Erasmus University Rotterdam

Erasmus School of Economics

Master Thesis Econometrics–Quantitative Finance

# Correcting Static Arbitrage in Machine Learning-Derived Implied Volatility Surfaces

Jakub Láža (680687)

**Abstract**

This thesis performs arbitrage correction of implied volatility surfaces of weekly options predicted by the model-guided Neural Network approach. Fitting Neural Networks on pricing errors of parametric models generate implied volatility surfaces with large arbitrage opportunities. The arbitrage violations are especially large for Calendar spread. In order to eliminate the arbitrage, we use Linear Programming and incorporation of arbitrage conditions into the Neural Network loss function. While the Linear Programming approach is more efficient in reducing arbitrage, the incorporation of arbitrage conditions into the Neural Network loss function reduces out-of-sample prediction errors, on average, by a further 8% for 1-day ahead prediction and by up to 16% for 1-month ahead predictions, making it superior to the model-guided Neural Network in terms of predictions accuracy based on Diebold-Mariano test, while still reducing arbitrage violations in IVS compared to the original model by 95%. Overall, we demonstrate that removing arbitrage in Machine Learning-derived IVS not only aligns models with the no-arbitrage principle but also enhances prediction accuracy.

| | |
|---|---|
| Supervisor: | dr. Maria Grith |
| Second assessor: | prof. dr. Chen Zhou |
| Date final version: | 14th August 2024 |

The content of this thesis is the sole responsibility of the author and does not reflect the view of the supervisor, second assessor, Erasmus School of Economics or Erasmus University.

# Acknowledgments

I would like to express my gratitude to my supervisor, dr. Maria Grith, for her guidance, insightful suggestions, and her dedicated time during our consultations. Her expertise and support helped to shape this work in its final form. Additionally, I extend my thanks to my family for their unconditional support and encouragement throughout my studies. Their constant belief in me has been greatly encouraging. Lastly, a great deal of thanks goes to my friends who contributed to making my academic endeavor an enjoyable experience and enriched me with new perspectives.

# Contents

# 1 Introduction

Volatility implied from option prices using the Black-Scholes formula contains information about the market's expectations for future price volatility. This information is crucial for gauging the forward-looking market sentiment, risk perception, and are used for option pricing. Precise prediction of implied volatility can thus help by making better-informed decisions about asset allocation and hedging strategies; traders and market makers apply implied volatility in their trading strategies. The implied volatility surface shows how volatility varies with respect to moneyness and time-to-maturity. Proper modeling and forecasting of the implied volatility surface (IVS) is of interest to both academics and investors. The parametric option pricing models are derived such that arbitrage is excluded (Harrison & Kreps, 1979). No-arbitrage is an important property as it ensures that riskless profit can not be made. Issuers of options could incur significant losses if they would price options based on models that admit arbitrage. While nearly all parametric pricing models generate arbitrage-free IVS, Machine Learning models that have gained popularity in recent years do not ensure to model IVS without arbitrage opportunities. The surge in the use of Machine Learning models for IVS modeling is mainly driven by their ability to handle complex non-linear relationships and large-dimensional data. These benefits result in increased prediction power compared to traditional models (Almeida *et al.*, 2023; Chen *et al.*, 2023). However, little attention is given to the ability of these models to generate arbitrage-free IVS, while the potential for arbitrage in Machine Learning-generated IVS can undermine the reliability of these models for practical applications. We adapt two arbitrage removal approaches in the literature to the Cont & Vuletić (2023) arbitrage penalties and more crucially to the Neural Network correction of parametric models as described in Almeida *et al.* (2023). Note that, we adjust the definition of the Butterfly spread of Cont & Vuletić (2023) by a more precise approximation of derivative on a discrete grid, making the Butterfly penalty more accurate. First, ex-post method focuses on removing arbitrage on the fitted implied volatility surfaces. An optimization problem is constructed such that one looks for the closest surface to the Machine Learning generated while considering arbitrage constraints (Cohen *et al.*, 2020). Secondly, the arbitrage can be tackled by penalizing the arbitrage constraints directly during the training of the machine learning model (Zhang *et al.*, 2023). This is done by including the static arbitrage constraints in the NN loss function as a regularization term. Since the Almeida *et al.* (2023) parametric correction is fitted on the pricing errors rather than directly, we propose a method based on Zhang *et al.* (2023) approach to remove arbitrage in the parametric corrected IVS. Note that the parametric correction of the Black-Scholes model is equivalent to fitting the IVS directly, hence it can serve as a baseline model for our analysis.

This thesis thus focuses on removing arbitrage based on the two approaches adjusted to the arbitrage penalties as defined in Cont & Vuletić (2023). Specifically, the arbitrage conditions are incorporated into ex-post static arbitrage removal that is solved using linear programming in the spirit of Cohen *et al.* (2020). We thus propose to use Linear Programming considering the constraints as defined in Cont & Vuletić (2023). Further, we extend the Zhang *et al.* (2023) method of including arbitrage conditions to a NN loss function to also work on the parametric corrected IVS. The implied volatility is modeled directly using Neural Networks as a function of moneyness and time-to-maturity and by fitting the Neural Network on the pricing errors made by parametric errors as was proposed by Almeida *et al.* (2023). Also, we use Gatheral & Jacquier (2014) approach of SSVI parametrization as a baseline model that fits arbitrage-free surfaces by penalizing for crossing of the slices of total implied variances lines. We discover that the magnitude of arbitrage violations do not substantially differ whether the surface was fit directly or by the model-guided approach of Almeida *et al.* (2023). Further, we reveal that modeling the IVS of weekly options using Neural Networks results in more arbitrage opportunities than when fitting monthly options. This is likely to be caused by the much shorter time-to-expiry of the weekly options. The rapid time decay, captured by the Greek Theta, can then lead to pricing inefficiencies that lead to arbitrage opportunities. More crucially, we show that correcting for the arbitrage leads to a reduction of the prediction error. The reduction in errors happens especially for out-of-money options, where also most arbitrage violations occur. We thus showcase that not only the removal of arbitrage in Machine Learning derived IVS can make the models in line with the no-arbitrage principle but also decrease the prediction errors.

The thesis is structured as follows: Section 2 introduces the relevant literature, Section 3 gives an overview of the data used and its transformation before the analysis, Section 4 is devoted to the description and implementation of the methods used, Section 5 then presents the most relevant results, and Section 6 concludes the thesis.

# 2 Literature Review

## 2.1 Arbitrage

Arbitrage opportunities offer profit without bearing any risks. It can be divided into static and dynamic arbitrage. Static arbitrage refers to exploiting price discrepancies between assets or derivatives at a specific point in time. While dynamic arbitrage involves taking advantage of mispricings over time, usually through strategies that adapt to changing market conditions. In this thesis, the no-arbitrage condition is defined by the absence of arbitrage opportunities in Calendar, Call, and Butterfly spreads, following Carr & Madan (2005) assertion that these criteria are sufficient to ensure no-arbitrage in option prices. The Calendar spread states that option prices are non-decreasing in time-to-maturity; to ensure no violations of the Call spread, the call/put option prices have to be increasing/decreasing in moneyness and the absence of Butterfly spread translates to option prices being convex in moneyness, for more details see Cousot (2007). The way how Calendar, Call and Butterfly spread penalties for option prices are computed is described in Cont & Vuletić (2023). Similarly, Zhang *et al.* (2023) use Calendar and Butterfly penalties in their adjusted loss arbitrage removal method, however the penalties are directly computed for the IVS rather than for the option prices.

No arbitrage properties are essential for option pricing models. The First Fundamental Theorem of Asset Pricing is built around no-arbitrage opportunities; crucially, it states the definition, which states when no-arbitrage is equivalent to the existence of equivalent martingale measure (Harrison & Kreps, 1979). Nearly all option pricing models follow no-arbitrage conditions, and thus its IVS are arbitrage-free. However, this is not the case for IVS derived using Machine Learning techniques. Therefore, there is a need to remove the arbitrage present in Machine Learning-generated IVS. The literature on removing arbitrage primarily focuses on smoothing and filtering the data (Cohen *et al.*, 2020). The smoothing approach employs non-parametric methods to remove arbitrage, e.g., Aıt-Sahalia & Duarte (2003), employing the polynomial kernel smoothing. Fengler (2009) estimates arbitrage-free IVS by incorporating the arbitrage-free constraints into the cubic spline smoothing. The cubic spline approach does not require the input data to be arbitrage-free, which poses a benefit over interpolation procedures to arbitrage removal such as Kahale (2004). Further, Orosi (2015) builds on Fengler (2009) by employing a non-parametric regression spline model to generate arbitrage-free call price surfaces. Interestingly, enforcing the arbitrage conditions can even lead to improved estimation of the pricing function (Fengler & Hin, 2015). The approach of Yatchew & Härdle (2006) use smoothed constraint estimate to enforce monotonicity and convexity of option prices. Gatheral & Jacquier (2014) define several SVI (stochastic volatility inspired) surfaces. The authors specify several arbitrage-free SVI surfaces that can be described by a closed-form solution. The smoothing approaches have in common that usually Euclidean norm ($\ell^2$ norm penalization) is applied over polynomial, spline, or kernel parameters to find the smoothed volatilities (Cohen *et al.*, 2020). As a result, the smoothing method changes most of the implied volatilities, while it is desirable to correct only the implied volatilities that violate the arbitrage conditions. In this thesis, the arbitrage correction is done by linear programming, in the spirit of Cohen *et al.* (2020), which only focuses on correcting arbitrage where it occurs, thus not changing implied volatilities that do not permit arbitrage. Filtering data is based on removing market observations based on several specifications, for example, dropping data points with low volume or based on any benchmarks, including other variables describing option contracts. The way filtering is done usually depends on empirical findings and may be subjective.

Recently, generative models such as Variational Autoencoders (VAN) and Generative adversarial networks (GAN) are being used to model implied volatility surface, given the generative nature of the models, sampling can be used to obtain IVS without arbitrage. In the volGAN model of Vuletić & Cont (2023), the authors use the Cont & Vuletić (2023) general re-weighting algorithm that assigns weight to the generated paths based on the violations of arbitrage correction and using these weights then samples the scenarios. Ning *et al.* (2023) then train VAE on SDE model parameters and then sample from the posterior distribution, which is then decoded to obtain SDE model parameters that define arbitrage-free IV surface.

