

Beyond Pricing Accuracy: A Multicriteria Diagnostic Benchmark of Neural Approaches to Financial PDE Approximation

Zakaria Elbayed^{1,*} and Abdelmjid Qadi El Idrissi¹

¹ Laboratory for Analysis and Modeling of Systems and Decision Support, National School of Applied Sciences of Berrechid (ENSAB), Hassan First University, Settat, Morocco.

* Corresponding author: Zakaria Elbayed (e-mail: z.elbayed.doc@uhp.ac.ma).

Received: 22 July 2026 Accepted: 23 September 2026 Published: 20 October 2026

DOI: 10.15849/ijasca.183 License: Open access under CC BY 4.0. Authors retain copyright.

Publisher: Al-Zaytoonah University of Jordan (ZUJ), Amman, Jordan.

Cite this article: Z. Elbayed and A. Qadi El Idrissi, "Beyond pricing accuracy: a multicriteria diagnostic benchmark of neural approaches to financial PDE approximation," *Int. J. Advance Soft Compu. Appl.*, vol. 18, no. 3, pp. 157–195, 2026, doi: <https://doi.org/10.15849/ijasca.183>.

ABSTRACT

Neural approximations of financial PDEs are often ranked by aggregate pricing error, although price accuracy alone does not establish differential, physical, or financial consistency. We present a multicriteria diagnostic benchmark of seven representative configurations in a common Black–Scholes setting. The primary comparison uses a frozen 20,000-point test set and five optimization seeds; targeted three-seed controls examine domain truncation, collocation density, DGM update allocation and loss scaling, checkpoint selection, Hybrid order and activation, and Monte Carlo path allocation. Evaluation covers price and Greek errors, PDE residuals and tails, financial-violation frequencies and magnitudes, regime performance, runtime, and seed variability. The MLP achieves the lowest primary price MAE, 0.0675 ± 0.0057 , whereas the Standard PINN at $S_{\max} = 400$ achieves the lowest primary PDE MAE, 0.2648 ± 0.0432 . The data-first Hybrid is competitive on both and obtains the lowest Greek errors and NEG, LBV, and monotonicity-violation frequencies. Moving $S_{\max}$ from 200 to 400 reduces PINN price MAE from 2.596 to 0.212, whereas extension to 800 raises it to 0.402; restoring collocation density does not recover $S_{\max} = 400$ performance. Increasing DGM allocation from one to 20 updates per epoch reduces its PDE MAE from 0.4539 to 0.1498, below the matched-seed PINN value of 0.2662, but raises training time from 230.6 to 3022.4 seconds. Hybrid and checkpoint controls reveal criterion-specific trade-offs, while path-count diagnostics quantify stochastic-target noise. No configuration is uniformly dominant. Reliable neural-PDE benchmarking requires joint reporting of accuracy, physical and financial consistency, protocol choices, and cost.

Keywords: Financial PDEs; option pricing; Black–Scholes equation; physics-informed neural networks; Deep Galerkin Method; financial admissibility; multicriteria benchmarking; protocol sensitivity.

1 Introduction

Financial partial differential equations (PDEs) remain central to derivative pricing, hedging, and risk management. Since the Black–Scholes framework [1] and Merton's option-pricing formulation [2], PDE-based models have provided a rigorous connection between derivative values and the dynamics of the underlying financial variables. More elaborate formulations, including the Heston stochastic-volatility model [3], accommodate richer market dynamics but also increase the computational burden of repeated pricing, sensitivity calculation, calibration, and stress testing over large parameter domains.

Neural approximation offers an alternative to repeatedly solving these equations with conventional numerical methods. Once trained, a neural model can provide a parametric approximation of the pricing function over a prescribed domain of market and contract variables. Relevant approaches include supervised neural pricers, physics-informed neural networks (PINNs) [4], residual-based methods such as the Deep Galerkin Method (DGM) [5], branch–trunk constructions inspired by DeepONet [6], attention-based architectures, and stochastic approaches derived from Feynman–Kac or backward stochastic differential equation representations [7]. These approaches differ not only in architecture but also in their learning signal. Depending on the configuration, optimization may use analytical or simulated prices, PDE residuals, terminal and boundary conditions, stochastic payoff estimates, or joint supervised and physical objectives.

Comparative studies commonly emphasize aggregate pricing errors such as MAE, RMSE, or R^2 . These measures are necessary, but they do not fully characterize a learned pricing surface. Models with similar price errors may produce substantially different Greeks, satisfy the governing PDE to different degrees, or violate monotonicity, convexity, and no-arbitrage-related price bounds in localized parts of the domain. Moreover, violation frequency alone does not indicate economic severity: two models can fail at comparable numbers of test points while producing violations of markedly different magnitudes. These distinctions matter because neural pricing functions are commonly differentiated for hedging and risk management and may be evaluated under financially sensitive maturity, volatility, and moneyness regimes.

The interpretation of a neural-PDE benchmark can also depend materially on its numerical protocol. Residual-based methods enforce the PDE over a computational domain that need not coincide with the predictive evaluation domain. In option pricing, the asset-price coordinate is generally truncated at an artificial upper boundary S_{max} , where an asymptotic far-field condition is imposed. Moving this boundary changes the problem presented to the network without changing the evaluation distribution. When the number of collocation points is held fixed, domain enlargement also reduces their density. Boundary placement and collocation allocation can therefore influence the apparent performance of a residual-trained configuration independently of its architecture.

Optimization introduces a related source of ambiguity. An “epoch” does not represent a common amount of computational work across neural PDE methods. One configuration may perform a single parameter update per epoch, whereas another may perform several updates using newly sampled interior, terminal, and boundary points. Loss scaling can additionally change the relative numerical importance of the residual and constraint terms. Consequently, observed differences between PINN and DGM may reflect optimizer-update allocation and loss formulation as well as network architecture.

Checkpoint selection constitutes another degree of freedom. A residual-trained model may be optimized without analytical price labels while its retained checkpoint is selected using an analytical validation set. Such a model is physics-trained but not fully label-independent at the model-selection stage. A physics-aligned checkpoint based on a fixed residual-validation sample may select a different epoch and produce a different balance between price and PDE accuracy. Analogous considerations apply to stochastic-target models selected using either analytical prices or fixed Monte Carlo validation targets.

Other design choices may also be confounded if they are varied simultaneously. For a Hybrid PINN, changing from data-first to physics-first training can also change the activation inherited from the source model. Likewise, the performance of a Feynman–Kac pricer depends on the variance of its finite-path Monte Carlo targets. Separating such effects requires targeted controls rather than attributing every observed difference directly to the nominal model family.

Against this background, this study presents a multicriteria diagnostic benchmark of seven representative neural configurations for financial PDE approximation: a supervised MLP, a Standard PINN, a Hybrid PINN, a DeepONet-style branch–trunk surrogate, DGM, a Transformer-inspired neural pricer, and a Feynman–Kac stochastic-target pricer. Black–Scholes provides the common analytical setting because prices, Greeks, and the differential operator can all be evaluated against explicit references. The seven primary configurations are trained over five optimization seeds, while targeted three-seed controls examine specific protocol factors. Alternative checkpoints extracted from the same training trajectories are treated as paired sensitivity evaluations rather than independently trained models.

The purpose is not to construct an intrinsic architecture leaderboard. The evaluated configurations retain heterogeneous learning signals, loss formulations, inductive biases, parameterizations, and optimization procedures. Instead, the study examines how conclusions change when the same learned pricing surfaces are assessed using pricing accuracy, Greek accuracy, PDE consistency, residual-tail behavior, financial admissibility, regime-wise performance, computational cost, and multi-seed variability. The experimental protocol is therefore treated as part of the object being evaluated.

The study addresses the following research questions:

RQ1: How do representative neural configurations for financial PDE approximation compare under a common multicriteria protocol covering price and Greek accuracy, PDE consistency, financial admissibility, regime-wise performance, computational cost, and multi-seed stability?

RQ2: To what extent do model rankings and conclusions change across evaluation criteria, and can favorable aggregate errors coexist with localized residual tails or financial violations of non-negligible magnitude?

RQ3: How sensitive are residual-based configurations to the upper computational boundary S_{max} , and to what extent can the observed domain effect be explained by changes in interior collocation density?

RQ4: How do optimizer-update allocation, residual-loss scaling, and checkpoint-selection criterion affect the comparison of residual-trained configurations?

RQ5: How do Hybrid training order and activation, as well as Monte Carlo path allocation in stochastic-target learning, influence the balance among pricing accuracy, PDE consistency, financial admissibility, and computational cost?

The principal contributions are as follows:

We establish a common five-seed benchmark for seven representative neural configurations spanning supervised, physics-informed, hybrid, residual-based, branch–trunk, attention-based, and stochastic-target learning paradigms.

We implement a shared evaluation interface that reports analytical price and Greek errors, PDE-residual distributions and tail diagnostics, financial-violation frequencies and conditional magnitudes, regime-wise performance, computational cost, and between-seed variability.

We separate the supervised evaluation domain from the computational PDE domain and examine PINN and DGM at different values of S_{max} . Additional density-controlled PINN experiments distinguish domain enlargement from collocation dilution. The results identify $S_{max} = 400$ as the best-performing tested truncation under the realized protocol, without interpreting it as a theoretically optimal boundary.

We cross DGM optimizer-update allocation with residual-loss scaling. Increasing the allocation from one to 20 updates per epoch substantially improves price and residual accuracy and, under matched control seeds, reverses the primary PINN–DGM residual ordering, although at a markedly higher computational cost.

We compare analytical-price and signal-aligned checkpoints for PINN, DGM, and Feynman–Kac trajectories. This paired analysis quantifies checkpoint sensitivity without counting alternative checkpoints as additional training runs.

We complete a 2×2 Hybrid control crossing training order with activation. This separates the effect of physics-first versus data-first training from the SiLU versus Tanh contrast and reveals an accuracy–admissibility trade-off.

We quantify finite-path Monte Carlo target noise and examine path allocations of 16, 64, and 256 antithetic simulations. This provides a direct diagnostic of the stochastic supervision used by the Feynman–Kac configuration.

Taken together, these experiments show why differences between neural financial PDE approximations should not be attributed to architecture without accounting for the learning signal, checkpoint criterion, numerical-domain specification, collocation allocation, optimization effort, and evaluation objective. A small pricing error alone does not establish differential, physical, or financial consistency.

The remainder of the paper is organized as follows. Section II reviews neural approaches to financial PDE approximation and positions the study relative to existing comparative work. Section III presents the benchmark design, architectures, computational domains, training and checkpoint protocols, and multicriteria evaluation procedure. Section IV reports and discusses the primary benchmark and protocol-sensitivity results. Section V summarizes the findings, states the limitations, and outlines directions for future research.

2 Related Work

Deep learning for derivative pricing has evolved from the early use of neural networks as supervised approximators of pricing functions toward a broader range of learning strategies. Current approaches include data-driven surrogate models, physics-informed neural networks (PINNs), residual-based methods such as the Deep Galerkin Method (DGM), neural operators, attention-based architectures, and probabilistic formulations based on Feynman–Kac representations or backward stochastic differential equations (BSDEs). A recent systematic review by Sharma and Verma illustrates this growing diversity of nonparametric and machine-learning methods for option pricing and also emphasizes the role of implementation and evaluation choices when such methods are compared [8].

2.1 Supervised and Data-Driven Neural Pricing

Supervised neural pricing treats derivative valuation as the approximation of a mapping from market and contractual variables to option prices. Early work by Hutchinson et al. showed that neural networks can act as flexible nonparametric approximators for derivative pricing and hedging [9]. The same principle remains relevant in more recent settings, although the emphasis has shifted toward larger parameter spaces, more demanding pricing models, and repeated evaluations after an offline training stage.

Glau and Wunderlich proposed the deep parametric PDE method, where a neural network approximates a parametric family of PDE solutions and subsequently provides rapid evaluations across state and model parameters [10]. This offline–online viewpoint is particularly attractive in applications in which pricing must be repeated over many market scenarios or parameter combinations. Rather than solving an individual numerical problem for each query, part of the computational effort is shifted to a reusable approximation learned in advance.

Pricing accuracy alone, however, does not determine the usefulness of such a surrogate. Hedging and risk management depend directly on derivatives of the pricing function. Ratku and Neumann showed that feed-forward neural networks trained for derivative pricing can also be differentiated to obtain first- and second-order sensitivities, with particular attention to the smoothness needed for higher-order derivatives [11]. This observation provides a direct motivation for assessing Greeks independently of pointwise price errors: two pricing networks with similar predictive accuracy need not produce sensitivities of comparable quality.

2.2 Physics-Informed and Residual-Based PDE Solvers

A different line of research incorporates the governing differential equation directly into the learning objective. PINNs [4] and DGM [5] are representative examples. Rather than relying exclusively on labeled option prices, these methods minimize PDE residuals together with terminal- and boundary-condition errors at points sampled from a computational domain.

The broader physics-informed learning literature has shown that this formulation is attractive but numerically non-trivial. Cuomo et al. review developments in PINNs and identify loss construction, optimization, boundary-condition treatment, network design, and convergence among the persistent challenges of physics-informed learning [12]. These issues are particularly relevant for financial PDEs because the computational domain and its artificial boundaries form part of the numerical problem presented to the network.

Several studies have adapted residual-based learning to financial models. Wang et al. investigated a PINN-based numerical PDE approach for option pricing through Black–Scholes-type equations [13]. Gatta et al. considered American option pricing using a meshless physics-informed formulation [14]. Hainaut and Casas subsequently developed a physics-inspired neural approach for European option pricing under the Heston model, with the objective of learning a parametric pricing function that can be reused across option and model specifications [15]. Pande et al. extended physics-informed residual learning to European option pricing under regime-switching dynamics [16], while more recent work has considered barrier-option pricing with PINNs over varying financial parameters [17].

Taken together, these contributions show that residual-based neural pricing extends well beyond the standard Black–Scholes setting. They also underline why observed performance cannot be attributed to network architecture alone. Collocation sampling, terminal and boundary constraints, loss scaling, optimization effort, and the definition of the computational domain jointly determine the numerical problem that is actually optimized.

This observation is directly relevant to the present benchmark. For residual-based configurations, the computational PDE domain is separated from the common domain used for evaluation. The influence of the artificial upper asset boundary S_{max} is then examined without changing the test distribution. In addition, optimization effort is treated explicitly because the same number of epochs can correspond to markedly different numbers of parameter updates across residual-based implementations.

2.3 Neural Operators and Attention-Based Pricing

Neural operators provide another perspective by targeting mappings between functions rather than approximating a single solution at isolated coordinates. DeepONet [6] is a prominent example and has motivated the use of branch–trunk constructions for parametric differential equations and surrogate modeling.

A recent study by Lee et al. is particularly relevant to the comparative perspective adopted here [18]. The authors compare DeepONet, PINN, and a deep BSDE configuration for European bond call options under the Hull–White and G2++ interest-rate models. Their analysis extends beyond out-of-sample price errors to include Vega accuracy obtained through automatic differentiation and performance under an out-of-distribution volatility stress. DeepONet performs favorably across the criteria considered in their experiments. More broadly, the study shows that recent financial-PDE comparisons are beginning to examine heterogeneous learning paradigms through criteria that go beyond price accuracy.

The branch–trunk configuration used in the present work is deliberately described as a DeepONet-style surrogate. Its branch receives a finite vector of pricing parameters rather than a discretized input function. Accordingly, it is used as a structured parametric pricing surrogate and is not presented as evidence about generic function-to-function neural operator learning.

Attention-based architectures have likewise entered the option-pricing literature. Høverstad et al. compared a Transformer-based model with an LSTM–MLP architecture and found that their relative performance depended on the market conditions considered [19]. The Transformer performed better during the extreme COVID-19 period in their study, whereas the LSTM–MLP was stronger overall. Their results provide another example of how a global error measure can conceal differences that become visible in particular market regimes.

For the same reason, the attention-based configuration examined in the present benchmark is described conservatively as Transformer-inspired. It is a supervised attention-based pricing model evaluated within the controlled Black–Scholes setting, rather than a claim about Transformer architectures as a general class of financial PDE solvers.

2.4 Probabilistic, Feynman–Kac, and BSDE-Based Approaches

A further family of methods exploits the relationship between parabolic PDEs and stochastic representations. Deep BSDE methods, following Han et al. [7], use simulated stochastic trajectories together with neural approximations of quantities appearing in the associated backward stochastic system. Such formulations provide an alternative to direct PDE residual minimization and are particularly attractive when conventional grid-based numerical methods become difficult to apply.

Recent studies continue to explore the connection between stochastic representations and neural pricing. Moreno et al. combined neural networks with a nonlinear Feynman–Kac formulation for a financial option-pricing problem [20]. Lee et al. also include a deep BSDE configuration alongside PINN and DeepONet in their cross-family comparison [18].

Stochastic approaches introduce an additional difficulty when compared with training against analytical labels. Their effective learning signal may depend on trajectory simulation, time discretization, Monte Carlo variance, variance-reduction techniques, and the simulation budget itself. Assabumrungrat et al. studied error sources in deep PDE solvers for option pricing and distinguished errors associated with problem specification, simulation, and neural-network training [21]. This distinction is important because prediction error can otherwise combine errors originating from the learned approximation with errors already present in the training signal.

The stochastic configuration considered here is therefore referred to as a Feynman–Kac stochastic-target pricer. The terminology is deliberately implementation-specific. The model is trained by regression on Monte Carlo estimates of discounted terminal payoffs and does not implement the complete temporal structure of a generic deep BSDE solver. The noise associated with these stochastic targets is consequently quantified as a separate diagnostic rather than being absorbed into the final prediction error.

