cdot 10^{-10}$
BS	Purged CV	10,000	$1.34 \cdot 10^{-6}$	$4.10 \cdot 10^{-9}$
BS	Purged CV	50,000	$1.30 \cdot 10^{-6}$	$8.34 \cdot 10^{-10}$
SVI	Temporal	10,000	$1.27 \cdot 10^{-6}$	$7.09 \cdot 10^{-9}$
SVI	Temporal	50,000	$1.26 \cdot 10^{-6}$	$1.52 \cdot 10^{-10}$
SVI	Purged CV	10,000	$1.34 \cdot 10^{-6}$	$4.37 \cdot 10^{-9}$
SVI	Purged CV	50,000	$1.31 \cdot 10^{-6}$	$2.45 \cdot 10^{-10}$

The fixed-degree polynomial has substantially larger value errors on these four-input targets than the lowest-error DML recipes. The difference from the one-input Heston comparison illustrates why a classical comparator must be assessed on the target and input representation actually used.

L.3 Atmospheric reconstruction and state information

The ERA5 experiment reconstructs Z500 geopotential using either location and seasonal coordinates alone or those coordinates augmented by 16 empirical orthogonal function coefficients of the contemporaneous state. The latter supply information about the field being reconstructed. We compare normalized field-value MSE in Table 55; the input and chronological split definitions are in Appendix B.

Table 55: ERA5 reconstruction value MSE, mean and sample standard deviation across five seeds. Same-day state features make this a reconstruction task. Both backbone columns use the same stated input configurations.

Inputs	Recipe	4×256 MLP	6×512 MLP
Bare	Value-only	0.174 ± 0.002	0.174 ± 0.002
Bare	Fixed	0.177 ± 0.004	0.177 ± 0.006
Bare	Half	0.174 ± 0.003	0.173 ± 0.001
State	Value-only	0.108 ± 0.005	0.108 ± 0.002
State	Fixed	0.128 ± 0.009	0.124 ± 0.009
State	Half	0.119 ± 0.009	0.116 ± 0.005

Transformer comparison. The same input comparison is repeated with a four-block, width-64 Transformer with four attention heads and GELU activations. Both derivative recipes increase value

Table 56: ERA5 Transformer reconstruction. Value MSE is in the normalized geopotential coordinates, with arithmetic mean and sample standard deviation over five paired seeds. (w_y, w_g) gives the aggregate coefficients on value and directional-derivative losses. Counts give pairs with lower value error than the paired value-only model. Each recipe uses its own weighted validation objective for scheduling, early stopping and checkpoint selection.

Inputs	Recipe	(w_y, w_g)	Value MSE ↓	Lower error
Coordinates	Value-only	$1, 0$	0.2829 ± 0.0580	–
	Equal weight	$1/2, 1/2$	0.2700 ± 0.0657	3/5
	Fixed	$1/3, 2/3$	0.2732 ± 0.0466	3/5
Coordinates + state	Value-only	$1, 0$	0.1781 ± 0.0187	–
	Equal weight	$1/2, 1/2$	0.2314 ± 0.0174	0/5
	Fixed	$1/3, 2/3$	0.2376 ± 0.0233	0/5

Table 57: ERA5 field reconstruction using a four-layer, width-256 MLP with coordinate-only or state-informed inputs.

Inputs	Normalized field MSE ($\cdot 10^{-1}$) Value-only	Fixed derivatives	Ratio [95% CI]	Lower error pairs
Coordinates	1.74	1.77	**1.018** [1.002, 1.034]	2/5
With state	1.08	1.28	**1.2** [1.1, 1.3]	0/5

Errors are arithmetic means; ratios are paired geometric means with 95% bootstrap intervals over five paired seeds. Rust marks intervals above one. The final column counts lower-error pairs.

error on every seed when contemporaneous state information is available (Table 56). With coordinates alone, their small mean improvements occur in three of five pairs. Thus the value cost of derivative supervision in the state-informed setting also appears with a different representation.

The Transformer has approximately $201{,}000$ parameters with coordinate inputs and $203{,}000$ with the 16 state coefficients. Training uses AdamW with learning rate 0.005, zero weight decay, batch size 256 and at most 200 epochs. The plateau scheduler halves the learning rate after 20 epochs without improvement; early-stopping patience is 50 epochs. Each epoch samples 100,000 training queries, with 20,000 queries per validation and test evaluation. This compares complete supervised-training procedures; the final value metric is common to all three.

State information lowers reconstruction value error for all three recipes in both MLP capacities and the Transformer. Within the state-informed task, value-only training has lower mean value error than either fixed derivative recipe, and the pattern persists across the tested capacities and backbones. Thus state inputs and derivative constraints have distinct effects: supplying the missing state improves reconstruction, while the spatial derivative objective does not improve this atmospheric target. The Burgers/FNO comparison demonstrates that derivative constraints can improve both sampled-point and full-field prediction in a different reconstruction task.