Further, with the rise of machine learning techniques in modeling volatility, different approaches to correct arbitrage have emerged. The arbitrage violations are usually treated during the model's training by incorporating the penalization for arbitrage constraints into the loss function. Zhang *et al.* (2023) and Ackerer *et al.* (2020) both employ similar arbitrage conditions into the loss function by considering calendar spread, Durleman's condition, and large moneyness behavior. Note that the smoothing is usually applied to the price surface that is then translated to implied volatilities, while in the adjustment of the loss function, the arbitrage conditions are defined on the implied volatility surface; detailed discussion on the conditions for IVS to be arbitrage-free is in Roper (2010). In this thesis, we extend the Zhang *et al.* (2023) approach to Almeida *et al.* (2023) non-parametric correction making it in line with the asset pricing principle. Further, we apply Linear Programming using Cont & Vuletić (2023) arbitrage constraints, making the two arbitrage method comparable. Further, we adjust the computation of Cont & Vuletić (2023) Butterfly spread by using central difference as opposed to forward/backward difference when approximating derivative on a discrete grid. Our adjustment of the Butterfly spread should thus be more precise and robust to extreme values. Lastly, the Linear Programming approach allows us to focus on correcting the arbitrage violations only where they occur, while the smoothing approach tend to change the whole IVS even where the IVS is not violating arbitrage conditions.

## 2.2 Implied Volatility

The literature on modeling and predicting implied volatility primarily consists of parametric models based on theory. These models provide a good theoretical foundation for understanding implied volatility, however their assumptions tend to be too restrictive to correctly model the dynamics of IV. The Black & Scholes (1973) seminal paper introduced an option pricing model that can be used to solve for implied volatility given the observed market data. The Black-Scholes world, however, assumes a constant implied volatility. That is, if markets would price options based on the Black-Scholes formula, implied volatility would be the same for all options on the same underlying asset. However, empirically, it is observed that implied volatility varies across moneyness and time-to-maturity, which presents the so-called "volatility smile" pattern. The Heston model approaches this issue by relaxing the strong parametric assumption of constant volatility and treats it as a stochastic process (Heston, 1993). Dumas *et al.* (1998), on the other hand, develop a straightforward model to capture the non-constant volatility by regressing polynomial features of the second order of time-to-maturity and moneyness on implied volatility. Carr & Wu (2005) then approach the problem of IVS modeling by directly modeling the implied volatility dynamics while considering no-arbitrage conditions.

Nonetheless, recently, with the availability of extensive datasets and increased computing power, data-driven Machine Learning (ML) methods have gained popularity. This results in Machine Learning methods beating traditional econometrics methods in predictive tasks (Vrontos *et al.*, 2021; Christensen *et al.*, 2023; Zhang *et al.*, 2024). Christensen *et al.* (2023) extensive study compares a broad range of ML methods in forecasting realized variance and shows that Neural Networks provide the best predictions. The flexibility and good generalization of Neural Networks are also utilized to fit the IVS. For example, Almeida *et al.* (2023) approach employs a feed-forward neural network to correct for the pricing errors made by the parametric models. The goal is to utilize the power of Machine Learning to model complex non-linear relationships that can improve the predictive power of parametric models. Similarly, Zhang *et al.* (2023) employ long-short term memory recurrent neural network to improve the predictions of IVS; however, compared to Almeida *et al.* (2023), Zhang *et al.* (2023) ensures that the predicted implied volatility surface is arbitrage-free.

While these studies focus on long-term implied volatility, little is known about the dynamics of short-term options. Andersen *et al.* (2017) showed that short-maturity options provide information on jump risks. Since the tenor is short, it is less prone to the expected volatility but rather describes the risk-neutral jump process. In contrast, the variability of future volatility can not be ignored for the long-term options. The consequent attractiveness of short-dated options is mainly in the form of allowing to hedge or bet on market movements over a short period of time. As a result, these options are used to hedge against heightened volatility around the earnings announcement days (EAD) (Alexiou *et al.*, 2021) as well as around Federal Open Market Committee (FOMC) announcements (Carr & Wu, 2005; Wright, 2020). The delta-neutral straddles, which can be used to lay off the risk of highly anticipated stock movements in any direction, are purchased by investors with substantial premiums. For example, Johannes *et al.* (2023) extracts the FOMC event risk from short-term options. The results show that the FOMC event risk is a good predictor of realized event risk and carries a large and significant event risk premia.

Given the different nature of short-term options compared to longer-dated options, new information can be obtained from short-term options. For instance, Todorov (2019) constructs spot volatility measures based on short-term options offering a precise volatility proxy. Further, Todorov (2022) leverages the information on jump risks and creates a non-parametric jump variation measure recovered from short-dated options data.

## 3 Data

The S&P 500 short-term options data are obtained from the OptionMetrics Ivy DB dataset available at the Wharton Research Data Services (WRDS). The time period spans from 1st January 2018 to 28th February 2023. For each day, information on options strikes, best bid/ask, implied volatility, time-to-maturity, and volume are gathered. The prices of the underlying S&P 500 index are obtained from Bloomberg terminal. The S&P 500 options are selected due to their higher liquidity compared to equity options and the fact they capture the US economy as a whole. Before proceeding with the analysis, the following filters are applied to the raw dataset of options obtained from OptionMetrics. Firstly, time-to-maturity is determined as the current date of an observation minus the expiration date of an option (taking into account only trading days). The moneyness is computed as $m_{i,t} = \frac{S_t}{K_{i,t}}$, while the market price of an option is determined as the midpoint of the bid-ask spread. To account for dividends being paid out, put-call parity is applied as is a common practice in option pricing literature; we follow the approach of Wallmeier (2024). Specifically, by considering put and call options that are closest to at-the-money (usually the most liquid options) and have the same time-to-maturity and moneyness, the put-call parity is employed to obtain dividend-adjusted price of the S&P 500 Index for each day in the following way:

$$S_t^*(K^*, T) = C_t(K^*, T) - P_t(K^*, T) + K^* e^{-r(T-t)} \tag{1}$$

where $K^*$ denotes the strike attributed to the call and put that are closest to the current price and $S_t^*(K^*, T)$ denotes the dividend-adjusted price. Hence, the $S_t^*$ is used instead of $S_t$, allowing us to set dividend yield to 0.

Using the adjusted data, we invert the Black Scholes formula to obtain implied volatility for each option contract; this is done by Brent's method using `SciPy` python package. Following the common practice in literature, only out-of-money (OOM) options are considered due to their liquidity (Almeida *et al.*, 2023). Therefore, for moneyness $\geq 1.03$, we use out-of-money Put options to obtain the ITM call price using Put-Call parity. Specifically, the options are grouped as in Almeida *et al.* (2023) to deep OTM call (DOTMC) if $m_{i,t} \in [0.80, 0.90)$, OTMcall (OTMC) if $m_{i,t} \in [0.90, 0.97)$, ATM if $m_{i,t} \in [0.97, 1.03)$, OTM put (OTMP) if $m_{i,t} \in [1.03, 1.10)$ and deep OTM put (DOTMP) if $m_{i,t} \in [1.10, 1.60]$. Lastly, options that were not traded, i.e., their volume is 0, are discarded, as well as options with a market price less than 0.125.

In total, we have a dataset with 729,980 observations. The descriptive statistics are presented in Table 1. It can be seen that most observations are for the at-the-money options, signaling that weekly options are mainly traded with a strike price around the current stock price, which is consistent with the UST option's limited time-to-maturity. Further, it can be noticed that the options with a time to maturity of less than 3 have higher mean volatility across all moneyness brackets and also a more intensive smile pattern of the implied volatility compared to options with a time to maturity between 4 to 7 days. In Figure 1 we plot the implied volatility surfaces of market data on two different days. The results for monthly options were obtained for the same period, and the same filters were applied.

| | Number | | Mean IV | | Std. dev. IV | |
|---|---|---|---|---|---|---|
| Time to maturity | 1-3 | 4-7 | 1-3 | 4-7 | 1-3 | 4-7 |
| Moneyness | | | | | | |
| DOTMC | 2,407 | 4,659 | 0.91 | 0.56 | 0.28 | 0.21 |
| OTMC | 24,611 | 59,299 | 0.37 | 0.27 | 0.21 | 0.13 |
| ATM | 144,779 | 202,732 | 0.23 | 0.22 | 0.12 | 0.10 |
| OTMP | 76,578 | 123,205 | 0.39 | 0.31 | 0.15 | 0.11 |
| DOTMP | 21,930 | 69,780 | 0.70 | 0.55 | 0.25 | 0.19 |
| Total | 270,305 | 459,675 | 0.34 | 0.30 | 0.21 | 0.17 |

Table 1: Descriptive Statistic. This table depicts total number of observations, mean implied volatility and its standard violations in buckets of moneyness over y-axis and time-to-maturity (days) over x-axis.

In addition to the options data, this thesis makes use of following variables: VIX index, that measures the market expectation of volatility over the next 30 years, is downloaded from the Bloomberg terminal; Spot Volatility Index (SpotVol) developed by Andersen *et al.* (2017), which provides an unbiased estimator of S&P 500 spot volatility with a minimal influence of price jumps, is downloaded from CBOE website[1]. The Jump risk measure constructed based on Bollerslev *et al.* (2015) describes the expected volatility caused by a sizeable negative jump in S&P Index price over a short period. Similarly to the Spot Volatility, it can be obtained from the CBOE website[2]. Note that both the Spot Volatility Index and Left Tail Volatility index are constructed from a portfolio of mainly out-of-money short-expiry S&P 500 options, and as To-

[1] Spot Volatility: `https://www.cboe.com/us/indices/dashboard/spotvol/`

[2] Left Tail Volatility: `https://www.cboe.com/us/indices/dashboard/LTV/`

dorov (2022) argues large portion of the IV of UST options could be explained by these indices.

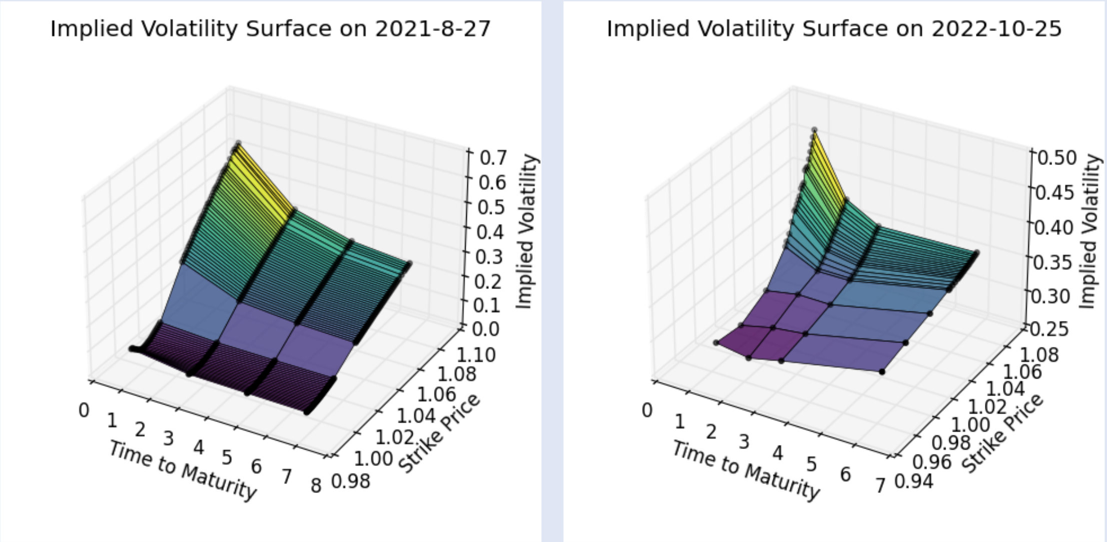


Figure 1: Snapshot of Implied Volatility Surfaces. This figure shows IVS of market data on 27th August, 2021 (left) and 25th October, 2022 (right).