2.5 Financial Consistency and the Remaining Evaluation Gap

Recent work increasingly recognizes that neural option-pricing models should not be assessed through pricing errors alone. The sensitivity analysis of Ratku and Neumann illustrates the importance of the differential structure of learned pricing functions [11]. From another perspective, Cohen et al. developed arbitrage-free Neural-SDE market models in which financial restrictions are incorporated directly into model construction [22]. These contributions demonstrate that financial consistency and differential accuracy constitute distinct dimensions of model quality.

Evaluation protocols have also become broader. Assabumrungrat et al. examine multiple sources of numerical error in deep PDE solvers [21]; Pande et al. investigate pricing under regime-switching dynamics [16]; and Lee et al. jointly consider pricing accuracy, Vega accuracy, and an out-of-distribution volatility stress across DeepONet, PINN, and a deep BSDE configuration [18]. The recent literature therefore cannot be characterized as relying exclusively on aggregate price-error measures such as MAE or RMSE.

The remaining gap is more specific. Relevant diagnostic dimensions are distributed across different studies and methodological families. Some works emphasize sensitivities, whereas others examine PDE consistency, stochastic-solver error, financial restrictions, regime dependence, or computational behavior. Cross-paradigm benchmarks that integrate several of these dimensions within a fixed evaluation protocol remain comparatively limited. Lee et al., whose study is among the closest to the present perspective, compare three learning paradigms through pricing accuracy, Vega accuracy, and robustness to a volatility distribution shift [18].

The present study addresses this narrower gap by evaluating seven implemented neural configurations spanning supervised, physics-informed, hybrid, residual-based, branch–trunk, attention-based, and stochastic-target learning. All configurations are assessed on the same frozen Black–Scholes test set. The evaluation covers price errors; Delta, Gamma, and Vega errors; PDE-residual distributions and tail behavior; financial-admissibility violations; regime-wise performance; computational effort; and variability across optimization seeds. The contribution therefore lies in integrating these diagnostic dimensions within a common protocol, rather than in introducing a new neural solver.

The analytical Black–Scholes solution makes it possible to evaluate prices, Greeks, PDE residuals, boundary behavior, and financial constraints against known references. The benchmark is not intended to suggest that neural approximators are preferable to the closed-form formula for this particular problem. Instead, Black–Scholes provides a controlled setting in which the diagnostic protocol can be developed and examined before its application to financial PDEs without analytical solutions.

The study also investigates the extent to which an observed comparison depends on selected numerical and optimization choices. For residual-based configurations, the evaluation domain is distinguished from the computational domain on which the PDE is enforced. Fixed-count and density-controlled experiments examine the effects of the artificial upper boundary S_{max} and collocation density. The

DGM controls vary optimizer-update allocation and loss scaling, while the Hybrid controls distinguish training-order sensitivity from activation effects. Checkpoint-selection sensitivity and Feynman–Kac path-count sensitivity are examined separately. These controls help determine whether differences among the tested configurations partly reflect protocol choices rather than architecture alone.

Accordingly, the conclusions concern the implemented configurations, the specified Black–Scholes domain, and the documented training and evaluation protocol. No universal ranking of the corresponding architecture families is claimed.

3 Methodology

This section presents the experimental protocol used to compare representative neural approaches to financial PDE approximation. The study is designed as a multicriteria diagnostic benchmark rather than an intrinsic ranking of neural architectures, since the evaluated configurations differ in both architecture and training information: analytical prices, PDE constraints, hybrid supervision, or stochastic payoff estimates. The supervised data domain is distinguished from the computational PDE domain used for residual enforcement. All retained models are evaluated through the same implementation on a fixed test set. The seven primary configurations are repeated over five optimization seeds, while targeted protocol-sensitivity controls use three prespecified seeds.

3.1 Black–Scholes Problem Formulation

We consider a European call option with value $V(S, K, \tau, r, \sigma)$, where S is the underlying asset price, K is the strike price, τ is the remaining time to maturity, r is the continuously compounded risk-free rate, and σ is the volatility. Using time to maturity as the temporal variable, the Black–Scholes equation is

$$-\frac{\partial V}{\partial \tau} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS \frac{\partial V}{\partial S} - rV = 0, \quad (1)$$

with terminal condition

$$V(S, K, 0, r, \sigma) = \max(S - K, 0). \quad (2)$$

For a non-dividend-paying European call, the static no-arbitrage price bounds are

$$\max(S - Ke^{-r\tau}, 0) \leq V(S, K, \tau, r, \sigma) \leq S. \quad (3)$$

The lower inequality combines non-negativity with discounted intrinsic value, whereas the upper inequality states that the call value cannot exceed the underlying asset price.

The Black–Scholes model provides an analytical price and closed-form Delta, Gamma, and Vega. It also permits direct evaluation of the PDE residual at arbitrary test points. Pricing accuracy, differential accuracy, PDE consistency, and financial admissibility can therefore be assessed against a common analytical reference.

3.2 Data Generation and Analytical References

The supervised observations are generated by independent uniform sampling over

$$S \in [40, 200], \quad K \in [60, 160], \quad \tau \in [0.05, 3], \quad r \in [0, 0.10], \quad \sigma \in [0.05, 0.80]. \quad (4)$$

Three independent datasets are generated: 100,000 training observations using dataset seed 42, 20,000 validation observations using seed 43, and 20,000 test observations using seed 44. The datasets are persisted before model training and reused unchanged across all configurations and optimization seeds. Dataset generation randomness is therefore separated from model optimization randomness.

Dataset identity is recorded through SHA-256 checksums. The abbreviated identifiers are 236f16212d98 for training, f34fecbf1c6d for validation, and e5603bbb63ba for testing; complete hashes are retained in the machine-readable experimental manifest and verified whenever the persisted datasets are reloaded. No model-specific resampling, repartitioning, or regeneration of test observations is performed.

The three splits were checked for missing and non-finite values and for analytical violations of the call-price and Greek constraints considered in this study. No inconsistencies were found. Their coverage of the six regimes introduced in Section III-G was also comparable: the largest difference between split-level regime proportions was 0.634 percentage points. These checks establish consistent coverage of the prescribed benchmark domain; they do not imply that uniform sampling represents an empirical market population.

For every observation, the analytical Black–Scholes price, Delta Δ , Gamma Γ , Vega ν , and the bounds in (3) are retained. The analytical references are checked for non-negative prices, lower- and upper-bound compliance, $\Delta \in [0,1]$, $\Gamma \geq 0$, and $\nu \geq 0$. At the reference point $(S, K, \tau, r, \sigma) = (100, 100, 1, 0.05, 0.20)$, centered finite differences reproduce the closed-form Delta, Gamma, and Vega with relative discrepancies below 1.5×10^{-8} .

The automatic-differentiation pipeline used for neural-model evaluation was validated separately. The analytical Black–Scholes function was implemented in double precision and differentiated with respect to its raw inputs at 2,048 validation points. The resulting derivatives were compared with the closed-form Greeks, and the Black–Scholes differential operator was evaluated using the same autograd derivatives. Table 1 shows discrepancies consistent with floating-point roundoff, confirming that the evaluation pipeline does not introduce material numerical error before being applied to the learned pricing functions.

Table 1. Double-precision validation of the analytical reference and automatic-differentiation pipeline over 2,048 points.

Quantity	Median	95th percentile	Maximum
$ \Delta_{\text{aD}} - \Delta_{\text{BS}} $	2.78×10^{-17}	2.36×10^{-16}	1.11×10^{-15}
$ \Gamma_{\text{aD}} - \Gamma_{\text{BS}} $	8.67×10^{-19}	1.13×10^{-17}	1.44×10^{-16}
$ \nu_{\text{aD}} - \nu_{\text{BS}} $	7.11×10^{-15}	3.20×10^{-14}	1.53×10^{-13}
$ \mathcal{L}_{\text{BS}}[\text{VBS}] $	9.99×10^{-16}	6.22×10^{-15}	8.06×10^{-14}

Normalization. Input and output standardization statistics are estimated exclusively from the training set. For (S, K, τ, r, σ) , the resulting means are approximately

$$(120.10, 109.93, 1.527, 0.0499, 0.4234), \quad (5)$$

with standard deviations

$$(46.16, 28.88, 0.851, 0.0289, 0.2166). \quad (6)$$

The price mean and standard deviation are 41.4546 and 36.5126, respectively. The fitted scaling objects are persisted, and round-trip transformations are verified before training. Validation and test observations do not contribute to their estimation.

Supervised configurations use standardized inputs and outputs. Residual-based configurations accept variables in their original units and return prices in financial units, with scaling performed internally where required. The common evaluation routine expresses every model as

$$x_{\text{raw}} \mapsto \hat{V}(x_{\text{raw}}) \quad (7)$$

in original financial units. For a standardized-output model,

$$\frac{\partial \hat{V}}{\partial s} = \sigma_y \frac{\partial}{\partial s} \left[f_{\theta} \left(\frac{x - \mu_x}{\sigma_x} \right) + \mu_y \right], \quad (8)$$

so the computational graph applies the chain rule through input and output scaling. The same raw-unit interface is used for finite-difference, autograd, and PDE-residual calculations. Consequently, prices, Greeks, and residuals are reported in original financial units for every configuration.

3.3 Computational PDE Domain and Boundary Specification

The supervised data domain and the computational domain used for PDE enforcement are treated as distinct objects. Residual-based configurations sample

$$\Omega(S_{max}) = [0, S_{max}] \times [60, 160] \times [0, 3] \times [0, 0.10] \times [0.05, 0.80]. \quad (9)$$

The ranges of K , τ , r , and σ are common to the supervised and residual-based experiments. Only the asset-price coordinate is extended from $S = 0$ to the artificial boundary S_{max} . Analytical validation and test observations remain restricted to $S \in [40, 200]$ in (4); changing S_{max} therefore does not alter the observations used for model comparison.

At the lower boundary,

$$V(0, K, \tau, r, \sigma) = 0. \quad (10)$$

At the artificial upper boundary, the far-field approximation is

$$V(S_{max}, K, \tau, r, \sigma) \approx \max(S_{max} - Ke^{-r\tau}, 0). \quad (11)$$

Unlike the condition at $S = 0$, (11) is asymptotic. Its accuracy depends on the location of S_{max} , which is therefore treated as an experimental factor rather than an implementation detail.

For the demanding parameter corner $(K, \tau, r, \sigma) = (160, 3, 0, 0.8)$, departure from the limiting Delta value is characterized using $1 - N(d_1)$, where

$$d_1(S_{max}) = \frac{\log(S_{max}/K) + (r + 1/2\sigma^2)\tau}{\sigma\sqrt{\tau}}. \quad (12)$$

For $S_{max} = 200, 400$, and 800 , the resulting values of $1 - N(d_1)$ are approximately 0.197, 0.088, and 0.032, respectively. This quantity is used only to characterize the candidate boundaries; it is neither a training objective nor a checkpoint-selection criterion.

Collocation-count control. Changing S_{max} while retaining a fixed number of collocation points moves the artificial boundary and changes the interior sampling density. The revised protocol therefore distinguishes fixed-count and density-controlled experiments. The fixed-count protocol uses $N_{PDE} = 4,096$ newly sampled interior points per optimizer update for every value of S_{max} . It preserves the allocation used in the submitted $S_{max} = 200$ and $S_{max} = 400$ configurations and extends it to $S_{max} = 800$.

The density-controlled protocol increases the number of interior points in proportion to the asset-price range. Using $S_{max} = 200$ and $N_{PDE} = 4,096$ as the reference gives 8,192 points at $S_{max} = 400$ and 16,384 points at $S_{max} = 800$. Terminal and boundary allocations remain fixed at 2,048 points per condition. Table 2 summarizes both protocols.

Table 2. Computational-domain and interior-collocation protocols. The far-field column reports $1 - N(d_1)$ at $(K, \tau, r, \sigma) = (160, 3, 0, 0.8)$.

Smax	Fixed count	Density controlled	1-N(d ₁)
200	4,096	4,096	0.197
400	4,096	8,192	0.088
800	4,096	16,384	0.032

Interior, terminal, and boundary samples are drawn from seed-controlled uniform distributions and renewed during optimization. For each trajectory, the physics-validation sample is generated deterministically from the model family, configuration variant, and optimization seed. It is independent of the renewed training samples and remains fixed across validation epochs.

The fixed-count sequence measures the practical effect of extending the domain under an unchanged collocation allocation. Comparisons between fixed-count and density-controlled configurations at the same S_{max} assess sensitivity to interior-collocation dilution. These experiments do not establish a theoretically optimal truncation point.

3.4 Evaluated Neural Configurations

The primary benchmark contains seven implemented neural configurations: a supervised multilayer perceptron (MLP), a standard physics-informed neural network (PINN), a Hybrid PINN, a DeepONet-style branch–trunk surrogate, a Deep Galerkin Method (DGM) configuration, a Transformer-inspired neural pricer, and a Feynman–Kac stochastic-target pricer.

The primary instances retain the configurations examined in the submitted comparison: the SiLU MLP; the Tanh PINN with $S_{max} = 400$ and a fixed interior-collocation count; the data-first SiLU Hybrid PINN; the DeepONet-style surrogate; the Tanh DGM with $S_{max} = 400$, one optimizer update per epoch, and losses expressed in financial units; the Transformer-inspired pricer; and the Feynman–Kac model trained with 64 antithetic paths per target. Each primary configuration is evaluated over five optimization seeds.

The terminology is implementation-specific. The branch–trunk model receives a finite parameter vector rather than a discretized input function and is therefore described as a DeepONet-style parametric surrogate, not as a generic function-to-function neural operator. Likewise, the stochastic-target model regresses simulated Feynman–Kac payoff estimates and does not implement the complete temporal architecture of a deep BSDE solver.

Targeted revision controls. Sixteen additional configurations are used as targeted controls and are not included as separate methods in the primary seven-configuration comparison. They vary a documented domain, sampling, optimization, activation, training order, or stochastic-target factor while retaining the relevant architecture and other settings whenever possible. Unless otherwise stated, these controls use the three seeds {42,123,2025}. They assess protocol sensitivity rather than constitute an exhaustive hyperparameter search.

The optimization settings reported in Table 5 are inherited from the submitted comparison and were not re-tuned for the present revision. No hyperparameter selection was performed against the frozen test set; the only settings explored are the documented control factors listed in Table 4.

The Hybrid controls form a 2×2 design crossing training order with activation. Seed-matched MLP-Tanh and PINN-SiLU source configurations permit training-order comparisons at fixed activation. The DGM controls distinguish computational-domain effects from optimizer-update allocation and loss scaling. The Feynman–Kac controls vary the number of simulated paths per target, while the PINN density controls address collocation dilution as the asset-price domain is enlarged. Detailed control definitions are provided in Section III-H.

3.5 Learning Objectives and Training Protocol

Analytically supervised configurations. The MLP, DeepONet-style surrogate, and Transformer-inspired pricer are trained against analytical Black–Scholes prices. Their common objective is the mean squared error between standardized predictions and analytical targets:

$$\mathcal{L}_{\text{data}} = \frac{1}{N_b} \sum_{i=1}^{N_b} [f_{\theta}(\tilde{x}_i) - \tilde{V}_i]^2, \quad (13)$$

where $\tilde{x}_i$ and $\tilde{V}_i$ are the standardized input vector and analytical price, respectively, and N_b is the mini-batch size. The MLP serves as the conventional data-driven baseline.

For the DeepONet-style surrogate, the branch network receives (K, r, σ) and the trunk network receives (S, τ) . Their 128-dimensional representations are combined through an inner product and an additive trainable bias.

The Transformer-inspired pricer treats the five standardized inputs as scalar tokens. Each token is embedded in 64 dimensions and processed by three pre-normalized self-attention encoder layers with four heads. Mean pooling and an MLP head produce the final prediction. The mathematical scaled-dot-product attention backend is used so that the second derivatives required for Gamma and PDE-residual evaluation remain available through automatic differentiation.

Residual-trained configurations. The PINN objective combines the Black–Scholes residual with terminal and boundary conditions:

$$\mathcal{L}_{\text{PINN}} = \lambda_{\text{PDE}} \mathcal{L}_{\text{PDE}} + \lambda_T \mathcal{L}_T + \lambda_B \mathcal{L}_B, \quad (14)$$

where

$$\mathcal{L}_{\text{PDE}} = \frac{1}{N_\Omega} \sum_{i=1}^{N_\Omega} \left(\frac{R_\theta(x_i)}{\sigma_y} \right)^2, \quad \mathcal{L}_T = \frac{1}{N_T} \sum_{i=1}^{N_T} \left(\frac{\hat{V}_\theta(x_i) - V_T(x_i)}{\sigma_y} \right)^2, \quad \mathcal{L}_B = \frac{1}{N_B} \sum_{i=1}^{N_B} \left(\frac{\hat{V}_\theta(x_i) - V_B(x_i)}{\sigma_y} \right)^2. \quad (15)$$

Here, R_θ is the Black–Scholes operator applied to the prediction, V_T is the terminal payoff, V_B is the applicable boundary target, and σ_y is the training-set price standard deviation. The weights are

$$(\lambda_{\text{PDE}}, \lambda_T, \lambda_B) = (1, 10, 10). \quad (16)$$

The primary and fixed-count PINNs use 4,096 newly sampled interior points, 2,048 terminal points, and 2,048 boundary points per optimizer update. Density-controlled variants use 8,192 interior points at $S_{\max} = 400$ and 16,384 at $S_{\max} = 800$, while retaining the same terminal and boundary sample sizes.

Table 3. Architectures and primary instances of the seven neural configurations used in the main Black–Scholes benchmark.

