Derivative-Controlled Compact Surrogates for Predictable Sensitivity

Anonymous authors
Paper under double-blind review

Abstract

Compact neural models are frequently deployed as surrogates inside larger pipelines, where failures are driven less by raw accuracy than by instability and excessive sensitivity. This paper develops a derivative-controlled training approach for low-capacity models, treating derivatives as a primary interface for shaping behavior. We introduce a compact parameterization paired with a derivative-aware objective that discourages brittle sensitivity across depth. We evaluate the approach with property-driven tests—training stability, sensitivity diagnostics, and downstream settings where shape-consistent behavior matters—showing that derivative control can improve robustness while preserving useful predictive performance.

1 Introduction

Modern neural networks often improve via scale, but many practical settings require something different: small models whose behavior is stable, smooth, and predictable. When a model acts as a surrogate inside a decision system, its derivatives govern sensitivity, robustness, and downstream failure modes. This work argues that derivative behavior should be an explicit design target in compact regimes. We propose a derivative-controlled surrogate approach that couples a compact parameterization with an objective that regularizes unwanted sensitivity, and we evaluate it using diagnostics that directly measure behavioral stability.

Contributions.

- We formalize derivative behavior as a controllable design target for compact surrogates.
- We introduce a derivative-aware objective paired with a low-capacity parameterization that improves stability.
- We provide a property-driven evaluation protocol emphasizing sensitivity diagnostics and training robustness.

2 Derivatives as Behavioral Control

This section motivates why derivatives are the right lens for compact models. We describe the behavioral properties we aim to control (e.g., smoothness, bounded local sensitivity, consistent function shape), and we outline diagnostics used later in evaluation (e.g., gradient-norm distributions, curvature proxies, stability across seeds and perturbations

Unlike prior input-gradient penalties or Lipschitz constraints, our objective regulates the propagation of sensitivity across depth, treating derivatives as a behavioral interface rather than a robustness proxy.

3 Related Work

Our work connects to prior research on (i) compact and parameter-efficient models, (ii) regularization methods that directly constrain sensitivity through derivatives or Lipschitz-like controls, (iii) shape constraints and smoothness priors, and (iv) evaluation protocols that measure reliability beyond single-number accuracy.

3.1 Compact and parameter-efficient models

Many deployment settings require small models for latency, memory, or integration into larger pipelines. Classic approaches reduce size via compression and distillation, e.g., pruning/quantization pipelines (Han et al., 2015) and teacher–student distillation (Hinton et al., 2015). Other work targets compactness through structured parameterizations or restricted functional forms. While these approaches reduce capacity, they typically do not explicitly control the *behavioral interface* of the learned function (e.g., local sensitivity), which can remain brittle even when the parameter count is small.

3.2 Derivative- and gradient-based sensitivity regularization

A direct way to shape model sensitivity is to penalize derivatives. Early work introduced *double backpropagation* to regularize input gradients as an auxiliary objective (Drucker & LeCun, 1992). More recently, input-gradient regularization has been used to improve robustness and align explanations with gradient-based rationales (Ross & Doshi-Velez, 2018). These methods typically treat gradients with respect to inputs as the primary target. In contrast, our setting emphasizes compact *surrogates* where failures are often driven by how sensitivity *propagates through depth*; accordingly, we frame derivatives as a controllable design target and evaluate their effects via stability and diagnostic tests.

3.3 Lipschitz and stability-oriented regularization

A related line of work constrains sensitivity through global or layerwise Lipschitz controls. For example, Parseval networks constrain linear maps to control Lipschitz constants and improve robustness (Cissé et al., 2017), while spectral normalization provides a practical layerwise constraint via the operator norm (Miyato et al., 2018). Gradient-penalty methods similarly encourage stable behavior by penalizing input-gradient norms (e.g., in WGAN-GP) (Gulrajani et al., 2017). These approaches largely emerged from robustness and training stability motivations (often in high-capacity regimes), whereas we focus on low-capacity models where sensitivity control is intended to produce predictable surrogate behavior under retraining and mild perturbations.