# 4 Methodology

## 4.1 Parametric models

### 4.1.1 Black & Scholes

Black & Scholes (1973) seminal paper introduced a model for determining the price of European options. This model sets a landmark in derivatives pricing by setting the theoretical foundations for deriving a fair price for a European-style option. The general idea of option pricing is that the price of an option equals its expected discounted payoff; hence, the price at any time $t$ equals

$$C_t = e^{-r(T-t)}\mathbb{E}_{\mathbb{Q}}\left[\max(S_T - K, 0)\right] \tag{2}$$

where $C_t$ denotes a put call price, $T$ is the time of option expiration, $K$ is the strike price and $\mathbb{Q}$ is the risk-neutral measure. Assuming the price process $S_t$ follows a geometric Brownian motion, the Black-Scholes closed-form solution for a fair price of a European call option is:

$$\begin{aligned} C_t &= S_t\Phi(d_1) - e^{-r(T-t)}K\Phi(d_2), \\ d_1 &= \frac{\log\left(\frac{S_t}{K}\right) + (r + \frac{\sigma^2}{2})(T-t)}{\sigma\sqrt{T-t}}, \\ d_2 &= d_1 - \sigma\sqrt{T-t}. \end{aligned} \tag{3}$$

Having defined the analytical solution of the Black-Scholes option pricing model, it can be noticed that the price depends on 4 parameters: strike price $(K)$, risk-free rate $r$, time to expiry, defined as $T - t$, and lastly, the volatility $\sigma$. Most importantly, the volatility parameter is constant. Therefore, the IVS surface based on Black-Scholes is flat. To obtain the implied volatility given the observed market price, the following equation needs to be solved

$$\sigma_t = C_{BS}^{-1}(C_t, S_t, K, r, T) \tag{4}$$

where $C_t$ is the observed market price of a call option. Since the equation has no analytical solution, numerical methods such as Brents's or Newton-Raphson have to be used to obtain the volatility that gives the observed market price when plugged into the Black-Scholes formula. Brent's solver, available in Python package `SciPy`, was used due to its better performance than the Newton-Raphson method on this thesis options data. Fitting the BS model on a group of options for the same day can be done by simple average of the options implied volatilities: $\hat{\sigma}_{BS_t} = \bar{\sigma}(m_{i,t}, \tau_{i,t})$, where $\sigma(m_{i,t}, \tau_{i,t})$ is the market observed data and the average for day $t$ is taken over the option contracts $i = 1, \ldots, N$.

The BS volatility parameter does not change with respect to the time-to-expiry or strike price. Such property is shown to be limiting as the empirical observations show that volatility varies across different time-to-expiry and strike prices. Therefore, the assumption of constant volatility is too restrictive and is likely to result in an incorrect valuation of options.

### 4.1.2 Adhoc Black & Scholes

Building on the limitations of the Black & Scholes model that assumes constant implied volatility, the AdHoc Black & Scholes model estimates the IV as a function of moneyness and time-to-maturity. Dumas *et al.* (1998) develop this approach to mimic the way option pricing is done in practice: first, the implied volatility is modeled, and then the IV is translated into option prices. For the cross-section of options, the model for the day $t$ is written as:

$$\sigma_{i,t} = a_{0,t} + a_{1,t}m_{i,t} + a_{2,t}m_{i,t}^2 + a_{3,t}\tau_{i,t} + a_{4,t}\tau_{i,t}^2 + a_{5,t}m_{i,t}\tau_{i,t} + \epsilon_{i,t}, \quad i = 1, \ldots, N \tag{5}$$

where $\sigma_{i,t}$ is the observed market implied volatility, $m_{i,t}$ and $\tau_{i,t}$ denote the moneyness and time-to-maturity. As can be seen, the model is a simple linear regression with moneyness, time-to-maturity, and their squared and interaction terms. This model is able to account for the smile and term structure of implied volatility while being simple and easy to estimate by ordinary least squares. Note, however, that the simplified regression approach does not ensure that the fitted IVS using the estimated parameters $\hat{\boldsymbol{\alpha}}$ will be arbitrage-free.

### 4.1.3 Carr and Wu model

The Carr & Wu (2016) model is designed to tackle the option pricing problem as is done by practitioners; the focus is on describing the dynamics of the implied volatility rather than the process of an underlying asset. Crucially, the near-term dynamics of the implied volatility surface are modeled while incorporating no-arbitrage conditions that are directly applied to the IVS. Contrary to traditional option pricing frameworks that impose no-arbitrage restrictions on the option prices. By following the assumptions of Carr and Wu model, the model considers the dynamics of the price process $S_t$

$$\frac{dS_t}{S_t} = \sqrt{v_t}dW_t, \tag{6}$$

where $v_t$ is the instantaneous variance of $log(S_t)$ and $W_t$ denotes Brownian motion. While the instantaneous variance $v_t$ has no specification, Carr & Wu (2016) directly model the dynamics of IVS by process, which, under risk-neutrality, has the following form

$$d\sigma_t(K,T) = \mu_t dt + \omega_t dZ_t, \tag{7}$$

$\mu_t$ being the drift, $\omega_t$ denotes the volatility of volatility, $K$ and $T$ are strike and time-to-maturity, respectively, lastly, $Z_t$ is Brownian motion. Writing the IVS process dynamics proportional to the IVS at time $t$ gives

$$\frac{d\sigma_t(K,T)}{\sigma_t(K,T)} = e^{-\eta_t(T-t)}(m_t dt + \omega_t dZ_t), \tag{8}$$

here, the new term $m_t$ is the average drift of IVS, $w_t$ vol of vol, $\eta_t$ introduces the empirical observation that long-dated IV tend to move less. Note that the two random elements in the form of Brownian motions have correlation $\rho_t$, which follows a stochastic process. The fact that $m_t$, $w_t$ and $\eta_t$ are independent of K, $\tau$ and $\sigma_t(K, \tau)$ simplifies the estimation.

Under the Carr & Wu (2016) no-arbitrage condition, the entire IVS can be obtained by solving the following quadratic equation

$$\frac{1}{4}e^{-2\eta_t\tau}w_t^2\tau^2\sigma_t^4 + \left(1 - 2e^{-\eta_t\tau}m_t\tau - e^{-\eta_t\tau}w_t\rho_t\sqrt{v_t}\tau\right)\sigma_t^2 \tag{9}$$

$$- \left(v_t + 2e^{-\eta_t\tau}w_t\rho_t\sqrt{v_t}k + e^{-2\eta_t\tau}w_t^2k^2\right) = 0 \tag{10}$$

with $k = log(K/S_t)$ and $\tau = T - t$. As Almeida *et al.* (2023) point out, the solution to the quadratic equation depends on the values of the stochastic process only at time $t$. Therefore, since the solution for $\sigma^2(k, \tau)$ does not depend on any dynamics of these processes when calibrating the model for a specific day $t$, the stochastic process values $(v_t, m_t, w_t, \eta_t, \rho_t)$ can be considered as parameters. Resulting in a parametric space $\mathbf{\Theta}_t = (v_t, m_t, w_t, \eta_t, \rho_t)$ over which one minimizes the following objective function to obtain the best fit for the $\sigma_t^2(\mathbf{\Theta}_t, k, \tau)$

$$\hat{\theta}_t = \arg\min_{\theta_t} \sum_{i=1}^{n} \Big[\frac{1}{4}e^{-2\eta_t\tau_{i,t}}w_t^2\tau_{i,t}^2\sigma_{i,t}^4 + \left(1 - 2e^{-\eta_t\tau_{i,t}}m_t\tau_{i,t} - e^{-\eta_t\tau_{i,t}}w_t\rho_t\sqrt{v_t}\tau_{i,t}\right)\sigma_{i,t}^2 \tag{11}$$

$$- \left(v_t + 2e^{-\eta_t\tau_{i,t}}w_t\rho_t\sqrt{v_t}k_{i,t} + e^{-2\eta_t\tau_{i,t}}w_t^2k_{i,t}^2\right)\Big]^2 \tag{12}$$

where $\sigma_{i,t}$ is the observed implied volatility for option with $k_{i,t}$ and $\tau_{i,t}$. In order to implement the minimization problem, we employ the `SciPy` package in Python – the differential evolution solver is used due to its robust convergence. After obtaining the parameters, we obtain $\sigma^2(\Theta_t, k, \tau)$ by solving the quadratic equation Equation 10 and subsequently take a square root $\sqrt{\sigma^2(\Theta_t, k, \tau)}$ to obtain the IV as per Carr and Wu model.

#### 4.1.4 SVI parametrizaiton

The stochastic volatility inspired parametrization, described by Gatheral 2004, aims to model the volatility smile. The implied volatility is modeled as a function of log-moneyness, hence a smile for a given maturity is modeled. This model also captures practitioners' approach to implied volatility modeling as the origins of the models were developed at Merrill Lynch. The appealing properties of this method are two-fold: firstly, the implied variance $\sigma(k,t)$ is linear in the log-strike $k$ as $k \to \infty$, given a fixed expiry $t$. Secondly, the model can be fit such that the calendar spread arbitrage is not violated. The raw SVI parametrization is defined as

$$w(k;\chi_R) = a + b\left\{\rho(k-m) + \sqrt{(k-m)^2 + \sigma^2}\right\}, \tag{13}$$

where $\chi_R = \{a, b, \rho, m, \sigma\}$; $a \in \mathbb{R}$ influences the general level of variance across all strike prices, shifting the volatility smile vertically, $b \geq$controls the steepness of the slopes for both the put and call wings of the volatility smile, thus tightening the smile, $|\rho| < 1$ adjusts the asymmetry of the smile. An increase in $\rho$ enhances the slope of the right wing, increasing volatility for higher strikes, and decreases the slope of the left wing, reducing volatility for lower strikes, resulting in a counter-clockwise rotation of the smile. The parameter $m \in \mathbb{R}$ acts as the modal or peak point of the smile and shifts the entire structure left or right along the strike price axis, aligning the peak with various market expectations or asset values. Lastly, $\sigma > 0$ determines the curvature at the money, around the point $m$.

Since in the raw formulation, it is nearly impossible to pose restrictions on the parameters such that the generated smiles are arbitrage-free, we work with the natural formulations (Gatheral & Jacquier, 2014):

$$w(k;\chi_N) = \Delta + \frac{\omega}{2}\left\{1 + \zeta\rho(k-\mu) + \sqrt{\left(\zeta(k-\mu)+\rho\right)^2 + (1-\rho^2)}\right\} \tag{14}$$

where the parameters, $\chi_N = \{\Delta, \mu, \rho, \omega, \zeta\}$, live in the following domains $\omega \geq 0, \Delta \in \mathbb{R}, \mu \in \mathbb{R}, |\rho| < 1$, and $\zeta > 0$. To see how the parameters $\chi_N$ relate to the parameters of the raw formulation, refer to Gatheral & Jacquier (2014).