Configuration	Architecture	Activation	Input/output convention	Parameters
Supervised MLP	5 hidden layers, 128 units/layer	SiLU	standardized → standardized	66,945
Standard PINN, $S_{\max}=400$	5 hidden layers, 128 units/layer	Tanh	raw → financial units	66,945
Hybrid PINN, data-first	5 hidden layers, 128 units/layer	SiLU	raw → financial units	66,945
DeepONet-style surrogate	branch (K,r,σ) and trunk (S,τ); 4 layers each; latent dim. 128	SiLU	standardized → standardized	132,993
DGM, $S_{\max}=400$	3 gated DGM layers, width 128	Tanh	raw → financial units	208,257
Transformer-inspired	d_model=64, 4 heads, 3 encoder layers, d_ff=128	GELU / SiLU	standardized → standardized	125,953
Feynman–Kac, M=64	5 hidden layers, 128 units/layer	SiLU	standardized → standardized	66,945

The primary Hybrid PINN is initialized from the seed-matched supervised SiLU MLP and subsequently fine-tuned with the hybrid objective.

Table 4. Targeted control configurations introduced for the revised analysis.

Control block	Configurations	Factor varied	Reference
MLP activation	MLP-Tanh	SiLU versus Tanh	Primary MLP-SiLU
PINN truncation	$S_{\max} \in \{200, 800\}$, fixed count	Upper PDE boundary	Primary PINN, $S_{\max}=400$
PINN density	$S_{\max}=400$: 8,192; $S_{\max}=800$: 16,384	Interior-collocation density	Fixed-count PINNs
PINN activation	PINN-SiLU, $S_{\max}=400$	Tanh versus SiLU	Primary PINN-Tanh
DGM domain	$S_{\max}=200$, one update, real scale	Upper PDE boundary	Primary DGM
DGM budget and scale	1 or 20 updates; real or standardized loss	Updates and loss scaling	Primary DGM
Feynman–Kac paths	$M \in \{16, 256\}$	Monte Carlo path count	Primary M=64
Hybrid factorial	data-first/physics-first; SiLU/Tanh	Training order and activation	Primary data-first/SiLU

DGM uses the same physical conditions and weights, but its submitted configuration expresses the loss components directly in financial units:

$$\mathcal{L}_{\text{DGM}} = \lambda_{\text{PDE}} \frac{1}{N_{\Omega}} \sum_{i=1}^{N_{\Omega}} R_{\theta}(x_i)^2 + \lambda_T \frac{1}{N_T} \sum_{i=1}^{N_T} [\hat{V}_{\theta}(x_i) - V_T(x_i)]^2 + \lambda_B \frac{1}{N_B} \sum_{i=1}^{N_B} [\hat{V}_{\theta}(x_i) - V_B(x_i)]^2. \quad (17)$$

The primary DGM performs one optimizer update per reported epoch. Its direct comparison with PINN therefore combines architecture, loss scaling, and optimization allocation. The revision controls examine update rates of one and 20 steps per epoch, financial-unit and standardized losses, and their combination. These controls do not impose equal total compute because the maximum reported epoch count is retained.

Hybrid physics-informed training. The Hybrid configuration combines analytical supervision with the PDE, terminal, and boundary objectives:

$$\mathcal{L}_H = \lambda_{\text{data}} \mathcal{L}_{\text{data}} + \lambda_{\text{PDE}} \mathcal{L}_{\text{PDE}} + \lambda_T \mathcal{L}_T + \lambda_B \mathcal{L}_B, \quad (18)$$

with

$$(\lambda_{\text{data}}, \lambda_{\text{PDE}}, \lambda_T, \lambda_B) = (1, 0.05, 0.02, 0.02). \quad (19)$$

For each supervised mini-batch, 256 interior points, 128 terminal points, and 128 boundary points are newly sampled on the common $S_{\text{max}} = 400$ domain.

The primary data-first Hybrid is initialized from the seed-matched SiLU MLP. The submitted physics-first direction is initialized from the seed-matched Tanh PINN. Because these paths differ in both pretraining order and activation, the revision adds the two missing factorial cells: data-first/Tanh and physics-first/SiLU. For each cell, parameters are transferred exactly from the corresponding seed-matched source model, and numerical equality is verified before fine-tuning.

Feynman–Kac stochastic targets. The stochastic-target configuration uses the Feynman–Kac representation rather than analytical prices during optimization. For $x = (S, K, \tau, r, \sigma)$,

$$S_T = \text{Sexp} \left[\left(r - \frac{1}{2} \sigma^2 \right) \tau + \sigma \sqrt{\tau} Z \right], \quad Z \sim \mathcal{N}(0, 1), \quad (20)$$

and the regression target is

$$\hat{Y}_{\text{MC}} = \frac{1}{M} \sum_{m=1}^M e^{-r\tau} \max(S_T^{(m)} - K, 0). \quad (21)$$

Antithetic Gaussian draws reduce estimator variance. The primary configuration uses $M = 64$ paths per input, and the sensitivity experiment uses $M = 16$ and $M = 256$. Targets are regenerated during optimization and standardized with the training-set price statistics. Analytical Black–Scholes prices are not used in the training gradients; they are used for primary checkpoint selection and common test evaluation. A separate fixed Monte Carlo validation target is used for the signal-aligned checkpoint.

All neural modules use their default PyTorch initialization; no bespoke initialization scheme is applied. Random-number generators are reset before model construction for every seed-level trajectory.

3.6 Optimization and Checkpoint Selection

Table 5. Optimization settings of the primary benchmark configurations. Patience is expressed in validation checks when validation is not performed after every epoch.

Configuration	Optimizer	LR	Weight decay	Max. epochs	Validation / patience	Updates/epoch	Grad. clip
MLP	Adam	10^{-3}	10^{-6}	300	every epoch / 30	98	–
PINN	Adam	10^{-3}	0	3000	every 50 / 20 checks	20	10
Hybrid PINN	Adam	5×10^{-4}	10^{-7}	80	every 5 / 12 checks	98	10
DeepONet-style	Adam	10^{-3}	10^{-6}	300	every epoch / 30	98	–
DGM	Adam	10^{-3}	0	3000	every 50 / 25 checks	1‡	10

Configuration	Optimizer	LR	Weight decay	Max. epochs	Validation / patience	Updates/epoch	Grad. clip
Transformer-inspired	AdamW	10^{-3}	10^{-6}	300	every epoch / 30	98	10
Feynman–Kac target	Adam	10^{-3}	10^{-6}	250	every epoch / 30	98	10

‡ The primary DGM performs one optimizer update per reported epoch. The budget controls use 20 updates per epoch while retaining the remaining DGM settings and maximum epoch count; they therefore assess increased update allocation rather than equal total computational cost.

Table 5 reports the optimization settings of the primary configurations. Supervised mini-batches contain 1,024 observations, producing 98 optimizer updates per complete pass through the 100,000-point training set. Unless varied by a specified control, the submitted optimization settings are retained.

PINN and DGM use cosine learning-rate annealing from the initial value to 10^{-2} of that value over the maximum training horizon. Gradient norms are clipped at 10 for PINN, DGM, Hybrid, Transformer-inspired, and Feynman–Kac configurations; no clipping is applied to the MLP or DeepONet-style surrogate.

Primary and signal-aligned checkpoints. The primary checkpoint for each configuration minimizes analytical price error on the frozen validation set. For standardized-output models, the criterion is standardized price MSE; other predictions are converted through the common financial-unit interface in Section III-B.

Validation observations do not enter optimization mini-batches or contribute to the gradients of residual-trained or stochastic-target models. However, analytical checkpoint selection means that PINN and DGM are physics-trained but not fully label-independent at the model-selection stage. Similarly, the Feynman–Kac model is trained using simulated payoffs while its primary checkpoint is selected using analytical prices.

For every primary PINN and DGM trajectory, a second checkpoint minimizes the combined PDE, terminal, and boundary loss on a fixed physics-validation sample. This sample is generated deterministically from the model family, configuration variant, and optimization seed, independently of renewed training points, and remains fixed throughout the trajectory. The physics-validation sample comprises 4,096 interior, 2,048 terminal, and 2,048 boundary points for both PINN and DGM. These sizes are held constant across all computational-domain and collocation-density variants, so that the physics-validation criterion remains comparable even when the training-time interior allocation changes. Its seed is derived by SHA-256 from the model family, configuration variant, optimization seed, and a fixed label, which makes the sample reproducible across sessions. The primary Feynman–Kac trajectory analogously retains a Monte Carlo-aligned checkpoint selected on a fixed target generated using 2,048 paths per validation input.

The retained checkpoint criteria are

$$\begin{aligned} \theta_{\text{price}}^* &= \underset{\theta_t}{\operatorname{argmin}} \mathcal{L}_{\text{val}}^{\text{analytical}}(\theta_t), & \theta_{\text{physics}}^* &= \underset{\theta_t}{\operatorname{argmin}} \mathcal{L}_{\text{val}}^{\text{physics}}(\theta_t) \text{ (PINN and DGM)}, & \theta_{\text{MC}}^* &= \\ & \underset{\theta_t}{\operatorname{argmin}} \mathcal{L}_{\text{val}}^{\text{MC}}(\theta_t) \text{ (Feynman–Kac)}. & & & (22) \end{aligned}$$

Analytical-price checkpoints constitute the primary seven-configuration benchmark. Signal-aligned checkpoints are evaluated separately on the same test set. Because both checkpoints originate from the same optimization trajectory, their comparison is paired by configuration and seed and does not represent independently trained models.

Early stopping. PINN and DGM training stops only after both the analytical-price and fixed physics-validation criteria remain unimproved for their respective patience windows. Feynman–Kac applies the analogous rule to analytical-price and fixed Monte Carlo criteria. The retained checkpoints may consequently correspond to different epochs. Because training continues until both criteria have ceased

to improve, the dual-criterion rule can extend a trajectory beyond the epoch at which a single-criterion rule would have stopped. Differences with respect to the submitted single-criterion configuration therefore reflect both the selection criterion and, where the stopping epoch differs, the additional optimization performed.

The MLP, DeepONet-style surrogate, and Transformer-inspired pricer use conventional early stopping based on analytical validation-price MSE. The Hybrid configuration uses the same criterion because analytical supervision is included in its training objective.

Repeated optimization and aggregation. The seven primary configurations are independently trained using

$$\mathcal{S}_{\text{main}} = \{7,42,123,999,2025\}, \quad (23)$$

whereas targeted controls use

$$\mathcal{S}_{\text{control}} = \{42,123,2025\}. \quad (24)$$

Optimization seeds control parameter initialization, mini-batch ordering, and stochastic sampling; the training, validation, and test datasets remain unchanged. Each Hybrid trajectory uses a source checkpoint with the same seed, and exact equality of transferred parameter tensors is verified before fine-tuning.

Primary results are reported as means and standard deviations over five independent trajectories. Three-seed controls are interpreted as targeted sensitivity analyses. Alternative checkpoints from the same trajectory are treated as paired checkpoint-selection results rather than independent repetitions.

3.7 Multicriteria Evaluation Protocol

Every retained checkpoint is evaluated using the same implementation and the same frozen 20,000-point test set. Model outputs are converted to original financial units through the common raw-input/raw-output interface introduced in Section III-B. No model-specific post-processing is used for the reported pricing, Greek, PDE-residual, financial-admissibility, or regime-wise metrics.

a) Pricing accuracy. Let V_i and $\hat{V}_i$ denote the analytical and predicted prices for test observation i . Pricing accuracy is assessed using MAE,

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |\hat{V}_i - V_i|, \quad (25)$$

RMSE,

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (\hat{V}_i - V_i)^2}, \quad (26)$$

the coefficient of determination R^2 , signed bias,

$$\text{Bias} = \frac{1}{N} \sum_{i=1}^N (\hat{V}_i - V_i), \quad (27)$$

filtered MAPE, and symmetric MAPE. MAPE is computed only where $V_i \geq 10^{-2}$ because percentage errors become ill-conditioned as the analytical price approaches zero. The number of retained observations is reported with the metric. Because R^2 is close to one for several configurations, $1 - R^2$ is also examined when resolving small differences among high-accuracy models. Conclusions are not based on R^2 alone.

b) Greek accuracy. Delta, Gamma, and Vega are computed by automatic differentiation of the composed raw-unit pricing function:

$$\hat{\Delta} = \frac{\partial \hat{V}}{\partial S}, \quad \hat{\Gamma} = \frac{\partial^2 \hat{V}}{\partial S^2}, \quad \hat{V} = \frac{\partial \hat{V}}{\partial \sigma}. \quad (28)$$

For each Greek, the absolute-error distribution is summarized by its mean, RMSE, median, 95th percentile, and maximum, together with R^2 . Relative Gamma errors are reported only for observations satisfying

$$|F_i| \geq 10^{-4}, \quad (29)$$

because division by values close to zero can produce large relative errors without a correspondingly large error in financial units. Delta, Gamma, and Vega errors are not combined into an unscaled score because the three quantities have different units and numerical ranges. The automatic-differentiation validation reported in Section III-B separates approximation errors from errors in the derivative-evaluation implementation.

c) PDE consistency. For every test observation, the Black–Scholes residual is

$$R_\theta = -\hat{V}_t + \frac{1}{2}\sigma^2 S^2 \hat{V}_{SS} + rS\hat{V}_S - r\hat{V}. \quad (30)$$

All derivatives are evaluated with respect to inputs in their original financial units. The absolute residual distribution is characterized by its mean, RMSE, median, 95th percentile, and maximum. The additional tail ratio

$$\text{TR}_R = \frac{\max|R_\theta|}{\text{median}|R_\theta|} \quad (31)$$

is reported when the denominator is numerically non-zero. It is used only as a descriptive indicator of concentrated residual tails and is interpreted jointly with the other distributional statistics. Pricing accuracy and PDE consistency remain separate evaluation dimensions because neither guarantees the other.

d) Financial admissibility. For a European call, the principal price constraints are

$$0 \leq V(S, K, \tau, r, \sigma) \leq S \quad (32)$$

and

$$V(S, K, \tau, r, \sigma) \geq L(S, K, \tau, r) = \max(S - Ke^{-r\tau}, 0). \quad (33)$$

The corresponding negative-price, lower-bound, and upper-bound violation rates are

$$\text{NEG} = \Pr(\hat{V} < -\eta), \quad \text{LBV} = \Pr(\hat{V} < L - \eta), \quad \eta = 10^{-6}. \quad \text{UBV} = \Pr(\hat{V} > S + \eta), \quad (34)$$

Monotonicity in the asset price and convexity are examined through

$$\text{MVR}_\varepsilon = \Pr(\hat{\Delta} < -\varepsilon), \quad \text{CVR}_\varepsilon = \Pr(\hat{f} < -\varepsilon), \quad (35)$$

with $\varepsilon = 10^{-3}$. Raw zero-threshold rates are retained as complementary diagnostics. The larger shape tolerance reduces the influence of negligible automatic-differentiation fluctuations around zero.

Violation frequencies are supplemented by their magnitudes:

$$d_i^{\text{NEG}} = \max(-\hat{V}_i, 0), \quad d_i^{\text{LB}} = \max(L_i - \hat{V}_i, 0), \quad d_i^{\text{UB}} = \max(\hat{V}_i - S_i, 0), \quad d_i^{\text{MON}} = \max(-\hat{\Delta}_i, 0), \quad d_i^{\text{CONV}} = \max(-\hat{f}_i, 0). \quad (36)$$

For each constraint, the conditional mean, median, 95th percentile, and maximum magnitude are computed over the violating observations. If $\mathcal{J}_c$ denotes the observations violating constraint c after applying its tolerance, the conditional mean severity is

$$\bar{d}_c^+ = \frac{1}{|\mathcal{J}_c|} \sum_{i \in \mathcal{J}_c} d_i^c. \quad (37)$$

Frequency and severity are reported together because similar violation rates may conceal substantially different magnitudes. LBV is also interpreted with signed bias and absolute pricing error, since systematic overpricing can lower the one-sided violation rate without improving overall accuracy.

e) Regime-wise performance. Global results are complemented by six overlapping subsets of the frozen test set. These subsets are descriptive slices of the original in-distribution domain and are therefore described as regime-wise performance rather than out-of-distribution robustness. No model is retrained or reselected within a regime.

For each regime, pricing errors, signed bias, and principal financial violations are calculated using the global definitions. Greek and PDE-residual summaries are retained where they help diagnose regime-specific deterioration. Differences in regime sample size are considered when interpreting stability.

Table 6. Evaluation tolerances, filtering thresholds, and realized regime sizes in the frozen test set.

Quantity	Definition/value	Test points
Shape tolerance	$\varepsilon = 10^{-3}$	–
Price-bound tolerance	$\eta = 10^{-6}$	–
MAPE price threshold	$V \geq 10^{-2}$	reported after filtering
Gamma relative-error floor		Γ
Near-the-money		S-K
In-the-money	$S/K > 1.05$	10,554
Out-of-the-money	$S/K < 0.95$	8,104
Short maturity	$\tau < 0.25$	1,368
Long maturity	$\tau > 1.50$	10,321
High volatility	$\sigma > 0.50$	7,996

f) Variability across optimization seeds. Every seed-level checkpoint is evaluated independently. Primary scalar metrics are reported as means and standard deviations over the five seeds in (23). Targeted controls are summarized over the three seeds in (24) and interpreted as sensitivity analyses. Differences comparable to the observed between-seed variability are not treated as stable configuration advantages. Alternative checkpoints are paired by configuration and seed with their analytical-price counterparts.

g) Computational cost. Training wall-clock time, price-only inference time, parameter count, retained checkpoint epoch, and total optimizer-update count are recorded. Update counts are required because an epoch does not represent the same optimization effort across model families.

Price-only inference is timed on the complete test set in batches of 4,096. CUDA operations are synchronized before and after measurement. Three complete passes are performed, and the minimum elapsed time is retained to reduce one-off runtime overhead. Greek and PDE-residual computations are excluded because they require first- and second-order automatic differentiation.