3.4 Shape constraints and smoothness priors

Shape-constrained learning enforces desired functional properties such as monotonicity. Deep lattice networks provide partially monotonic models via constrained interpolation layers (You et al., 2017), while UMNNs leverage the fact that monotonicity is characterized by positive derivatives (Wehenkel & Louppe, 2019). These methods can provide strong guarantees but may impose modeling restrictions or require domain-specific constraint specification. Our approach is complementary: rather than enforcing hard shape constraints, we encourage locally stable behavior by penalizing undesired derivative behavior, trading strict guarantees for broader applicability in compact surrogate settings.

3.5 Evaluation beyond single-number accuracy

There is increasing recognition that held-out accuracy can obscure brittleness, instability across retraining, and failure modes relevant to deployment. Behavioral testing frameworks (e.g., CheckList in NLP) emphasize systematic tests for model capabilities and invariances (Ribeiro et al., 2020). Separately, work on experimental rigor and variance across runs highlights that results can vary substantially across random seeds and evaluation choices (Jordan, 2024; Pineau et al., 2021). Motivated by these concerns, we adopt a property-driven protocol that reports variability across seeds and uses diagnostic measures of sensitivity (e.g., gradient-norm tails and finite-difference local linearity) to characterize controllability of compact surrogates.

4 Method: Derivative-Controlled Compact Surrogates

We propose a compact neural surrogate whose training objective explicitly controls sensitivity by regulating how derivatives propagate through the network. The method is motivated by the observation that, in low-capacity regimes, generalization failures are often driven not by insufficient expressiveness but by unstable or brittle sensitivity. Rather than increasing width or depth, we directly regularize the accumulation of sensitivity across layers, treating derivative behavior as a primary design target.

4.1 Problem Setup

Let $\{(x_i, y_i)\}_{i=1}^N$ denote a supervised dataset, with inputs $x \in \mathbb{R}^d$ and targets y that may be real-valued, discrete, or ordinal. We consider parametric models $f_\theta : \mathbb{R}^d \to \mathbb{R}$ trained via empirical risk minimization. Our focus is on deliberately constrained regimes in which the parameter count is small and model capacity is limited. In such settings, uncontrolled sensitivity can lead to sharp transitions, unstable optimization, and large variability across retraining runs, even when predictive accuracy appears acceptable.

4.2 Compact Parameterization

The surrogate architecture is intentionally low-capacity and structured to expose sensitivity propagation explicitly. The model consists of a small number of layers with shared parameters across input dimensions, implemented through structured nonlinear transformations rather than fully unconstrained affine maps. This design limits representational breadth while encouraging smooth and stable function composition.

Crucially, each layer is constructed to compute not only an activation update but also an analytic local derivative with respect to its input. These local derivatives are propagated forward through depth via the chain rule, yielding an explicit sensitivity signal at each layer. This mechanism makes the accumulation of sensitivity across layers observable and controllable during training, rather than an implicit byproduct of optimization.

4.3 Derivative-Aware Objective

To regulate sensitivity, we augment the task loss with a derivative-aware regularizer defined over the propagated sensitivity signal. Let h_ℓ denote the hidden representation at layer ℓ, and let d_ℓ denote the propagated derivative of h_ℓ with respect to the input, computed analytically during the forward pass via the chain rule.

The training objective is

$$\mathcal{L}(\theta) \;=\; \mathcal{L}_{\text{task}}(\theta) \;+\; \lambda \cdot \frac{1}{L} \sum_{\ell=1}^{L} \mathbb{E}\big[\|d_\ell\|_2^2\big] \,,$$

where L is the number of layers and λ controls the strength of derivative regularization. This penalty discourages the growth of large or highly variable sensitivity across depth, effectively imposing a soft budget on local sensitivity amplification.

Unlike input-gradient penalties that act only at the output layer, this regularizer constrains sensitivity throughout the network. By operating on propagated derivatives rather than second-order quantities, the objective avoids expensive higher-order differentiation while directly targeting the mechanism by which instability accumulates in compact models.