Now, we write the Surface SVI (SSVI), introduced by Gatheral & Jacquier (2014), that can be adjusted such that it produces arbitrage-free surfaces. The SSVI is a special case of the natural formulation in Equation 14 where $\chi_N = \{0, 0, \rho, \theta_t, \varphi(\theta_t)\}$:

$$w(k;\theta_t) = \frac{\theta_t}{2}\left\{1 + \rho\varphi(\theta_t)k + \sqrt{\left(\varphi(\theta_t)k+\rho\right)^2 + (1-\rho^2)}\right\} \tag{15}$$

where $\theta_t = \sigma^2(0,t)t$ is ATM implied total variance and $\varphi$ is a smooth function. We apply the power-law parametrization for the $\varphi$ function, thus $\varphi(\theta) = \eta / \left(\theta^\gamma (1+\theta)^{(1+\gamma)}\right)$. Using the power-law parametrization, the static arbitrage is guaranteed if $\eta\,(1+|\rho|) \leq 2$.

Since the values of the parameters in the above-mentioned specification can be unstable and unintuitive, SVI Jump-Wings (SVI-JW) parametrization is used. The SVI-JW works with implied variance instead of total implied variance, making the parameter more intuitive for traders. Lastly, the SVI-JW parameters depend on time to expiration $\tau$. To obtain the SVI-JW, only the following transformation of the already obtained parameters from SSVI parametrization needs to be done:

$$\begin{aligned} v_t &= \theta_t/\tau & \psi_t &= \frac{1}{2} \\ p_t &= \frac{1}{2}\varphi(\theta_t)(1-\rho) & c_t &= \frac{1}{2}\varphi(\theta_t)(1+\rho) \\ &\tilde{v}_t = \frac{\theta_t}{\tau}(1-\rho^2) \end{aligned} \tag{16}$$

The SVI-JW parameters can also be derived from the raw parametrization; see Gatheral & Jacquier (2014).

Now, the calibration of the SSVI model that generates arbitrage-free surfaces is done in three steps. Firstly, we estimate the parameters for the formulation in Equation 17, so we obtain the initial estimates $\hat{v}_t, \hat{\psi}_t, \hat{p}_t, \hat{c}_t$ and $\hat{\tilde{v}}_t$. Each of these parameter groups define a single slice of the volatility surface. Secondly, we eliminate butterfly arbitrage by fixing the $v_t, \psi_t$, and $p_t$ and compute the other two parameters in the following way

$$c'_t = \hat{p}_t + 2\hat{\psi}_t \tag{17}$$

$$v'_t = \hat{v}_t \frac{4\hat{p}_t c'_t}{(\hat{p}_t + c'_t)^2} \tag{18}$$

such parameters define a butterfly arbitrage-free implied volatility smile. Now, due to the continuity of parameters as described in Gatheral & Jacquier (2014) we can look for $(c^*_t, v^*_t)$ that are as close as possible to the original ones by using the sum of squared option price differences as objective function and bounds for $c_t$ defined as $c_t \in (\min(\hat{c}_t, c'_t), \max(\hat{c}_t, c'_t))$ and $v_t \in (\min(\hat{v}_t, v'_t), \max(\hat{v}_t, v'_t))$. Using a sequential least squares solver, we discover the parameters free of butterfly arbitrage and as close as possible to the initial smile. Thirdly, we remove the calandar spread. Using the estimated parameters up to this point as initial guesses, we find parameters that minimize the sum of squared distances of the fitted IVs and the observed market IVs subject to large penalties for violation of calendar arbitrage. This violation happens if the total variance lines cross, i.e., when looking at two SVI slices that are both described by different parameters $\chi_1$ and $\chi_2$. To quantify the crossings, the following points $k_i$, where $i = 1, \ldots, n$ with $n \leq 4$, are determined based on where the slices cross. Next, $\tilde{k}_i$ points are computed as

$$\begin{aligned}
\tilde{k}_1 &= k_1 - 1, \\
\tilde{k}_i &= \frac{1}{2}(k_{i-1} + k_i), \quad \text{if } 2 \leq i \leq n, \\
\tilde{k}_{n+1} &= k_n + 1.
\end{aligned}$$

Now, the amounts describing how much the slices cross are given as

$$x_i = \max\Big(0, w(\tilde{k}_i; \chi_1) - w(\tilde{k}_i; \chi_2)\Big) \tag{19}$$

Hence, the crossedness $x_i$ is used to impose a penalty on the crossing of two slices, which is equivalent to a violation of calendar spread.

## 4.2 Non-Parametric Correction

Following Almeida *et al.* (2023) non-parametric correction approach of parametric models, we first estimate the IVS of the following parametric models:

- Black-Scholes (equivalent to directly fitting IVS)
- Ad Hock Black Scholes
- Carr-Wu model
- SSVI

The pricing errors of the parametric models, $\hat{\epsilon}_h(m_{i,t}, \tau_{i,t})$, are defined as the difference between the fitted values, $\hat{\sigma}_h(m_{i,t}, \tau_{i,t})$, and the observed values $\sigma_h(m_{i,t}, \tau_{i,t})$, where $h$ denotes a parametric model. The goal of the non-parametric correction is then to minimize the following function

$$\frac{1}{n}\sum_{i=1}^{n}\left[\hat{\epsilon}_h(m_{i,t}, \tau_{i,t}) - f(m_{i,t}, \tau_{i,t})\right]^2 . \tag{20}$$

This optimization problem is approached by a feedforward neural network that estimates the pricing error function $\hat{f}(m, \tau)$. The corrected volatility surface is then defined as $\hat{\sigma}_h(m, \tau) + \hat{f}(m, \tau)$. One important note of Almeida *et al.* (2023) is that the non-parametric correction to Black Scholes errors is equivalent to fitting the implied volatility surface directly – stemming from the fact that the IV errors in Black Scholes ($\hat{\epsilon}_{BS}(m_{i,t}, \tau_{i,t}) = \sigma_{BS}(m_{i,t}, \tau_{i,t}) - \hat{\sigma}_{BS}(m_t, \tau_t)$) are only shifted by a constant BS implied volatility ($\hat{\sigma}_{BS}(m_t, \tau_t)$), which does not influence the optimization problem. Hence, essentially, we are fitting directly the market observed data ($\sigma_{BS}(m_{i,t}, \tau_{i,t})$). Therefore, the results presented for Black Scholes model can be thought of as a case when the IVS is fitted directly.

### 4.2.1 Neural Networks

The approximation of the pricing error function is done via Neural Network, which based on the Universal Approximation Theorem should be able to approximate any well-measurable function. That is, theoretically, NNs should be able to learn the mapping from inputs to implied volatility. In the feed-forward NN architecture, the data moves in one direction, starting in the input layer (input node for each feature) and passing through $n$ hidden layers until an output layer. The nodes of input, hidden, and output layers are connected, and each connection has its specific weight; the value of a node is then computed as the sum of the previous nodes multiplied by the corresponding weights plus bias, and finally, the activation function is used to obtain the node value. This can be written as:

$$\boldsymbol{h}_i = z\left(\boldsymbol{W}_{i-1}\boldsymbol{h}_{i-1} + \boldsymbol{b}_{i-1}\right), \quad \text{for} \quad i = 1, \ldots, H, \quad \boldsymbol{h}_0 = \boldsymbol{x} \tag{21}$$

$$f(\boldsymbol{x}_{i,t}) = \boldsymbol{W}_H \boldsymbol{h}_H + \boldsymbol{b}_H \tag{22}$$

where $\boldsymbol{x}_{i,t} = (m_{i,t}, \tau_{i,t})'$ is an input variable, $H$ is the total number of hidden layers, matrix $\boldsymbol{W}_i$ denotes given weights, $\boldsymbol{b}_i$ are the biases (constants), $z(\cdot)$ is the activation function at the hidden layer. Lastly, $h_i$ reflects the hidden layer value, and $f(\boldsymbol{x_{i,t}})$ is the learnt pricing function. The activation function is a crucial part of the NN as it introduces the non-linear nature of the model. ReLU$(x)$, defined as $\max(0, x)$, is a popular activation function; also, sigmoid and hyperbolic tangent are widely used. Note that no activation function is used at the output given the regression task at hand.

Since the main focus of this paper is the arbitrage correction, we employ the architectures as in Almeida *et al.* (2023). While we experimented with different number of hidden layers, batch-sizes and number of epochs. The final Neural Network architecture, for which results are presented, has 3 hidden layers consisting of 32, 16 and 8 neurons (same as the best performing NN in Almeida *et al.* (2023)). The decrease of number of neurons in each hidden layers is motivated by the proven rule of geometric pyramid described in Masters (1993). Further, as is common practice, activation function at the hidden layers is ReLU. While we experimented with Elu activation function based on the Horvath *et al.* (2021) that used Elu to train Neutral Networks to calibrate parametric models. For this problem it did not yield better results than ReLU. The Neural Network thus has the following architecture:

| Layer | Type | Number of Neurons | Activation Function |
|---|---|---|---|
| 1 | Input Layer | n | - |
| 2 | Hidden Layer 1 | 32 | ReLU |
| 3 | Hidden Layer 2 | 16 | ReLU |
| 4 | Hidden Layer 3 | 8 | ReLU |
| 5 | Output Layer | 1 | Linear |

Table 2: Neural Network Architecture. This table presents the general architecture used for non-parametric correction of parametric models. $n$ is the number of features.

For the training of the NN, the number of minibatches and epochs needs to be also determined. In each epoch, a minibatch, a subset of the dataset, is used to look for the minimimu of a loss function. The more epochs, the longer the algorithm looks for the optimal minimum. The number of minibatches influences the speed as well as overfitting of the network. If a large batchsize is selected than a large proportion of the training data are used for the computation of gradient leading to a higher chance of overfitting, on the other hand, too small gradient might lead to too much noise in the gradient steps. In the general case we set batch size to 64 and number of epochs to 100. However, when running the NN with the customized loss function, the batch size in increased to 128, such that we have sufficient amount of data to construct relevant IVS in the loss function for which we compute the arbitrage penalties. Further, we use early stopping that stops the training process once the model's loss has not improved in last 15 epochs. The best weights are then retrieved as the final weights. Lastly, learning rate, the length of a step in each iteration, is set to a cosine decay. That is at the beginning of the learning process, the algorithm makes large steps, but as the training progresses the steps are shortening such that the algorithm is able to find and stay in a optimal minimal of the loss function. Before training all input data are standardized.

