

Prices or implied volatilities? Choosing the loss function in machine learning option pricing

Panayiotis C. Andreou^{*†}

Chulwoo Han[‡]

Nan Li[†]

This version: September 2026

Abstract

This paper investigates the importance of the loss function in machine learning option pricing by asking whether a model should be trained on pricing errors or on implied-volatility errors. Both choices appear in the literature, usually without justification, and a meta-analysis of the field shows an increasing share of studies training on implied-volatility errors, whilst pricing accuracy remains the criterion on which their models are judged. We answer the question empirically by estimating the same machine learning model under two loss functions that differ only in the space where the error is measured, using 8.67 million S&P 500 index-option observations priced out of sample from 1997 through 2025. The price-space loss lowers pooled pricing RMSE by 25.3%, nearly twice what the network itself contributes over the parametric benchmark, and attains a lower error in every sample year and on the implied-volatility criterion that its rival optimizes. Conditional predictive ability tests show that the advantage varies with contract vega but not with market volatility, so it follows from the loss function rather than the sample period. The training space is therefore a first-order design choice, and researchers who price options with machine learning are better off measuring the error where the model will be judged.

Keywords: option pricing, machine learning, loss function, optimization target, vega weighting, out-of-sample pricing accuracy

JEL Classification: C45, C52, C58, G12, G13

^{*}Cyprus University of Technology, Department of Finance, Accounting and Management Science, Archbishop Kyprianou 30, Limassol 3603, Cyprus.

[†]Durham University Business School, Mill Hill Lane, Durham, DH1 3LB, United Kingdom.

[‡]Department of Economics, Sungkyunkwan University, Jongno-gu, Seoul, South Korea.

E-mail addresses: panayiotis.andreou@cut.ac.cy, chulwoo.han@skku.edu, nanli.2@durham.ac.uk

Acknowledgements: Panayiotis C. Andreou dedicates this paper to the memory of Spiros H. Martzoukos, who passed away on 28 July 2026. He and Chris Charalambous supervised the doctoral work at the University of Cyprus from which the GPF framework studied here emerged (Andreou, Charalambous and Martzoukos, 2010). He was a meticulous scholar with a deep knowledge of option theory and a kind and generous teacher to whom Andreou remains permanently indebted.

1 Introduction

Over the last three decades, machine learning has moved from a curiosity to a toolbox of empirical option pricing. What began with a handful of neural network applications in the early 1990s has grown into a large and rapidly expanding literature that spans pricing, hedging, calibration, and the modeling of implied volatility (Ruf and Wang, 2020). It now forms one of the most active branches of the broader machine learning turn in asset pricing (Gu et al., 2020; Kelly and Xiu, 2023). For all this activity, the field has largely left unexamined a question that arises in every implementation. Should the model be trained in *option price space*, where it fits market option prices, or in *implied volatility space*, where it fits the implied volatilities behind them? The choice is made both ways in the literature, usually without discussion.

The two candidate loss functions weight the same contracts differently. BS (Black and Scholes, 1973) maps implied volatility into price through a nonlinear transformation whose slope, vega, varies widely across the traded cross-section. In our S&P 500 sample, for instance, vega ranges from below 0.01 to approximately 25 index points per volatility point, with a median near 3.4, so a one-percentage-point volatility error costs a few cents on some contracts and more than twenty dollars on others. A loss function defined on volatilities treats these errors as equally important. A loss defined on prices weights each contract by the size of the resulting pricing error, so a contract on which the volatility error translates into 20 index points receives far more weight in estimation than one on which it translates into a few cents. Whether that difference in weighting matters for out-of-sample accuracy, however, is not something the algebra settles. If the machine learning model class were flexible enough to fit every contract exactly, or if vega were constant across contracts, the two targets would select the same model. Neither condition holds in practice, but theory alone does not indicate which target then produces the more accurate prices on unseen data. To our knowledge, the literature has not made it the object of a controlled test; ergo, this paper asks whether it matters out

of sample and answers empirically.

The literature’s objectives have instead evolved along other margins through several recognizable phases. The founding studies asked whether flexible approximators could price at all. [Hutchinson et al. \(1994\)](#), for instance, showed that learning networks trained on prices can approximate and hedge BS-type valuation, and [Yao et al. \(2000\)](#) applied networks to option price forecasting.² A second wave built economic structure into the learner. For instance, [Garcia and Gençay \(2000\)](#) imposed a homogeneity hint and improved both pricing and hedging, whilst [Andreou et al. \(2008\)](#) and [Andreou et al. \(2010\)](#) developed hybrid and semi-parametric designs in which a parametric model supplies pricing structure and the network supplies flexibility, the family of designs this paper builds on.³

The modern phase has diversified the objectives whilst keeping that broad architecture. For instance, [Yang et al. \(2017\)](#) design gated networks whose predictions respect rationality restrictions; [Liu et al. \(2019\)](#) embed no-arbitrage structure and target calibration speed; [Ackerer et al. \(2020\)](#) fit arbitrage-free deep representations of the implied volatility surface; [Büchner and Kelly \(2022\)](#) carry the factor approach to option returns; and [Almeida et al. \(2023\)](#) train networks on the implied-volatility errors of parametric models to correct their misspecification. Across this literature, the modal objectives are, *inter alia*, pricing accuracy against parametric benchmarks, volatility and surface forecasting, hedging performance, recovery of risk-neutral densities, American-style exercise, and fast calibration, pursued by varying architectures, feature sets, parametric

² Other early contributions in this vein include [Anders et al. \(1998\)](#), who selected a parsimonious network by statistical specification tests and beat Black–Scholes out of sample on DAX calls; [Carelli et al. \(2000\)](#), who fixed the architecture by profile methods and priced off the fitted volatility surface; [Raberto et al. \(2000\)](#), who trained a perceptron on fat-tailed simulated prices; [Zapart \(2002\)](#), who paired wavelet pre-processing with networks under stochastic volatility; and [Amilon \(2003\)](#), who benchmarked networks against Black–Scholes in both pricing and delta hedging.

³ Similar ideas appear in the non-parametric literature of the same period. [Fan and Mancini \(2009\)](#) used a parametric model to guide the fit and let a non-parametric term absorb its pricing errors. [Aït-Sahalia and Duarte \(2003\)](#) and [Birke and Pilz \(2009\)](#) imposed the monotonicity and convexity conditions implied by no arbitrage on the estimator itself, whilst [Kohler et al. \(2010\)](#) put networks inside the backward dynamic programming recursion to price high-dimensional American options.

components, economic constraints, *et cetera*.

What the literature has varied deliberately, in other words, is nearly everything except the space in which the training loss is computed. Few of these studies explain why they train in one space rather than the other. The choice usually follows from how the model is built. A model that outputs prices is trained on price errors. A model built around the implied-volatility surface is trained on implied-volatility errors, and when a price is needed, it is obtained by inserting the fitted volatility into the BS formula. There is already evidence that this choice matters. The closest forerunners of this paper are [Christoffersen and Jacobs \(2004\)](#) and [Bams et al. \(2009\)](#), who establish that, for parametric models calibrated by nonlinear least squares, the estimation loss should match the evaluation loss and that mismatching the two is economically costly. That principle has received far less attention in the machine learning field, where, under limited capacity of the machine learning model, the loss determines how approximation accuracy is allocated across millions of contracts. Ergo, the intent of this study is to isolate that choice and measure what it is worth.

In this study, we document this gap with a qualitative meta-analysis of 268 studies published between 1994 and mid 2026 that apply machine learning to options. Of these, 117 price options face the choice of training space; most are trained on prices, a burgeoning minority is trained on implied volatilities, whilst none of the studies we review compares the two with the same model on the same data.

To fill the gap, our investigation holds everything fixed except for the loss function. Within the Generalized Parametric Function (GPF) framework of [Andreou et al. \(2010\)](#), we estimate a single model architecture under two alternative loss functions: one defined on pricing errors and the other on implied-volatility errors. Under GPF^{OPS} the network output is converted to a price through an embedded BS model before the error is evaluated, so that estimation minimizes squared pricing errors. Under GPF^{IVS} estimation minimizes squared implied-volatility errors measured against market implied

volatilities, and prices are calculated through an external BS model. Both are estimated and evaluated on the same 8.67 million S&P 500 index-option observations, priced out of sample from January 1997 through August 2025 under a strict 10-2-1 month rolling protocol that yields 344 monthly out of sample test blocks. Nothing else differs between them, ergo, any difference in the results comes from the loss function alone.

We evaluate pricing performance on three metrics whose vega weights rise from zero (IVRMSE) through squared (RMSE) to cubed (VRMSE) and test the differences with Diebold–Mariano statistics clustered by trading date. These statistics compare average accuracy, and an average can conceal variation across market conditions, so we complement them with the conditional predictive ability tests of [Giacomini and White \(2006\)](#). The tests ask whether the accuracy difference between the two models varies with the information available when the forecast is made. We sort trading days into ten states by the previous day’s implied volatility and contracts into ten states by their vega at the forecast date, and within each state, we measure the difference in mean squared pricing error between the two models. Two hypotheses are then tested, namely that the two models are equally accurate in every state and that the advantage of one over the other is the same size in every state. The first asks whether the ranking holds throughout. The second bears directly on the mechanism, since an advantage that comes from the vega weighting should vary with the vega of the contract being priced rather than with the level of market volatility.

Training on pricing errors produces more accurate out-of-sample prices, and the margin is large. GPF^{OPS} reduces pooled out-of-sample pricing RMSE by 25.3% relative to GPF^{IVS} (26.2% for calls and 24.5% for puts), with rejections at the 1% level throughout, and it posts the lower RMSE in every one of the 29 calendar years, significantly in all but one. For perspective, the gain from changing the loss function exceeds the 13.5% improvement that introducing the network in volatility space delivers over the parametric benchmark itself. Within this framework, the choice of training space contributes

more to pricing accuracy than the machine learner *per se*.

Where the advantage comes from is a separate question. The two estimations share the network, the inputs, and the sample, so greater flexibility cannot be the answer. What they do not share is the weight the loss places on each volatility error. To first order, a pricing error is a volatility error multiplied by vega, so the pricing loss weights volatility errors by vega, whilst the volatility loss weights every contract equally. If this weighting is the source, the advantage should be larger the more weight the evaluation criterion gives to vega, and it should be concentrated on high-vega contracts. Both patterns are observed in the data. The advantage is smallest on the criterion that gives vega the least weight and largest on the criterion that gives it the most: 7.4% on the implied-volatility RMSE, 25.3% on the pricing RMSE, and 31.5% on the vega-weighted RMSE. Along moneyness, the reduction in pricing RMSE for calls rises from 1.9% deep out of the money, where vega is smallest, to 44.0% deep in the money. Along maturity, it rises from 10.9% for calls and 5.9% for puts with fewer than 60 days to maturity, where vega is small, to 36.1% and 34.5% beyond 120 days, where vega is large. The Giacomini–White tests point in the same direction. When contracts are sorted by vega, the advantage differs across the deciles at the 1% level; when trading days are sorted by the previous day’s implied volatility, no such variation is found for calls or for the pooled sample. The advantage moves with the vega of the contract being priced, not with the volatility of the market.

The paper makes two contributions. First, it informs the large-scale comparison that the literature has lacked. Recent studies continue to refine machine learning option pricing along other dimensions; they propose new network architectures for valuing market options (*e.g.*, [Vinje et al., 2025](#)), impose no-arbitrage restrictions on network outputs (*e.g.*, [Cao et al., 2021](#)), and use network approximations to make structural models orders of magnitude faster to calibrate and estimate (*e.g.*, [Horvath et al., 2021](#); [Chen et al., 2026](#)). None of this recent work examines the space in which its loss function

is defined. The paper thereby extends the loss-function principle of [Christoffersen and Jacobs \(2004\)](#) and [Bams et al. \(2009\)](#) from the calibration of low-dimensional parametric models to the training of high-dimensional learners.

The paper also offers two reasons why the choice should matter. Vega measures how much an option’s price moves when its volatility changes, so an error in the fitted volatility produces a pricing error of roughly vega times its size. The price-space loss, therefore, weights each volatility error by its price consequence, whilst the volatility-space loss treats all volatility errors alike. And because implied volatilities are inverted from prices that are themselves recorded with error ([Andersen et al., 2021](#)), the price-space weighting also places the least weight on the volatilities measured with the most noise, as generalized least squares would. Giacomini–White conditional predictive ability tests find that the advantage varies with the vega of the contract being priced but not with the volatility of the market, a pattern that the weighting argument implies.

Second, the evidence in this study rests on a longer sample than the literature has brought to this question. What limits the existing evidence is not the number of observations; [Ulrich et al. \(2023\)](#) construct implied volatility surfaces from 388 million option price observations, of which 14.3 million are on the S&P 500 index and the remainder on 499 individual stocks, and [Vinje et al. \(2025\)](#) value more than 11 million S&P 500 call options. It is, however, the length of the evaluation window. Among the 67 studies in our survey that price index options, the median sample covers five years, and none is evaluated out of sample over a period approaching 29 calendar years. Therefore, the ranking reported here is the first to be established across the calm markets of the mid-2000s and the crises of 2008 and 2020 alike.⁴

The remainder of the paper proceeds as follows. Section 2 presents the qualitative meta-analysis of the literature’s loss functions. Section 3 develops the two specifications

⁴ [Ulrich et al. \(2023\)](#) and [Vinje et al. \(2025\)](#) use more option observations than we do, over samples of 16 and 12 years. Liquidity screens are also applied unevenly in this literature. We require positive open interest and positive trading volume, whereas neither of those two studies removes contracts on which no open interest is recorded.

and the evaluation metrics. Section 4 describes the data and the sample. Section 5 reports the pooled, yearly, and cross-sectional results, and Section 6 examines the source of the advantage. Section 7 reports the robustness exercises, and Section 8 concludes.