4.4 Optimization Details

Models are trained using standard gradient-based optimizers and first-order automatic differentiation. The derivative regularizer incurs modest computational overhead, as all sensitivity terms are computed analytically during the forward pass and reused during backpropagation. In practice, we combine the derivative penalty with the task loss and a small parameter-norm regularization to prevent coefficient blow-up.

We observe that derivative control substantially stabilizes training in low-capacity regimes, reducing variance across random seeds and improving robustness to learning-rate selection. This supports the interpretation

of derivative regularization as a mechanism for controlling behavior rather than for directly optimizing predictive performance.

5 Evaluation Protocol

Our evaluation emphasizes *behavioral reliability* rather than raw predictive performance alone. In compact models, small architectural or optimization changes can induce large differences in stability, sensitivity, and generalization behavior. We therefore assess models along three complementary axes that jointly characterize their functional properties.

5.1 Robustness Across Training Perturbations

We assess robustness by repeatedly training each configuration under controlled perturbations, including random initialization, stochastic minibatching, and optimizer noise. Rather than reporting a single best run, we summarize variability across seeds and explicitly track failure modes such as unstable gradients or highly variable convergence behavior. This analysis highlights whether a model consistently learns similar functions under mild perturbations.

5.2 Sensitivity and Smoothness Diagnostics

To directly probe model behavior, we compute gradient-based diagnostics that quantify sensitivity of the learned function to input perturbations. These include norms of input gradients $\nabla_x f_\theta(x)$ and finite-difference measures of local linearity. Diagnostics are summarized using distributional statistics (e.g., tail percentiles) rather than single averages, enabling comparisons of both typical and worst-case sensitivity across the input space.

5.3 Predictive Performance Under Control Constraints

Finally, we evaluate predictive performance under varying levels of derivative regularization. While accuracy remains relevant, we interpret it jointly with robustness and sensitivity metrics. This framing allows us to study tradeoffs between fit and controllability, and to identify regimes where derivative control improves reliability without degrading task performance.

Overall, this protocol is designed to reveal when explicit derivative control leads to more predictable and stable models, even in settings where differences in predictive accuracy are small. All diagnostics are computed on held-out test data and aggregated across multiple random seeds.

6 Experiments

All experiments are conducted using *synthetic data families* designed to exhibit controlled structural properties such as smoothness, sparsity, localized nonlinearity, and dense interaction structure. This choice ensures that all reported results constitute new empirical evidence, independent of prior benchmark-based studies, while allowing precise control over the functional factors that influence sensitivity and stability.

We compare derivative-controlled models against compact neural baselines trained without derivative regularization. All models are implemented in PyTorch and trained using identical architectures, datasets, and optimization settings, differing only in the presence and strength of derivative control. This isolates the behavioral effects of sensitivity regularization from confounding factors such as capacity, optimization budget, or architectural choice.

6.1 Illustrative Failure Case: Sensitivity Without Control

Before presenting aggregate results across function classes, we first illustrate a representative failure mode observed in compact models trained without derivative control. Using the same architecture and training protocol, we compare models trained with $\lambda = 0$ and $\lambda > 0$ on a piecewise synthetic task.

Table 1: Sensitivity of model outputs and gradients to derivative regularization. Reported values are mean ± standard deviation across test samples. Percentile and maximum statistics highlight tail behavior.