## 4.3 Static Arbitrage Correction

Since non-parametric error correction does not ensure that the IVS will be arbitrage-free, we employ static-arbitrage correction. First, let us define the conditions that must be satisfied such that the implied volatility surface is arbitrage-free. Following Davis & Hobson (2007), the call/put option prices should follow the following conditions: increasing in time-to-maturity and increasing/decreasing and convex in moneyness. These conditions can be written as 3 separate arbitrage penalties[3] (Cont & Vuletić, 2023):

$$p_1(\sigma(m,\tau)) = \sum_{i=1}^{N_m}\sum_{j=1}^{N_\tau}\left(\tau_j \frac{c(m_i,\tau_j) - c(m_i,\tau_{j+1})}{\tau_{j+1}-\tau_j}\right)^+, \tag{23}$$

$$p_2(\sigma(m,\tau)) = \begin{cases} \sum_{i=1}^{N_m}\sum_{j=1}^{N_\tau}\left(\frac{c(m_{i+1},\tau_j)-c(m_i,\tau_j)}{m_i-m_{i+1}}\right)^+ & \text{if } \ \mathrm{c}(\mathrm{m}_{i+1},\tau_j) \text{ and } c(m_i,\tau_j) \text{ Calls} \\ \sum_{i=1}^{N_m}\sum_{j=1}^{N_\tau}\left(\frac{c(m_i,\tau_j)-c(m_i+1,\tau_j)}{m_i-m_{i+1}}\right)^+ & \text{if } \ \mathrm{c}(\mathrm{m}_{i+1},\tau_j) \text{ and } c(m_i,\tau_j) \text{ Puts.} \end{cases} \tag{24}$$

$$p_3(\sigma(m,\tau)) = \sum_{i=1}^{N_m}\sum_{j=1}^{N_\tau}\left(\frac{c(m_i,\tau_j)-c(m_{i-1},\tau_j)}{m_i-m_{i-1}} - \frac{c(m_{i+1},\tau_j)-c(m_i,\tau_j)}{m_{i+1}-m_i}\right)^+. \tag{25}$$

where $p_1$ describes the violation of calendar spread, $p_2$ represents the call constraint and lastly $p_3$ measures the butterfly spread. Since, contrary to Cont & Vuletić (2023), we also consider put options, the definition for $p_2$ is split between call options and put options. Further, these arbitrage conditions apply to the prices of the call/put options, more specifically to relative call/put price defined as $c(m_i,\tau_j) = \frac{1}{S}C_{BS}(S,K_i,\tau_j,\sigma)$ to make the arbitrage conditions comparable across different option prices. Since the market prices are interpolated as the midpoint price of the market quotes provided by OptionMetric, we might not necessarily observe the true price. This can lead to the introduction of arbitrage even in the market data, especially the Butterfly spread computed as backward minus forward differentiation, which can be very sensitive to such slight discrepancies. For an example of market price with respect to moneyness see Figure 8, while in general the shape is convex, when zooming on individual observation, small deviations can be detected. In order to limit the small discrepancies mostly caused by the interpolation of market quotes, we propose an adjusted Butterfly measure $p_3$ that uses central differences to approximate the derivation at each point. The central differences are a superior

[3] Note, that our condition differs in the denominator of the call constraint $p_2$ due to the fact that we define moneyness as price/strike, while Cont & Vuletić (2023) use strike/price definition. The other constraints remain unchanged under the opposed definition.

approach to numerically approximate derivatives compared to forward and backward differences as it considers both the point before and the point after the point we approximate the derivative (Turner, 1994). The adjusted Butterfly spread measure for market data is computed as:

$$p_3^*(\sigma(m,\tau)) = \sum_{i=1}^{N_m}\sum_{j=1}^{N_\tau}\left(\frac{c(m_i,\tau_j)-c(m_{i-2},\tau_j)}{m_i-m_{i-2}} - \frac{c(m_{i+1},\tau_j)-c(m_{i-1},\tau_j)}{m_{i+1}-m_{i-1}}\right)^+ . \tag{26}$$

the first term approximate the derivative at point $m_{i-1}$ while the second term at point $m_i$. Hence, the penalty still checks for the convexity of prices with respect to moneyness. However, we utilize a more precise approximation of the derivatives, such as our measure is more robust to extreme values. When computing the arbitrage penalties, first, Black Scholes with corresponding implied volatility need to be inverted to obtain option prices. The implied volatility surface is said to be static-arbitrage-free if

$$\Phi(\sigma(m,\tau)) = p_1(\sigma(m,\tau)) + p_2(\sigma(m,\tau)) + p_3((m,\tau)) = 0. \tag{27}$$

Since it is not guaranteed that the implied volatility surface will have the same shape each day, to make the penalties comparable across days, we divide the sum of all penalties by the number of observations in a given implied volatility surface. In this way, we obtain the arbitrage violation measure per observation, and thus, larger implied volatility surfaces can be compared with smaller ones.

In order to correct the arbitrage violations as defined by the above-mentioned measures, we include them in a linear programming problem and apply the customized loss function as defined by Zhang *et al.* (2023).

### 4.3.1 Linear Programming

Cohen *et al.* (2020) use linear programming to obtain an arbitrage-free surface. Given that this optimization problem is run on the price surface, firstly, the implied volatilities are transformed to prices, then the arbitrage-free prices are estimated, and finally, we invert BS to transform the prices back into implied volatilities. Having $\hat{\sigma}(m,\tau)$, generated by our Neural Network, that violates arbitrage-free conditions, the goal is to find the closest $\sigma(m,\tau)$ that is arbitrage-free. We adjust the optimization problem to the arbitrage penalties of Cont & Vuletić (2023). The problem is thus written as:

$$\min_{\sigma(m_i,\tau_j),1\leq i\leq N_m,1\leq j\leq N_\tau} \sum_{i=1}^{N_m}\sum_{j=1}^{N_\tau} |\hat{\sigma}(m_i,\tau_j) - \sigma(m_i,\tau_j)| \tag{28}$$

subject to the following constraints,

$$\begin{aligned}
&\frac{\sigma(m_i,\tau_j)-\sigma(m_i,\tau_{j+1})}{\tau_{j+1}-\tau_j} \leq 0, && \forall 1\leq i < N_m, 1\leq j\leq N_\tau,\\
&\frac{\sigma(m_{i+1},\tau_j)-\sigma(m_i,\tau_j)}{m_i-m_{i+1}} \leq 0, \text{ if } c(m_{i+1},\tau_j) \text{ and } c(m_i,\tau_j) \text{ Calls}, && \forall 1\leq i < N_m, 1\leq j\leq N_\tau\\
&\frac{\sigma(m_i,\tau_j)-\sigma(m_{i+1},\tau_j)}{m_i-m_{i+1}} \leq 0, \text{ if } c(m_{i+1},\tau_j) \text{ and } c(m_i,\tau_j) \text{ Puts}, && \forall 1\leq i < N_m, 1\leq j\leq N_\tau\\
&\frac{\sigma(m_i,\tau_j)-\sigma(m_{i-1},\tau_j)}{m_i-m_{i-1}} \leq \frac{\sigma(m_{i+1},\tau_j)-\sigma(m_i,\tau_j)}{m_{i+1}-m_i}, && \forall 1\leq i < N_m, 1\leq j\leq N_\tau.
\end{aligned}$$

Since this is an ex-post method, it can be applied to all the methods that generate the corrected parametric IVS. Also, this approach is expected to yield minimal changes to the root-mean squared error, since we look for the closest price surface that does not violate arbitrage to the generated one.

**Implementation:** Since the second constraint changes depending on whether the option is a call or put – call prices should be increasing in moneyness, while put should be decreasing. Hence, we apply the correct constraint to each option type. In a case when $\sigma(m_i,\tau_j)$ is a call and $\sigma(m_i,\tau_{j+1})$ is a put, or vice versa, we do not compute the constraint. Hence, in theory, we run the optimization two times – one with call price surface and the other time with put price surface with only difference in the second constraint. To translate the minimization into linear programming, firstly, the objective function in Equation 28 needs to be linearized due to the non-linear absolute value; this is done by introducing auxiliary variable $z_{i,j}$, which has following constraints

$$z_{i,j} \geq \hat{\sigma}(m_i,\tau_j) - \sigma(m_i,\tau_j), \tag{29}$$

$$z_{i,j} \geq \sigma(m_i,\tau_j) - \hat{\sigma}(m_i,\tau_j), \tag{30}$$

given that $z_{i,j}$ is a non-negative variable, the constraints ensure that $z_{i,j}$ is equal to the absolute value of the difference in Equation 28. Now, the objective function becomes

$$\min_{z_{i,j},1\leq i\leq N_m,1\leq j\leq N_\tau} \sum_{i=1}^{N_m}\sum_{j=1}^{N_\tau} z_{i,j}. \tag{31}$$

The constraints are then expressed as $\boldsymbol{Ax} \leq \boldsymbol{b}$, where $\boldsymbol{x}$ is a vector of $\sigma$ and $z$. The vector $\boldsymbol{x}$ denoting the variables to minimize, matrix $\boldsymbol{A}$ and vector $\boldsymbol{b}$ describing the constraints are imputed into the *lingprog* function of `SciPy` Python package, together with bounds for $\boldsymbol{x}$, the lower bound is not 0, but 0.125 since the data for options are filtered to contain only options with a price higher than 0, thus $\boldsymbol{x} \geq 0.125$. After we obtain the corrected price surface, the prices are converted back to implied volatilities that should not be arbitrage-free.

### 4.3.2 Customized Loss Function

This approach implements the arbitrage-free conditions directly into the loss function of the NN that is being used to fit the implied volatility surface (Zhang *et al.*, 2023). The goal is to learn the Neural Network to generate arbitrage-free surfaces by penalizing for the arbitrage violations during the model's training. For the Neural Network to learn properly, the loss function needs to be differentiable, as the updates of the weights are based on the gradient of the loss function. Therefore, we can not directly implement the above-mentioned calendar, call, and butterfly penalty measures that are based on the prices of the options since the training is done on volatilities: the issue is that the penalties need to invert Black Scholes to turn the volatilities into prices. However, this operation is not differentiable hence it can not be part of the loss function. To overcome this issue, we use two conditions defined in Zhang *et al.* (2023) that are closely related to the $p_1, p_2$ and $p_3$ penalties, however they are directly applied to the IVS surface. Firstly, $\ell_{cal}$ addresses the monotonicity of option prices with respect to time-to-maturity and can be related to the calendar constraint $p_1$. The IVS calendar constraint $\ell_{cal}$ can be written as

$$\ell_{cal}(m,\tau) = \sigma(m,\tau) + 2\tau\partial_\tau\sigma(m,\tau) \geq 0 \tag{32}$$

Secondly, the $p_2$ and $p_3$ constraints can be captured by Durleman's Condition, which states that for every $(m,\tau)$, we have

$$\begin{aligned}\ell_{\mathrm{dur}}(m,\tau) = \left(1 - \frac{m\partial_m\sigma(m,\tau)}{\sigma(m,\tau)}\right)^2 - \frac{\left(\sigma(m,\tau)\tau\partial_m\sigma(m,\tau)\right)^2}{4} \\ + \tau\sigma(m,\tau)\partial_{mm}\sigma(m,\tau) \geq 0.\end{aligned} \tag{33}$$