2 Qualitative meta-analysis of optimization criteria

How the literature has handled the choice of training space can be measured. To do so, we assembled the body of work that this question concerns and read it with the question in mind. We searched for applications of machine learning and statistical learning to option valuation, hedging, and implied volatility published between 1994 and 2026 in journals rated 2 and above in the Chartered Association of Business Schools Academic Journal Guide, together with the leading field journals in mathematical finance, statistics, and operations research that the Guide does not cover. We read the full text of the 268 studies identified.⁵ We review these studies along three dimensions: the vocabulary with which they describe their work, the space in which their estimation criterion is defined, and the error criteria they use to estimate and judge fit.

Figure 1 depicts the cloud of terms that occur most often across the full text of these studies, with term sizes proportional to the number of studies using each term. The vocabulary is dominated by what is modeled and with which tools, including “option price” and “option pricing” (212 studies each), “neural network” (180), “implied volatility” (174), “call option” (171), and “stochastic volatility” (149), followed by architectures, estimation tools, and market descriptors. Terms naming the estimation objective are present but far less prominent, and the terms that would name the space in which the objective is measured are rarer still, with “loss function” occurring in 117 of the 268 studies, “objective function” in 79, and “pricing error” in 88, whilst a phrase denominating the error in implied volatilities occurs in 16 and a vega weighting in 5.

⁵ See the Internet Appendix for the 268 studies with the classification described below recorded for each.

The literature describes at length what it models and how it minimizes; where the error is measured is seldom mentioned.

[Insert Figure 1, here]

Panel (a) of Figure 2 counts the 117 studies whose loss function, as stated in their methodology sections, operates in option-price space, implied-volatility space, or both. The remaining 151 pursue objectives that lie in neither space, among them volatility forecasting, risk-neutral density estimation, hedging-error minimization, and entropic valuation, and are set aside because the question of the training space does not arise for them. The literature has grown rapidly, from three classified studies in 1994 to 1998, to 34 in 2019, to 2023, and a further 42 in the three years since; yet the growth has produced no convention on the training space. Price-space optimization is the modal choice in every period; implied-volatility optimization, present since the volatility-surface work of the early 2000s, rises from 26% of studies before 2012 to 35% from 2021 onward with the turn toward surface modeling, deep calibration, and residual correction; and only eight of the 117 classified studies estimate in both spaces, in every case for different model components rather than as a comparison, so that none confronts the two criteria with the same machine learning model on the same market data. There is, in short, a paucity of evidence on the one design choice every implementation must make. The closest such comparison we find is a parametric one involving deterministic volatility functions estimated by ordinary least squares on implied volatilities *vis-à-vis* nonlinear least squares on prices (Christoffersen and Jacobs, 2004; Andreou et al., 2014).

Panel (b) reports the error criteria that these studies define and report in the same sections and suggests why the choice of space draws so little attention. The criteria are standardized and are stated in generic units, which leaves the space in which they are evaluated implicit. Squared-error loss functions govern estimation in 81% of the classified studies, and the share rises to 89% when the studies that minimize the squared residuals of a pricing partial differential equation or a backward stochastic differential equation are

also counted. Fit is judged almost entirely by generic measures, with RMSE appearing in 44% of the studies, MSE in 43%, mean absolute errors in 26%, and percentage errors in 23%, whilst an error measure explicitly denominated in implied volatilities appears in six studies, and a vega-weighted criterion in one. Reporting the same model in both price and implied-volatility units, as this paper does, is therefore not the convention that the literature has settled on.

[Insert Figure 2, here]

A final dimension of the meta-analysis concerns the samples themselves. Of the 117 studies classified by loss-function space in Panel (a) of Figure 2, 83 are based on market option data, and 67 of these examine price index options, the setting examined in this paper. Their median sample covers five years, and 51 of the 67 use ten years or less. The longest spans 28 years, but its out-of-sample test holds out alternate strikes within the same trading day rather than forecasting forward, and the longest forward-looking evaluation in the survey covers 24 years. None, therefore, evaluates out of sample over a period approaching the 29 calendar years examined here. Observation counts are less comparable because studies differ in what they count and in how they screen. The median reported count is roughly 71,000, and two studies report more than the 8.67 million observations used here: one with 14.3 million index prices over 16 years that applies almost no screening and validates within each trading day rather than forward in time, and one with 11.3 million S&P 500 call options over 12 years. Neither varies the space in which its model is trained.

3 Methodology

3.1 Ad hoc Black-Scholes model

We begin with the ad hoc Black-Scholes (BS) model. Under the BS assumptions, the price of a European call is

$$c^{\text{BS}} = Se^{-q\tau}\Phi(d_1) - Ke^{-r\tau}\Phi(d_2), \quad (1)$$

and the price of a European put is

$$p^{\text{BS}} = Ke^{-r\tau}\Phi(-d_2) - Se^{-q\tau}\Phi(-d_1), \quad (2)$$

with

$$d_1 = \frac{\ln(S/K) + (r - q + \frac{1}{2}\sigma^2)\tau}{\sigma\sqrt{\tau}}, \quad d_2 = d_1 - \sigma\sqrt{\tau}. \quad (3)$$

Here S is the spot price of the underlying index; K is the strike price of the contract; τ is its time to maturity, measured in years; r is the continuously compounded risk-free rate to maturity; q is the continuously compounded dividend yield of the index; σ is the volatility of the index return, which under the BS assumptions is a single constant common to every contract written on the index on a given date; and $\Phi(\cdot)$ is the standard normal cumulative distribution function.

Two conventions used throughout follow from *Eq. (1)–(3)*. First, i indexes an option contract and u a trading date, and $\mathcal{I}_u$ denotes the cross-section of contracts traded on date u that survive the filters of Section 4. Where contract- and date-specific values are required, we write S_u , r_u , and q_u for the index level, risk-free rate, and dividend yield on date u ; K_i for the strike of contract i ; and $\tau_{i,u}$ for its remaining time to maturity on date u ; the construction of r_u and q_u from the OptionMetrics term structures is described in Section 4. The observed price of contract i on date u is the bid–ask midpoint, written as

$V_{i,u}$. Second, we define the general BS pricing function as $BS(S, K, r, q, \tau, \sigma, cp)$ where cp denotes the call/put indicator. Specifically, when cp identifies a call option, the function corresponds to $c^{BS}(S, K, r, q, \tau, \sigma)$ in Eq. (1), whereas for a put option, it corresponds to $p^{BS}(S, K, r, q, \tau, \sigma)$ in Eq. (2). Both pricing functions use the common definitions of d_1 and d_2 in Eq. (3).

On each date $u - 1$, we estimate a single cross-sectional volatility parameter $\hat{\sigma}_{u-1}$ by nonlinear least squares, minimizing the sum of squared deviations between the model prices implied by Eq. (1)–(2) and the observed midpoints across all contracts in $\mathcal{I}_{u-1}$. This estimate is then treated as the volatility prevailing on date u , so that the BS prices reported for date u are strictly out of sample. The limitation of this benchmark is well known. A single σ implies an implied-volatility (IV) surface that is flat in both moneyness and maturity, contrary to the systematic smile and skew documented by Rubinstein (1994).

To relax the constant-volatility restriction whilst preserving the analytical tractability of Eq. (1)–(2), we employ an ad hoc BS model in which the volatility argument is itself a deterministic function of the contract’s coordinates. Following Andreou et al. (2010, 2014), we adopt the nonlinear deterministic volatility function (DVF)

$$\sigma^{\text{DVF}}(m, \tau; \mathbf{a}) = \max\{0.01, a_0 + a_1 \ln(m) + a_2 \ln^2(m) + a_3 \tau + a_4 \ln(m) \tau + a_5 \tau^2\}, \quad (4)$$

where $\mathbf{a} = (a_0, a_1, \dots, a_5)'$ is the parameter vector to be estimated and

$$m_{i,u} = \frac{S_u}{K_i} \exp\{(r_u - q_u) \tau_{i,u}\} \quad (5)$$

is the forward-adjusted moneyness of contract i on date u , that is, the ratio of the forward price of the index at the contract’s maturity to its strike.⁶ Contracts at the

⁶ We adopt the nonlinear specification following Andreou et al. (2010, 2014), who show that nonlinear DVFs yield substantially lower pricing errors and superior IV surface fit relative to linear alternatives (Almeida et al., 2023), both in-sample and out-of-sample. The optimization is implemented with the

money forward have $m_{i,u} = 1$ and hence $\ln m_{i,u} = 0$, so the quadratic in *Eq. (4)* is centered at the money. The quadratic in $\ln(m)$ accommodates the smile and skew in the strike dimension, the terms in τ the term structure of at-the-money volatility, and the interaction $\ln(m)\tau$ the flattening of the skew with maturity. The floor at 0.01 (one volatility point) keeps the volatility argument strictly positive, so that *Eq. (1)–(3)* remains well defined.

On each date $u - 1$ we estimate $\mathbf{a}_{u-1}$ by nonlinear least squares, minimizing squared pricing errors across the cross-section:

$$\hat{\mathbf{a}}_{u-1} = \arg \min_{\mathbf{a}} \sum_{i \in \mathcal{I}_{u-1}} \left[c_{i,u-1} - \text{BS}(S_{u-1}, K_i, r_{u-1}, q_{u-1}, \tau_{i,u-1}, \sigma_{i,u-1}^{\text{DVF}}(\mathbf{a}), cp_i) \right]^2, \quad (6)$$

where the summation runs over the call options in the cross-section $\mathcal{I}_{u-1}$. On date u we form the out-of-sample price prediction for each contract by evaluating *Eq. (1)* at the contract-specific volatility $\sigma^{\text{DVF}}(m_{i,u}, \tau_{i,u}, \hat{\mathbf{a}}_{u-1})$, so that the volatility surface uses only information available on date $u - 1$ whilst the pricing inputs S_u , r_u , q_u and $\tau_{i,u}$ are contemporaneous. An analogous procedure applies to put options through *Eq. (2)*, *mutatis mutandis*.

The DVF specification nests the constant-volatility model as a special case. Setting $a_1 = a_2 = a_3 = a_4 = a_5 = 0$ collapses *Eq. (4)* to $\sigma = a_0$. Throughout the remainder of the paper, DVF denotes the nonlinear specification of *Eq. (4)*, and BS denotes the constant-volatility benchmark estimated as described above; both appear as reference specifications in the tables that follow.

Sequential Least Squares Programming (SLSQP) algorithm via `scipy.optimize.minimize` in Python. To mitigate convergence to local minima, we employ multiple initialization strategies: (i) parameter values based on daily averages reported in prior literature, (ii) optimized parameter vectors from the preceding five trading days, and (iii) the mean of these five historical parameter estimates.

3.2 Feature set

Both machine learning specifications receive the same six-element input vector, written $\mathbf{X}_{i,u}$ for contract i on date u :

$$\mathbf{X}_{i,u} = (m_{i,u}, \tau_{i,u}, r_u, q_u, \sigma_{i,u}^{\text{DVF}}, cp_i)^\top, \quad (7)$$

where $m_{i,u}$ is the forward-adjusted moneyness of *Eq. (5)*, $\tau_{i,u}$ the time to maturity, r_u and q_u the risk-free rate and dividend yield of *Section 3.1*, cp_i a call/put indicator, and

$$\sigma_{i,u}^{\text{DVF}} = \sigma^{\text{DVF}}(m_{i,u}, \tau_{i,u}, \hat{\mathbf{a}}_{u-1})$$

the DVF surface *Eq. (4)*, estimated on the previous trading day and evaluated at the contract's own coordinates. We write $\mathbf{X}_{i,u}^{(-\sigma)}$ for the same vector with the volatility element removed.

Only the volatility feature carries information from beyond the contract itself. It summarizes the shape of the previous day's cross-section of implied volatilities, whilst the other features describe the contract or the market conditions on the date it is priced. Lagging it by one day keeps contemporaneous information out of the input set. Features are standardized on training-window statistics alone, so nothing from the validation or test month enters the standardization.

The feature set is deliberately small and identical across the two specifications. The object of study is the loss function space, and enriching the inputs of one specification or both would confound the comparison with a difference in the information the network has to work with. This is a restriction relative to the richer input sets used elsewhere in the machine-learning option-pricing literature, and it is a deliberate one: a larger feature set would very likely lower both specifications' errors, but it would not sharpen the comparison between them.

3.3 The unified GPF framework

Andreou et al. (2010) introduce the Generalized Parametric Function (GPF) architecture, a semi-parametric option-pricing method that combines the complementary strengths of a parametric model and a machine learning algorithm. The *raison d'être* of the architecture is that the parametric model supplies the pricing structure, whilst the learner delivers a more accurate prediction, so that the corrected prices remain well behaved where data are sparse and flexible where data are rich. The two models we study are both instances of this architecture. They take the same six input features, the same DVF surface, and the same network up to its output, and both price through the BS formula. They differ in one respect only. GPF^{OPS} measures its training loss in price space, whereas GPF^{IVS} measures it in implied-volatility space.

Figure 3 illustrates the two specifications. Throughout the paper, $\mathbf{X}$ denotes the common six-feature input vector defined in Section 3.2, and $\mathbf{X}^{(-\sigma)}$ denotes the same vector excluding the volatility feature. Both specifications use the same contract information and the same BS pricing function, but they differ in where the loss function is defined. The notation follows the convention that a hat denotes a model-generated quantity, whereas an unhatted variable denotes its observed market counterpart. In particular, V represents the observed option price.

[Insert Figure 3, here]

The two panels differ in where the pricing map sits. Panel (a) describes the structure of GPF^{OPS} . The neural network $F(\cdot; \theta)$ processes contract features $\mathbf{X}$ to produce enhanced implied volatility ($\hat{\sigma}^{\text{enh}}$). The parentheses of ($\hat{\sigma}^{\text{enh}}$) indicate that this variable is embedded within the model architecture. Both the features $\mathbf{X}^{(-\sigma)}$ and the enhanced volatility flow into the embedded BS pricing layer, which produces the predicted option price $\hat{V}^{\text{OPS}}$. The red dotted box encompasses the entire GPF architecture, highlighting that GPF is an end-to-end model. Panel (b) describes the structure of GPF^{IVS} . The

neural network $G(\cdot; \phi)$ processes contract features $\mathbf{X}$ to produce an implied volatility residual output $\hat{\epsilon}^{\sigma}$. Both $\mathbf{X}$ and the predicted $\hat{\epsilon}^{\sigma}$ are passed to an external BS model, which combines the baseline parametric estimates with the learned volatility residual to generate the final option price $\hat{V}^{IVS}$. The red dotted box highlights the nonparametric learning component, emphasizing its two-step approach.