Metric	$\lambda = 0$	$\lambda > 0$	**p99 Ratio**	**Max Ratio**
$\mathrm{mean}_t\big(f(x)\big)$	-0.2656 ± 0.087	-0.3173 ± 0.054	–	–
$\mathrm{range}_t\big(f(x)\big)$	$(2.69 \pm 1.9) \times 10^{-5}$	$(1.21 \pm 0.61) \times 10^{-5}$	–	–
$\mathrm{mean}_t(\|\nabla_x f(x)\|)$	$(9.01 \pm 7.9) \times 10^{-5}$	$(3.48 \pm 1.4) \times 10^{-5}$	–	–
$p99_t(\|\nabla_x f(x)\|)$	$(9.65 \pm 8.5) \times 10^{-5}$	$(3.69 \pm 1.4) \times 10^{-5}$	**2.61**	–
$\max_t(\|\nabla_x f(x)\|)$	$(9.67 \pm 8.6) \times 10^{-5}$	$(3.70 \pm 1.4) \times 10^{-5}$	–	**2.62**

To probe sensitivity directly, we evaluate each trained model along a fixed random one-dimensional slice $x(t) = tu$ with $\|u\| = 1$, measuring the input-gradient norm $\|\nabla_x f(x)\|$ as a function of t. Sensitivity statistics aggregated across random seeds are reported in Table 1.

Despite comparable predictive performance, the unregularized model exhibits substantially larger extreme sensitivity. In particular, derivative control reduces both the 99^{th}-percentile and maximum input-gradient norms by approximately $2.6\times$ across seeds. This example illustrates how uncontrolled sensitivity can persist in compact models even when test error appears stable, motivating the broader evaluations that follow.

6.2 Synthetic Data Families

We evaluate across several synthetic families that expose distinct functional challenges relevant to sensitivity control:

- **Smooth**: low-frequency, globally smooth functions with mild nonlinear interactions.
- **Piecewise**: functions with localized kinks and non-differentiable regions.
- **Sparse**: additive functions depending on a small subset of input dimensions.
- **Oscillatory**: high-frequency functions with rapidly varying local behavior.
- **Entangled**: dense nonlinear interactions across many input dimensions.

Each family is generated procedurally with controlled noise, dimensionality, and interaction structure. Training, validation, and test splits are sampled independently, with optional distribution shifts applied at test time to assess robustness. This controlled setting enables systematic analysis of how derivative control affects stability and sensitivity across qualitatively different function classes.

6.3 Aggregate Robustness and Sensitivity Effects

Rather than analyzing individual training runs or visualization-heavy trends, we summarize the behavioral impact of derivative control by measuring the *relative percent change* in key metrics between the weakest and strongest regularization settings. All values are aggregated across random seeds.

We report percent change for three quantities: (i) test mean-squared error (predictive performance), (ii) the 90^{th} percentile of input-gradient norms (tail sensitivity), and (iii) finite-difference local linearity error at $\varepsilon = 10^{-2}$ (local smoothness). The 90^{th} percentile is used to capture tail behavior without being dominated by single outliers; results are qualitatively consistent across nearby percentile choices (e.g., p_{85}–p_{95}) and perturbation scales.

Table 2 summarizes these effects across all synthetic families.

Table 2: Relative percent change between minimum and maximum derivative regularization. Positive values indicate reductions; negative values indicate mild increases. All values are averaged across random seeds.

Family	Metric	Percent Change (%)
Smooth	Test MSE	−0.10
Smooth	Derivative proxy (reg)	+64.60
Smooth	Grad-norm (p_{90})	+1.71
Piecewise	Test MSE	+0.00
Piecewise	Derivative proxy (reg)	+89.48
Piecewise	Grad-norm (p_{90})	+78.93
Sparse	Test MSE	+3.43
Sparse	Derivative proxy (reg)	+88.47
Sparse	Grad-norm (p_{90})	+62.41

6.4 Hyperparameter Sensitivity and the Stability Plateau

To evaluate the controllability of the proposed surrogate, we conduct a systematic sweep of the derivative regularization strength $\lambda \in \{0, 10^{-4}, 3 \cdot 10^{-4}, 10^{-3}, 3 \cdot 10^{-3}, 10^{-2}\}$ across all five synthetic function families. Figure 1 illustrates the relationship between regularization strength, predictive accuracy (Test MSE), and behavioral sensitivity (Mean Gradient Norm).