We notice that both penalties $\ell_{cal}$ and $\ell_{dur}$ are non-linear and contain a derivative of the implied volatility surface. Since we observe a discrete grid of $(m,\tau)$ and thus the IVS is not continuous, the derivatives $\partial_\tau, \partial_m, \partial_{mm}$ need to be approximated. Following Turner (1994), for the first derivative and second derivative, respectively, we employ centered numerical differentiation for $t \in \{1, \dots, n-1\}$ we have

$$\frac{\partial \sigma(m, \tau_t)}{\partial \tau_t} = \frac{\sigma(m, \tau_{t+1}) - \sigma(m, \tau_{t-1})}{\tau_{t+1} - \tau_{t-1}}, \quad \frac{\partial^2 \sigma(m_0, \tau)}{\partial m_t^2} = \frac{\sigma(m_n, \tau) - 2\sigma(m_{n-11}, \tau) + \sigma(m_{n-2}, \tau)}{(m_n - m_{n-1})^2}.$$

Edges of the IVS surface, i.e., when $t \in \{0, n\}$, need special treatment as the edge implied volatility does not have a neighbor value to the left or the right. Hence, we use the forward difference ($t = 0$),

$$\frac{\partial \sigma(m, \tau_0)}{\partial \tau_t} = \frac{\sigma(m, \tau_1) - \sigma(m, \tau_0)}{\tau_1 - \tau_0}, \quad \frac{\partial^2 \sigma(m_0, \tau)}{\partial m_t^2} = \frac{\sigma(m_2, \tau) - 2\sigma(m_1, \tau) + \sigma(m_0, \tau)}{(m_1 - m_0)^2},$$

and backward difference when $t = n$,

$$\frac{\partial \sigma(m, \tau_n)}{\partial \tau_t} = \frac{\sigma(m, \tau_n) - \sigma(m, \tau_{n-1})}{\tau_n - \tau_{n-1}}, \quad \frac{\partial^2 \sigma(m_0, \tau)}{\partial m_t^2} = \frac{\sigma(m_n, \tau) - 2\sigma(m_{n-11}, \tau) + \sigma(m_{n-2}, \tau)}{(m_n - m_{n-1})^2}.$$

Similar computations also apply to the first derivative of IVS with respect to moneyness. Now, it is possible to cast the calendar and Durleman's condition losses of IVS using `Tensorflow` Python package operations, which ensures that the loss is differentiable and, thus, gradient for the updating step can be computed. By adding these penalization terms to the loss function defined in Equation 27, we obtain the following customized loss function

$$\mathcal{L} = \mathcal{L}_1 + \lambda \left(\ell_{cal} + \ell_{dur}\right) \tag{34}$$

where $\mathcal{L}$ is the total loss function, $\mathcal{L}_1$ is the loss function as defined in Equation 20. Lastly, to ensure smooth convergence, the $\ell_{cal}$ and $\ell_{dur}$ penalties are scaled by $\lambda$ such that they do not dominate the $\mathcal{L}_1$ or the other way are not so small that the penalties are negligible in the training process. In other words, $\lambda$ represents the strength of the penalization of the violations of the static arbitrage. For simplicity, $\lambda$ is the same for both penalties, however, it is possible to tune different values for each penalty.

Applying this method to the IVS error-correction of Almeida *et al.* (2023) needs extra attention, as the error-correction done does not fit the implied volatility surface directly but estimates the pricing errors that are then added to the parametric IVS. Although a parametric model's IVS is arbitrage-free, the arbitrage conditions of IVS do not apply to the pricing errors. Therefore, when the estimated pricing errors are passed into the loss function, firstly the parametric error-corrected IVS is constructed inside the loss $\left(\hat{\sigma}_h(m,\tau)+\hat{f}(m,\tau)\right)$ and after that the values for $\ell_{cal}$ and $\ell_{dur}$ can be computed.

To evaluate the performance of the arbitrage corrections, the arbitrage penalty, as defined in Equation 27, will be used. In this way, it will be possible to distinguish among the static arbitrage correction methods performances (where possible) and plot the arbitrage violations on the $(m,\tau)$ grid.

### 4.4 Empirical Settings

**Cross-Section:** The predictions are made in the option cross-section. Each day, a parametric model is calibrated, and its pricing errors are computed. Consequently, to perform the parametric error correction, for each day $t$ in our dataset, the neural network takes as input and outputs the following variables:

**INPUTS**

- moneyness $(m)$
- time-to-maturity $(\tau)$

**OUTPUT**

- IVS error-correction $(\hat{f}(m_i,\tau_i))$

Hence, we estimate $t$ NNs for each day to obtain the corrected implied volatility surface. This setting does not take into consideration any dynamics but rather focuses on the interpolation performance. Further, the setting can be extended to $h$-ahead predictions by training the models on all data belonging to day $t$ and predicting on the $t+h$ data. Thus, it is possible to assess how well the non-parametric correction at time $t$ generalizes to future periods.

**Implementation:** Firstly, the data for each day are divided into train/test sets. For the interpolation results on the same day, the test set consists of every fifth observation on a given day, which results in approximately 80/20% train/test split. Given that the dataset is sorted, it ensures that we train the neural network on options that span most of the moneyness and time-to-maturity grid. Further motivation for this split ratio is to have enough data points to calibrate the parametric models consistently. Especially since the Carr and Wu model is not converging well for the weekly options and thus needs a sufficient number of observations. For the $h$-ahead predictions, the train set consists of options belonging to a whole day $t$, and the test set is the $t + h$-th day. The parametric models are calibrated on the train set, next we proceed with training the neural network on the pricing errors using the loss in Equation 20 with $n = 2$ using $\boldsymbol{x_{i,t}} = (m_{i,t}, \tau_{i,t})'$ as an input vector to learn the pricing error function $f(m_{i,t}, \tau_{i,t})$. The evaluation is done on the test set by firstly fitting a parametric model to the test data using the parameters obtained from calibration on the train test and then applying the estimated pricing error function $\hat{f}(m_{i,t}, \tau_{i,t})$ to correct for the pricing errors made by a parametric model. Having estimated the Machine learning-derived IVS, we apply the 3 arbitrage measures to discover the arbitrage opportunities in the error-corrected models. In order to remove the arbitrage present in the IVS, the Linear Programming approach is employed. While in the case of the Customized Loss Function method, the Neural Network for the non-parametric correction is trained using the loss with regularization of arbitrage penalties (Equation 34). After the arbitrage correction, we again calculate the arbitrage measures to investigate the effectiveness of the arbitrage correction methods.

**Option Panel**: The options Panel also considers time-varying variables, and instead of training the neural network each day, it is trained once on a panel of option data. The inputs and output of the NN are:

**INPUTS**

- moneyness ($m$)
- time-to-maturity ($\tau$)
- state variables: VIX, VIX9, Spot Volatility, Jump Risk Measure ($\boldsymbol{y_t}$)

**OUTPUT**

- IVS error-correction ($\hat{f}(\boldsymbol{y_t}, m_i, \tau_i)$)

Thus, this setting allows us to explain the mispricing using the time-varying state of economy variables; i.e., it is possible to learn the dynamics of the parametric models' mispricing conditional on several variables.

**Implementation**: The dataset is split into a train set that consists of options ranging from the beginning of our dataset to 18th October 2021; the test set then spans from the end of the train test up until 28th February of 2023. Since we are estimating the IVS over several dates, we need to slightly adjust the loss in Equation 20 by also summing over the dates. Now, we fit the

IVS directly using only moneyness and time-to-maturity features (denoted as Black Scholes as correcting for Black Scholes pricing errors is equivalent to fitting the surface directly (Almeida *et al.*, 2023)). Further, time-varying state variables VIX, VIX9, SPOTVOL, and Jump risk measures are added to investigate how information describing the current state of the economy can help with fitting the IVS. The feature importance is employed to investigate the most relevant feature for IVS modeling. When applying the arbitrage correction in the panel setting, the distinction to the cross-section setting is that the test set includes options at various dates. Hence, when using Linear Programming as a correction of arbitrage, firstly, options are grouped by the same date, and Linear Programming is applied to each day. In the case of the Customized Loss Function, the approach is similar, i.e., in the loss, the arbitrage penalization is computed for each date and then summed up.

## 4.5 Evaluation

In order to compare performances among our models, we employ the root mean squared error (RMSE) as the evaluation metric. This is a common practice in a regression task such as the one at hand. To ensure the comparability to Almeida *et al.* (2023), we apply the RMSE to the % of the implied volatility, making it equivalent to the implied volatility RMSE (IVRMSE). The evaluation metrics, for $n$ observations, is thus defined:

$$\text{(IV)RMSE} = \sqrt{\frac{1}{n}\sum_{i=0}^{n}(100 \cdot \sigma_i - 100 \cdot \hat{\sigma}_i)^2} \tag{35}$$

To compare the forecasting accuracy of two models we incorporate the Diebold and Mariano test (Diebold & Mariano, 2002). The idea is to define a loss differential between prediction errors of two models

$$\ell = e_{1t}^2 - e_{2t}^2 \tag{36}$$

where $e_{it}^2$ is the prediction error, defined as the difference between the actual value and the prediction, for model $i$ and time $t$. The hypothesis test is designed in a following way:

$$H_0 : \mathbb{E}[\ell_t] = 0 \quad \text{(The two models have equal predictive accuracy)} \tag{37}$$

$$H_1 : \mathbb{E}[\ell_t] \neq 0 \quad \text{(The two models do not have equal predictive accuracy)} \tag{38}$$

To evaluate the hypothesis, the Diebold-Mariano statistic, for $h \geq 1$, is defined as

$$DM = \frac{\bar{d}}{\sqrt{\left[\gamma_0 + 2\sum_{k=1}^{h-1}\gamma_k\right]/T}} \tag{39}$$

where

$$\bar{d} = \frac{1}{T}\sum_{i=1}^{T} d_i \qquad \mu = \mathbb{E}[d_i]$$

$$\gamma_k = \frac{1}{T}\sum_{i=k+1}^{T}\left(d_i - \bar{d}\right)\left(d_{i-k} - \bar{d}\right) \qquad \text{for } n > k \geq 1.$$

Under the null hypothesis the DM statistics is then asymptotically standard normally distributed. We thus reject the null if $|DM| > z_{\alpha/2}$, with $\alpha$ being the significance level and $z$ is the corresponding values of the standard normal distribution.

Lastly, in the options panel, we want to determine the most important features of modeling the IVS. This is done using Shapley Values, which are based on the Game Theory game proposed by Shapley (1953). The idea is to assess the contribution of each individual agent (feature) to the final payoff (final prediction). Let $N$ be a set of agents (features) and $v$ function mapping subset of agents to $\mathbb{R}$ such that $v(\emptyset) = 0$. For a feature $i \in N$ the Shapley value is defined as

$$S_i(N, v) = \sum_{A \subseteq N\setminus\{i\}} \frac{(n-1-|A|)!|A|!}{n!}[v(A \cup \{i\}) - v(A)], \tag{40}$$

$A$ denotes the subset of features not including feature $i$, hence we measure the contribution of feature $i$ to group $A$ by $v(A \cup \{i\}) - v(A)$. The preceding term serves as a weight to each permutation of the possible subset groups A.