3.3.1 Price-space specification (GPF^{OPS})

The Generalized Parametric Function framework of [Andreou et al. \(2010\)](#) combines a parametric option pricing model with a neural network to achieve both structural discipline and flexible nonparametric fitting. In our implementation, the parametric component is the ad hoc BS model with the nonlinear DVF of Section 3.1, whilst the network learns an enhanced volatility, and the embedded BS model imposes the pricing structure.

Estimation. The network $F(\cdot; \theta)$, parameterized by weights θ , transforms the feature vector into an enhanced implied volatility,

$$\hat{\sigma}_{i,u}^{\text{enh}} = F(\mathbf{X}_{i,u}; \theta), \quad (8)$$

which is converted to a strike-normalized price through an embedded BS model,

$$\hat{V}_{i,u} = \frac{1}{K_i} BS(S_u, K_i, r_u, q_u, \tau_{i,u}, \hat{\sigma}_{i,u}^{\text{enh}}, cp_i). \quad (9)$$

The estimation target is the observed price on the same scale,

$$V_{i,u} = \frac{V_{i,u}^{mkt}}{K_i}, \quad (10)$$

where $V_{i,u}^{mkt}$ denotes the observed midpoint. Strike normalization stabilizes the magnitude of prices across strike levels and improves the numerical conditioning of the opti-

mization. The parameters are estimated by minimizing the mean squared normalized pricing error over the training window,

$$\hat{\boldsymbol{\theta}}_{\mathcal{T}} = \arg \min_{\boldsymbol{\theta}} \sum_{u \in \mathcal{W}_{\mathcal{T}}} \sum_{i \in \mathcal{I}_u} \left(V_{i,u} - \hat{V}_{i,u}(\boldsymbol{\theta}) \right)^2, \quad (11)$$

where $\mathcal{W}_{\mathcal{T}}$ denotes the trading dates in the ten-month training window $[\mathcal{T} - 12, \mathcal{T} - 2)$, which rolls forward one month for each test month $\mathcal{T}$, and the double summation aggregates squared errors across dates and contracts.

The embedded BS model. The distinctive feature of the architecture is that *Eq. (9)* is embedded in the neural network rather than after it. This serves two purposes. First, it constrains the network’s output to respect the BS pricing structure so that predicted prices inherit the monotonicity and no-arbitrage properties of the formula. Second, because $BS(\cdot)$ is analytically differentiable in its volatility argument, the pricing error propagates back to the network’s weights during training, with the derivative of the loss with respect to $\hat{\sigma}_{i,u}^{\text{enh}}$ scaled by the BS vega at the predicted volatility. The enhanced volatility is therefore never an estimation target in its own right but an internal quantity the network selects to minimize pricing error, and the vega scaling arises mechanically from the chain rule rather than being imposed. The Internet Appendix gives the gradient calculation in full.

Out-of-sample prediction. For each contract observed on a trading day u of the test month, we construct $\mathbf{X}_{i,u}$ from the market conditions of date u and the DVF surface of date $u - 1$, and pass it through the network estimated at $\mathcal{T}$. The price forecast is the rescaled network output,

$$\hat{V}_{i,u}^{OPS} = \frac{1}{K_i} BS(S_u, K_i, r_u, q_u, \tau_{i,u}, \hat{\sigma}_{i,u}^{\text{enh}}, cp_i). \quad (12)$$

3.3.2 Volatility-space specification (GPF^{IVS})

GPF^{IVS} is the reading of the framework that the recent neural-network literature has adopted, in which the network corrects the volatility error of a fitted surface, is trained on that residual, and prices only afterwards. Its antecedent is the hybrid network of Andreou et al. (2008), and a recent prominent example is the two-step error-correction network of Almeida et al. (2023), whose implementation we follow.⁷ Unlike GPF^{OPS}, which embeds the pricing formula in the network architecture, this specification decomposes the task into two stages: fitting a parametric model to obtain baseline volatilities and training a network to learn the residual in implied-volatility space. The parametric component is the same DVF surface.

Estimation. For each date u in the training window, the target is constructed in two steps.

Step 1: parametric baseline. The DVF parameter vector $\hat{\mathbf{a}}_{u-1}$ is estimated on the cross-section of the previous trading date by solving Eq. (6), giving the contract-specific parametric volatility

$$\sigma_{i,u}^{\text{DVF}} = \sigma^{\text{DVF}}(m_{i,u}, \tau_{i,u}, \hat{\mathbf{a}}_{u-1}), \quad i \in \mathcal{I}_u. \quad (13)$$

⁷ See Almeida et al. (2023) for the complete methodology and for evidence of cross-sectional and panel improvements over both uncorrected parametric models and direct nonparametric fitting of the IV surface. Their design is well suited to its purpose, correcting the misspecification of parametric volatility surfaces; our interest is in what the volatility-space loss implies when pricing accuracy is the criterion by which the corrected model is judged. Our implementation keeps their two-step structure and squared-error loss on volatility residuals but departs from it in four respects: (i) of their four parametric baselines, the closest to ours is the Dumas et al. (1998) quadratic in spot moneyness, fitted to each day’s surface by ordinary least squares on implied volatilities, whereas our DVF is quadratic in log forward moneyness and fitted by nonlinear least squares on the previous day’s prices, so our residuals are one-day-ahead rather than same-day errors; (ii) their network sees only moneyness and maturity, joined by eight daily state variables such as the VIX in their panel exercise, rather than the six features of Eq. (7); (iii) they compare one- to five-layer sigmoid networks trained by scaled conjugate gradient, rather than a single batch-normalized layer sized on a validation window, trained with Adam and averaged over five seeds; and (iv) they use out-of-the-money options with 20 to 240 days to maturity from January 2016 to June 2019, fitted day by day or on one two-year panel and evaluated by implied-volatility RMSE, rather than calls and puts from 1996 to 2025 under the rolling protocol of Section 3.4.

Step 2: residual construction. The observed implied volatility $\sigma_{i,u}^{\text{BS}}$ is recovered from the mid-quote $V_{i,u}$ by numerically inverting the BS formula, and the target is the difference between it and the parametric baseline,

$$\epsilon_{i,u}^{\sigma} = \sigma_{i,u}^{\text{BS}} - \sigma_{i,u}^{\text{DVF}}. \quad (14)$$

The network $G(\cdot; \phi)$, parameterized by weights ϕ , is trained to predict this residual from $\mathbf{X}_{i,u}$ by minimizing the mean squared error in volatility space,

$$\hat{\phi}_{\mathcal{T}} = \arg \min_{\phi} \sum_{u \in \mathcal{W}_{\mathcal{T}}} \sum_{i \in \mathcal{I}_u} (\epsilon_{i,u}^{\sigma} - G(\mathbf{X}_{i,u}; \phi))^2, \quad (15)$$

over the same window $\mathcal{W}_{\mathcal{T}}$. The pricing map does not enter *Eq. (15)* because the residual is taken at the volatility, not at the price, which is the single respect in which this specification differs from *Eq. (11)*.

Out-of-sample prediction. For each contract observed on a trading day u of the test month, the forecast is again formed in two steps. *Step 1: parametric baseline.* The DVF surface estimated on date $u - 1$ is evaluated at the contract's coordinates, as in *Eq. (13)*. *Step 2: nonparametric correction.* The trained network supplies the volatility correction, giving the corrected volatility

$$\hat{\sigma}_{i,u}^{\text{IVS}} = \sigma_{i,u}^{\text{DVF}} + G(\mathbf{X}_{i,u}; \hat{\phi}_{\mathcal{T}}), \quad (16)$$

and the price forecast follows by substituting it into the BS formula,

$$\hat{V}_{i,u}^{\text{IVS}} = BS(S_u, K_i, r_u, q_u, \tau_{i,u}, \hat{\sigma}_{i,u}^{\text{IVS}}, cp_i). \quad (17)$$

3.4 Neural network specification and implementation details

The machine learning model comprises a single fully connected hidden layer with batch normalization and a sigmoid activation, followed by a linear output layer. A single hidden layer keeps the design close to the semi-parametric networks of [Hutchinson et al. \(1994\)](#), [Garcia and Gençay \(2000\)](#) and [Andreou et al. \(2010\)](#), in which the parametric component rather than network depth supports the pricing structure. Five networks are estimated for each rolling window, differing in the random seed that governs weight initialization and batch shuffling. For each seed, a network is trained at every candidate size in $\{32, 16, 8\}$ neurons and the size attaining the lowest validation loss is retained, so the five retained networks need not share the same size. These are the layer sizes from which [Almeida et al. \(2023\)](#) assemble their architectures, following [Gu et al. \(2020\)](#). As [Almeida et al. \(2023\)](#) observe, searching over the full space of architectures is not feasible, so a small fixed candidate set is the usual compromise. The prediction for each contract is the equal-weighted average of the five outputs, an ensemble device used by [Gu et al. \(2020\)](#) so that predictions do not rest on a single random initialization.

We adopt a strict 10-2-1 month rolling protocol. For each test month $\mathcal{T}$ the models are trained on the window $[\mathcal{T} - 12, \mathcal{T} - 2)$, validated on $[\mathcal{T} - 2, \mathcal{T})$, and used out of sample to price options in month $\mathcal{T}$, giving 344 monthly test blocks. The design follows the rolling-forward partition of [Andreou et al. \(2010\)](#), who estimate the GPF on successive training and validation windows each followed by a separate and non-overlapping monthly test period. Keeping the three windows in chronological order, so that hyperparameter selection never sees the evaluation period, is also standard practice in machine learning asset pricing ([Gu et al., 2020](#); [Chen et al., 2024](#); [Bali et al., 2023](#)), though those studies refit annually on an expanding window rather than monthly on a fixed-length one.

Table 1 reports the configuration in full. Every setting in it applies to both specifications, so the comparison that follows isolates the loss function.

[Insert Table 1, here]

3.5 Evaluation metrics

The models' out-of-sample performance is evaluated with three error metrics, reported for each model, option type, and evaluation subset.

The first metric is the standard root mean squared error (RMSE), in index points, which is the criterion GPF^{OPS} minimizes:

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (\hat{V}_i - V_i)^2}, \quad (18)$$

where $\hat{V}_i$ is the predicted price, V_i the market price, and N the number of observations in the evaluation set.

The second metric is the implied-volatility root mean squared error (IVRMSE), in percentage points, which is the criterion GPF^{IVS} minimizes:

$$\text{IVRMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (\hat{\sigma}_i - \sigma_i^{\text{BS}})^2}, \quad (19)$$

where $\hat{\sigma}_i$ is the model-implied volatility, obtained by inverting the predicted dollar price through Newton's method with a bisection fallback, and σ_i^{BS} is the observed implied volatility. IVRMSE assigns no vega weight to the volatility residual.

The third metric is the vega-weighted root mean squared error ($\mathcal{V}$ RMSE), in index points, which weights the squared pricing error by vega:

$$\mathcal{V}\text{RMSE} = \sqrt{\frac{\sum_{i=1}^N \mathcal{V}_i (\hat{V}_i - V_i)^2}{\sum_{i=1}^N \mathcal{V}_i}}, \quad (20)$$

where $\mathcal{V}_i$ is the BS vega⁸, computed at the observed implied volatility. $\mathcal{V}$ RMSE tilts the pricing criterion toward the contracts whose values respond most strongly to volatility and is a natural measure of hedging-relevant accuracy.

The three metrics are not independent measurements; they are three views of the same per-contract volatility error, seen through the pricing map with increasing vega loadings. Write $\epsilon_i = \hat{\sigma}_i - \sigma_i^{\text{BS}}$ for the volatility error of contract i . Because the observed price satisfies $V_i = \text{BS}(\sigma_i^{\text{BS}}, \cdot)$, a first-order expansion of BS in its volatility argument around σ_i^{BS} gives $\hat{V}_i - V_i \approx \mathcal{V}_i \epsilon_i$, so the squared pricing error is, to leading order, the squared volatility error weighted by squared vega. The three metrics, therefore, span a weighting spectrum over the same residual. IVRMSE weights ϵ_i^2 equally across contracts, RMSE weights it, to leading order, by $\mathcal{V}_i^2$, and $\mathcal{V}$ RMSE, which applies one further vega weight to the squared pricing error, weights it by $\mathcal{V}_i^3$. The RMSE that GPF^{OPS} minimizes is thus a vega-squared-weighted volatility criterion expressed in price units, whereas the IVRMSE that GPF^{IVS} minimizes weights every contract equally. The two loss functions coincide only in the knife-edge case of vega being constant across contracts; however, in practice, option pricing data are characterized by a wide empirical range of vega, so the weighted and unweighted criteria differ materially in our sample.⁹

This link generates a testable prediction. If the advantage of the price-space loss operates through the vega weighting, it should increase with the vega weight of the evaluation metric, from IVRMSE through RMSE to $\mathcal{V}$ RMSE, and it should be larger for high-vega contracts than for low-vega contracts. The empirical sections test both parts of this prediction.

For statistical inference, we apply Diebold–Mariano tests ([Diebold and Mariano](#),

8

$$\frac{\partial \text{BS}}{\partial \sigma} = S e^{-q\tau} \phi(d_1) \sqrt{\tau} \equiv \mathcal{V}(\sigma). \quad (21)$$

Formulas express vega per unit of implied volatility; dollar magnitudes quoted in the text and tables follow the per-volatility-point convention, with the formula value divided by 100.

⁹ The Internet Appendix states the expansion formally, gives the conditions under which the two loss functions share their minimizer, and discusses what the observed metric ordering can and cannot establish.

1995) of equal predictive accuracy between GPF^{IVS} and GPF^{OPS} , computed on the loss differentials underlying each of the three metrics, with standard errors clustered by trading date.¹⁰ Section 6.1 adds Giacomini–White conditional predictive ability tests (Giacomini and White, 2006), which ask whether the accuracy difference varies with lagged implied volatility and with ex-ante vega.