For Hybrid models, fine-tuning time excludes source-model training. End-to-end cost therefore adds the time and updates of the seed-matched source model. Alternative checkpoints incur no additional training cost because they are retained during the same trajectory.

3.8 Protocol-Sensitivity Experiments

The primary benchmark is supplemented by targeted controls for computational domain, collocation density, DGM update allocation and loss scaling, Hybrid training order and activation, checkpoint selection, and Feynman–Kac path count. These controls assess protocol dependence and are not counted as additional methods in the primary comparison.

Computational domain and collocation density. The primary PINN and DGM use $S_{max} = 400$. Both are additionally evaluated at $S_{max} = 200$, and PINN is further examined at $S_{max} = 800$. Within each family, the architecture, loss weights, optimization seeds, evaluation domain, and frozen supervised datasets remain unchanged.

The fixed-count protocol uses 4,096 interior points per update at every S_{max} . It therefore varies both boundary placement and sampling density. Density-controlled PINNs use

$$N_{\Omega}(200) = 4,096, \quad N_{\Omega}(400) = 8,192, \quad N_{\Omega}(800) = 16,384. \quad (38)$$

Terminal and boundary allocations remain fixed at 2,048 points per condition. Comparing fixed-count and density-controlled variants at the same S_{max} helps distinguish domain extension from collocation dilution. The tested values are sensitivity settings, not theoretically optimal boundaries.

DGM update allocation and loss scaling. The primary PINN uses 20 optimizer updates per epoch and standardized physical losses. The submitted DGM uses one update and losses in financial units. Consequently, their direct comparison combines architecture, scaling, and optimization-allocation differences.

At $S_{max} = 400$, DGM is examined under

$$\text{updates per epoch} \in \{1, 20\}, \quad \text{loss scaling} \in \{\text{financial units, standardized}\}. \quad (39)$$

All other DGM settings are retained. A 20-update DGM at $S_{max} = 200$ is also included to reassess domain sensitivity under increased optimization effort. Because architectures, computational graphs, per-update costs, and maximum epoch counts differ, these controls do not constitute an equal-total-cost PINN–DGM comparison.

Hybrid training order and activation. The Hybrid controls cross two factors:

$$\text{training order} \in \{\text{data-first, physics-first}\}, \quad \text{activation} \in \{\text{SiLU, Tanh}\}. \quad (40)$$

Data-first models use seed-matched supervised checkpoints, whereas physics-first models use seed-matched $S_{max} = 400$ PINN checkpoints. All four cells use the same objective in (18), the same computational domain, loss weights, physics-sampling sizes, optimizer, validation schedule, and fine-tuning protocol. At fixed activation, the comparison assesses training-order sensitivity; at fixed training order, it assesses activation sensitivity within the implemented Hybrid protocol.

Checkpoint selection. For the primary PINN and DGM, analytical-price and physics-aligned checkpoints from the same trajectory are compared. The primary Feynman–Kac comparison uses its analytical-price and Monte Carlo-aligned checkpoints. Their selection criteria are defined in Section III-F. All checkpoints are evaluated on the same frozen test set, and differences are paired by configuration and optimization seed.

3.9 Monte Carlo Target-Noise and Path-Count Diagnostic

The Feynman–Kac configuration receives noisy simulated-payoff targets. Their noise is characterized independently of network approximation using 4,096 fixed inputs sampled from the training domain. For each Gaussian draw Z_m , the antithetic value $-Z_m$ is used. For even M ,

$$\hat{Y}^{MC}(x) = \frac{2}{M} \sum_{m=1}^{M/2} \frac{e^{-r\tau}}{2} [\max(S_T(Z_m) - K, 0) + \max(S_T(-Z_m) - K, 0)]. \quad (41)$$

Relative to the analytical price V_i , the target error is

$$e_i^{MC} = \hat{Y}_i^{MC} - V_i. \quad (42)$$

Its distribution is summarized using empirical bias, standard deviation, MAE, RMSE, median absolute error, and the 95th percentile of absolute error:

$$\begin{aligned} \text{Bias}_{MC} &= \frac{1}{N} \sum_{i=1}^N e_i^{MC}, \quad \text{SD}_{MC} = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (e_i^{MC} - \bar{e}^{MC})^2}, \quad \text{MAE}_{MC} = \\ &= \frac{1}{N} \sum_{i=1}^N |e_i^{MC}|, \quad \text{RMSE}_{MC} = \sqrt{\frac{1}{N} \sum_{i=1}^N (e_i^{MC})^2}. \end{aligned} \quad (43)$$

The diagnostic and network-training experiments use

$$M \in \{16, 64, 256\}, \quad (44)$$

with antithetic sampling throughout and $M = 64$ as the primary configuration. Since the targets are regenerated during training, the network may average part of their noise across observations and updates. Target MAE therefore characterizes the training signal but is not a lower bound on the error attainable by the trained model.

3.10 Experimental Environment and Reproducibility

All experiments were conducted in Kaggle Notebooks using an NVIDIA Tesla T4 GPU. Table 7 records the software and hardware environment.

Table 7. Experimental software and hardware environment.

Component	Configuration
Platform	Kaggle Notebooks
GPU	NVIDIA Tesla T4
CUDA	12.8
Python	3.12
PyTorch	2.10.0+cu128
NumPy	2.0.2
Primary optimization seeds	7, 42, 123, 999, 2025
Control-experiment seeds	42, 123, 2025
Dataset seeds	42 / 43 / 44
cuDNN deterministic mode	enabled
cuDNN benchmarking	disabled
Supervised mini-batch size	1,024

Python, NumPy, and PyTorch random-number generators are reset before every trajectory, and models are constructed only after the reset. Deterministic cuDNN behavior is enabled and cuDNN benchmarking is disabled. Dataset seeds remain distinct from optimization seeds.

The pipeline persists configuration dictionaries, scaling statistics, dataset and protocol hashes, checkpoints, training histories, aggregate and regime-specific metrics, and pointwise test outputs. Pointwise files retain predicted prices, model-derived Greeks, PDE residuals, and financial-violation indicators, permitting diagnostic recomputation without retraining.

A trajectory is marked complete only after its required checkpoints, metrics, predictions, and metadata have been written and verified. Status is persisted after every trajectory, and verified trajectories are skipped on resume. Source-model dependencies are checked before Hybrid fine-tuning. Each execution batch is exported with file-level and archive-level SHA-256 checksums for restoration and final consolidation.

The complete orchestration notebook, including the batch definitions, the aggregation cells, and the figure-generation cells that produce the tables and figures of Section IV, is archived on Zenodo under the DOI 10.5281/zenodo.22539850 (<https://doi.org/10.5281/zenodo.22539850>), an identifier that always resolves to the most recent version of the record.

4 Results and Discussion

Preliminary Note on Benchmark Interpretation. The evaluated models represent distinct implemented configurations rather than training-budget- or parameter-matched realizations of neural architecture families. They differ in learning signal, loss scaling, activation function, parameter count, optimizer-update allocation, and, for residual-trained methods, computational PDE specification. The supervised MLP, DeepONet-style surrogate, and Transformer-inspired pricer learn from analytical prices; PINN and DGM use residual, terminal, and boundary losses; the Hybrid PINN combines analytical and physical terms; and the Feynman–Kac configuration learns from renewed Monte Carlo payoff targets.

All configurations are nevertheless assessed on the same frozen test set through a common implementation for prices, Greeks, PDE residuals, financial admissibility, regime-wise performance, and inference timing. This shared evaluation protocol supports a controlled diagnostic comparison of the realized configurations, but it does not convert the experiment into an intrinsic ranking of

architecture families. Conclusions are therefore stated at the configuration level and interpreted jointly with the training protocol, computational effort, and between-seed variability.

4.1 Dataset Integrity and Analytical-Reference Validation

Before aggregating the model results, the integrity of the frozen datasets and the numerical correctness of the common evaluation pipeline were verified. The stored training, validation, and test artifacts contain 100,000, 20,000, and 20,000 observations, respectively, with 17 recorded variables per observation. For each split, the recomputed SHA-256 digest matched the digest recorded when the dataset was generated. No missing or non-finite values were detected.

Because the three datasets were sampled independently from the same uniform parameter distribution, their realized regime proportions are not expected to be exactly identical. Nevertheless, the distributions were closely aligned. Across the six predefined regimes, the largest absolute difference between any two splits was 0.634 percentage points for the long-maturity regime. The corresponding differences for the remaining regimes did not exceed 0.414 percentage points. Together with the recorded split-specific hashes, these checks provide no indication of corrupted observations or a substantial distributional imbalance among the three frozen splits.

The automatic-differentiation and PDE-residual implementation was validated independently of the trained neural networks. The analytical Black–Scholes pricing function was evaluated at 2,048 points and differentiated using the same automatic-differentiation operations subsequently applied to the learned pricing functions. The resulting Delta, Gamma, and Vega values were compared with their closed-form analytical counterparts. The analytical price was also inserted into the implemented Black–Scholes residual evaluator.

The discrepancies in Table 8 are at or close to floating-point numerical precision and are several orders of magnitude below the model errors analyzed in the following subsections. This experiment therefore validates the differentiation, scaling, and residual-assembly implementation used by the common evaluator. It does not, by itself, assess the approximation quality of any trained model.

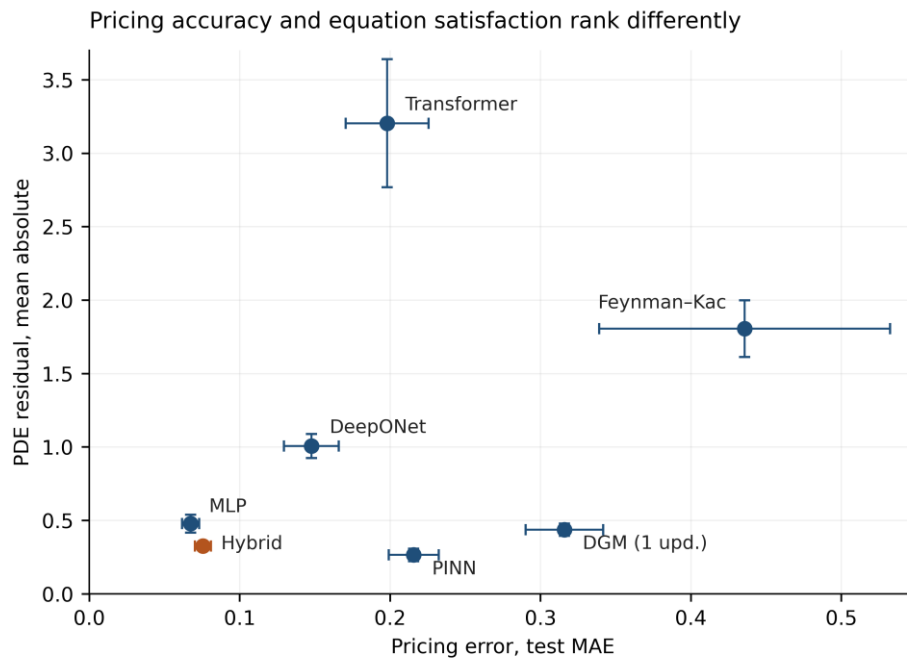
Finally, the filtered MAPE criterion retained 18,827 of the 20,000 test observations, corresponding to 94.135% of the test set. The remaining 1,173 observations had an analytical option price below 10^{-2} and were excluded only from MAPE because percentage errors are ill-conditioned near zero. This filtering count is identical for every configuration because it depends exclusively on the frozen analytical reference prices. All other principal price, Greek, residual, and admissibility metrics were computed on the complete test set.

Table 8. Numerical validation of the automatic-differentiation and Black–Scholes residual evaluation pipeline over 2,048 observations. Reported values are absolute discrepancies; q0.95 denotes their 95th percentile.

Validation quantity	N	Mean absolute	Median	q0.95	Maximum
Delta autograd error	2,048	7.640×10^{-17}	2.776×10^{-17}	2.359×10^{-16}	1.110×10^{-15}
Gamma autograd error	2,048	2.785×10^{-18}	8.674×10^{-19}	1.128×10^{-17}	1.440×10^{-16}
Vega autograd error	2,048	1.057×10^{-14}	7.105×10^{-15}	3.197×10^{-14}	1.528×10^{-13}
Analytical PDE residual	R		2,048	1.939×10^{-15}	9.992×10^{-16}

4.2 Global Black–Scholes Benchmark

Table 9 reports the main results of the Black–Scholes benchmark over five training seeds. For the residual-based configurations, the primary comparison uses $S_{max} = 400$, while the influence of the computational domain is examined separately in Section IV-H. The reported Hybrid PINN corresponds to the data-first variant. The DGM row corresponds to its primary one-update-per-epoch configuration; the effect of increasing its optimizer-update allocation is analyzed in Section IV-I and shows that this row should be read as the lower-update DGM configuration actually used in the primary benchmark, rather than as a budget-matched comparison with PINN.



Means over five training seeds; bars are between-seed standard deviations. Source: Created by the authors based on the experimental results.

Fig. 1. Price MAE versus PDE residual MAE across the evaluated Black–Scholes configurations, with between-seed standard deviations. If the two criteria were equivalent, the configurations would align along a diagonal. Source: Created by the authors based on the experimental results.

No configuration is uniformly dominant across all criteria. Under dense analytical supervision, the MLP baseline remains the most accurate pricing model, with a price MAE of 0.0675 ± 0.0057 . The data-first Hybrid PINN is close behind at 0.0756 ± 0.0053 while attaining a lower PDE residual than the MLP, 0.3249 ± 0.0153 against 0.4781 ± 0.0605 . The Standard PINN with $S_{max} = 400$ is markedly less accurate in price, 0.2158 ± 0.0167 , yet it attains the lowest PDE residual MAE among the seven primary configurations, 0.2648 ± 0.0432 .

The DeepONet-style surrogate achieves reasonable pricing performance, 0.1477 ± 0.0182 , with a substantially higher residual, 1.0066 ± 0.0821 . The Transformer-inspired pricer provides the clearest example of criterion-dependent performance: its price MAE remains competitive at 0.1980 ± 0.0276 , yet it produces the largest mean PDE residual of the benchmark, 3.2037 ± 0.4357 , and the heaviest residual tail. The Feynman–Kac stochastic-target pricer is the least competitive in direct analytical accuracy under the present configuration, and its results must be read together with the noise level of its Monte Carlo training targets, discussed in Section IV-L.

Fig. 1 summarizes this first result. The configuration with the lowest price error is not the one with the lowest PDE residual, and the configurations do not align along a diagonal. The two criteria therefore provide distinct information, and Section IV-M quantifies the extent of their disagreement across the full set of evaluation criteria.

One configuration is competitive on both axes at once. The data-first Hybrid PINN ranks second in price error and second in PDE residual, and it attains the lowest Delta, Gamma, and Vega errors of the benchmark. This qualifies the divergence rather than contradicting it: price accuracy and equation consistency are not intrinsically opposed, although their joint improvement depends on the realized training design. Section IV-F accounts for the cost of this two-stage construction, and Section IV-K separates the effects of training order and activation in the hybrid controls.

Table 9. Global Black–Scholes benchmark. Results are reported as mean $\pm$ standard deviation over five training seeds {7, 42, 123, 999, 2025}. Lower values are better for all reported error metrics.

Panel A: price accuracy and PDE consistency

Model	Price MAE	Price RMSE	PDE MAE
MLP baseline	0.0675 ± 0.0057	0.1005 ± 0.0068	0.4781 ± 0.0605
Hybrid PINN, data-first	0.0756 ± 0.0053	0.1281 ± 0.0058	0.3249 ± 0.0153
DeepONet-style surrogate	0.1477 ± 0.0182	0.2269 ± 0.0279	1.0066 ± 0.0821
Transformer-inspired pricer	0.1980 ± 0.0276	0.2543 ± 0.0296	3.2037 ± 0.4357
Standard PINN, Smax=400	0.2158 ± 0.0167	0.4245 ± 0.0281	0.2648 ± 0.0432
DGM, Smax=400, one update†	0.3160 ± 0.0258	0.5139 ± 0.0230	0.4367 ± 0.0411
Feynman–Kac stochastic-target pricer	0.4358 ± 0.0967	0.5646 ± 0.1206	1.8050 ± 0.1924

Panel B: Greek accuracy

Model	Delta MAE	Gamma MAE	Vega MAE
MLP baseline	0.0045 ± 0.0005	0.00051 ± 0.00006	0.7826 ± 0.0823
Hybrid PINN, data-first	0.0038 ± 0.0003	0.00036 ± 0.00003	0.7086 ± 0.0622
DeepONet-style surrogate	0.0109 ± 0.0011	0.00117 ± 0.00007	1.3714 ± 0.1331
Transformer-inspired pricer	0.0093 ± 0.0009	0.00131 ± 0.00019	1.3225 ± 0.1933
Standard PINN, Smax=400	0.0063 ± 0.0007	0.00043 ± 0.00006	2.1087 ± 0.2712
DGM, Smax=400, one update†	0.0104 ± 0.0007	0.00073 ± 0.00004	2.8657 ± 0.1046
Feynman–Kac stochastic-target pricer	0.0169 ± 0.0023	0.00137 ± 0.00015	2.5623 ± 0.5097

† Primary lower-update DGM configuration; see Section IV-I.

4.3 Financial Admissibility and Shape Constraints

Price accuracy alone does not determine whether a learned pricing surface is financially admissible. For a European call option, predicted prices should remain non-negative, stay above the lower no-arbitrage bound $\max(S - Ke^{-rt}, 0)$, remain below the upper bound S , be non-decreasing in S , and remain convex in S . We therefore examine negative-price violations (NEG), lower-bound violations (LBV), upper-bound violations (UBV), and thresholded monotonicity and convexity violation ratios, denoted MVR_ε and CVR_ε .