# 5 Results

## 5.1 Cross-Section

The Table 3 presents the out-of-sample RMSE results for the same-day and 1-day ahead predictions. The results are displayed for the parametric models, the non-parametric correction, and subsequently for 2 static arbitrage correction methods – Linear Programming (LP) and Custom-

| | Parametric | Error correction | Arbitrage Correction (LP) | Arbitrage Correction (LF) |
|---|---|---|---|---|
| | | Panel A: Same-day | | |
| BS | 10.34 | 0.68 | 0.60 | **0.59** |
| AHBS | 2.62 | 0.61 | 0.61 | 0.60 |
| CW | 3.39 | 0.63 | 0.61 | 0.63 |
| SSVI | 3.42 | 0.64 | 0.60 | 0.61 |
| | | Panel B: 1-day ahead | | |
| BS | 10.72 | 3.37 | 3.36 | 2.80 |
| AHBS | 4.54 | 2.91 | 2.92 | **2.64** |
| CW | 5.13 | 3.02 | 3.01 | 2.91 |
| SSVI | 5.26 | 2.97 | 2.91 | 2.84 |
| | | Panel C: 5-days ahead | | |
| BS | 13.93 | 6.34 | 6.28 | **5.85** |
| AHBS | 7.88 | 6.88 | 6.55 | 6.37 |
| CW | 8.42 | 6.18 | 6.23 | 6.05 |
| SSVI | 8.33 | 6.26 | 6.37 | 6.11 |
| | | Panel D: 21-days ahead | | |
| BS | 14.56 | 7.91 | 7.44 | **6.81** |
| AHBS | 11.26 | 8.95 | 7.61 | 7.13 |
| CW | 11.89 | 8.18 | 7.67 | 7.09 |
| SSVI | 11.56 | 8.05 | 7.71 | 7.24 |

Table 3: RMSE (%) OF PREDICTION IN THE OPTION CROSS-SECTION. This table depicts the RMSE of prediction in the cross section for Black-Scholes, Adhoc BS, and Carr and Wu model. LP denotes the Linear Programming method and LF the customized loss approach to removing arbitrage. The numbers in **bold** stress the best-performing models.

ized Loss Function (LF). In contrast, the values of arbitrage penalties for the error correction and violations after the arbitrage correction are in Table 5. Overall, we can notice, that the Linear Programming approach is more efficient in removing arbitrage than the Customized Loss Function method, however the Customized Loss method yileds the lowest RMSE. This trade-off can be seen in Figure 2. The LF approach always has lower RMSE, but also higher magnitude of arbitrage. Further, naturally, with the increase in the prediction period the RMSE increases, while the arbitrage after the correction stays at a similar level. Suggesting the magnitude of arbitrage after the correction is rather invariant to the number of prediction steps ahead. The values of the arbitrage violations of the baseline parametric models can be seen at Table 11. The Carr-Wu and SSVI which should generate arbitrage-free surfaces have only minimal violations of the three arbitrage spreads, while the Adhoc BS model, that does not guarantee arbitrage-free IVS, violates all three spreads; however at lower magnitude than the error correction.

As Table 4 presents the decrease in RMSE between parametric correction (NN) and the Linear Programming method of removal arbitrage (NNLP) is statistically insignificant for any predic-

| Model Pair | h = 1 | h = 5 | h = 21 |
|---|---|---|---|
| NN/NNLP | -1.23 (0.22) | -0.91 (0.36) | -0.55 (0.57) |
| NN/NNLF | -2.25**(0.02) | — | — |
| NNLP/NNLF | -1.75*(0.08) | — | — |
| Note: *p<0.1; **p<0.05; ***p<0.01 | | | |

Table 4: Diebold-Mariano (DM) Test Results. This table presents the DM test results comparing the forecasting accuracy of the models NN, NNPL, and NNLF across different forecast horizons (h = 1, 5, 21) with Black Scholes being the parametric model. The DM statistic values corresponding *p*-values in bracket indicate the relative predictive performance of the models.

tion horizont based on the Diebold-Mariano test. However, the adjustment of the loss function approach to arbitrage removal shows a significantly lower RMSE at a 5% significance level compared to the parametric correction. The difference between Linear Programming and Loss Function has *p*-value of 0.08, suggesting that the statistical significance at $\alpha = 10$. We thus show that the Loss Function arbitrage removal yields a significant increase in predictive performance compared to the parametric correction.

| | Error Correction | | | Arbitrage Correction (LP) | | | Arbitrage Correction (LF) | | |
|---|---|---|---|---|---|---|---|---|---|
| | Calendar | Call | Butterfly | Calendar | Call | Butterfly | Calendar | Call | Butterfly |
| Panel A: Same Day | | | | | | | | | |
| BS | 0.02 | $8.0 \times 10^{-8}$ | 0.05 | $6.4 \times 10^{-4}$ | 0.0 | $2.2 \times 10^{-15}$ | 0.01 | 0.00 | 0.03 |
| AHBS | 0.06 | $1.0 \times 10^{-3}$ | 0.01 | $1.1 \times 10^{-4}$ | 0.0 | $1.0 \times 10^{-16}$ | $1.7 \times 10^{-3}$ | 0.00 | 0.03 |
| CW | 0.05 | 0.09 | 0.08 | $7.5 \times 10^{-3}$ | 0.0 | $3.4 \times 10^{-17}$ | 0.02 | 0.00 | 0.01 |
| SSVI | 0.05 | $5.1 \times 10^{-9}$ | 0.03 | $3.2 \times 10^{-4}$ | 0.0 | $8.7 \times 10^{-17}$ | 0.01 | 0.00 | 0.02 |
| Panel B: 1 day ahead | | | | | | | | | |
| BS | 10.9 | 0.91 | 15.4 | 0.19 | $1.3 \times 10^{-9}$ | $2.5 \times 10^{-9}$ | 0.47 | 0.00 | **2.01** |
| AHBS | **39.3** | 0.22 | 25.1 | **0.26** | $8.8 \times 10^{-18}$ | $8.7 \times 10^{-8}$ | 0.43 | 0.00 | **2.44** |
| CW | 9.23 | 3.42 | 14.2 | 0.12 | $4.4 \times 10^{-17}$ | $8.8 \times 10^{-8}$ | **0.93** | 0.00 | **1.53** |
| SSVI | 14.1 | 0.29 | 13.4 | 1.22 | 0.0 | $7.23 \times 10^{-8}$ | 1.89 | 0.00 | 0.79 |
| Panel C: 5 day ahead | | | | | | | | | |
| BS | 89.2 | 0.00 | 29.3 | 0.15 | 0.00 | $5.62 \times 10^{-9}$ | 0.88 | 0.00 | **2.01** |
| AHBS | **118** | 0.65 | 87.7 | 0.49 | 0.00 | $2.35 \times 10^{-8}$ | 0.43 | 0.00 | **2.44** |
| CW | **103** | 22.7 | 54.4 | **0.78** | 0.00 | $4.23 \times 10^{-8}$ | **2.74** | 0.00 | **1.53** |
| SSVI | 77.5 | 0.83 | 65.7 | 0.54 | 0.0 | $4.6 \times 10^{-8}$ | 1.53 | 0.00 | **2.84** |
| Panel D: 21 day ahead | | | | | | | | | |
| BS | 15.4 | 0.75 | 15.8 | 0.27 | 0.00 | $1.63 \times 10^{-8}$ | 0.88 | 0.00 | **2.01** |
| AHBS | 68.8 | 0.98 | 34.4 | 0.23 | 0.00 | $8.63 \times 10^{-8}$ | 0.43 | 0.00 | **2.44** |
| CW | 56.3 | 9.00 | 30.9 | 0.37 | 0.00 | $1.98 \times 10^{-8}$ | **2.74** | 0.00 | **1.53** |
| SSVI | 43.8 | 1.77 | 18.4 | 0.11 | 0.0 | $3.8 \times 10^{-8}$ | 0.35 | 0.00 | 1.12 |
| Market | 0.02 | 0.16 | 14.5 | 0.02 | 0.16 | 14.5 | 0.02 | 0.16 | 14.5 |

Table 5: Mean Arbitrage Errors. This table depicts the mean arbitrage errors for the market price and for the error and arbitrage correction using linear programming. The average is taken over 1st of February 2018 to 28th of February 2023. The Arbitrage penalizations are based on the one-day-ahead test predictions. The numbers in **bold** signify large arbitrage violations for a given model and method.

It is apparent that the non-parametric correction generates IVS surfaces with many arbitrage opportunities. The most prominent violations are in the calendar spread ($p_1$); this is likely caused by the fact that the Calendar spread is based on the arbitrage violations over the time-to-maturity, which, in terms of the weekly option, has a limited number of possible values. On average, we observe 4 unique time-to-maturity values each day, while the moneyness spans a much wider grid. Hence, when correcting only using moneyness and time-to-maturity, many options have similar values for $\tau$, making it harder for the Neural Network to learn the effect of the time-to-expiration. This results in the NNs likely performing poorly over the $\tau$ variable, leading to increased arbitrage violations of the Calendar spread.

As can be seen in the Table 5, the Linear Programming (LP) approach in removing arbitrage is more efficient than the Customized Loss Function (LF). The main difference is in the ability to remove the Butterfly spread arbitrage ($p_3$). While LP removes nearly all arbitrage in $p_3$, the LF method is able to reduce the Butterfly spread only to a value of 2. In contrast, the $p_3$ values for error correction attain values as high as 87.7, in the case of the Adhoc Black Scholes, hence the LF is still able to considerably reduce the Butterfly spread arbitrage. The Calendar spread arbitrage, the most violated spread, is also the hardest to remove; however, both methods perform quite similarly, with LP slightly better. The fact that Linear Programming is superior to the LF in removing arbitrage is no surprise. The adjusted loss function approach is based on training NN to minimize the RMSE together with arbitrage penalties on the train set, which may then not translate to the test set. On the other hand, Linear Programming is employed ex-post and directly targeted to the arbitrage violations in the test sets.