4 Data and sample construction

4.1 Data source and filters

We obtain end-of-day S&P 500 index-option (SPX) data from OptionMetrics for January 1996 through August 2025; the first twelve months serve only to initialize the rolling estimation windows, so the evaluation test months span January 1997 through August 2025. The data contain best bid and ask quotes, strike prices, expiration dates, trading volume, open interest, and the OptionMetrics interest-rate and dividend-yield inputs. We define the observed option price as the bid–ask midpoint. To place the two loss functions on exactly the same footing, we recover the observed implied volatility from the midpoint using the same BS formula that converts each model’s predicted volatility into a predicted dollar price.

We first remove observations with a missing OptionMetrics implied volatility, a non-positive best bid, a negative bid–ask spread, a vendor-supplied implied volatility outside $[2\%, 300\%]$, non-positive open interest, or non-positive trading volume. We retain contracts with 6 to 365 calendar days to maturity and additionally require the absolute BS delta to lie in $[0.05, 0.95]$. The short end of the maturity window follows Bakshi et al. (1997) and Dumas et al. (1998), who exclude options with fewer than six days to expiration to alleviate expiration-related liquidity biases; the one-year cap removes

¹⁰ Results are qualitatively unchanged when clustering by calendar month, which is uniformly more conservative.

thinly traded long-dated contracts. The delta screen removes contracts at the extreme tails for which quotes are particularly sensitive to discreteness and microstructure noise. The midpoint-implied volatility used as the model target is recovered subsequently and independently, so it is not mechanically truncated at the vendor-field screening bounds. These filters define the paper’s filtered universe. Within it, the analysis applies no further moneyness, option-price, or call/put sampling rules.

Within each training window, we retain only contracts whose strike lies on the \$10 grid. SPX strikes are listed at \$5 increments near the money and at wider increments in the wings, so the density of listed strikes varies systematically across the moneyness range; restricting to the \$10 grid equalizes that density and prevents the densely-listed near the money region from dominating the training loss through sheer contract count. The rule follows [Almeida et al. \(2023\)](#). It is applied to the training cross-section only, so the out-of-sample evaluation universes are unaffected.

4.2 Summary statistics

Table 2 summarizes the evaluation sample, which contains 3,686,008 call and 5,007,928 put observations over 7,208 trading days.¹¹ Three features of the table inform the analysis that follows. First, the distribution of observed implied volatility concentrates near 17%. The median is 16.1% for calls and 18.7% for puts, with inter-quartile ranges of roughly 13% to 21% and 14% to 24%, and a long right tail. Second, prices span more than four orders of magnitude, from a quarter of an index point to more than five thousand. A one-point pricing error is therefore a large error on the cheapest contracts and a negligible one on the most expensive, so a criterion defined on prices and a criterion defined on volatilities do not rank the same set of errors in the same way. Third, and central to the paper, BS vega ranges from below 0.01 to approximately 25 index points per volatility point, with a median near 3.4. The vega-squared weight that separates

¹¹ The performance tables report observations of 3,676,030 calls and 4,991,872 puts, the marginally smaller common sample on which all four models produce a price forecast.

the two loss functions in Section 3.5 therefore varies across contracts by several orders of magnitude, which is what gives the choice of loss space its empirical content.

[Insert Table 2, here]

5 Empirical results

This section reports the out-of-sample comparison. Section 5.1 presents the pooled results, Sections 5.2 and 5.3 the same comparison by calendar year and by contract characteristics; Section 6 then examines the source of the advantage. Throughout, the two GPF specifications are estimated on identical samples with identical features, architectures, tuning rules, and rolling dates, so differences in their performance identify the effect of the loss function.

5.1 Pooled performance

Table 3 reports pooled out-of-sample performance for the four models on the common evaluation sample of 8,667,902 observations. The table shows three things. First, the familiar ranking of model classes is reproduced, with both learners improving substantially on their parametric counterparts and the DVF improving on the constant-volatility BS benchmark, consistent with the machine-correction evidence of Andreou et al. (2010). Second, the loss space contributes more to pricing accuracy than the introduction of the learner itself. Moving from DVF to GPF^{IVS} , that is, adding the network in volatility space, lowers pooled RMSE by 13.5% (15.27 to 13.21 index points). Moving from GPF^{IVS} to GPF^{OPS} , a single change in the loss function lowers it by a further 25.3% (13.21 to 9.87), with reductions of 26.2% for calls (14.33 to 10.57) and 24.5% for puts (12.33 to 9.31). The Diebold–Mariano tests against the GPF^{IVS} benchmark reject equal accuracy at the 1% level in every block.

Third, the pattern across metrics is exactly the one the vega weighting of Section 3.5 predicts. The pooled advantage of GPF^{OPS} *vis-à-vis* GPF^{IVS} rises monotonically with the vega loading of the criterion, from 7.4% on the unweighted IVRMSE (2.99 to 2.77 percentage points) to 25.3% on RMSE and to 31.5% on the vega-weighted $\mathcal{V}$ RMSE (19.58 to 13.42 index points). The same monotonicity holds within calls (10.0%, 26.2%, 32.9%) and within puts (4.9%, 24.5%, 29.9%). This pattern is informative in two respects. First, the pricing accuracy of GPF^{OPS} does not come at the expense of volatility accuracy, as it attains the lower IVRMSE as well. Second, the advantage of GPF^{OPS} grows with the weight the evaluation metric places on vega. Since the two specifications share the same architecture, this pattern points to the vega weighting rather than to one model simply being more flexible than the other.

[Insert Table 3, here]

5.2 Performance over time

Table 4 decomposes the pooled comparison by calendar year. GPF^{OPS} posts the lower pricing RMSE in each of the 29 calendar years, with proportional reductions ranging from roughly 2 – 3% (1998, 2002) to 51.8% (2017), and the Diebold–Mariano test rejects equal accuracy at the 1% level in 25 years, at 5% in 2009, and at 10% in 1997 and 1998; 2002 is the single year in which the difference is not significant at conventional levels. On the vega-weighted criterion, the dominance is complete. GPF^{OPS} attains the lower $\mathcal{V}$ RMSE in all 29 calendar years, significantly at the 5% level or better in every one. The unweighted IVRMSE is, as expected, the closest contest. GPF^{IVS} attains the lower value in eight years (1997, 1998, 2002, 2005, 2008, 2009, 2015, 2018), significantly in five, whilst GPF^{OPS} is lower in the remaining 21. That the price-space specification concedes only the criterion its rival optimizes, only occasionally, and by economically small margins, whilst winning the price-based criteria essentially always, is the prediction of Section 3.5 expressed year by year.

[Insert Table 4, here]

The observations column adds a compositional caution for interpreting pooled magnitudes. The panel is strongly unbalanced, with yearly counts ranging from 24,586 in 2000 to 1,113,532 in 2024, and the six most recent years contribute roughly two thirds of the pooled observations, so the pooled figures of Table 3 are weighted toward the post-2020 regime of elevated volatility and expanded strike listings. The year-by-year decomposition shows that the ranking does not depend on that weighting: the advantage is present from 1997 onward, in calm years and in crisis years, and its proportional size is, if anything, largest in the ordinary-volatility years examined further in Section 6.1.

5.3 Performance across moneyness and maturity

Table 5 reports the same comparison across moneyness and maturity, the two contract dimensions that drive vega. For calls, the reduction in pricing RMSE rises steadily as the contract moves into the money, from 1.9% deep out of the money to 9.4% out of the money, 23.8% at the money, 38.6% in the money, and 44.0% deep in the money. For puts, the reduction rises from 17.8% deep out of the money to 18.1% out of the money, 27.9% at the money and 35.3% in the money, and then falls to 21.3% deep in the money, the smallest bucket in the table with 84,243 contracts. In both cases the advantage is smallest in the out-of-the-money wing, where contracts are cheapest, and largest at or inside the money. The pattern across maturity is the same. The reduction in pricing RMSE grows from 10.9% for calls and 5.9% for puts with fewer than 60 days to maturity, to 14.2% and 21.4% between 60 and 120 days, and to 36.1% and 34.5% beyond 120 days. Vega rises with maturity, so this is the cross-sectional counterpart of the ordering across metrics reported in Section 5.1. The differences are larger still on the vega-weighted criterion. $\mathcal{V}$ RMSE falls by half on deep in-the-money calls, from 46.72 to 23.47 index points, and by 38% on long-dated calls, from 33.58 to 20.65.

The exceptions are as informative as the rule because they sit exactly where the

mechanism says they should. In five cells, all of them low-vega, the IVRMSE advantage of GPF^{OPS} disappears. For deep out-of-the-money calls, GPF^{IVS} attains a significantly lower IVRMSE (3.27 against 3.37) whilst the RMSE difference is small and insignificant; for short-dated puts its IVRMSE is lower at the 10% level (3.00 against 3.06); and for out-of-the-money calls, deep out-of-the-money puts, and out-of-the-money puts the gap is within 0.03 percentage points and insignificant (2.35 against 2.32, 2.72 against 2.69, and 2.21 against 2.22). These are the cheap, low-vega contracts on which volatility errors barely move prices and on which, per Section 6.2, the inverted volatilities are noisiest. In the remaining eleven cells, GPF^{OPS} is more accurate on all three metrics at the 5% level.

[Insert Table 5, here]

6 The source of the price-space advantage

The results so far establish that training on prices produces more accurate out-of-sample prices and that the margin is large and present in every sample year. This section asks where the advantage comes from. We first test whether it varies with conditions known at the forecast date, sorting trading days by lagged implied volatility and contracts by ex-ante vega. We then develop a statistical interpretation of the pattern those tests reveal, in which the price-space loss acts as a feasible generalized least squares criterion on noisily measured volatilities.

6.1 Conditional predictive ability across implied-volatility and vega states

Following [Giacomini and White \(2006\)](#), we use Giacomini–White tests to examine whether the relative predictive performance of GPF^{OPS} and GPF^{IVS} varies with lagged implied

volatility and with ex-ante vega. For option-type group $g \in \{C, P, A\}$, where A denotes the pooled sample, define the daily pricing-loss differential as

$$d_u^g = \text{MSE}_{\text{GPF-IVS},u}^g - \text{MSE}_{\text{GPF-OPS},u}^g, \quad (22)$$

where $\text{MSE}_{\cdot,u}^g$ is the mean squared pricing error of the indicated model over the group- g contracts observed on date u . A positive value of d_u^g indicates that GPF^{OPS} has the lower squared pricing error on that date. As a reference point for the conditional analysis, we first test the unconditional null $\mathbb{E}(d_u^g) = 0$ with a t statistic computed from the daily differential series, using the same 22-lag Newey–West variance estimator as the Wald tests below; this statistic is reported as Unconditional t in Table 6.

For each conditioning variable, observations are assigned to ten ordered states. Lagged-IV states are defined across forecast dates. Ex-ante vega is calculated from the lagged-IV input and sorted within date and option type, so the vega classification uses only information available when forecasts are formed. Let D_s denote the event that a forecast date falls in state $s \in \{1, \dots, 10\}$, and let $\mu_s^g = \mathbb{E}(d_u^g \mid D_s)$ denote the mean loss differential in state s .

The conditional analysis asks two questions about the state means $(\mu_1^g, \dots, \mu_{10}^g)$. The conditional-equality (CE) null hypothesis states that the two models are equally accurate in every state. The state-dependence (SD) null hypothesis states that the advantage of GPF^{OPS} is the same size in every state. Formally,

$$H_{0,\text{CE}} : \mu_1^g = \mu_2^g = \dots = \mu_{10}^g = 0, \quad (23)$$

$$H_{0,\text{SD}} : \mu_1^g = \mu_2^g = \dots = \mu_{10}^g. \quad (24)$$

Rejecting $H_{0,\text{CE}}$ indicates that predictive accuracy differs between the models in at least one state. Rejecting $H_{0,\text{SD}}$ indicates that the size of the advantage varies with the state, which raises the sharper question here, since an advantage that comes from the

vega weighting should vary across vega states. Because the CE null imposes ten zero restrictions on the ten state means, whilst the SD null imposes only their mutual equality—nine restrictions—the corresponding Wald statistics are asymptotically $\chi^2(10)$ and $\chi^2(9)$, with the degrees of freedom reported in Table 6.

Both hypotheses are evaluated using the HAC Wald statistic

$$W = (\mathbf{R}\hat{\boldsymbol{\mu}} - \mathbf{r})' \left[\widehat{\mathbf{RVar}(\hat{\boldsymbol{\mu}})} \mathbf{R}' \right]^{-1} (\mathbf{R}\hat{\boldsymbol{\mu}} - \mathbf{r}) \stackrel{a}{\sim} \chi_k^2, \quad (25)$$

where $\hat{\boldsymbol{\mu}} = (\hat{\mu}_1, \dots, \hat{\mu}_{10})'$ collects the estimated state means, the matrix $\mathbf{R}$ and the vector $\mathbf{r}$ encode the restrictions of Eq. (23) or Eq. (24), and k is the number of restrictions. Newey–West HAC covariances use 22 lags.

[Insert Table 6, here]

Table 6 reports the results. The unconditional mean loss differential is positive and highly significant for calls, puts, and the pooled sample, and the conditional-equality null is rejected at the 1% level under both circumstances. Every estimated state mean is positive; ergo, the rejections have a clear direction. There is no volatility regime and no vega decile in which GPF^{IVS} prices more accurately on average.

The two sorts give sharply different answers on the second null, and the contrast is the central message of the table. Conditioning on lagged implied volatility, a constant differential cannot be rejected for calls ($p = 0.422$) or for the pooled sample ($p = 0.178$), with modest evidence of variation for puts ($p = 0.028$). Conditioning on ex-ante vega, state dependence is rejected at the 1% level for every group, and the Wald statistics under the vega sort, 47.7 to 65.5 on nine degrees of freedom, are several times those under the volatility sort, 9.2 to 18.7. The advantage of the price-space loss is therefore approximately invariant to the aggregate volatility environment but varies systematically with the vega of the contract being priced. This is the pattern that the vega weighting predicts, operating through the cross-sectional dispersion of vega rather than through

the level of market volatility.