Following the reviewer’s suggestion, Table 10 reports violation magnitudes alongside violation frequencies. The two are not redundant. The Standard PINN and the MLP baseline produce negative prices at almost the same rate, 4.32% against 4.75% of test points, but the magnitude of those violations differs by a factor of two in the mean, 0.046 against 0.095, and at the 95th percentile, 0.128 against 0.314. Conversely their lower-bound frequencies are nearly identical, 10.31% against 11.54%, while the PINN’s lower-bound violations are the larger in magnitude, with a mean of 0.166 against 0.097 and a maximum of 2.23 against 1.09. A frequency-only comparison would rank these two configurations as almost equivalent on both criteria, which the magnitudes contradict.

The convexity diagnostic separates the families most sharply. The Transformer-inspired pricer violates the thresholded convexity condition at 7.40% of test points, an order of magnitude more often than the Standard PINN at 0.30%, and with the largest magnitudes recorded on that criterion. This is also the configuration with the largest PDE residual. The association is consistent with the fact that convexity is a second-derivative property and that the Black–Scholes residual contains the term $\frac{1}{2}\sigma^2 S^2 V_{SS}$; it should nevertheless be read as a descriptive association rather than as evidence of a causal mechanism.

Upper-bound violations are absent everywhere. Every configuration respects $V \leq S$ at every test point, which is the only admissibility criterion on which the benchmark records unanimous success. This is reported for completeness and is omitted from the table.

The threshold $\varepsilon = 10^{-3}$ reduces the influence of very small negative derivatives in regions where the corresponding analytical quantity is numerically close to zero. Raw sign violations remain useful as diagnostics, but the thresholded quantities provide a more conservative measure of persistent shape

defects. LBV must also be interpreted carefully because it is one-sided: a model that systematically overprices can satisfy the lower bound while remaining inaccurate. NEG, LBV, derivative-shape diagnostics, pricing error, signed bias, and PDE residuals should therefore be considered jointly rather than as interchangeable criteria.

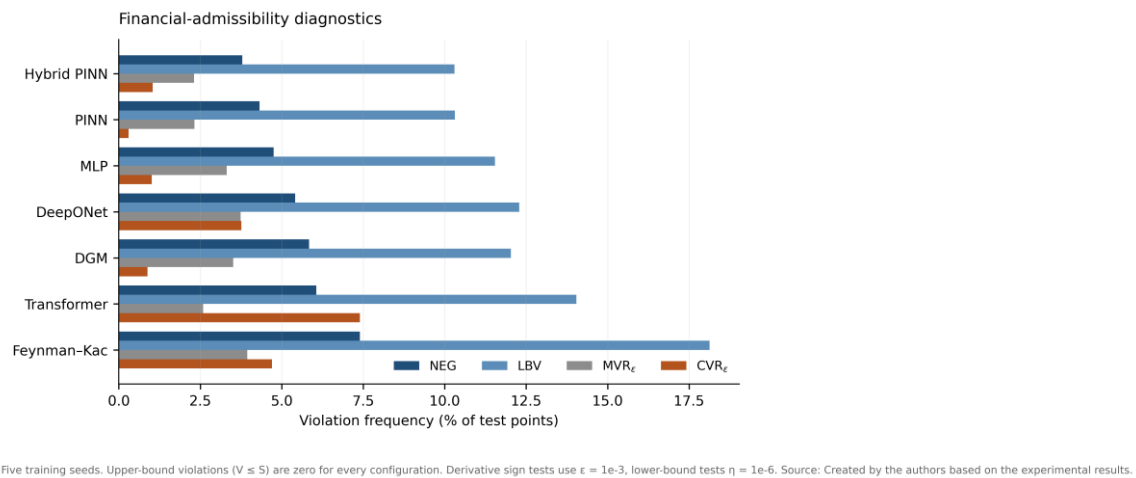


Fig. 2. Financial-admissibility and shape-constraint diagnostics over five training seeds. Lower values are better.
Source: Created by the authors based on the experimental results.

Fig. 2 provides a compact visual comparison and further illustrates that admissibility-based conclusions need not coincide with price-based conclusions. The Hybrid PINN has the lowest NEG, LBV, and MVR frequencies, whereas the Standard PINN has the lowest CVR frequency and the DGM has the smallest mean CVR magnitude. Admissibility is therefore criterion-dependent even before it is compared with price accuracy.

Table 10. Financial-admissibility violations over five training seeds. Frequencies are percentages of test observations. Magnitudes are reported in price units for NEG and LBV and in the units of the corresponding derivative for MVR and CVR. Derivative sign tests use $\epsilon = 10^{-3}$, and lower-bound tests use $\eta = 10^{-6}$. Lower values are better.

Negative price (NEG) and Lower-bound violation (LBV)

Configuration	NEG %	NEG Mean	NEG q0.95	NEG Max.	LBV %	LBV Mean	LBV q0.95	LBV Max.
Hybrid PINN, data-first	3.79	0.0917	0.277	0.785	10.30	0.0892	0.245	0.805
Standard PINN	4.32	0.0460	0.128	0.541	10.31	0.1663	0.660	2.227
MLP baseline	4.75	0.0952	0.314	1.049	11.54	0.0970	0.313	1.090
DeepONet-style surrogate	5.41	0.2596	0.806	2.477	12.29	0.2447	0.782	2.477
DGM, one update	5.84	0.3957	1.152	2.525	12.03	0.3955	1.191	2.907
Transformer-inspired pricer	6.06	0.1532	0.309	0.835	14.04	0.2300	0.555	1.367
Feynman-Kac pricer	7.40	0.4738	0.899	2.110	18.13	0.5953	1.330	3.239

Monotonicity (MVRε) and Convexity (CVRε)

Configuration	MVR %	MVR Mean	MVR q0.95	MVR Max.	CVR %	CVR Mean	CVR q0.95	CVR Max.
Hybrid PINN, data-first	2.31	0.0098	0.0343	0.0650	1.04	0.00239	0.00509	0.00870
Standard PINN	2.32	0.0028	0.0069	0.0155	0.30	0.00175	0.00343	0.00464
MLP baseline	3.31	0.0091	0.0312	0.0736	1.01	0.00230	0.00476	0.00934
DeepONet-style surrogate	3.74	0.0258	0.0796	0.1592	3.76	0.00271	0.00697	0.01913

Configuration	MVR %	MVR Mean	MVR q0.95	MVR Max.	CVR %	CVR Mean	CVR q0.95	CVR Max.
DGM, one update	3.51	0.0147	0.0432	0.0765	0.88	0.00147	0.00224	0.00421
Transformer-inspired pricer	2.59	0.0052	0.0131	0.0295	7.40	0.00562	0.01409	0.02042
Feynman–Kac pricer	3.94	0.0150	0.0339	0.0551	4.70	0.00191	0.00424	0.00865

Upper-bound violations (UBV , $V \leq S$) are zero for every configuration and are omitted.

4.4 Greek Accuracy

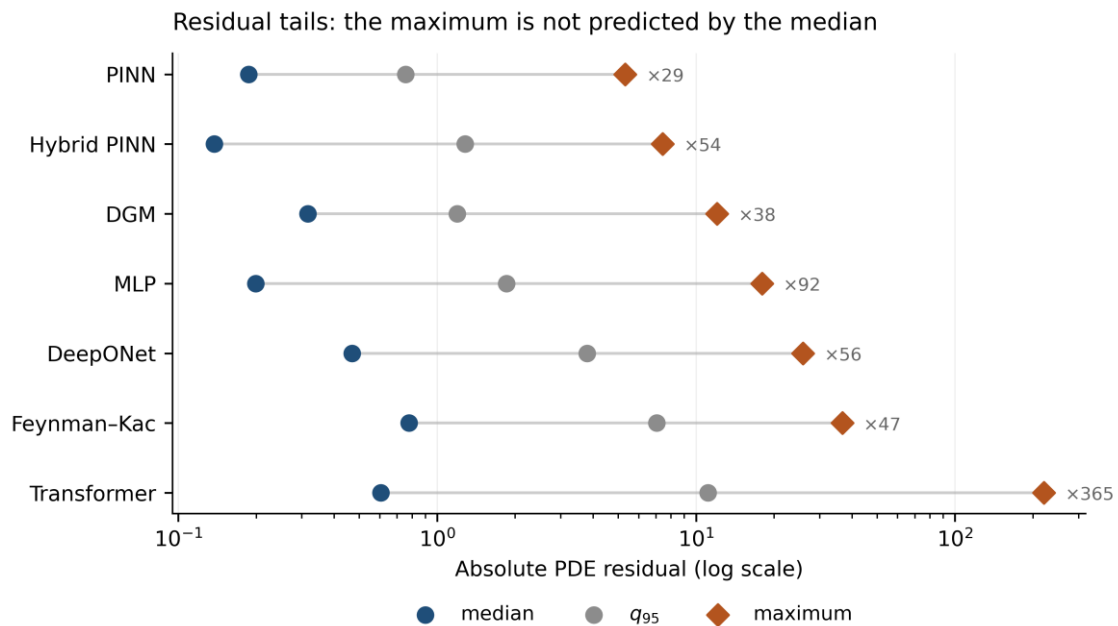
Since option-pricing models are frequently differentiated to compute sensitivities, Greek accuracy should be evaluated separately from price accuracy. Table 9 shows that the ranking changes once the learned pricing surface is differentiated. The data-first Hybrid PINN attains the lowest Delta, Gamma, and Vega errors, 0.0038, 0.00036, and 0.7086, respectively. The MLP baseline follows at 0.0045, 0.00051, and 0.7826. The Standard PINN achieves a Gamma MAE of 0.00043, despite a price error more than three times larger than that of the MLP. This is compatible with its training objective, since the Black–Scholes residual explicitly contains the second-order term $\frac{1}{2}\sigma^2 S^2 V_{SS}$.

Vega has the largest absolute error for every configuration, although its scale and units differ from those of Delta and Gamma. Its MAE ranges from 0.7086 for the Hybrid PINN to 2.8657 for the DGM. The PDE residual contains derivatives with respect to S and τ , but no derivative with respect to σ ; consequently, a small residual does not directly guarantee an accurate Vega. The PINN illustrates this distinction by combining a low Gamma error with a substantially larger Vega error than the supervised MLP.

The DeepONet-style and Transformer-inspired configurations have Gamma MAEs of 0.00117 and 0.00131, respectively. The Feynman–Kac configuration has the largest Delta error, while DGM has the largest Vega error; thus no single configuration is worst for every Greek. Overall, price accuracy does not determine differential accuracy, and the three Greeks should not be collapsed into a single unscaled score.

4.5 PDE Residual Distribution and Tail Behavior

Mean residuals alone are not sufficient to characterize PDE consistency. Two models may have similar average residuals while exhibiting substantially different residual distributions. Table 11 therefore reports the mean absolute residual, the median, the 95th percentile, the maximum, and the ratio between the maximum and the median.



Five training seeds; annotations give the maximum-to-median ratio. Source: Created by the authors based on the experimental results.

Fig. 3. Residual distribution and tail behavior across the evaluated configurations, on a logarithmic scale. Annotations give the max-to-median ratio. Source: Created by the authors based on the experimental results.

The Standard PINN with $S_{max} = 400$ provides the lowest mean residual, the smallest maximum residual, and the lightest relative tail, with a max-to-median ratio of 29.2. The Hybrid PINN follows on the mean and the maximum. The DGM occupies an intermediate position, and the DeepONet-style, Transformer-inspired, and stochastic-target configurations show larger residual levels throughout.

The ordering by tail ratio is not the ordering by mean residual, and this is the informative part of the table. The MLP baseline has a lower mean residual than the DeepONet-style surrogate, 0.478 against 1.007, but a heavier relative tail, 91.9 against 55.7, with a maximum residual of 17.98 against a median of 0.199. Good supervised price approximation can therefore produce moderate average differential consistency without guaranteeing uniformly small residuals over the domain.

The Transformer-inspired pricer is the clearest case in which the residual distribution adds information. Its residual MAE is already the highest of the benchmark, but its maximum reaches 220.8 and its max-to-median ratio is 365, approximately four times the next-largest ratio. An unfavorable average residual is therefore compounded by a particularly heavy tail, indicating localized regions of much stronger PDE inconsistency rather than uniformly degraded consistency.

Fig. 3 reinforces this distinction. Mean PDE consistency and residual-tail behavior are related but not interchangeable, and both are useful when characterizing learned pricing surfaces.

Table 11. PDE residual distribution diagnostics, means over five training seeds. Lower values are better.

Model	Mean	Median	q0.95	Max	Max/Med
Standard PINN, $S_{max}=400$	0.265	0.187	0.756	5.32	29.2
Hybrid PINN, data-first	0.325	0.138	1.282	7.44	54.4
DGM, $S_{max}=400$, one update	0.437	0.317	1.197	12.05	37.6
MLP baseline	0.478	0.199	1.854	17.98	91.9
DeepONet-style surrogate	1.007	0.469	3.799	25.91	55.7
Feynman–Kac stochastic-target pricer	1.805	0.779	7.055	36.72	46.7
Transformer-inspired pricer	3.204	0.607	11.120	220.82	365.4

4.6 Efficiency and Multi-Seed Stability

Table 12 reports training time, price-only inference time, parameter count, and selected training epoch for the seven primary configurations, together with a total-cost column that accounts for the training of source checkpoints where applicable.

The Feynman–Kac stochastic-target pricer has the lowest mean training time, although its between-seed dispersion is also the largest. The Standard PINN with $S_{max} = 400$ has the largest training cost in the primary benchmark, 955.6 seconds, approximately seven times the cost of the MLP baseline for a price error more than three times larger; it is also the configuration that attains the lowest PDE residual. This equation-consistency–accuracy–cost trade-off is concealed by a single-metric comparison.

Epoch counts and wall-clock times do not rank the configurations in the same order, and the epoch is not a transferable unit of cost. The single-update DGM runs 2,770 epochs in 230.6 seconds while the Transformer-inspired pricer runs 93.2 epochs in 222.2 seconds, a factor of thirty in epoch count for similar wall-clock budgets. Reported epoch counts are therefore comparable only within a configuration family, and the cost axis of this benchmark uses wall-clock time.

The Hybrid PINN illustrates why the accounting must be explicit. Its fine-tuning stage requires 165.9 seconds, but the seed-matched MLP source checkpoint must also be trained, bringing the end-to-end mean to 301.5 seconds. This is approximately twice the cost of the MLP baseline, which remains slightly more accurate in price. Reporting only the fine-tuning stage would understate the computational cost of the transfer-based configuration.

Price-only inference remains inexpensive for all configurations relative to training. The Transformer-inspired pricer is the slowest at inference, whereas the MLP and PINN are among the fastest.

Between-seed variability also differs among configurations, and it is not uniform across metrics. The MLP baseline and the Hybrid PINN remain stable in price MAE, with standard deviations of 0.0057 and 0.0053. The Feynman–Kac pricer is the least stable, 0.0967, an order of magnitude larger, which is consistent with the renewed stochastic targets described in Section IV-L. For residual-based configurations that reach or approach the maximum training horizon, these results should not be interpreted as evidence of complete optimization convergence.

Table 12. Efficiency and selected training statistics, reported as mean $\pm$ standard deviation over five training seeds where applicable. Inference time is the mean, across seeds, of the minimum of five synchronized price-only passes over the complete test set. The Total column includes the seed-matched source-checkpoint cost for the transfer-based configuration.

Model	Training time (s)	Total (s)	Inference time (s)	Parameters	Best epoch
Feynman–Kac stochastic-target pricer	82.3 $\pm$ 41.2	82.3 $\pm$ 41.2	0.0032	66,945	63.8 $\pm$ 46.8
MLP baseline	135.6 $\pm$ 27.3	135.6 $\pm$ 27.3	0.0032	66,945	141.6 $\pm$ 36.2
DeepONet-style surrogate	157.0 $\pm$ 30.1	157.0 $\pm$ 30.1	0.0063	132,993	137.0 $\pm$ 32.6
Hybrid PINN, data-first	165.9 $\pm$ 0.1	301.5 $\pm$ 27.3	0.0032	66,945	64.0 $\pm$ 13.9
Transformer-inspired pricer	222.2 $\pm$ 31.6	222.2 $\pm$ 31.6	0.0392	125,953	93.2 $\pm$ 17.6
DGM, $S_{max}=400$, one update	230.6 $\pm$ 2.6	230.6 $\pm$ 2.6	0.0088	208,257	2770.0 $\pm$ 168.1
Standard PINN, $S_{max}=400$	955.6 $\pm$ 48.4	955.6 $\pm$ 48.4	0.0027	66,945	2320.0 $\pm$ 485.5

4.7 Checkpoint-Selection Sensitivity

The primary benchmark uses the frozen analytical validation set to select one checkpoint from every training trajectory. This provides a common model-selection criterion across configurations, but it also means that the PINN, DGM, and Feynman–Kac configurations are not fully independent of analytical labels at the checkpoint-selection stage. To quantify the effect of this choice, each of their five primary trajectories additionally retains the signal-aligned checkpoint defined in (22).

For PINN and DGM, the alternative checkpoint minimizes the combined PDE, terminal, and boundary loss on a fixed physics-validation sample. For each trajectory, this sample is generated once using a deterministically recorded seed and remains unchanged across validation epochs. It is independent of the collocation points resampled during optimization. Different configurations and optimization seeds may nevertheless use different fixed physics-validation realizations. For the Feynman–Kac configuration, the alternative checkpoint minimizes prediction error against a fixed Monte Carlo validation target generated using 2,048 paths per input.

The analytical and signal-aligned checkpoints are extracted from the same optimization trajectory. They are therefore paired observations rather than independently trained models. Both checkpoints are evaluated using the same frozen 20,000-point analytical test set and the same multicriteria evaluation routine.