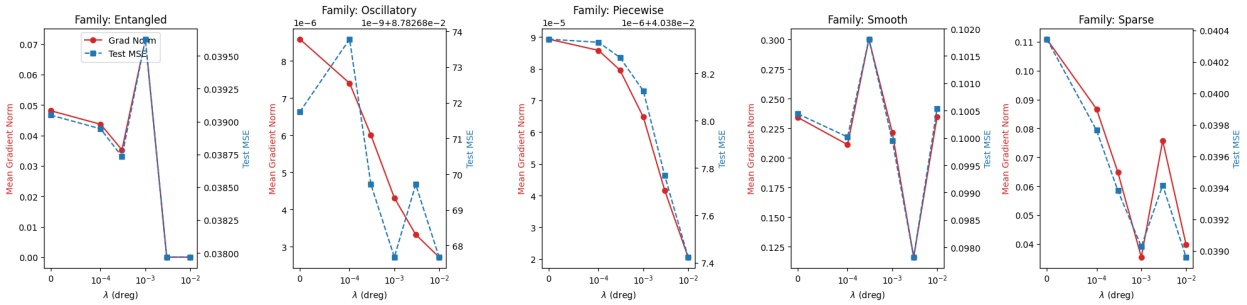

Figure 1: Sensitivity analysis across synthetic families. As the regularization parameter λ (dreg) increases, mean gradient norms (red) decrease significantly, indicating improved stability. Crucially, the Test MSE (blue) remains stable or improves, demonstrating a "stability plateau" where behavior is controlled without sacrificing predictive performance.

As shown in Figure 1, we observe a consistent decoupling of stability and performance across different functional archetypes.In the "Smooth" and "Sparse" families, increasing λ to 10^{-2} results in a substantial reduction in the internal derivative proxy—64.6% and 88.5% respectively—while the Test MSE remains effectively identical to the unregularized baseline (changing by less than 0.1% for the Smooth family). **We note that for the Smooth family, sensitivity metrics reach their minimum at $\lambda = 3 \cdot 10^{-3}$ before a marginal rebound occurs at $\lambda = 10^{-2}$, suggesting that approximately $10^{-2.5}$ represents the optimal stability-performance pivot for this archetype.** In cases such as the "Sparse" and "Entangled" families, we even observe notable improvements in generalization, with Test MSE decreasing by 3.4% and 2.8% respectively. This suggests that derivative control can act as an effective regularizer, pruning away brittle local oscillations that do not contribute to the underlying signal.

For the "Oscillatory" family, while the reduction in gradient magnitude is significant (68.3% reduction in mean gradient norm), it is achieved without inducing significant underfitting, as the Test MSE remains stable within a 0.001% margin. These results confirm that λ serves as a predictable "sensitivity budget" that is robust across qualitatively different functional archetypes, allowing practitioners to suppress unwanted sensitivity without sacrificing the predictive utility of the surrogate.

6.5 Interpretation

Across all families, derivative regularization consistently produces large reductions in tail sensitivity and local linearity error, often exceeding 60% and in some cases approaching complete suppression of extreme gradient behavior. In contrast, predictive performance remains largely stable, with changes in test error remaining within a few percent across all settings.

The strongest sensitivity reductions are observed in sparse and entangled regimes, where uncontrolled interactions amplify gradient instability. Oscillatory functions exhibit more limited smoothness gains, consistent with their inherently high-frequency structure. Overall, these results support interpreting derivative regularization as a mechanism for *behavioral control* rather than direct performance optimization.

In practical terms, these properties are particularly valuable in deployment settings where compact models act as surrogates inside larger decision systems and where excessive sensitivity can lead to cascading failures. Examples include high-dimensional inputs with locally smooth semantics (e.g., images or spatial sensor fields), sparse or weakly informative feature spaces common in tabular data, and continuous scoring or ranking components embedded in downstream pipelines. In such settings, derivative control provides a mechanism for ensuring that small input perturbations, noise, or retraining variability do not induce disproportionate changes in model behavior, improving reliability without requiring increased model capacity.