Figure 4 plots the pooled error curves across the ten lagged-IV states and shows what lies behind the non-rejection. Both curves slope upward, with pricing RMSE three to four times larger in the most turbulent state than in the calmest. The slope itself is expected and carries no information about the ranking. High-volatility states raise option premia, widen quoted spreads, and speed up the movement of the surface relative to the one-day-lagged deterministic volatility function, so the pricing errors of any model grow with the volatility level, and panel (b) shows that volatility errors rise across the same states. What matters is the gap between the two curves.

[Insert Figure 4, here]

The gap is most informative in the states where the market spends most of its time. Table 2 places the center of the observed implied-volatility distribution near 17%, with medians of 16.1% for calls and 18.7% for puts, and the state means of the lagged-IV sort are correspondingly compressed, six of the ten lying below 19% and only the top state exceeding 25%. The interior states of Figure 4 therefore describe the market’s ordinary condition rather than an unusual one. Precisely in these states the GPF^{IVS} curve is most erratic. Its pooled RMSE spikes to roughly 14 index points in the state centered near 14.5%, against a value below 8 for GPF^{OPS} , and remains elevated in the adjacent state, whereas the GPF^{OPS} curve rises with the volatility level. The spike has no counterpart in panel (b), where the two models are separated by no more than a few tenths of a percentage point in the same states. A modest gap in volatility space can become a near-twofold gap in price space only if the volatility errors of GPF^{IVS} fall disproportionately on high-vega contracts, which is the weighting argument of Section 3.5 operating in the middle of the volatility distribution. *Prima facie* the spike suggests state dependence, yet the formal test cannot reject a constant differential across volatility states, so it is a feature of the point estimates rather than of the test; what the curves add is the location of the instability.

The calendar-year evidence of Table 4 points the same way. The years in which GPF^{IVS} trails by the widest proportional margins, 2013, 2014, 2017, and 2023, where its pooled RMSE exceeds that of GPF^{OPS} by roughly 50% to 110%, are years of low or moderate volatility, whereas in the crisis years 2008 and 2020 the proportional gap is a comparatively modest 14% to 17%. The advantage of the price-space loss is therefore not a product of stress episodes. It is largest, and GPF^{IVS} most erratic, under the ordinary conditions in which the market spends most of its time, and it persists in smaller proportional form in the most turbulent states.

Figure 5 repeats the exercise of sorting contracts by ex-ante vega, the dimension on which the vega weighting operates directly. In panel (a), pricing RMSE rises steeply with vega for both models, which is expected because vega converts volatility errors into price errors. What that does not explain is that the two curves move apart. Through the middle deciles, the gap is one to two index points, and in the top decile, it widens sharply to roughly 29 index points for GPF^{IVS} against 18 for GPF^{OPS} . Panel (b) contains the sharper result. Volatility errors decline with vega for both models. This is the expected direction since a volatility inverted from a noisy quote is least reliable when vega is small, an argument that Section 6.2 develops formally. In the bottom vega state, where the volatility signal is weakest, the two curves move in lockstep at roughly 4.9 percentage points. As vega rises, the curves part. The GPF^{OPS} error declines through the top decile to roughly 1.8 percentage points, whilst the GPF^{IVS} error flattens and then turns upward, ending near 2.7. In the top vega decile, GPF^{OPS} is therefore roughly 30% more accurate in the criterion that GPF^{IVS} itself minimizes. The volatility loss spends much of the model’s limited capacity fitting inversion noise on the numerous low-vega contracts and gives correspondingly little weight to the contracts whose volatilities are measured most precisely and on which volatility errors translate into the largest pricing errors. The price loss, which to first order weights squared volatility errors by squared vega, allocates accuracy in the opposite direction. The prediction of Section 3.5 is therefore

confirmed in its strongest form. On high-vega contracts, the price-space loss does not trade volatility accuracy for pricing accuracy; it improves both.

[Insert Figure 5, here]

6.2 A measurement-error interpretation of the price-space loss

The vega-squared weighting that the price-space loss applies to volatility residuals also has a statistical interpretation. The observed midpoint is a noisy measurement of the option's latent value, since microstructure effects, quote discreteness, and asynchronous updating place the midpoint within the spread rather than at the value itself. Write

$$V_i = V_i^* + \omega_i \eta_i, \quad \mathbb{E}[\eta_i \mid \mathcal{F}_i] = 0, \quad \mathbb{E}[\eta_i^2 \mid \mathcal{F}_i] = 1, \quad (26)$$

where V_i^* is the latent value, η_i is a standardized noise innovation, ω_i scales the quoted noise of contract i , and $\mathcal{F}_i$ collects the contract's characteristics. The observed implied volatility is obtained by inverting the BS formula at the noisy price, and a delta-method expansion of the inverse formula around V_i^* gives

$$\sigma_i^{\text{BS}} \approx \sigma_i^* + \frac{\omega_i}{\mathcal{V}_i} \eta_i, \quad \text{Var}(\sigma_i^{\text{BS}} \mid \mathcal{F}_i) \approx \frac{\omega_i^2}{\mathcal{V}_i^2}, \quad (27)$$

where σ_i^* is the volatility implicit in the latent value. The observed implied volatility is therefore heteroskedastic by construction. Its measurement error is inversely proportional to vega, so the volatility data are noisiest exactly where vega is smallest, deep in the wings and at the shortest maturities.

Viewed through *Eq. (27)*, least squares on implied volatilities is a regression on data whose error variance is known up to scale, and the efficient generalized least squares weight on the squared volatility residual is the inverse variance $\mathcal{V}_i^2/\omega_i^2$. If quote noise is roughly constant in price units across contracts, so that $\omega_i \equiv \omega$, the efficient weight

reduces, up to scale, to $\mathcal{V}_i^2$, which is exactly the weight the price-space loss applies to first order (Section 3.5). Under this reading, GPF^{OPS} approximates feasible generalized least squares in volatility space, whilst GPF^{IVS} is the ordinary least squares analog, which places its greatest effective weight on the observations whose volatilities are measured with the least precision.

This reading fits the evidence of Section 6.1. It implies that the price-space loss should fit the volatility surface at least as well where vega is large, since those observations carry the highest signal-to-noise ratio, and that is what panel (b) of Figure 5 shows. It also motivates a robustness exercise in which each squared volatility residual is weighted by $\mathcal{V}_i^2/\hat{\omega}_i^2$, with $\hat{\omega}_i$ proxied by the contract’s bid–ask spread, relaxing the assumption that quote noise is constant across contracts.

Altogether, the conditional tests, the state curves, and the measurement-error interpretation give the choice of loss function its economic content. Conditioned on the aggregate volatility state, the price-space advantage is pervasive and statistically indistinguishable from constant. Conditioned on contract-level vega, it is pervasive, steeply increasing, and present in the volatility metric itself on the contracts whose volatilities are measured most precisely. These patterns follow jointly from defining the loss function in the space in which pricing performance is measured, a choice that weights each volatility error by how much it moves the price and, under quote noise that is roughly constant in price units, by how precisely the volatility is measured.

7 Robustness

The year-by-year and bucket-level decompositions of Section 5 already establish that the ranking of the two targets is stable across time and across the contract cross-section. We offer a range of robustness tests to buttress our interpretation, examining two remaining implementation choices on which the comparison might conceivably rest: the frequency

at which the networks are re-estimated and the parametric benchmark around which both specifications are built.

7.1 Sparser estimation: Wednesday-only training

The baseline protocol re-estimates both models monthly on all trading days of the rolling window. Table 7 repeats the exercise using only Wednesday cross-sections for estimation, roughly one fifth of the training observations, with the evaluation panel unchanged. Both specifications degrade only modestly, which is itself useful to know. Pooled RMSE rises from 9.87 to 10.85 index points for GPF^{OPS} and from 13.21 to 13.68 for GPF^{IVS} . The comparison of interest is unaffected. GPF^{OPS} retains the lower error on every metric in every block, with pooled reductions of 20.7% in RMSE, 5.0% in IVRMSE, and 26.3% in $\mathcal{V}$ RMSE, all significant at the 1% level, and the monotone widening of the advantage in the vega loading of the metric is preserved. The price-space advantage does not depend on daily estimation data.

[Insert Table 7, here]

7.2 Alternative parametric benchmark

Both baseline specifications are built around the nonlinear DVF surface, and one may ask whether the price-space advantage reflects an interaction between that particular benchmark and the loss. Table 8 replaces the DVF benchmark with the constant-volatility BS benchmark of Section 3.1 in both specifications, leaving the features, architectures, and rolling protocol unchanged. The benchmark is now deliberately crude. It provides no smile and no term structure, so the networks must learn the entire shape of the surface. The ranking is undisturbed. GPF^{OPS} attains pooled errors of 10.39 index points (RMSE), 2.77 percentage points (IVRMSE), and 14.19 index points ($\mathcal{V}$ RMSE), against 13.33, 2.94, and 20.14 for GPF^{IVS} , reductions of 22.1%, 5.8%, and 29.5%, all significant

at the 1% level. Two further readings are worth recording. First, GPF^{OPS} with the BS benchmark (10.39) still prices more accurately than GPF^{IVS} with the DVF benchmark (13.21 in Table 3). Within this framework, the choice of loss space matters more than the choice of parametric benchmark. Second, the advantage is nearly invariant to the benchmark (25.3% under DVF, 22.1% under flat BS), so it is a property of the optimization target rather than of any particular parametric starting point.

[Insert Table 8, here]

8 Conclusions

Machine learning option pricing models are evaluated on prices, and the literature has heretofore divided on whether to train on prices or on implied volatilities without examining the consequences. The conversion between the two is not an accounting step but a reweighting. To first order, the squared pricing error is the squared volatility error weighted by squared vega, so the two training targets pose different statistical problems, and theory alone does not say which one prices more accurately out of sample. Holding every other element of the GPF framework fixed, we measure the difference on 8.67 million S&P 500 index-option observations priced out of sample between 1997 and 2025.

The price-space target reduces pooled pricing RMSE by 25.3% relative to the volatility-space target, nearly twice what the network itself adds over the parametric benchmark, attains the lower RMSE in every sample year, and delivers the lower implied-volatility RMSE as well. The advantage rises with the vega loading of the evaluation metric and with contract vega, and it survives estimation on Wednesday-only training data and a flat constant-volatility benchmark, a pattern consistent with the pricing loss acting as feasible generalized least squares on noisily measured volatilities. Adopting it amounts to a single line in the loss function.

Our evidence comes from one underlying asset, one family of architectures, and pricing-based criteria; delta-hedging and trading performance, asset classes with larger quote noise, and richer architectures are natural extensions. For research practice, the training space is a first-order design choice that belongs alongside architecture, features, and sample construction in the description of any machine learning pricing model.

References

- Ackerer, D., Tagasovska, N., and Vatter, T. (2020). Deep smoothing of the implied volatility surface. In *Advances in Neural Information Processing Systems*, volume 33, pages 11552–11563.
- Aït-Sahalia, Y. and Duarte, J. (2003). Nonparametric option pricing under shape restrictions. *Journal of Econometrics*, 116(1–2):9–47.
- Almeida, C., Fan, J., Freire, G., and Tang, F. (2023). Can a machine correct option pricing models? *Journal of Business & Economic Statistics*, 41(3):995–1009.
- Amilon, H. (2003). A neural network versus Black–Scholes: A comparison of pricing and hedging performances. *Journal of Forecasting*, 22(4):317–335.
- Anders, U., Korn, O., and Schmitt, C. (1998). Improving the pricing of options: A neural network approach. *Journal of Forecasting*, 17(5–6):369–388.
- Andersen, T. G., Fusari, N., Todorov, V., and Varneskov, R. T. (2021). Spatial dependence in option observation errors. *Econometric Theory*, 37:205–247.
- Andreou, P. C., Charalambous, C., and Martzoukos, S. H. (2008). Pricing and trading European options by combining artificial neural networks and parametric models with implied parameters. *European Journal of Operational Research*, 185(3):1415–1433.
- Andreou, P. C., Charalambous, C., and Martzoukos, S. H. (2010). Generalized parameter functions for option pricing. *Journal of Banking & Finance*, 34(3):633–646.
- Andreou, P. C., Charalambous, C., and Martzoukos, S. H. (2014). Assessing the performance of symmetric and asymmetric implied volatility functions. *Review of Quantitative Finance and Accounting*, 42(3):373–397.
- Bakshi, G., Cao, C., and Chen, Z. (1997). Empirical performance of alternative option pricing models. *The Journal of Finance*, 52(5):2003–2049.

- Bali, T. G., Beckmeyer, H., Mörke, M., and Weigert, F. (2023). Option return predictability with machine learning and big data. *Review of Financial Studies*, 36:3548–3602.
- Bams, D., Lehnert, T., and Wolff, C. C. P. (2009). Loss functions in option valuation: A framework for selection. *Management Science*, 55(5):853–865.
- Birke, M. and Pilz, K. F. (2009). Nonparametric option pricing with no-arbitrage constraints. *Journal of Financial Econometrics*, 7(2):53–76.
- Black, F. and Scholes, M. (1973). The pricing of options and corporate liabilities. *Journal of Political Economy*, 81(3):637–654.
- Büchner, M. and Kelly, B. (2022). A factor model for option returns. *Journal of Financial Economics*, 143(3):1140–1161.
- Cao, Y., Liu, X., and Zhai, J. (2021). Option valuation under no-arbitrage constraints with neural networks. *European Journal of Operational Research*, 293(1):361–374.
- Carelli, A., Silani, S., and Stella, F. (2000). Profiling neural networks for option pricing. *International Journal of Theoretical and Applied Finance*, 3(2):183–204.
- Chen, H., Didisheim, A., and Scheidegger, S. (2026). Deep surrogates for finance: With an application to option pricing. *Journal of Financial Economics*, 177:104222.
- Chen, L., Pelger, M., and Zhu, J. (2024). Deep learning in asset pricing. *Management Science*, 70(2):714–750.
- Christoffersen, P. and Jacobs, K. (2004). The importance of the loss function in option valuation. *Journal of Financial Economics*, 72(2):291–318.
- Diebold, F. X. and Mariano, R. S. (1995). Comparing predictive accuracy. *Journal of Business & Economic Statistics*, 13(3):253–263.