For the PINN, the analytical and physics criteria select the same epoch for only one of the five trajectories. On average, physics-aligned selection reduces the test PDE MAE from 0.265 ± 0.043 to 0.251 ± 0.023 , a reduction of approximately 5.3%, while increasing the price MAE from 0.216 ± 0.017 to 0.234 ± 0.020 , or approximately 8.5%. This direction is broadly consistent with the distinction between the two selection criteria. However, the change in test PDE MAE is not uniform across seeds: it improves for three trajectories, worsens for one, and is unchanged for one because the same epoch is selected. The modest aggregate difference should therefore be interpreted as a checkpoint-sensitivity result, not as evidence that one selection rule uniformly dominates the other.

For DGM, the two criteria select the same epoch for two of the five trajectories. The physics-aligned checkpoint does not improve the aggregate test PDE residual: PDE MAE changes from 0.437 ± 0.041 to 0.469 ± 0.077 , while price MAE changes from 0.316 ± 0.026 to 0.330 ± 0.040 . At the seed level, test PDE MAE improves for two trajectories, is unchanged for two, and worsens substantially for one. This heterogeneity explains the larger standard deviation of the physics-selected result.

The DGM outcome does not imply that the physics-validation criterion failed to select its own minimum. By construction, the physics-aligned checkpoint minimizes the recorded loss on its trajectory-specific fixed validation sample. Table 13, however, reports the interior PDE residual on the common analytical test set. A minimum on one finite physics-validation realization need not correspond to the smallest residual on a distinct set of test points, particularly when residual training remains variable across epochs and seeds.

For the Feynman–Kac configuration, the analytical and fixed Monte Carlo criteria select the same epoch for all five trajectories. Exact comparison of the retained parameter tensors confirms that the two checkpoint versions are identical within each seed, explaining the equality of every reported test metric. This result indicates strong empirical alignment between the two validation criteria under the 2,048-path Monte Carlo validation protocol. It does not establish that analytical and finite-path Monte Carlo checkpoint selection must coincide under other samples, path budgets, or optimization trajectories.

The checkpoint analysis refines, but does not overturn, the interpretation of the primary benchmark. PINN exhibits the expected aggregate trade-off between analytical-price selection and physics-aligned selection, although its seed-level behavior is not uniform. DGM displays greater selection variability, and the alternative criterion does not improve its aggregate test residual under the evaluated protocol. Feynman–Kac is insensitive to the selection rule in the five observed trajectories because both criteria retain the same parameters.

The analytical checkpoint is retained as the primary benchmark checkpoint because it provides a common selection criterion across all seven configurations. The signal-aligned checkpoints are reported only as sensitivity evaluations and are not counted as additional training runs. Accordingly, PINN and DGM remain physics-trained, and Feynman–Kac remains stochastic-target trained, but none of these three configurations is described as fully label-independent at the model-selection stage.

Table 13. Sensitivity to checkpoint-selection criterion over the five primary training seeds {7, 42, 123, 999, 2025}. Values are means $\pm$ standard deviations. “Analytical” denotes selection by analytical validation-price MSE, “Physics” denotes selection by the fixed physics-validation loss, and “Monte Carlo” denotes selection by the fixed 2,048-path stochastic validation target. Selected epochs are averages over the same five trajectories.

Configuration	Checkpoint	Selected epoch	Price MAE	PDE MAE	Delta MAE	Gamma MAE	LBV
PINN, $S_{max}=400$	Analytical	2320 $\pm$ 486	0.216 $\pm$ 0.017	0.265 $\pm$ 0.043	0.00630 $\pm$ 0.00072	0.000429 $\pm$ 0.000062	0.1031 $\pm$ 0.0059
PINN, $S_{max}=400$	Physics	2190 $\pm$ 275	0.234 $\pm$ 0.020	0.251 $\pm$ 0.023	0.00643 $\pm$ 0.00077	0.000436 $\pm$ 0.000055	0.1144 $\pm$ 0.0280
DGM, $S_{max}=400$	Analytical	2770 $\pm$ 168	0.316 $\pm$ 0.026	0.437 $\pm$ 0.041	0.01044 $\pm$ 0.00065	0.000730 $\pm$ 0.000035	0.1203 $\pm$ 0.0420
DGM, $S_{max}=400$	Physics	2660 $\pm$ 622	0.330 $\pm$ 0.040	0.469 $\pm$ 0.077	0.01080 $\pm$ 0.00152	0.000757 $\pm$ 0.000092	0.1136 $\pm$ 0.0214
Feynman–Kac, $M=64$	Analytical	63.8 $\pm$ 46.8	0.436 $\pm$ 0.097	1.805 $\pm$ 0.192	0.01687 $\pm$ 0.00226	0.001369 $\pm$ 0.000147	0.1813 $\pm$ 0.0393
Feynman–Kac, $M=64$	Monte Carlo	63.8 $\pm$ 46.8	0.436 $\pm$ 0.097	1.805 $\pm$ 0.192	0.01687 $\pm$ 0.00226	0.001369 $\pm$ 0.000147	0.1813 $\pm$ 0.0393

4.8 Computational-Domain and Collocation-Density Sensitivity

The computational-domain analysis examines whether the performance of residual-trained configurations depends on the artificial upper boundary S_{max} . All comparisons in this subsection use the same frozen analytical test set and the matched control seeds {42, 123, 2025}. Restricting the aggregation to these three seeds is necessary because the domain and density controls were not trained with the two additional seeds used in the primary benchmark.

For the PINN, three fixed-count domains, $S_{max} \in \{200, 400, 800\}$, are compared using 4,096 interior collocation points per optimizer update. Two additional density-controlled configurations use 8,192 points at $S_{max} = 400$ and 16,384 points at $S_{max} = 800$, thereby maintaining the interior sampling density along the asset-price dimension relative to the $S_{max} = 200$ configuration. The terminal and boundary allocations remain fixed at 2,048 points per condition. For DGM, the domain comparison uses the primary real-unit loss and one-update-per-epoch protocol; optimization-budget effects are examined separately in Section IV-I.

Under the fixed-count protocol, moving the artificial boundary from $S_{max} = 200$ to $S_{max} = 400$ substantially improves both residual-trained configurations. The PINN price MAE decreases by a factor of approximately 12.2, from 2.596 ± 0.239 to 0.212 ± 0.018 , while its Vega MAE decreases from 11.990 ± 0.840 to 1.981 ± 0.084 . The corresponding DGM price MAE decreases by a factor of approximately 7.4, from 2.415 ± 0.052 to 0.325 ± 0.026 , and its Vega MAE decreases from 11.652 ± 0.195 to 2.861 ± 0.107 .

At $S_{max} = 200$, the artificial upper boundary coincides with the upper edge of the evaluation domain $S \in [40, 200]$. The asymptotic condition in (11) is therefore imposed directly adjacent to observations on which the model is evaluated. The DGM results illustrate why the interior PDE residual cannot diagnose this issue by itself: moving from $S_{max} = 200$ to $S_{max} = 400$ reduces its price MAE by a factor of 7.4, whereas its PDE MAE changes only from 0.488 ± 0.157 to 0.454 ± 0.047 . Moreover, the DGM price bias is negative for all three control seeds at $S_{max} = 200$, with an aggregate value of -2.054 ± 0.129 , compared with 0.005 ± 0.119 at $S_{max} = 400$. These observations are consistent with an

influence of the nearby artificial boundary, although they do not isolate boundary placement from every other domain-dependent training effect.

The truncation effect is also heterogeneous across financial regimes. Table 15 reports the DGM price MAE under $S_{max} = 200$ and $S_{max} = 400$ using the same three control seeds. The regimes overlap and should therefore be interpreted as diagnostic slices rather than as a partition of the test set.

The deterioration at $S_{max} = 200$ is smallest under short maturity and largest under high volatility. Large effects are also observed for long maturities and for near- and in-the-money options. The out-of-the-money regime has a smaller degradation despite having a volatility–maturity distribution comparable to several of the other moneyness regimes. Domain sensitivity therefore cannot be characterized by volatility or maturity alone; it is jointly associated with volatility, maturity, and moneyness.

The dependence on domain size is not monotonic. With a fixed number of collocation points, enlarging the PINN domain from $S_{max} = 400$ to $S_{max} = 800$ increases the price MAE from 0.212 ± 0.018 to 0.402 ± 0.066 , the PDE MAE from 0.266 ± 0.039 to 0.595 ± 0.077 , and the Vega MAE from 1.981 ± 0.084 to 3.333 ± 0.701 . Thus, placing the upper boundary farther from the evaluation region does not necessarily improve the learned approximation under a fixed sampling and optimization protocol. Because the asymptotic condition is imposed farther from the test domain at $S_{max} = 800$, its misspecification is unlikely to be the sole explanation. The wider training region, the allocation of collocation points, the fixed network capacity, and the available optimization effort are plausible contributing factors, but the present experiment does not identify their individual causal contributions.

The density-controlled configurations directly examine one of these factors. At $S_{max} = 400$, doubling N_Ω changes the price MAE only from 0.212 ± 0.018 to 0.217 ± 0.018 , while improving the PDE MAE from 0.266 ± 0.039 to 0.243 ± 0.010 . The unrounded price-MAE difference is 0.0043, smaller than the between-seed standard deviation 0.0180 of either configuration. At $S_{max} = 800$, quadrupling N_Ω relative to the fixed-count configuration produces only a modest price-MAE reduction, from 0.402 ± 0.066 to 0.388 ± 0.053 , and leaves the PDE MAE nearly unchanged.

The density control nevertheless improves the two reported admissibility frequencies at $S_{max} = 800$: LBV decreases from 0.1664 ± 0.0226 to 0.1049 ± 0.0117 , and NEG decreases from 0.0747 ± 0.0074 to 0.0476 ± 0.0068 . This criterion-specific improvement reinforces the need to evaluate sampling changes using more than price and residual errors alone. At $S_{max} = 400$, by contrast, the corresponding admissibility differences are small relative to their between-seed variability.

Overall, restoring the interior collocation density does not recover the accuracy obtained at $S_{max} = 400$ and therefore does not support collocation dilution as the sole explanation for the deterioration at $S_{max} = 800$. The density-controlled experiments are not computationally cost-matched, however, because their optimizer updates process more collocation points. The results establish $S_{max} = 400$ as the best-performing truncation among the tested values under the present protocol, but they do not establish it as a theoretically optimal boundary.

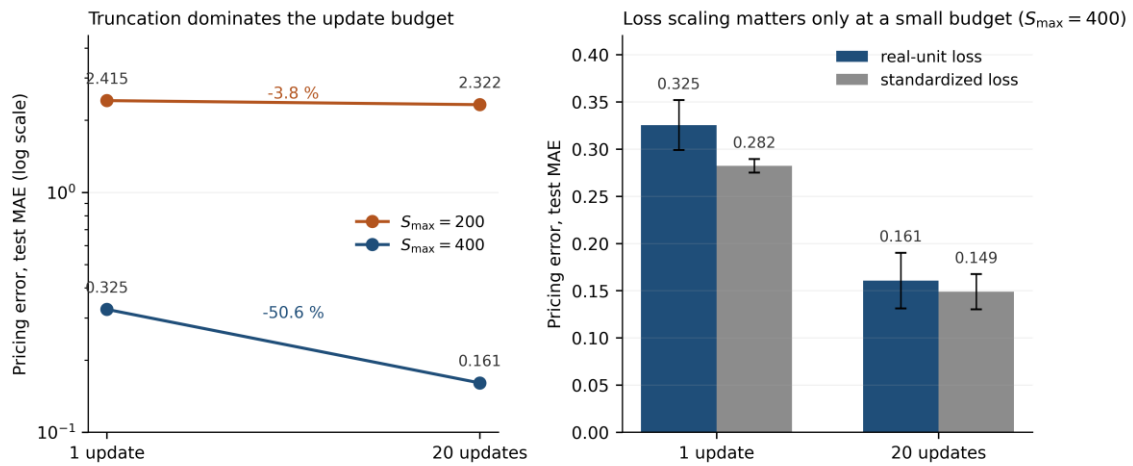
Table 14. Computational-domain and collocation-density sensitivity. Results are mean $\pm$ standard deviation over the matched control seeds {42, 123, 2025}. N_Ω denotes the number of interior collocation points per optimizer update. LBV and NEG are reported as proportions. Lower values indicate better performance for all reported quantities.

Model	Smax	Protocol	N_Ω	Price MAE	PDE MAE	Vega MAE	LBV	NEG
PINN	200	Fixed	4,096	2.596 ± 0.239	0.969 ± 0.453	11.990 ± 0.840	0.0240 ± 0.0197	0.0125 ± 0.0126
PINN	400	Fixed	4,096	0.212 ± 0.018	0.266 ± 0.039	1.981 ± 0.084	0.1051 ± 0.0074	0.0412 ± 0.0203
PINN	400	Density	8,192	0.217 ± 0.018	0.243 ± 0.010	2.019 ± 0.170	0.0983 ± 0.0459	0.0437 ± 0.0286
PINN	800	Fixed	4,096	0.402 ± 0.066	0.595 ± 0.077	3.333 ± 0.701	0.1664 ± 0.0226	0.0747 ± 0.0074

Model	Smax	Protocol	N Ω	Price MAE	PDE MAE	Vega MAE	LBV	NEG
PINN	800	Density	16,384	0.388 $\pm$ 0.053	0.591 $\pm$ 0.038	3.485 $\pm$ 0.260	0.1049 $\pm$ 0.0117	0.0476 $\pm$ 0.0068
DGM	200	Fixed	4,096	2.415 $\pm$ 0.052	0.488 $\pm$ 0.157	11.652 $\pm$ 0.195	0.0995 $\pm$ 0.0499	0.0506 $\pm$ 0.0082
DGM	400	Fixed	4,096	0.325 $\pm$ 0.026	0.454 $\pm$ 0.047	2.861 $\pm$ 0.107	0.1315 $\pm$ 0.0523	0.0614 $\pm$ 0.0164

Table 15. Regime-wise localization of DGM domain sensitivity. Values are price MAE means $\pm$ standard deviations over the matched control seeds {42, 123, 2025}. The final column is the ratio of the two reported means.

Regime	Smax=200	Smax=400	Ratio
Short maturity	0.404 $\pm$ 0.127	0.275 $\pm$ 0.033	1.47
Out of the money	1.205 $\pm$ 0.040	0.297 $\pm$ 0.031	4.06
Near the money	3.022 $\pm$ 0.047	0.375 $\pm$ 0.037	8.05
In the money	3.266 $\pm$ 0.065	0.341 $\pm$ 0.026	9.57
Long maturity	3.612 $\pm$ 0.013	0.385 $\pm$ 0.010	9.37
High volatility	4.724 $\pm$ 0.038	0.349 $\pm$ 0.011	13.55
Global	2.415 $\pm$ 0.052	0.325 $\pm$ 0.026	7.42



Three common seeds {42, 123, 2025}; bars are between-seed standard deviations. Source: Created by the authors based on the experimental results.

Fig. 4. Left: computational domain crossed with update allocation, logarithmic scale. Right: update allocation crossed with loss scaling at $S_{\max} = 400$. Matched control seeds {42, 123, 2025}. Source: Created by the authors based on the experimental results.

4.9 Optimizer-Update Sensitivity

The primary PINN and DGM configurations differ substantially in the amount of optimization performed within one epoch. The PINN uses twenty optimizer updates per epoch, whereas the primary DGM configuration uses one. DGM was therefore retrained with twenty updates per epoch while retaining its architecture, collocation sizes, computational domain, optimizer, maximum epoch count, and control seeds. At $S_{\max} = 400$, update allocation was also crossed with real-unit and standardized residual-loss scaling. These controls distinguish two protocol factors that were confounded in the submitted comparison.

Under real-unit scaling at $S_{\max} = 400$, increasing the DGM update rate reduces its price MAE from 0.3254 ± 0.0264 to 0.1606 ± 0.0295 , and its PDE residual MAE from 0.4539 ± 0.0466 to 0.1498 ± 0.0138 . The corresponding Gamma R^2 increases from 0.8342 ± 0.0010 to 0.9574 ± 0.0199 .

This result materially revises the interpretation of the primary PINN–DGM gap. At the matched update allocation, the DGM attains a lower PDE residual than the Standard PINN, 0.1498 ± 0.0138 against 0.2662 ± 0.0394 on the same seeds. The primary difference in residual consistency therefore cannot be attributed to architecture alone. However, the twenty-update DGM requires 3022.4 ± 2.4 seconds, compared with 230.6 ± 3.2 seconds for its one-update counterpart and 979.1 ± 7.6 seconds for PINN on the matched seeds. Matching the number of optimizer updates does not match wall-clock cost.

Crossing the update allocation with the loss scaling separates the two factors. At one update per epoch, replacing the real-unit loss by its standardized form lowers the price MAE from 0.3254 ± 0.0264 to 0.2822 ± 0.0072 , a difference of 0.0432. At twenty updates, the same substitution changes the price MAE from 0.1606 ± 0.0295 to 0.1489 ± 0.0186 , a smaller difference of 0.0117. The residual comparison is not uniformly ordered: the standardized variant is better at one update, whereas the real-unit variant is better at twenty updates. With only three matched seeds, these patterns are treated as descriptive evidence that loss scaling interacts with update allocation, not as a formal test of an interaction.

The training and validation losses of the real-unit and standardized variants are expressed on different scales and are therefore not comparable across variants. Only the evaluation metrics, computed in price units on the frozen test set, are comparable.

The $S_{max} = 200$ experiment provides the most informative observation of this subsection, and it requires both evaluation axes to be visible. Raising the update allocation reduces the PDE residual MAE from 0.4877 ± 0.1569 to 0.2744 ± 0.1263 , a reduction of 43.7%, while the price MAE decreases only from 2.4145 ± 0.0523 to 2.3219 ± 0.0523 , a reduction of 3.8%, for an approximately elevenfold increase in mean cost. At $S_{max} = 400$, the same change reduces the residual by 67.0% and the price error by 50.6%.