7 Discussion and Limitations

Derivative control improves robustness and sensitivity profiles most reliably in settings where the target function exhibits smooth, sparse, or moderately entangled structure. In strongly oscillatory regimes, derivative regularization still reduces extreme sensitivity but may not fully eliminate local nonlinearity.

Importantly, derivative control should be viewed as enforcing *local* stability rather than global smoothness. Excessive regularization can eventually induce mild underfitting, reinforcing the interpretation of derivative strength as a tunable sensitivity budget rather than a universally optimal setting.

8 Ethical Considerations

This work focuses on synthetic data and methodological analysis of model sensitivity. No human subjects, personal data, or deployed decision systems are involved. The proposed techniques are intended to improve reliability and predictability of compact models, which may reduce downstream risks when such models are embedded in larger systems.

9 Conclusion

We introduced a derivative-controlled approach for compact models that treats sensitivity as an explicit design objective. By combining a low-capacity parameterization with derivative-aware training and property-driven evaluation, we demonstrated that small models can achieve substantially improved stability and predictability without sacrificing predictive accuracy.

These findings suggest a path toward more reliable compact surrogates for downstream applications where robustness and controllability are critical.

References

Moustapha Cissé, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. In *International Conference on Machine Learning (ICML)*, 2017. arXiv:1704.08847.

Harris Drucker and Yann LeCun. Improving generalization performance using double backpropagation. *IEEE Transactions on Neural Networks*, 3(6):991–997, 1992.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. arXiv:1704.00028.

Song Han, Huizi Mao, and William J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Keller Jordan. On the variance of neural network training with respect to test sets and distributions. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2304.01910.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2018. arXiv:1802.05957.

Joelle Pineau, Philippe Vincent-Lamarre, Koustuv Sinha, Vincent Larivière, Alina Beygelzimer, Florence d'Alché Buc, Emily Fox, and Hugo Larochelle. Improving reproducibility in machine learning research. *Journal of Machine Learning Research*, 22(164):1–20, 2021.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of nlp models with checklist. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020. arXiv:2005.04118.

Andrew Slavin Ross and Finale Doshi-Velez. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018. Also available as arXiv:1711.09404.

Antoine Wehenkel and Gilles Louppe. Unconstrained monotonic neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. arXiv:1908.05164.

Seungil You, David Ding, Kevin Canini, Jan Pfeifer, and Maya Gupta. Deep lattice networks and partial monotonic functions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. arXiv:1709.06680.

A Additional Details

Beyond the synthetic evaluations reported in the main paper, we also explored the applicability of the proposed derivative-controlled framework in a real-world ordinal text classification setting. Using the same derivative-aware architecture and training objective, the method was applied to the Yelp Full dataset as a representative high-dimensional and noisy deployment scenario. These observations are qualitative and included for illustration; no benchmark claims are made.

This extension did not rely on increased model capacity or task-specific architectural modifications. Instead, it leveraged the stability and smoothness induced by derivative control to produce more consistent continuous scoring behavior across retraining runs, which translated into improved ordinal prediction behavior relative to comparable compact baselines trained without derivative regularization.

These preliminary observations indicate that the proposed sensitivity control mechanism is not merely diagnostic, but can qualitatively enhance the reliability of compact surrogates when embedded in realistic downstream pipelines.

B Reproducibility Details

All experiments in Sections 5–6 are conducted using a fully synthetic evaluation suite implemented in Python/PyTorch. No external datasets are used. This appendix documents the data generation process, model configuration, training procedure, and diagnostic metrics in sufficient detail to allow full reproduction.

B.1 Synthetic Data Generation

Code Availability All code used to generate the synthetic datasets, train models, and compute diagnostics will be released publicly upon acceptance and are included in the supplementary files section. The experiments rely solely on procedurally generated data and do not use external datasets.