- Dumas, B., Fleming, J., and Whaley, R. E. (1998). Implied volatility functions: Empirical tests. *The Journal of Finance*, 53(6):2059–2106.
- Fan, J. and Mancini, L. (2009). Option pricing with model-guided nonparametric methods. *Journal of the American Statistical Association*, 104(488):1351–1372.
- Garcia, R. and Gençay, R. (2000). Pricing and hedging derivative securities with neural networks and a homogeneity hint. *Journal of Econometrics*, 94(1–2):93–115.
- Giacomini, R. and White, H. (2006). Tests of conditional predictive ability. *Econometrica*, 74(6):1545–1578.
- Gu, S., Kelly, B., and Xiu, D. (2020). Empirical asset pricing via machine learning. *The Review of Financial Studies*, 33(5):2223–2273.
- Horvath, B., Muguruza, A., and Tomas, M. (2021). Deep learning volatility: A deep neural network perspective on pricing and calibration in (rough) volatility models. *Quantitative Finance*, 21:11–27.
- Hutchinson, J. M., Lo, A. W., and Poggio, T. (1994). A nonparametric approach to pricing and hedging derivative securities via learning networks. *The Journal of Finance*, 49(3):851–889.
- Kelly, B. and Xiu, D. (2023). Financial machine learning. *Foundations and Trends® in Finance*, 13(3-4):205–363.
- Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.
- Kohler, M., Krzyżak, A., and Todorovic, N. (2010). Pricing of high-dimensional American options by neural networks. *Mathematical Finance*, 20(3):383–410.
- Liu, S., Oosterlee, C. W., and Bohte, S. M. (2019). Pricing options and computing implied volatilities using neural networks. *Risks*, 7(1):16.

- Raberto, M., Cuniberti, G., Riani, M., Scalas, E., Mainardi, F., and Servizi, G. (2000). Learning short-option valuation in the presence of rare events. *International Journal of Theoretical and Applied Finance*, 3(3):563–564.
- Rubinstein, M. (1994). Implied binomial trees. *The Journal of Finance*, 49(3):771–818.
- Ruf, J. and Wang, W. (2020). Neural networks for option pricing and hedging: a literature review. *Journal of Computational Finance*, 24(1):1–46.
- Ulrich, M., Zimmer, L., and Merbecks, C. (2023). Implied volatility surfaces: A comprehensive analysis using half a billion option prices. *Review of Derivatives Research*, 26:135–169.
- Vinje, J., Rygg, E. S., Wu, C., Rissstad, M., Pimentel, R., Westgaard, S., and Ewald, C. O. (2025). Merged LSTM-MLP for option valuation. *Quantitative Finance*, 25:1679–1694.
- Yang, Y., Zheng, Y., and Hospedales, T. (2017). Gated neural networks for option pricing: Rationality by design. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.
- Yao, J., Li, Y., and Tan, C. L. (2000). Option price forecasting using neural networks. *Omega*, 28(4):455–466.
- Zapart, C. (2002). Stochastic volatility options pricing with wavelets and artificial neural networks. *Quantitative Finance*, 2(6):487–495.

Figures

Figure 1: Keywords used in the machine learning option valuation literature

This figure depicts the terms that occur most frequently in the full text of the machine learning option valuation literature located by our search. The corpus comprises 268 studies published between 1994 and 2026 whose text could be extracted; two image-only scans and one misfiled duplicate are excluded. Terms are one- to three-word phrases extracted from the complete text of each study, normalized for case, plurals, and hyphenation, with journal and publisher boilerplate removed; a study counts once for a term if the term occurs at least twice in it, and a phrase is suppressed when a longer phrase containing it retains at least 75% of its studies. The 200 most frequent terms are shown, with size proportional to the number of studies using the term. The largest terms are “option price” and “option pricing” (212 studies each), “neural network” (180), “implied volatility” (174), and “call option” (171). For comparison, “loss function” occurs in 117 studies, “objective function” in 79, “pricing error” in 88, a phrase denominating the error in implied volatilities in 16, and a vega weighting in 5.

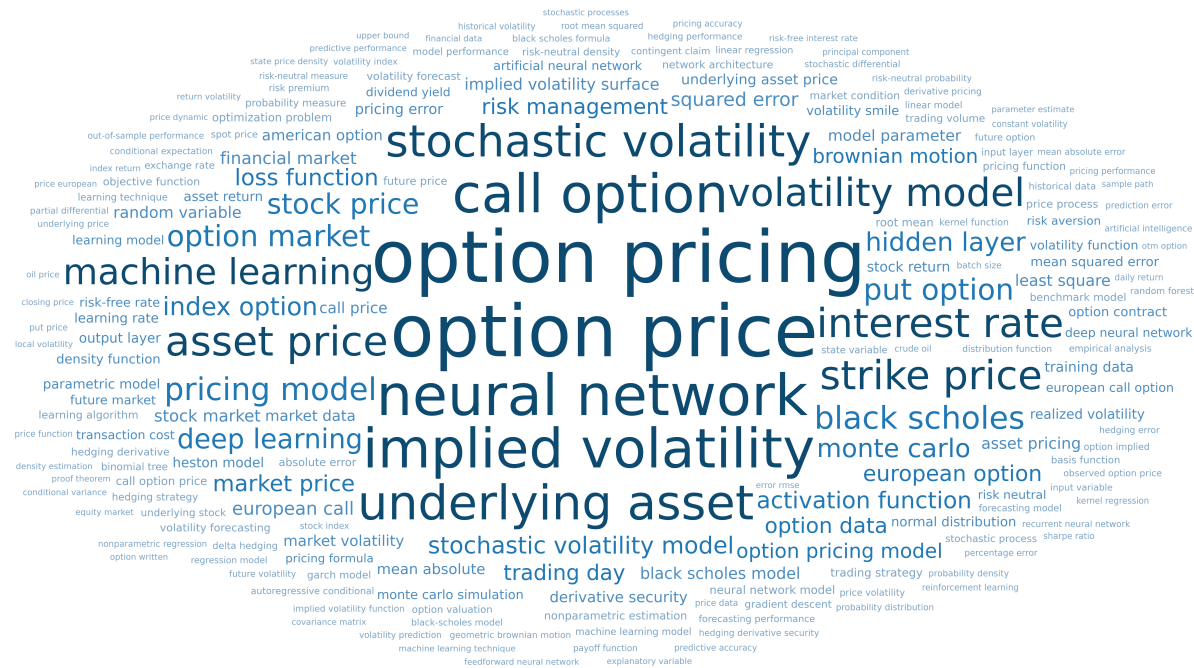


Figure 2: Training spaces and error criteria in the machine learning option pricing literature

Both panels cover the 117 studies, drawn from the 268 readable studies located by our search, whose loss function operates in option-price space, implied-volatility space, or both. Classification is made from the methodology sections of each study, that is, the text between the end of the introduction and the conclusion or reference list, so that criteria attributed to other papers in introductions and literature reviews do not enter the counts. Studies whose objective lies in neither space are excluded: 43 forecast realized or index volatility, 40 estimate risk-neutral densities or pricing kernels, 30 minimize hedging error or a reinforcement-learning reward, 8 apply entropy or canonical valuation, 7 derive pricing bounds or approximation results, 4 predict option returns, 3 are surveys, and 16 pursue other objectives. (a) reports the number of the 117 studies in five-year periods, stacked by training space, with the total above each bar; the final period covers 2024 to 2026. “Both spaces” indicates that the study estimates in the two spaces, in every case for different model components. (b) reports the share of the same 117 studies that define or report each error criterion in those sections, with the number of studies next to each bar; a study counts once per criterion, criteria are not mutually exclusive, and matches that only describe other papers, enumerate losses generically, or arise from adjacent table columns are excluded by hand. Percentage errors comprise MAPE, MPE, and related relative measures; the implied-volatility RMSE bar counts studies reporting an error explicitly denominated in implied volatilities, whilst studies that fit volatility functions but label their errors generically are counted under the generic criterion; the vega-weighted bar counts studies whose criterion weights errors by vega. No search of this kind is exhaustive, and the counts should be read as indicative of practice rather than as a census.

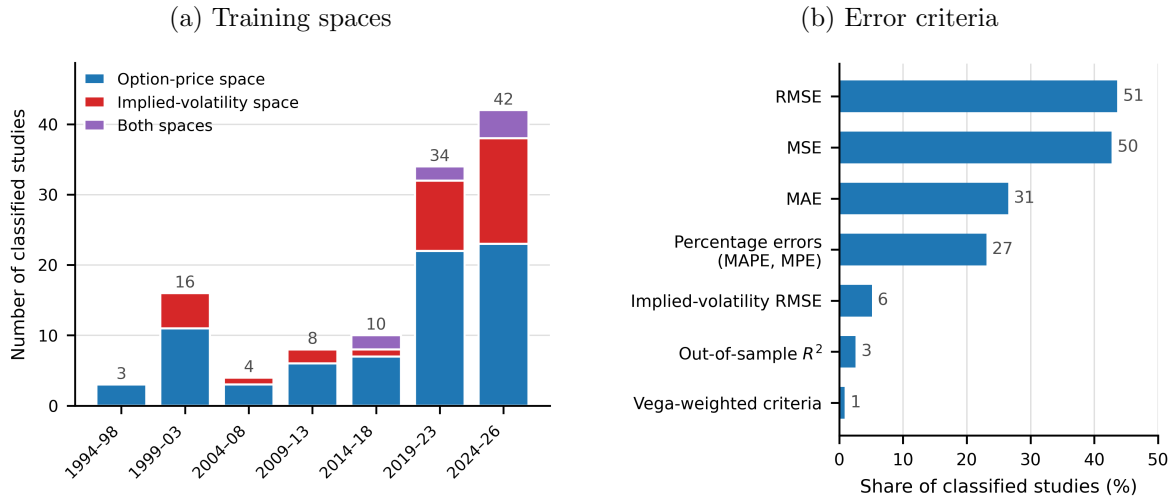


Figure 3: The two GPF specifications

This figure illustrates the architecture of the two GPF specifications, which share the same features, network, and BS pricing function and differ only in the space in which the training loss is evaluated. $\mathbf{X}$ is the complete set of input features of Eq. (7), and $\mathbf{X}^{(-\sigma)}$ is the same set excluding the implied-volatility feature. A hat denotes a model-generated quantity and an unhatted variable its observed market counterpart. **(a) GPF^{OPS}.** The neural network $F(\cdot; \theta)$ processes contract features $\mathbf{X}$ to produce an enhanced implied volatility ($\hat{\sigma}^{enh}$), an internal variable of the model architecture rather than an estimation target. Both the features $\mathbf{X}^{(-\sigma)}$ and the enhanced volatility flow into the embedded BS pricing layer, which applies the Black–Scholes formula of Eq. (1)–(2) to produce the option price $\hat{V}^{OPS}$. The training loss is evaluated on this price, as per Eq. (11). The red dotted box encompasses the entire architecture, highlighting that GPF^{OPS} is an end-to-end model. **(b) GPF^{IVS}.** The neural network $G(\cdot; \phi)$ processes contract features $\mathbf{X}$ to produce an implied-volatility residual $\hat{\epsilon}^{\sigma}$, on which the training loss is evaluated, as per Eq. (15). Both $\mathbf{X}$ and the predicted residual are then passed to an external BS pricing layer, which combines the parametric baseline with the learned residual to generate the option price $\hat{V}^{IVS}$. The red dotted box encloses the nonparametric learning component alone, emphasizing the two-step construction.

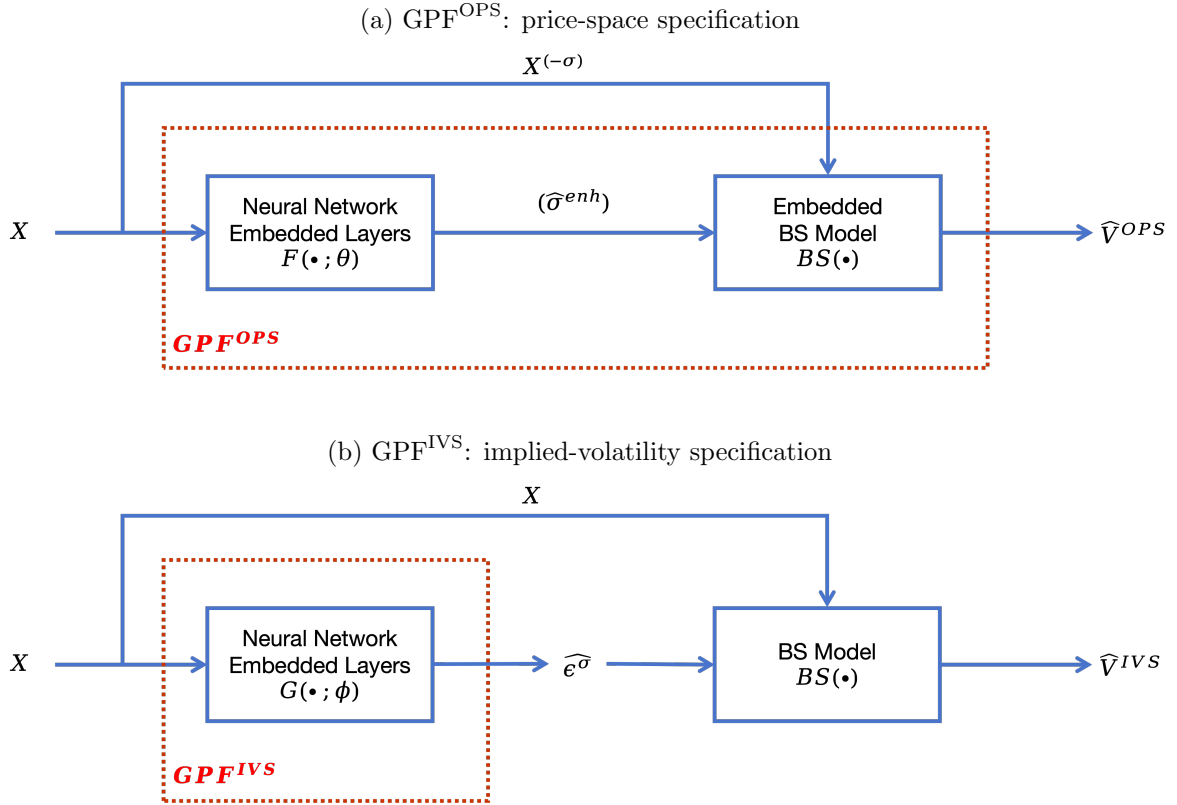


Figure 4: Out-of-sample pricing accuracy across lagged implied-volatility states

The figure reports pooled out-of-sample error curves across ten lagged implied-volatility states for the full option universe, January 1997 through August 2025, calls and puts combined. GPF^{IVS} and GPF^{OPS} differ only in the space in which the training loss is evaluated, volatility residuals for GPF^{IVS} as per *Eq. (15)* and strike-normalized prices for GPF^{OPS} as per *Eq. (11)*. The conditioning variable is the one-day-lagged implied volatility used as a model input, which is constant across contracts observed on a given date. Forecast dates are sorted into ten equal-frequency states, and the horizontal axis reports the mean lagged implied volatility within each state in percentage points. (a) reports the root mean squared pricing error in S&P 500 index points, with the \$100 contract multiplier not applied, and (b) the implied-volatility RMSE in percentage points. GPF^{OPS} attains the lower pricing RMSE in every state and its curve rises with the volatility level, whereas the GPF^{IVS} curve is irregular and spikes in the third state, at a mean lagged implied volatility near 14.5%. The volatility-error curves in panel (b) remain close throughout, so the pricing gap in panel (a) reflects where volatility errors fall across contracts rather than their overall size.