The two evaluation criteria therefore respond differently at $S_{max} = 200$: additional optimization substantially improves equation satisfaction but barely changes price accuracy. The individual trajectories also show divergent checkpoint timing. For seed 2025, for example, the best analytical-price epoch remains 150 while the best physics-validation epoch moves to 2800; for seed 42, the corresponding epochs are 350 and 950. These examples are consistent with continued improvement of the residual objective after price validation has plateaued, but they should not be generalized to a single common checkpoint pattern across all seeds.

A residual-only diagnostic would have declared the $S_{max} = 200$ configuration substantially improved. Domain adequacy and optimization adequacy therefore affect the benchmark through distinct mechanisms, and a larger optimization budget cannot compensate for a truncation boundary placed inside the region where the option value is still economically active. The residual at $S_{max} = 200$ is also strongly dispersed across seeds, from 0.129 to 0.348. This variation reinforces the need for repeated optimization seeds.

This experiment should not be interpreted as a fully equalized architecture-level comparison between PINN and DGM. Their architectures remain different, as do aspects of their loss formulation. It shows that optimizer-update allocation is a consequential benchmark variable, that its effect can interact with loss scaling and the computational domain, and that epoch counts alone are insufficient to characterize optimization effort.

Table 16. Optimizer-update allocation crossed with loss scaling for the residual-driven configurations. Results are mean $\pm$ standard deviation over the matched control seeds {42, 123, 2025}. At the maximum epoch horizon, the one- and twenty-update configurations permit 3,000 and 60,000 optimizer updates, respectively; early stopping can reduce these totals.

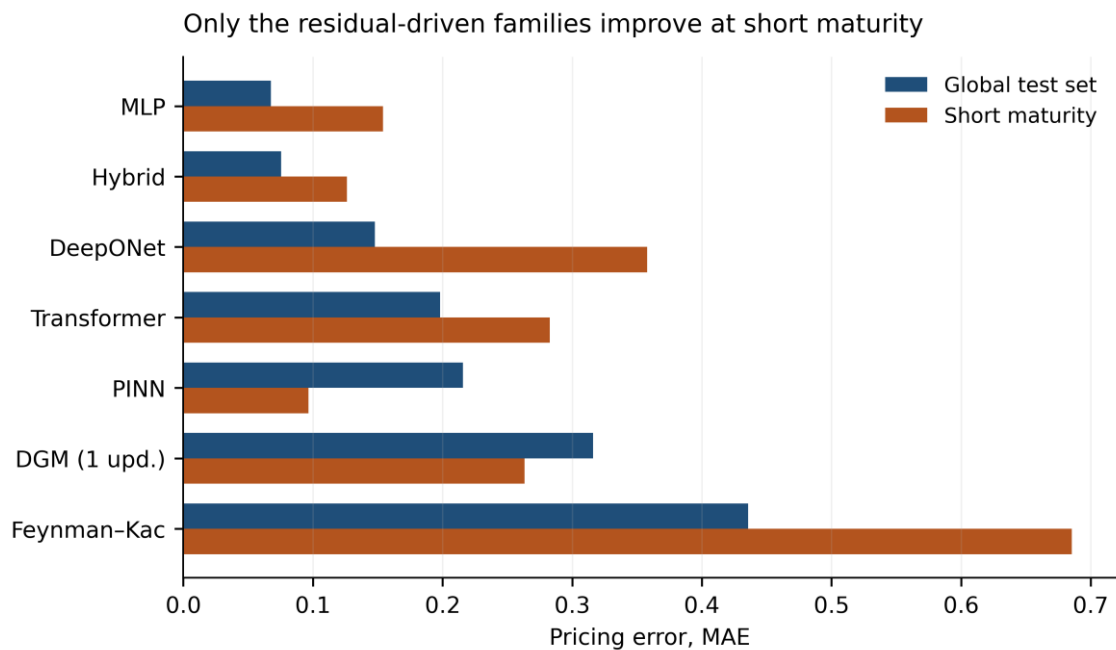
Model	Smax	Updates/epoch	Loss scale	Price MAE	PDE MAE	Training time (s)
DGM	400	1	real unit	0.3254 ± 0.0264	0.4539 ± 0.0466	230.6 ± 3.2
DGM	400	1	standardized	0.2822 ± 0.0072	0.3990 ± 0.0321	235.1 ± 2.8
DGM	400	20	real unit	0.1606 ± 0.0295	0.1498 ± 0.0138	3022.4 ± 2.4
DGM	400	20	standardized	0.1489 ± 0.0186	0.1622 ± 0.0394	2776.0 ± 138.7
PINN	400	20	real unit	0.2124 ± 0.0180	0.2662 ± 0.0394	979.1 ± 7.6

Model	Smax	Updates/epoch	Loss scale	Price MAE	PDE MAE	Training time (s)
DGM	200	1	real unit	2.4145 ± 0.0523	0.4877 ± 0.1569	230.2 ± 3.3
DGM	200	20	real unit	2.3219 ± 0.0523	0.2744 ± 0.1263	2587.0 ± 442.0

4.10 Regime-Wise Performance

Global metrics provide only an average view of performance and may hide localized failures in financially important parts of the parameter domain. We therefore evaluate each configuration across six regimes: near-the-money, in-the-money, out-of-the-money, short maturity, long maturity, and high volatility. The regimes overlap and should be interpreted as diagnostic slices rather than as a partition of the test set.

Table 17 and Fig. 5 show that pricing errors are not uniformly distributed. The leading configuration changes in the near-the-money, out-of-the-money, and short-maturity slices, with the most pronounced reversal occurring under short maturity. The Standard PINN and the DGM are the only primary configurations whose short-maturity error falls below their own global price error: 0.0966 against 0.2158 for the PINN, a ratio of 0.45, and 0.2632 against 0.3160 for the DGM. Every data-driven and hybrid configuration degrades in that regime, most severely the DeepONet-style surrogate, 0.3579 against 0.1477, and the Feynman–Kac pricer, 0.6853 against 0.4358. In absolute terms the PINN obtains a short-maturity MAE of 0.0966 against 0.1540 for the MLP baseline, reversing their global ordering.



Means over five training seeds. Source: Created by the authors based on the experimental results.

Fig. 5. Global and short-maturity price MAE across the evaluated configurations. Only the Standard PINN and DGM have lower short-maturity than global MAE. Source: Created by the authors based on the experimental results.

This pattern is consistent with the explicit terminal-condition term used by the residual-trained configurations: $V(S, K, 0) = \max(S - K, 0)$. The supervised models instead learn the near-expiry region from sampled price labels, while the target approaches a non-smooth payoff profile as τ tends to zero. The experiment does not isolate terminal-condition enforcement from the other differences among configurations, so this explanation is interpretive rather than causal.

The advantage does not extend across the domain. Long-maturity and high-volatility regions remain the most difficult for the PINN and the DGM, with errors of 0.2881 and 0.2917 for the PINN against 0.0573

and 0.0644 for the MLP baseline. Proximity to the terminal condition therefore does not translate into uniformly better performance.

These results confirm that global averages alone are insufficient for model selection. In applications focused on a particular maturity, volatility, or moneyness region, local performance may be more relevant than modest differences in global MAE; under short maturity, the Standard PINN moves from fifth in the global price-MAE ranking to first.

Table 17. Regime-wise price MAE across the Black–Scholes configurations, means over five training seeds. Lower values are better.

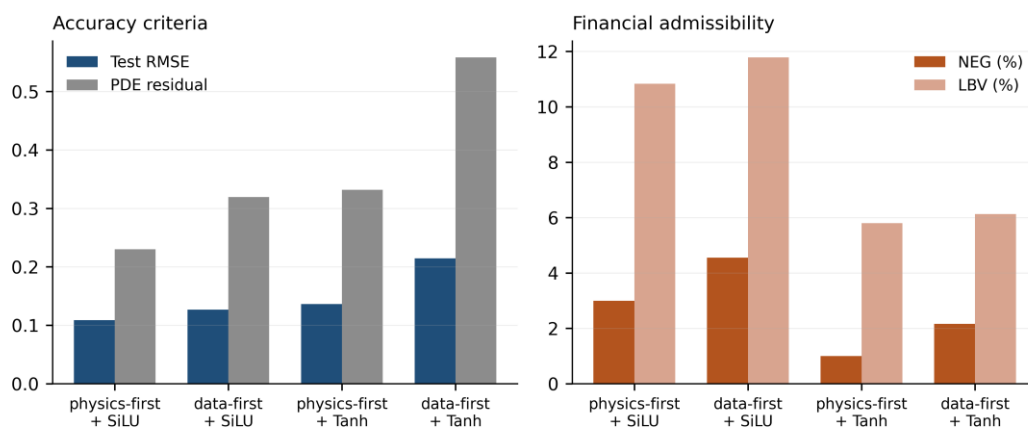
Model	Global	Near ATM	ITM	OTM	Short	Long	High Vol.
MLP baseline	0.0675	0.0875	0.0655	0.0669	0.1540	0.0573	0.0644
Hybrid PINN, data-first	0.0756	0.0831	0.0820	0.0659	0.1261	0.0793	0.0912
DeepONet-style surrogate	0.1477	0.1912	0.1405	0.1499	0.3579	0.1221	0.1132
Transformer-inspired pricer	0.1980	0.2112	0.2170	0.1712	0.2826	0.1894	0.1944
Standard PINN, Smax=400	0.2158	0.2744	0.2550	0.1550	0.0966	0.2881	0.2917
DGM, Smax=400, one update	0.3160	0.3466	0.3360	0.2848	0.2632	0.3804	0.3456
Feynman–Kac stochastic-target pricer	0.4358	0.4520	0.4501	0.4145	0.6853	0.4184	0.4126

4.11 Directional Hybrid Fine-Tuning: A Factorial Control

The submitted version compared two directional fine-tuning paths, data-first and physics-first. As the reviewer noted, that comparison confounded the training order with the activation function, since the data-first configuration inherited the SiLU activation of its MLP source and the physics-first configuration the Tanh activation of its PINN source. We therefore implemented the complete two-by-two control, crossing the training order with the activation function, using seed-matched source checkpoints in all four cells.

Table 18 shows descriptive effects for both factors, so the submitted comparison cannot be interpreted as an initialization-only ablation. Averaged across activations, physics-first ordering lowers price RMSE by 0.048, PDE MAE by 0.158, NEG by 1.35 percentage points, and LBV by 0.64 percentage points. Within each activation, physics-first also improves all four of these mean criteria. The best cell on the two accuracy criteria is physics-first with SiLU, with price RMSE 0.1086 and PDE MAE 0.2301.

The two factors improve accuracy and degrade admissibility



Three common seeds {42, 123, 2025}. Cells ordered by RMSE. Source: Created by the authors based on the experimental results.

Fig. 6. The two-by-two factorial control. Left: accuracy criteria. Right: financial admissibility. The activation contrast produces different orderings on the two panels. Source: Created by the authors based on the experimental results.

The activation factor exhibits a different pattern. Averaged across training orders, replacing SiLU with Tanh increases price RMSE by 0.058 and PDE MAE by 0.170, but decreases NEG by 2.19 percentage points and LBV by 5.35 percentage points. Under physics-first ordering, Tanh reduces the negative-price frequency from 2.99% to 1.00% and the lower-bound violation frequency from 10.84% to 5.80%, while degrading both the price error and the residual. Under data-first ordering the same substitution reduces NEG from 4.55% to 2.16% and LBV from 11.79% to 6.12%, again at the cost of both accuracy criteria.

The factorial control therefore yields two distinct conclusions. Physics-first ordering is favorable across the reported mean criteria under both activations, whereas activation presents an accuracy–admissibility trade-off: SiLU is preferable for price and residual errors, and Tanh for NEG and LBV. Accordingly, the preferred hybrid cell depends on the evaluation priority: physics-first SiLU for price and PDE accuracy, or physics-first Tanh for the two reported admissibility frequencies.

The data-first cell is retained as the primary benchmark entry because it corresponds to the submitted configuration and because it is the cell repeated with the five main seeds. Hybridization should nevertheless be interpreted in relation to both the starting configuration and the criterion of interest.

Table 18. Factorial control on the Hybrid configuration: training order crossed with activation function, over the matched control seeds {42, 123, 2025}. Cells are ordered by price RMSE.

Order	Activation	Price MAE	Price RMSE	PDE MAE	NEG (%)	LBV (%)	Training time (s)
Physics-first	SiLU	0.0597	0.1086	0.2301	2.99	10.84	166.3
Data-first	SiLU	0.0740	0.1268	0.3195	4.55	11.79	165.9
Physics-first	Tanh	0.0895	0.1364	0.3318	1.00	5.80	139.4
Data-first	Tanh	0.1563	0.2145	0.5583	2.16	6.12	139.4
<i>Main effect of order (physics-first – data-first)</i>		–0.041	–0.048	–0.158	–1.35	–0.64	0.2
<i>Main effect of activation (Tanh – SiLU)</i>		+0.056	+0.058	+0.170	–2.19	–5.35	–26.7

4.12 Monte Carlo Target-Noise and Path-Count Diagnostic

The Feynman–Kac stochastic-target configuration is trained using Monte Carlo estimates of discounted terminal payoffs rather than analytical Black–Scholes prices. Its performance must therefore be interpreted jointly with the sampling noise of the training targets. Two complementary experiments are reported below. The first measures the target error directly on a fixed set of 4,096 inputs, independently of neural-network approximation. The second trains the same network configuration with $M \in \{16, 64, 256\}$ antithetic paths per input.

Table 19 reports the distribution of the Monte Carlo target error relative to the analytical price. Increasing the number of paths progressively reduces all dispersion measures. The target MAE decreases from 8.5040 at $M = 16$ to 4.5077 at $M = 64$ and 2.2840 at $M = 256$, corresponding to an overall reduction of approximately 73.1%. The standard deviation decreases from 16.4806 to 4.5571 over the same range.

The reduction in target dispersion is broadly consistent with the $M^{-1/2}$ convergence expected for a Monte Carlo sample mean. In particular, the ratio of the standard deviations at $M = 64$ and $M = 256$ is 1.994, close to the theoretical factor of 2. The corresponding ratio between $M = 16$ and $M = 64$ is 1.814, showing that the finite-sample behavior does not follow the asymptotic rate exactly at every path count.

The observed biases are small relative to the dispersion of the target errors. Using the empirical standard deviation divided by $\sqrt{4096}$ as the standard error of the mean target error, the absolute bias remains below 1.8 standard errors for all three path counts. The finite-sample results are therefore compatible

with the theoretical unbiasedness of the Monte Carlo estimator, but they should not be interpreted as empirical proof of exact zero bias.

Table 19. Monte Carlo target-noise sensitivity on a fixed diagnostic sample of 4,096 inputs. The quantities compare antithetic Monte Carlo targets directly with analytical Black–Scholes prices and are independent of neural-network predictions. q0.95 denotes the 95th percentile of the absolute target error.

Paths M	Bias	Std. deviation	MAE	RMSE	Median AE	q0.95 AE
16	-0.4470	16.4806	8.5040	16.4846	3.3179	34.3351
64	-0.2499	9.0875	4.5077	9.0898	1.6164	18.6609
256	-0.0459	4.5571	2.2840	4.5568	0.8292	9.5411

To determine how this target noise propagates to the learned pricing surface, the Feynman–Kac network was trained separately with $M \in \{16, 64, 256\}$. Table 20 reports results over the matched control seeds $\{42, 123, 2025\}$. The $M = 64$ row in this table is intentionally restricted to the same three seeds as the two path controls; the five-seed estimate for the primary $M = 64$ configuration is reported separately in the global benchmark.

All five reported mean errors improve as the path count increases. Moreover, price MAE is non-increasing with M for each of the three individual control seeds, while PDE MAE decreases for every seed. From $M = 16$ to $M = 256$, the aggregate price MAE decreases by approximately 38.6%, whereas PDE MAE decreases by approximately 64.1%. Delta, Gamma, and Vega errors decrease by approximately 51.0%, 54.1%, and 44.2%, respectively. The smoother targets produced by the larger simulation budget therefore improve not only pointwise pricing but also the differential and residual properties of the learned surface.

The improvement in model accuracy is nevertheless smaller than the reduction in raw target noise. Increasing the path count from 16 to 256 reduces target MAE by a factor of approximately 3.72, whereas model price MAE decreases by a factor of approximately 1.63. Consequently, finite-path target noise accounts for part of the performance limitation, but it does not by itself determine the final approximation error. Network approximation, finite optimization, and the interaction between noisy renewed targets and model fitting remain contributing factors.

The training histories provide complementary evidence. The mean training loss over the final 20 recorded epochs is approximately 0.262, 0.0658, and 0.0166 for $M = 16$, $M = 64$, and $M = 256$, respectively. The successive reductions are close to a factor of four, consistent with a substantial $1/M$ contribution from Monte Carlo variance under a squared-error objective. Because the observed training loss also includes network approximation and optimization error, this pattern should not be interpreted as a pure measurement of target variance.

These results establish path count as a consequential component of the Feynman–Kac learning protocol. The $M = 64$ configuration remains the primary benchmark implementation, preserving the submitted training design, whereas $M = 16$ and $M = 256$ quantify its sensitivity to lower and higher simulation budgets. The comparison does not establish an optimal path count, since increasing M also raises the simulation cost of every generated training target.

Table 20. Feynman–Kac path-count sensitivity over the matched control seeds $\{42, 123, 2025\}$. Values are means $\pm$ standard deviations. All configurations use antithetic sampling and differ only in the number of Monte Carlo paths used to generate each renewed training target. Lower values indicate better performance.