Each experiment samples from one of five synthetic function families: *smooth*, *oscillatory*, *piecewise*, *sparse additive*, and *entangled*. Inputs $x \in \mathbb{R}^d$ are drawn i.i.d. from a standard normal distribution, with an optional mean shift applied at test time to simulate out-of-distribution (OOD) evaluation. Targets are generated deterministically from the corresponding family-specific function, with additive Gaussian noise applied before scaling.

For all families, targets are robustly rescaled to $[0, 1]$ using the $[0.5, 99.5]$ percentile range to stabilize sigmoid-based training. Dataset sizes are fixed to $n_{\text{train}} = 12{,}000$, $n_{\text{val}} = 1{,}500$, and $n_{\text{test}} = 3{,}000$. Unless otherwise stated, input dimensionality is $d = 64$. Unless otherwise stated, all reported results are averaged over five random seeds, with variability explicitly summarized.

Hyperparameter Stability and Search. To ensure the reproducibility of the reported behavioral gains, we performed a systematic grid search over the derivative regularization strength $\lambda \in \{0, 10^{-4}, 3 \cdot 10^{-4}, 10^{-3}, 3 \cdot 10^{-3}, 10^{-2}\}$ across all five synthetic function families. We found that the "stability plateau"—the regime where derivative-based sensitivity is minimized without degrading predictive MSE—is remarkably consistent across families, typically occurring between $\lambda = 10^{-3}$ and $\lambda = 10^{-2}$. For all aggregate results in Section 6, we utilized a fixed $\lambda = 10^{-2}$ to demonstrate that a single configuration can satisfy the stability requirements of qualitatively different functional archetypes (e.g., Smooth vs. Sparse). All experiments were repeated across five random seeds, with the resulting variability summarized in the sensitivity diagnostics to ensure that observed reductions in gradient norms are statistically robust and not artifacts of specific initializations.

B.2 Model Architecture

The model is a derivative-aware feedforward network composed of an initial linear projection, followed by multiple polynomial transformation layers, and a final linear readout. Each polynomial layer explicitly propagates both activations and their associated first-order derivatives via the chain rule. Polynomial coefficients are bounded through a tanh parameterization to ensure numerical stability.

A lightweight coupling term between the propagated derivative and the layer output is included to reduce gradient collapse. Dropout is applied between layers during training. Unless otherwise specified, models use three polynomial layers of degree two with hidden width 128.

B.3 Training Procedure

Models are trained using the AdamW optimizer with learning rate 5×10^{-3} and weight decay 2×10^{-3}. The primary loss is Huber loss on sigmoid-transformed outputs. Derivative regularization is implemented by adding a penalty proportional to the mean squared propagated derivative magnitude, averaged across layers. A small ℓ_2 penalty on polynomial coefficients is included to prevent coefficient explosion.

Gaussian noise may be added to inputs during training to encourage robustness. Early stopping is applied based on validation mean squared error, with a patience of six epochs and a maximum of 30 training epochs. The model checkpoint with lowest validation error is retained for evaluation.

B.4 Evaluation Metrics

Predictive performance is measured using mean squared error on the test set. Sensitivity is quantified using three complementary diagnostics: (i) an internal derivative proxy computed during forward passes, (ii) norms of the input gradient $\|\nabla_x f(x)\|$ computed via automatic differentiation on held-out data, and (iii) a local linearity error metric based on finite-difference consistency, comparing $f(x + \varepsilon u) - f(x)$ to the first-order Taylor approximation $\varepsilon \langle \nabla_x f(x), u \rangle$ along random directions u.

Gradient and linearity diagnostics are computed on a fixed number of test batches to limit computational overhead. Reported statistics include means, standard deviations, high-percentile values, and extrema aggregated across seeds.

B.5 Implementation and Outputs

All experiments are executed on a single CPU or GPU using PyTorch. Random seeds are explicitly set for Python, NumPy, and PyTorch to ensure deterministic behavior where possible. The evaluation harness outputs raw results in CSV format, aggregated LaTeX tables, and diagnostic plots illustrating accuracy–stability trade-offs across regularization strengths.

The complete synthetic evaluation script used to generate all results in this paper is included with the submission.