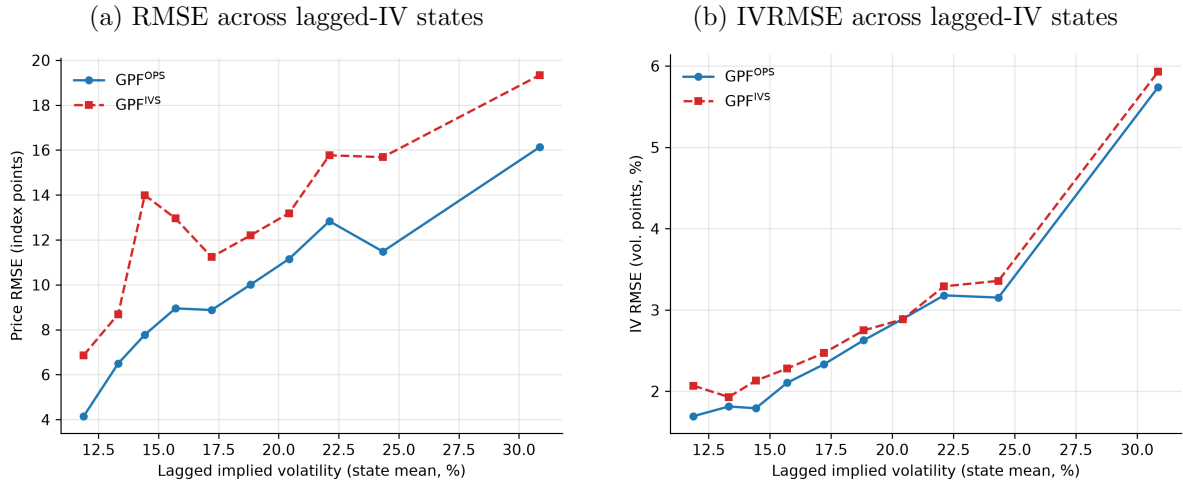
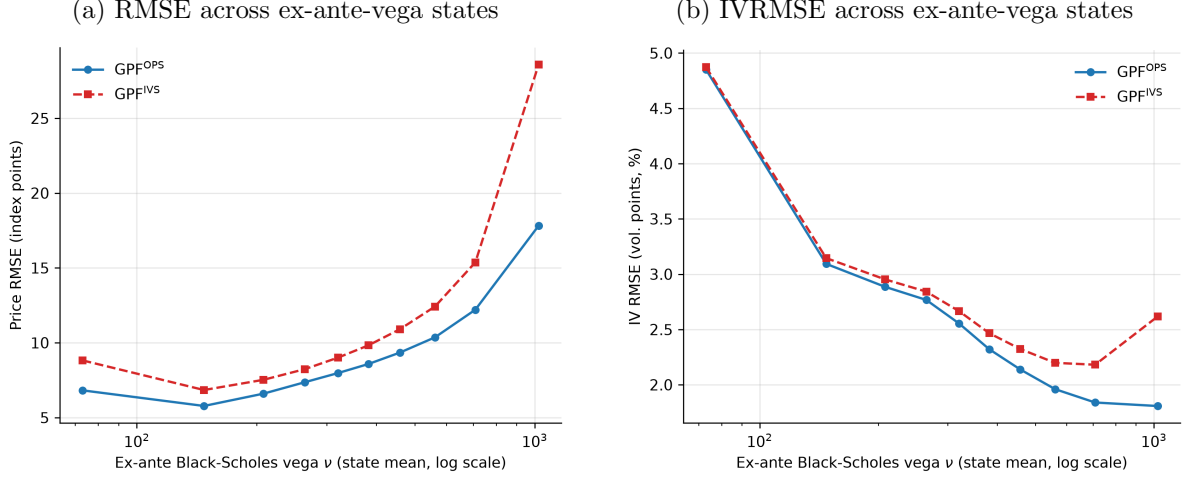


Figure 5: Out-of-sample pricing accuracy across ex-ante-vega states

The figure reports pooled out-of-sample error curves across ten ex-ante vega states for the full option universe, January 1997 through August 2025, calls and puts combined. GPF^{IVS} and GPF^{OPS} differ only in the space in which the training loss is evaluated, volatility residuals for GPF^{IVS} as per *Eq. (15)* and strike-normalized prices for GPF^{OPS} as per *Eq. (11)*. For each contract, BS vega is computed from S , K , T , r , q , and the one-day-lagged implied volatility input, so the sorting variable uses only information available when the forecast is formed. Contracts are sorted into ten equal-frequency states within each date and option type, which removes variation that merely tracks the level of the index. The horizontal axis reports the state-mean vega on a logarithmic scale. Vega is expressed per unit of implied volatility, one hundred times the per-volatility-point convention used elsewhere in the paper; state assignments and error statistics do not depend on this scaling. **(a)** reports the root mean squared pricing error in S&P 500 index points, with the \$100 contract multiplier not applied, and **(b)** the implied-volatility RMSE in percentage points. Pricing errors rise with vega for both models and the gap between them widens in the highest states. In panel (b), the GPF^{OPS} volatility error declines with vega whilst the GPF^{IVS} error turns upward in the top states, so the price-space loss is more accurate on both metrics precisely where vega, and with it the effect of a volatility error on the price, is greatest.



Tables

Table 1: Estimation protocol and machine learning configuration

Every setting is identical across GPF^{OPS} and GPF^{IVS} , which differ only in the space in which the loss function is evaluated. $\mathcal{T}$ denotes the test month, so the training window $[\mathcal{T} - 12, \mathcal{T} - 2)$ covers the ten months ending two months before it and the validation window $[\mathcal{T} - 2, \mathcal{T})$ the two months immediately preceding it; the three windows are disjoint and ordered in calendar time, so no information from the test month enters estimation or model selection. The rolling-forward partition follows [Andreou et al. \(2010\)](#). The hidden-layer size is selected separately for each of the five seeds, which govern weight initialization and batch shuffling, so the retained networks need not share a size. The GPF^{OPS} loss is evaluated on strike-normalized prices as per [Eq. \(11\)](#); the GPF^{IVS} loss is evaluated on volatility residuals in levels as per [Eq. \(15\)](#).

Panel A: Rolling protocol

Training, validation, and test windows	$[\mathcal{T} - 12, \mathcal{T} - 2)$, $[\mathcal{T} - 2, \mathcal{T})$, and $\mathcal{T}$ (10, 2, and 1 months)
Refitting	Monthly, fixed-length window
Test blocks	344, non-overlapping, January 1997 to August 2025
Deterministic volatility function	Lagged one day
Training cross-section	Strikes on the \$10 grid only

Panel B: Neural network

Hidden layers	One, with 32, 16, or 8 neurons, selected per seed by lowest validation loss
Hidden-layer normalization	Batch normalization
Hidden-layer activation	Sigmoid
Output layer	Linear
Random seeds	5
Prediction	Equal-weighted average across the 5 seeds
Feature standardization	Training-window statistics only

Panel C: Optimization

Optimizer and learning rate	Adam (Kingma and Ba, 2015), 10^{-3} as in Cao et al. (2021)
Batch size, clipping, weight decay	1,024, 1.0, 10^{-4}
Epochs	At most 200, stopping after 5 without validation improvement

Table 2: Summary statistics

The sample contains all S&P 500 index-option contracts from January 1996 to August 2025 surviving the filters of Section 4.1, namely a positive best bid, a non-negative bid–ask spread, positive open interest and trading volume, a vendor implied volatility within $[2\%, 300\%]$, 6 to 365 calendar days to maturity, and an absolute BS delta within $[0.05, 0.95]$. The market price V is the bid–ask midpoint (index points), and the implied volatility σ is recovered from it by inverting the BS formula at the row-specific interest rate and dividend yield. σ^{DVF} is the deterministic volatility function of Eq. (4), estimated on the previous trading day and evaluated at the contract’s own coordinates. Moneyness is reported as S/K ; the network’s input is the forward-adjusted moneyness $m = (S/K) \exp[(r - q)\tau]$ of Eq. (5). Volatilities are in decimal form and maturity in calendar days. Vega is the BS vega of Eq. (21), expressed as the index-point price change associated with a one-percentage-point change in implied volatility.

Variable	Mean	Std. Dev.	Min	p25	Median	p75	Max
Panel A: Calls ($n = 3,686,008$)							
Mid price V (pts)	97.78	144.49	0.25	17.25	51.90	121.30	5354.65
Implied volatility σ	0.178	0.079	0.046	0.126	0.161	0.212	7.635
Moneyness S/K	0.989	0.085	0.579	0.961	0.988	1.013	22.75
Time to maturity T (days)	68.0	74.1	6	21	39	84	365
DVF prior-day IV σ^{DVF}	0.166	0.072	0.010	0.118	0.150	0.197	2.999
BS vega $\mathcal{V}$ (\$)	4.46	3.52	0.005	2.02	3.41	5.77	25.45
Panel B: Puts ($n = 5,007,928$)							
Mid price V (pts)	61.93	75.47	0.38	15.15	36.10	79.45	1826.00
Implied volatility σ	0.201	0.086	0.017	0.143	0.187	0.242	2.405
Moneyness S/K	1.059	0.085	0.571	1.006	1.039	1.090	2.284
Time to maturity T (days)	71.5	74.2	6	22	44	91	365
DVF prior-day IV σ^{DVF}	0.206	0.082	0.010	0.148	0.190	0.245	1.556
BS vega $\mathcal{V}$ (\$)	4.20	3.31	0.098	1.88	3.22	5.47	25.44

Table 3: Pooled out-of-sample performance

All four models are estimated on the filtered universe of S&P 500 index options under the rolling protocol of Table 1 and evaluated out of sample over the 344 test months from January 1997 to August 2025. BS is the constant-volatility benchmark and DVF the ad hoc Black–Scholes model with the nonlinear deterministic volatility function of Eq. (4), both estimated on the preceding trading day’s cross-section. GPF^{IVS} and GPF^{OPS} share the same features, architecture, tuning rules, and rolling dates, and differ only in the space in which the training loss is evaluated, volatility residuals for GPF^{IVS} as per Eq. (15) and strike-normalized prices for GPF^{OPS} as per Eq. (11). RMSE is the pricing-error root mean squared error and $\mathcal{V}\text{RMSE}$ the normalized vega-weighted pricing RMSE of Eq. (20), both in index points with the \$100 contract multiplier not applied; IVRMSE is the implied-volatility RMSE in percentage points. Vega is evaluated at the observed implied volatility. Obs. is the number of contracts on which all four models produce a price forecast. Bold entries identify the lowest error within each option-type block. Superscripts on the GPF^{OPS} row denote Diebold–Mariano significance against the GPF^{IVS} benchmark, with standard errors clustered by trading date: $*p < 0.10$, $**p < 0.05$, $***p < 0.01$. No tests against BS or DVF are reported.

	Model	RMSE (pts)	IVRMSE (pp)	$\mathcal{V}\text{RMSE}$ (pts)
Calls (Obs. = 3,676,030)	BS	23.35	5.59	34.89
	DVF	17.46	3.57	27.05
	GPF^{IVS}	14.33	3.19	21.58
	GPF^{OPS}	10.57***	2.87***	14.48***
Puts (Obs. = 4,991,872)	BS	20.96	6.13	28.83
	DVF	13.42	2.79	20.24
	GPF^{IVS}	12.33	2.83	17.87
	GPF^{OPS}	9.31***	2.69***	12.52***
Pooled (Obs. = 8,667,902)	BS	22.01	5.91	31.63
	DVF	15.27	3.15	23.48
	GPF^{IVS}	13.21	2.99	19.58
	GPF^{OPS}	9.87***	2.77***	13.42***

Table 4: Year-by-year pooled out-of-sample performance

The table decomposes by calendar year the pooled comparison of Table 3. GPF^{IVS} and GPF^{OPS} share the same features, architecture, tuning rules, and rolling dates, and differ only in the space in which the training loss is evaluated, volatility residuals for GPF^{IVS} as per Eq. (15) and strike-normalized prices for GPF^{OPS} as per Eq. (11). All entries pool calls and puts. RMSE is the pricing-error root mean squared error and $\mathcal{V}\text{RMSE}$ the normalized vega-weighted pricing RMSE of Eq. (20), both in index points with the \$100 contract multiplier not applied; IVRMSE is the implied-volatility RMSE in percentage points. Vega is evaluated at the observed implied volatility. Obs. is the number of contracts in the year on which both models produce a price forecast, and the yearly counts sum to 8,667,902, the pooled total in Table 3; 2025 covers January to August. Bold marks the lower error within each metric-year pair. Superscripts report Diebold–Mariano tests of equal predictive accuracy between GPF^{IVS} and GPF^{OPS} , with standard errors clustered by trading date and attached to the more accurate model: $^*p < 0.10$, $^{**}p < 0.05$, $^{***}p < 0.01$.