Paths M	Price MAE	PDE MAE	Delta MAE	Gamma MAE	Vega MAE
16	0.517 $\pm$ 0.083	3.186 $\pm$ 0.254	0.0243 $\pm$ 0.0023	0.00213 $\pm$ 0.00011	3.135 $\pm$ 0.563
64	0.391 $\pm$ 0.081	1.734 $\pm$ 0.206	0.0161 $\pm$ 0.0028	0.00129 $\pm$ 0.00012	2.392 $\pm$ 0.531
256	0.317 $\pm$ 0.036	1.145 $\pm$ 0.072	0.0119 $\pm$ 0.0012	0.00098 $\pm$ 0.00003	1.750 $\pm$ 0.136

4.13 Implications for Benchmark Design and Model Selection

The preceding subsections establish criterion-dependent behavior case by case. Table 21 complements those comparisons with selected pairwise Spearman rank correlations across the seven primary configurations. With only $n = 7$ configurations, the coefficients are descriptive summaries of the realized ranking and no inferential claim is made.

Four conclusions emerge from the combined experiments.

First, price MAE and PDE residual produce markedly different rankings. Their Spearman correlation is only $\rho = +0.14$ in these seven configurations. Sections IV-B and IV-E show the underlying cases: low price error can coexist with moderate or heavy-tailed residuals, whereas the lowest residual does not imply the lowest price error.

Second, the residual ranking is more closely aligned with the negative-price rate than the price-error ranking is in this sample ($\rho = +0.82$ versus $+0.64$). Its correlation with the thresholded convexity-violation rate is $+0.89$. This latter association is consistent with the appearance of V_{SS} in the Black–Scholes operator, but the small number of configurations and the multiple components of admissibility preclude treating the residual as a substitute for direct financial checks.

Third, greater wall-clock cost is associated more strongly with lower PDE residual than with lower price MAE across the primary configurations: $\rho = -0.75$ and -0.04 , respectively. The within-DGM control reaches a similar descriptive conclusion: at $S_{max} = 200$, an approximately elevenfold increase in mean training time reduces the residual by 43.7% but the price error by only 3.8%. Optimization effort and problem specification are therefore separate benchmark dimensions.

Fourth, the data-first Hybrid PINN is competitive on both accuracy axes and has the lowest NEG, LBV, and MVR frequencies among the primary configurations, at approximately twice the end-to-end training cost of the MLP baseline once the source checkpoint is included. The factorial control in Section IV-K further separates this result: physics-first ordering improves the reported mean criteria under both activations, whereas SiLU improves price and residual accuracy at the cost of higher NEG and LBV frequencies relative to Tanh.

Three further observations bound the interpretation of these results.

The computational PDE domain is not a secondary implementation choice for residual-trained models. Moving the upper boundary from $S_{max} = 200$ to $S_{max} = 400$ changes both pricing and residual performance by large factors even though the supervised information and the test distribution are unchanged, and the dependence is not monotonic: enlarging the domain further to $S_{max} = 800$ degrades the PINN price MAE from 0.212 ± 0.018 to 0.402 ± 0.066 under a fixed sampling protocol.

Model selection is itself a degree of freedom. Two defensible checkpoint criteria select epochs up to 1,100 apart on the same DGM trajectory and up to 700 apart on the same PINN trajectory (Section IV-G). Reporting a single figure without naming the selection criterion under-specifies the experiment.

Finally, five-seed estimates are reported for the primary benchmark and three-seed estimates for targeted controls. The resulting standard deviations show that some apparent differences are small relative to optimization variability. The controls are consequently used to qualify mechanisms and protocol sensitivity, rather than to expand the primary model ranking.

The benchmark should therefore be read as a diagnostic comparison of realized configurations rather than as an intrinsic ranking of neural architecture families. Taken together, the results show that reliable benchmarking of neural approaches to financial PDE approximation requires attention not only to what is measured, but also to how the numerical problem is posed, how the training signal is generated, how much optimization effort is allocated to solving it, and how the retained model is selected.

Table 21. Rank agreement between evaluation criteria across the seven primary configurations. Spearman rank correlations, reported descriptively ($n = 7$).

Criterion	Criterion	p
Price MAE	PDE residual	+0.14
Price MAE	Training time	−0.04
Price MAE	Negative-price rate	+0.64
Price MAE	Residual tail ratio	−0.61
PDE residual	Training time	−0.75
PDE residual	Negative-price rate	+0.82
PDE residual	Convexity violation	+0.89

5 Conclusion and Limitations

This study developed a multicriteria diagnostic benchmark for neural approximation of the Black–Scholes PDE. Seven primary configurations were evaluated over five optimization seeds using a common frozen test set, while targeted three-seed controls examined checkpoint selection, computational domain, collocation density, optimization effort, loss scaling, Hybrid training order and activation, and Monte Carlo path allocation. The evaluation extended beyond aggregate pricing error to include Greek accuracy, PDE consistency, residual-tail behavior, financial admissibility, regime-wise performance, computational cost, and between-seed variability.

The primary benchmark confirms that no configuration dominates all evaluation criteria. The supervised MLP attains the lowest global price MAE, 0.0675 ± 0.0057 , whereas the Standard PINN with $S_{max} = 400$ obtains the lowest PDE-residual MAE among the seven primary configurations, 0.2648 ± 0.0432 . The data-first Hybrid PINN is competitive on both axes, with a price MAE of 0.0756 ± 0.0053 and a PDE MAE of 0.3249 ± 0.0153 , while also producing the lowest Delta, Gamma, and Vega errors and the lowest NEG, LBV, and thresholded monotonicity-violation frequencies among the primary configurations. These results demonstrate that price accuracy, differential accuracy, equation consistency, and financial admissibility describe related but non-equivalent properties of a learned pricing surface.

Reporting violation severity alongside frequency further qualifies the model comparison. For example, the Standard PINN and the MLP exhibit similar negative-price frequencies, but the corresponding violation magnitudes differ by approximately a factor of two. Conversely, similar lower-bound violation rates can coexist with markedly different conditional severities. Upper-bound violations are absent for every primary configuration, but none of the remaining admissibility criteria produces a universally dominant model. Frequency-only reporting would therefore conceal economically relevant differences in the severity of localized failures.

The control experiments show that residual-based results depend materially on the computational and optimization protocol. Moving the artificial boundary from $S_{max} = 200$ to $S_{max} = 400$ reduces the PINN price MAE from 2.596 ± 0.239 to 0.212 ± 0.018 and the DGM price MAE from 2.415 ± 0.052 to 0.325 ± 0.026 on the matched control seeds. Further enlargement to $S_{max} = 800$ is not beneficial under the tested protocol: its price MAE increases to 0.402 ± 0.066 . Restoring the interior collocation density at $S_{max} = 800$ produces only a modest reduction to 0.388 ± 0.053 and does not recover the performance obtained at $S_{max} = 400$. Collocation dilution is therefore not supported as the sole explanation for the deterioration, although the density-controlled experiments are not computationally cost-matched.

Optimization effort is equally consequential. At $S_{max} = 400$, increasing the real-unit DGM allocation from one to 20 updates per epoch reduces its price MAE from 0.3254 ± 0.0264 to 0.1606 ± 0.0295 and its PDE MAE from 0.4539 ± 0.0466 to 0.1498 ± 0.0138 . Under this matched update allocation, DGM obtains a lower PDE residual than the Standard PINN on the same three seeds, 0.1498 against 0.2662. This improvement is accompanied by a substantial computational cost: mean DGM training time rises from 230.6 to 3022.4 seconds, compared with 979.1 seconds for the PINN. Moreover, the effect of standardized versus real-unit loss scaling changes with update allocation. The primary PINN–DGM

difference can consequently not be interpreted as an architecture effect alone, and epoch counts are insufficient indicators of optimization effort.

The completed 2×2 Hybrid control likewise refines the interpretation of hybridization. Crossing training order with activation shows that physics-first ordering improves the reported mean accuracy and admissibility criteria under both SiLU and Tanh. Physics-first SiLU provides the best price and PDE accuracy within the factorial control, whereas physics-first Tanh provides the lowest NEG and LBV frequencies. Activation therefore induces an accuracy–admissibility trade-off rather than a uniformly ordered effect. The data-first SiLU configuration remains the primary benchmark entry because it corresponds to the submitted configuration and was repeated over all five main seeds, but it should not be interpreted as the universally preferable Hybrid design.

Checkpoint selection and stochastic-target noise provide two additional sources of protocol dependence. Physics-aligned checkpoint selection produces a modest aggregate price–residual trade-off for PINN but does not uniformly improve the test residual across seeds. For DGM, it does not improve the aggregate test PDE residual, despite selecting the minimum recorded loss on its fixed physics-validation sample. The analytical and Monte Carlo criteria select identical Feynman–Kac checkpoints in the five observed trajectories. These findings support retaining the analytical-price checkpoint for the common benchmark while reporting signal-aligned checkpoints as paired sensitivity analyses. The Feynman–Kac results must also be interpreted in light of the finite-path target noise, which decreases substantially as the path count increases.

Several limitations delimit these conclusions. The benchmark compares realized training configurations rather than isolated architectures: learning signals, loss formulations, optimization procedures, parameter counts, and effective computational budgets remain heterogeneous. Although the main configurations use five seeds, most controls use only three and are interpreted descriptively; they do not provide exhaustive statistical evidence over the space of possible initializations and protocols. Analytical validation prices are used for primary checkpoint selection, meaning that the residual-trained configurations are physics-trained but not fully label-independent at the model-selection stage. The tested truncation boundaries, collocation allocations, loss scalings, and Hybrid designs are targeted controls rather than exhaustive hyperparameter searches.

The study is also restricted to a uniformly sampled, in-distribution Black–Scholes setting. The regime analyses identify localized behavior within that fixed domain and do not establish out-of-distribution robustness. Wall-clock measurements were obtained on a single NVIDIA Tesla T4 platform and should be interpreted as comparative measurements for this implementation rather than hardware-independent complexity estimates. Finally, conclusions obtained for a one-dimensional, analytically solvable PDE cannot be assumed to transfer unchanged to stochastic-volatility, local-volatility, higher-dimensional, or path-dependent pricing problems.

Future work should therefore extend the same diagnostic framework to more complex financial PDEs and contracts, with explicitly controlled optimization and computational budgets. Further priorities include adaptive truncation and collocation strategies, improved residual-loss balancing, variance-reduced stochastic supervision, constraint-preserving architectures, and external-domain evaluation. Within the scope of the present experiments, the principal conclusion remains that reliable evaluation of neural financial PDE solvers requires more than a single pricing metric. The learning signal, checkpoint criterion, computational domain, optimization effort, differential accuracy, financial admissibility, and computational cost must be reported jointly before differences between neural configurations can be interpreted.

DECLARATIONS

Ethics approval: Not applicable. This study is based on computational experiments using synthetically generated financial data and does not involve human participants, animals, or identifiable personal data.

Informed consent: Not applicable. No human participants were involved in this research.

Author contributions: Zakaria Elbayed: Conceptualization, methodology, software, investigation, formal analysis, experimental implementation, visualization, and writing – original draft. Abdelmjid Qadi El Idrissi: Supervision,

methodological guidance, validation, and writing – review and editing. Both authors reviewed and approved the final version of the manuscript.

Use of artificial intelligence: AI-assisted tools were used solely to support linguistic proofreading, grammatical correction, and improvements in the clarity and readability of selected passages of the manuscript. The authors reviewed and verified the resulting text and take full responsibility for the scientific content, methodology, analyses, results, and conclusions.

Funding: This research received no external funding.

Conflicts of interest: The authors declare that they have no conflict of interest.

Acknowledgements: The authors acknowledge the institutional support of the National School of Applied Sciences of Berrechid (ENSAB), Hassan First University, Morocco.

Data and code availability. The experimental code and the reproducibility material used in this study are permanently archived on Zenodo under the DOI 10.5281/zenodo.22539850 (<https://doi.org/10.5281/zenodo.22539850>), an identifier that always resolves to the most recent version of the record. The archive contains the orchestration notebook for the fifteen experimental batches, covering dataset construction, the SHA-256 dataset and protocol hashes, the training and evaluation pipeline, and the aggregation and figure-generation cells that produce the tables and figures reported in Section IV. The same notebook is additionally available in its original execution environment at <https://www.kaggle.com/code/zakariaelbayed/financial-pde-benchmark-black-scholes>, which provides the GPU runtime under which the campaign was executed. The Zenodo record is the citable and version-controlled reference; the Kaggle notebook is provided for convenience only.

REFERENCES

- [1] F. Black and M. Scholes, "The pricing of options and corporate liabilities," *Journal of Political Economy*, vol. 81, no. 3, pp. 637–654, 1973, doi: 10.1086/260062.
- [2] R. C. Merton, "Theory of rational option pricing," *The Bell Journal of Economics and Management Science*, vol. 4, no. 1, pp. 141–183, 1973, doi: 10.2307/3003143.
- [3] S. L. Heston, "A closed-form solution for options with stochastic volatility with applications to bond and currency options," *The Review of Financial Studies*, vol. 6, no. 2, pp. 327–343, 1993, doi: 10.1093/rfs/6.2.327.
- [4] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019, doi: 10.1016/j.jcp.2018.10.045.
- [5] J. Sirignano and K. Spiliopoulos, "DGM: A deep learning algorithm for solving partial differential equations," *Journal of Computational Physics*, vol. 375, pp. 1339–1364, 2018, doi: 10.1016/j.jcp.2018.08.029.
- [6] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis, "Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators," *Nature Machine Intelligence*, vol. 3, pp. 218–229, 2021, doi: 10.1038/s42256-021-00302-5.
- [7] J. Han, A. Jentzen, and W. E, "Solving high-dimensional partial differential equations using deep learning," *Proceedings of the National Academy of Sciences*, vol. 115, no. 34, pp. 8505–8510, 2018, doi: 10.1073/pnas.1718942115.
- [8] A. Sharma and C. Kumar Verma, "Non-parametric insights in option pricing: A systematic review of theory, implementation and future directions," *Artificial Intelligence Review*, vol. 58, 2025, Art. no. 252, doi: 10.1007/s10462-025-11249-z.
- [9] J. M. Hutchinson, A. W. Lo, and T. Poggio, "A nonparametric approach to pricing and hedging derivative securities via learning networks," *The Journal of Finance*, vol. 49, no. 3, pp. 851–889, 1994, doi: 10.1111/j.1540-6261.1994.tb00081.x.
- [10] K. Glau and L. Wunderlich, "The deep parametric PDE method and applications to option pricing," *Applied Mathematics and Computation*, vol. 432, 2022, Art. no. 127355, doi: 10.1016/j.amc.2022.127355.
- [11] A. Ratku and D. Neumann, "Derivatives of feed-forward neural networks and their application in real-time market risk management," *OR Spectrum*, vol. 44, no. 3, pp. 947–965, 2022, doi: 10.1007/s00291-022-00672-1.
- [12] S. Cuomo, V. Schiano Di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli, "Scientific machine learning through physics-informed neural networks: Where we are and what's next," *Journal of Scientific Computing*, vol. 92, 2022, Art. no. 88, doi: 10.1007/s10915-022-01939-z.
- [13] X. Wang, J. Li, and J. Li, "A deep learning based numerical PDE method for option pricing," *Computational Economics*, vol. 62, pp. 149–164, 2023, doi: 10.1007/s10614-022-10279-x.
- [14] F. Gatta, V. Schiano Di Cola, F. Giampaolo, F. Piccialli, and S. Cuomo, "Meshless methods for American option pricing through physics-informed neural networks," *Engineering Analysis with Boundary Elements*, vol. 151, pp. 68–82, 2023, doi: 10.1016/j.engabound.2023.02.040.
- [15] D. Hainaut and A. Casas, "Option pricing in the Heston model with physics inspired neural networks," *Annals of Finance*, vol. 20, pp. 353–376, 2024, doi: 10.1007/s10436-024-00452-7.

- [16] N. K. Pande, P. Pasricha, A. Kumar, and A. K. Gupta, "European option pricing in regime switching framework via physics-informed residual learning," *Expert Systems with Applications*, vol. 288, 2025, Art. no. 128226, doi: 10.1016/j.eswa.2025.128226.
- [17] Y. Chen, X. Lü, H. Tian, and R. H. Li, "Physics-informed neural network for barrier option pricing in coupled financial quantitative system with varying interest rate and volatility," *Engineering Analysis with Boundary Elements*, vol. 180, 2025, Art. no. 106457, doi: 10.1016/j.enganabound.2025.106457.
- [18] S. Lee, J. Huh, and S. Jeong, "DeepONet-based surrogate modeling for bond option pricing," *AIMS Mathematics*, vol. 11, no. 3, pp. 5853–5896, 2026, doi: 10.3934/math.2026242.
- [19] B. A. Høverstad, M. Ristad, and L. K. Sagen, "Transformers vs. LSTM-MLP for option pricing," *AIMS Mathematics*, vol. 10, no. 11, pp. 27152–27170, 2025, doi: 10.3934/math.20251193.
- [20] J. F. Moreno T., C. Zapata, and D. Aragon, "Neural networks and the nonlinear Feynman–Kac theorem applied to financial options pricing," *SN Computer Science*, vol. 5, 2024, Art. no. 697, doi: 10.1007/s42979-024-02931-0.
- [21] R. Assabumrungrat, K. Minami, and M. Hirano, "Error analysis of option pricing via deep PDE solvers: Empirical study," in *2024 16th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI)*. IEEE, 2024, pp. 329–336, doi: 10.1109/IIAI-AAI63651.2024.00068.
- [22] S. N. Cohen, C. Reisinger, and S. Wang, "Arbitrage-free Neural-SDE market models," *Applied Mathematical Finance*, vol. 30, no. 1, pp. 1–46, 2023, doi: 10.1080/1350486X.2023.2257217.