Year	Obs.	RMSE (pts)		IVRMSE (pp)		$\mathcal{V}\text{RMSE}$ (pts)	
		GPF^{IVS}	GPF^{OPS}	GPF^{IVS}	GPF^{OPS}	GPF^{IVS}	GPF^{OPS}
1997	33,884	3.22	3.11[*]	3.08	3.10	3.39	3.22^{***}
1998	33,244	3.20	3.13[*]	2.90[*]	2.93	3.81	3.67^{**}
1999	31,455	3.02	2.80^{***}	2.40	2.37^{***}	3.53	3.21^{***}
2000	24,586	5.19	4.83^{***}	3.06	3.05	6.62	6.09^{***}
2001	25,164	4.69	4.22^{***}	2.94	2.92	6.14	5.41^{***}
2002	28,409	3.22	3.13	2.84^{***}	2.95	4.08	3.80^{**}
2003	30,239	3.05	2.58^{***}	2.32	2.18^{***}	4.01	3.34^{***}
2004	30,739	4.20	3.09^{***}	2.50	2.29^{***}	5.51	3.79^{***}
2005	31,835	2.99	2.77^{***}	1.84^{***}	1.90	3.85	3.45^{***}
2006	36,963	3.04	2.37^{***}	1.69	1.64^{**}	4.01	2.93^{***}
2007	47,048	5.72	5.16^{***}	2.56	2.50	7.68	6.81^{***}
2008	60,060	6.14	5.37^{***}	4.16^{**}	4.41	8.01	6.14^{***}
2009	61,147	2.56	2.47^{**}	2.38	2.43	3.08	2.85^{***}
2010	64,295	3.48	3.04^{***}	2.50	2.43^{**}	4.40	3.64^{***}
2011	73,769	4.45	3.77^{***}	3.19	3.13	5.65	4.42^{***}
2012	86,438	2.88	2.10^{***}	1.80	1.65^{***}	3.83	2.57^{***}
2013	117,160	4.56	2.72^{***}	2.10	1.78^{***}	6.37	3.45^{***}
2014	158,897	4.29	2.85^{***}	2.10	1.88^{***}	6.04	3.59^{***}
2015	236,690	4.85	4.11^{***}	2.36	2.39	6.44	4.88^{***}
2016	294,773	4.57	3.40^{***}	2.05	1.88^{***}	6.25	4.25^{***}
2017	378,025	7.24	3.49^{***}	2.28	1.71^{***}	10.59	4.46^{***}
2018	526,994	9.81	8.42^{***}	3.13^{***}	3.37	13.78	10.87^{***}
2019	546,911	6.41	5.10^{***}	1.98	1.76^{***}	8.70	6.65^{***}
2020	766,356	16.63	14.26^{***}	5.52	5.33^{***}	20.71	16.80^{***}
2021	849,352	12.16	8.41^{***}	2.57	2.22^{***}	15.82	9.96^{***}
2022	1,060,997	15.87	11.44^{***}	2.64	2.39^{***}	22.12	14.99^{***}
2023	1,051,518	14.23	8.39^{***}	2.43	2.12^{***}	20.44	10.90^{***}
2024	1,113,532	14.76	11.16^{***}	2.23	2.08^{***}	20.39	14.48^{***}
2025	867,422	19.56	15.25^{***}	3.03	2.90	25.10	17.91^{***}

Table 5: Out-of-sample performance by moneyness and maturity

The table decomposes the pooled comparison of Table 3 across the two contract dimensions that drive vega. GPF^{IVS} and GPF^{OPS} share the same features, architecture, tuning rules, and rolling dates, and differ only in the space in which the training loss is evaluated, volatility residuals for GPF^{IVS} as per Eq. (15) and strike-normalized prices for GPF^{OPS} as per Eq. (11). For calls, DOTM, OTM, ATM, ITM, and DITM correspond to $S/K < 0.94$, $[0.94, 0.97)$, $[0.97, 1.03]$, $(1.03, 1.06]$, and > 1.06 ; for puts the economic labels are reversed along the same numerical intervals. Maturity is measured in calendar days. RMSE is the pricing-error root mean squared error and $\mathcal{V}\text{RMSE}$ the normalized vega-weighted pricing RMSE of Eq. (20), both in index points with the \$100 contract multiplier not applied; IVRMSE is the implied-volatility RMSE in percentage points. Vega is evaluated at the observed implied volatility. Obs. is the number of contracts in the bucket on which both models produce a price forecast. The two panels are separate one-dimensional decompositions of the same evaluation sample, so their counts each sum to the pooled total of Table 3 within option type. Bold marks the lower error within each metric-bucket pair. Superscripts report Diebold–Mariano tests of equal predictive accuracy between GPF^{IVS} and GPF^{OPS} , with standard errors clustered by trading date and attached to the more accurate model: $*p < 0.10$, $**p < 0.05$, $***p < 0.01$.

Option	Bucket	Obs.	RMSE (pts)		IVRMSE (pp)		VRMSE (pts)	
			GPF ^{IVS}	GPF ^{OPS}	GPF ^{IVS}	GPF ^{OPS}	GPF ^{IVS}	GPF ^{OPS}
Panel A: Moneyness buckets								
Call	DOTM	512,216	12.36	12.13	3.27**	3.37	16.52	16.08
Call	OTM	666,844	10.59	9.59**	2.35	2.32	16.11	13.79***
Call	ATM	1,962,515	12.38	9.43***	2.42	2.15***	18.41	12.57***
Call	ITM	293,669	17.31	10.63***	3.14	2.84***	29.01	15.62***
Call	DITM	240,786	29.71	16.64***	6.94	6.20***	46.72	23.47***
Put	DOTM	1,873,619	10.05	8.26***	2.72	2.69	13.65	11.01***
Put	OTM	940,010	9.34	7.65***	2.21	2.22	14.20	10.62***
Put	ATM	1,921,731	13.14	9.48***	2.42	2.19***	18.87	12.23***
Put	ITM	172,269	23.82	15.42***	4.22	3.78***	33.10	20.43***
Put	DITM	84,243	26.21	20.63***	8.67	8.53**	34.27	27.64***
Panel B: Maturity buckets								
Call	Short (< 60 days)	2,380,569	8.61	7.67**	3.15	3.01***	10.20	8.93**
Call	Medium (60–120 days)	699,096	12.30	10.55***	2.42	2.22***	14.23	12.02***
Call	Long (≥ 120 days)	596,365	28.15	18.00***	3.52	2.92***	33.58	20.65***
Put	Short (< 60 days)	3,092,135	7.51	7.07***	3.00*	3.06	9.07	8.39***
Put	Medium (60–120 days)	1,012,664	11.93	9.38***	2.27	1.90***	13.93	10.96***
Put	Long (≥ 120 days)	887,073	22.27	14.59***	2.62	1.96***	26.24	16.88***

Table 6: Conditional predictive ability across lagged-IV and ex-ante-vega states

The table reports Giacomini–White tests (Giacomini and White, 2006) of whether the relative accuracy of GPF^{IVS} and GPF^{OPS} varies with information available when the forecast is formed. The two specifications share the same features, architecture, tuning rules, and rolling dates, and differ only in the space in which the training loss is evaluated, volatility residuals for GPF^{IVS} as per Eq. (15) and strike-normalized prices for GPF^{OPS} as per Eq. (11). The daily loss differential d_u of Eq. (22) is the mean squared pricing error of GPF^{IVS} less that of GPF^{OPS} , so positive values favor GPF^{OPS} , and Mean d_u is its average over the 7,208 trading days from January 1997 to August 2025, in squared index points. In Panel A forecast dates are sorted into ten equal-frequency states by the one-day-lagged implied volatility used as a model input; in Panel B contracts are sorted into ten equal-frequency states by BS vega computed from that lagged input, within date and option type, so both conditioning variables are known ex ante. Conditional equality tests whether the ten state-specific mean loss differentials are jointly zero, as per Eq. (23); state dependence tests whether the ten means are equal, as per Eq. (24). Both are HAC Wald statistics of Eq. (25) with Newey–West covariances at 22 lags, and the unconditional t statistic uses the same variance estimator. All estimated state-specific differentials are positive. Parentheses contain p -values. *** and ** denote significance at the 1% and 5% levels, respectively.

Sample	Mean d_u	Unconditional t	Conditional equality $\chi^2(10)$	State dependence $\chi^2(9)$
Panel A: Lagged implied-volatility deciles				
Calls	33.20	6.73*** (< 0.001)	69.00*** (< 0.001)	9.17 (0.422)
Puts	23.42	5.86*** (< 0.001)	56.02*** (< 0.001)	18.70** (0.028)
All options	27.56	7.03*** (< 0.001)	72.79*** (< 0.001)	12.67 (0.178)
Panel B: Ex-ante-vega deciles				
Calls	33.20	6.73*** (< 0.001)	97.87*** (< 0.001)	47.70*** (< 0.001)
Puts	23.42	5.86*** (< 0.001)	67.12*** (< 0.001)	65.48*** (< 0.001)
All options	27.56	7.03*** (< 0.001)	88.34*** (< 0.001)	46.89*** (< 0.001)

Table 7: Robustness: Wednesday-only training

The table repeats Table 3 with the two machine learning specifications estimated on Wednesday cross-sections only, roughly one fifth of the training observations, leaving the features, architectures, tuning rules, and rolling dates unchanged; the evaluation panel is the full daily panel of Table 3. BS is the constant-volatility benchmark and DVF the ad hoc Black–Scholes model with the nonlinear deterministic volatility function of Eq. (4), both estimated on the preceding trading day’s cross-section. GPF^{IVS} and GPF^{OPS} differ only in the space in which the training loss is evaluated, volatility residuals for GPF^{IVS} as per Eq. (15) and strike-normalized prices for GPF^{OPS} as per Eq. (11). RMSE is the pricing-error root mean squared error and $\mathcal{V}\text{RMSE}$ the normalized vega-weighted pricing RMSE of Eq. (20), both in index points with the \$100 contract multiplier not applied; IVRMSE is the implied-volatility RMSE in percentage points. Vega is evaluated at the observed implied volatility. Obs. is the number of contracts on which all four models produce a price forecast. Bold entries identify the lowest error within each option-type block. Superscripts on the GPF^{OPS} row denote Diebold–Mariano significance against the GPF^{IVS} benchmark, with standard errors clustered by trading date: $*p < 0.10$, $**p < 0.05$, $***p < 0.01$. No tests against BS or DVF are reported.

	Model	RMSE (pts)	IVRMSE (pp)	$\mathcal{V}\text{RMSE}$ (pts)
Calls (Obs. = 3,676,030)	BS	23.40	5.59	34.92
	DVF	17.53	3.57	27.13
	GPF^{IVS}	15.29	3.21	23.25
	GPF^{OPS}	11.29***	2.94***	15.76***
Puts (Obs. = 4,991,872)	BS	21.02	6.12	28.88
	DVF	13.50	2.79	20.33
	GPF^{IVS}	12.36	2.81	18.04
	GPF^{OPS}	10.51***	2.74***	14.58***
Pooled (Obs. = 8,667,902)	BS	22.06	5.90	31.67
	DVF	15.34	3.15	23.56
	GPF^{IVS}	13.68	2.98	20.50
	GPF^{OPS}	10.85***	2.83***	15.11***

Table 8: Robustness: Alternative parametric benchmark

The table repeats Table 3 with the two machine learning specifications built around the constant-volatility BS benchmark in place of the DVF surface, leaving the features, architectures, tuning rules, and rolling dates unchanged; the evaluation panel is the full daily panel of Table 3. The replaced benchmark provides neither smile nor term structure, so the networks must learn the entire shape of the surface. BS is the constant-volatility benchmark and DVF the ad hoc Black–Scholes model with the nonlinear deterministic volatility function of Eq. (4), both estimated on the preceding trading day’s cross-section and reported here as reference specifications. GPF^{IVS} and GPF^{OPS} differ only in the space in which the training loss is evaluated, volatility residuals for GPF^{IVS} as per Eq. (15) and strike-normalized prices for GPF^{OPS} as per Eq. (11). RMSE is the pricing-error root mean squared error and $\mathcal{V}\text{RMSE}$ the normalized vega-weighted pricing RMSE of Eq. (20), both in index points with the \$100 contract multiplier not applied; IVRMSE is the implied-volatility RMSE in percentage points. Vega is evaluated at the observed implied volatility. Obs. is the number of contracts on which all four models produce a price forecast. Bold entries identify the lowest error within each option-type block. Superscripts on the GPF^{OPS} row denote Diebold–Mariano significance against the GPF^{IVS} benchmark, with standard errors clustered by trading date: $*p < 0.10$, $**p < 0.05$, $***p < 0.01$. No tests against BS or DVF are reported.

	Model	RMSE (pts)	IVRMSE (pp)	$\mathcal{V}\text{RMSE}$ (pts)
Calls (Obs. = 3,676,030)	BS	23.35	5.59	34.89
	DVF	17.46	3.57	27.05
	GPF^{IVS}	15.03	3.20	22.92
	GPF^{OPS}	11.21***	2.93***	15.21***
Puts (Obs. = 4,991,872)	BS	20.96	6.13	28.83
	DVF	13.42	2.79	20.24
	GPF^{IVS}	11.92	2.73	17.67
	GPF^{OPS}	9.75***	2.64***	13.33***
Pooled (Obs. = 8,667,902)	BS	22.01	5.91	31.63
	DVF	15.27	3.15	23.48
	GPF^{IVS}	13.33	2.94	20.14
	GPF^{OPS}	10.39***	2.77***	14.19***