Although the LF approach is not so efficient in removing arbitrage, it is able to decrease the RMSE of the error-corrected model more than the LP method. Interestingly, when looking at Table 6, we can notice that the decrease in RMSE is mainly caused by lower RMSE in the (deep) out-of-money puts. It is especially noticeable for the BS + NN + LF setting, where the lowest RMSE is for the deep out-of-money put options. For 5 & 21 days ahead decomposition of RMSE consult Table 12. Looking at the Figure 3, we can notice that most of the arbitrage violations for BS + NN happen for the deep out-of-money options Hence, this suggests that the removal of arbitrage opportunities in the DOTMP is connected with decreased RMSE. Even thought the highest magnitude of arbitrage violations is recorded for the close to expiry ATM options. These findings are observed for each of the parametric models. Hence, regularization

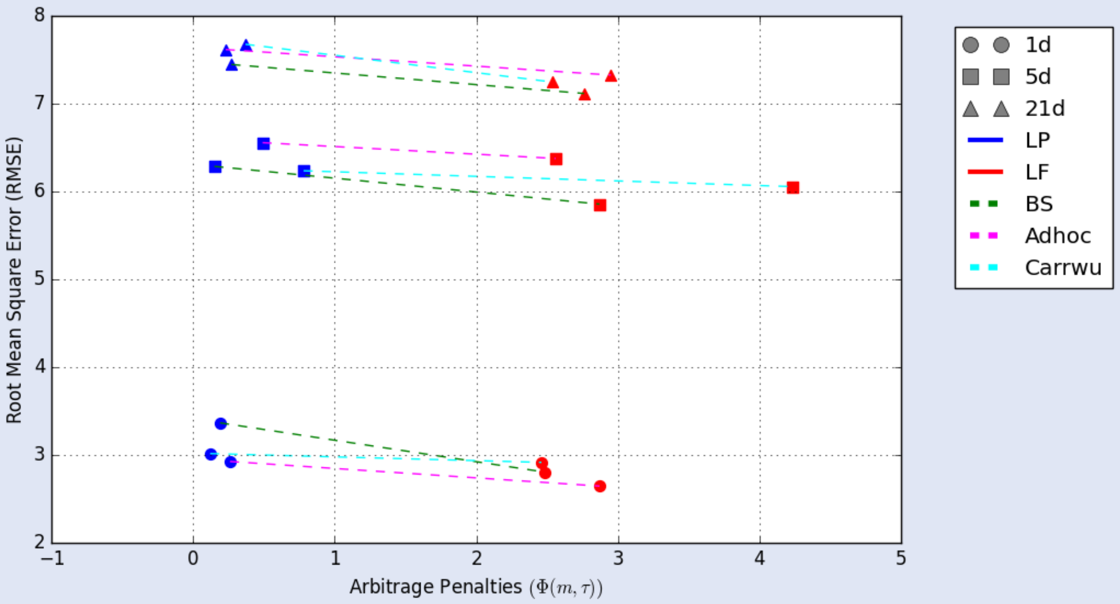
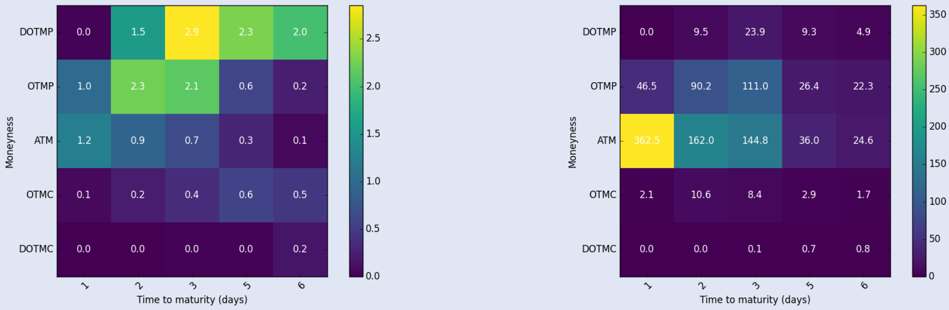


Figure 2: Trade-off of Arbitrage and RMSE of 1-day Ahead Predictions: This Figure plots the RMSE and total value of Arbitrage for the Linear Programming Correction (LP) and Customized Loss Function approach (LF).

of the arbitrage using adjusted loss can help with predictions of the IVS, especially for the out-of-money options. This is an important finding as the out-of-money parts of the IVS are the areas where the parametric models and its nonparametric correction perform the worst. The parametric models and its NN correction perform best for the at-the-money options, as is also the case for monthly options in Almeida *et al.* (2023); these options are usually the most traded, and thus, we possess the most observations for these options.

Unsurprisingly, the root mean squared error, among the parametric models, is highest for the BS model that has constant volatility. After the error correction, each model's RMSE remains at similar levels. Hinting that the NN can correct for the pricing errors at similar performance unconditional on the underlying parametric model. Looking at the 1-day ahead predictions, the errors for the error correction increase quite substantially compared to the same-day interpolation, suggesting that the short-term options error correction does not generalize well to future errors. This might be caused by the fact that the short-term options are noisier and more dependent on the specific time-varying state of the economy. This can also be supported by the higher RMSE for the parametric models for the weekly options compared to monthly options (see results for monthly options in Appendix A).



(a) % of occurrences

(b) Magnitude ($\Phi(m, \tau)$)

Figure 3: Arbitrage violations on $(m, \tau)$ grid. The left part (a) denotes the percetage number of violations of arbitrage contraints in the given grid, while the right part (b) presents the total magnitude of the violations as defined in Equation 27 over the whole period relative to the number of observations. The values are present for 1-day ahead predictions of BS + NN.

Touching on the arbitrage present in the market data, we can notice that the market observations of weekly options are not free of arbitrage based on our Calendar, Call, and Butterfly spreads arbitrage measures. For monthly options, Cont & Vuletić (2023) find that the market data also tend to not be free of arbitrage. Similarly, the most arbitrage violations for monthly options are in the Butterfly spread. Cont & Vuletić (2023) argue that it can be attributed to the noise in the data, especially the way market price is determined – an average of the lowest ask

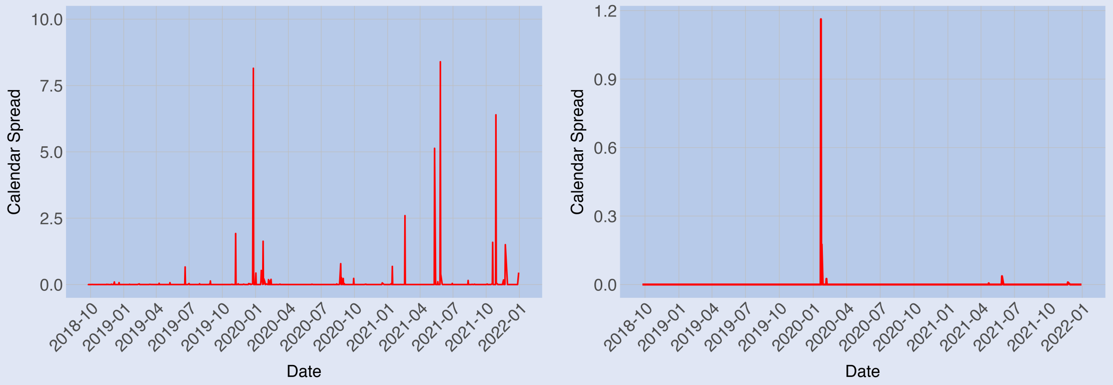
| Panel A: 1 day ahead | | | | | |
|---|---|---|---|---|---|
| | DOTMC | OTMC | ATM | OTMP | DOTMP |
| BS | 11.57 | 9.13 | 9.34 | **6.91** | 15.36 |
| BS + NN | 4.90 | 2.84 | **2.61** | 3.32 | 4.47 |
| BS + NN + LF | 4.61 | 2.88 | 2.33 | 2.20 | **1.99** |
| AdHoc BS | 8.41 | 4.20 | **2.91** | 4.48 | 6.13 |
| AdHoc BS + NN | 3.21 | 3.01 | 2.76 | **2.75** | 2.90 |
| AdHoc BS + NN + LF | 7.32 | 4.78 | **0.52** | **0.57** | **0.58** |
| Carr Wu | 8.99 | 4.76 | **2.72** | 4.07 | 6.35 |
| Carr Wu + NN | 4.52 | 2.52 | **2.33** | 2.99 | 4.01 |
| Carr Wu + NN + LF | 4.15 | 3.42 | **2.32** | **2.35** | **2.34** |
| SSVI | 8.78 | 4.21 | **2.57** | 4.24 | 6.01 |
| SSVI + NN | 4.73 | 2.32 | **2.09** | 2.56 | 4.86 |
| SSVI + NN + LF | 4.29 | 3.38 | **2.16** | **2.21** | **2.30** |
| Number of observations | 7 066 | 83 910 | 347 511 | 199 783 | 91 710 |

Table 6: Decomposition of RMSE. This table depicts decomposition of RMSE into (deep) out-of-money calls: (D)OTMC, at-the-money options: ATM and (deep) out-of-money puts: (D)OTMP. The RMSE for LP approach are omitted due to being very similar to RMSE of BS + NN. The number in model denotes the lowest RMSE for given models.

and highest bid. Therefore, we might not necessarily capture the actual market price. These minor discrepancies can be easily picked up when computing penalties. The fact that the Butterfly spread arbitrage in the market data is due to the inefficiencies of the collected data and not due to systematic arbitrage is supported by Figure 6, which shows that the $p_3$ errors are relatively evenly spread across the $(m, \tau)$ grid as opposed to error correction where the arbitrage opportunities usually arise for the out-of-money put options (Figure 3). Further, we argue that specifically since the Butterfly spread checks whether the prices are convex with respect to moneyness by looking at the differences between the forward and backward numerical differentiation in each point, it may introduce further error. Hence, we propose adjusted Butterfly spread $(p_3)$ computation by using central differences, which provide a more accurate approximation of a derivative at a particular point. Thus possibly reducing the errors introduced by deriving market price data from market quotes (the original and adjusted values for $p_3$ can be found in Table 10).

By looking at the arbitrage violation on the $(m, \tau)$ grid for each spread, i.e., the decomposition of Figure 3 at Figure 9. The Calendar spread is mainly violated for very short time-to-maturities (1-3) days of the deep out-of-money puts, given that the weekly options are heavily traded in this region due to the ability to hedge short-term events, the arbitrage correction is especially relevant for that region. However the magnitude is highest for the ATM options with 1-3 days time to maturity. One potential explanation could be that as the ATM options approach its expiration, changes in the S&P index price can have high impact on the options prices which could distort the market and create arbitrage opportunities. The Call spread $(p_2)$ is hardly ever violated, the only violations happen in the top right corner of the moneyness/time-to-maturity grid defining (deep) out-of-money puts with 5-6 days of time-to-maturity. Unlike Calendar and Butterfly spread, the amount of arbitrage is mainly concentrated also at the top-right corcer, however the magnitude is almost negligable compared to the other two spreads. Lastly, the violations of the Butterfly spread are present in multiple parts of the grid, however, as for other spreads, the most violations are present for the deep out-of-money puts and the magnitude is most apparent for the ATM options with the highest values for the close to expiry options that are popular for speculative trades, which can lead to short-term distortions.

In the Figure 4, the Calendar Spread violations of non-parametric error correction without (BS + NN) and with arbitrage correction (BS + NN + LF) are plotted over time. The arbitrage



(a) Without Arbitrage Correction (b) With Arbitrage Correction (LP)

Figure 4: Calendar Spread Violations in Time. The BS+NN model with and without Linear Programming (LP) Arbitrage Correction for 1-Day Ahead Prediction

correction is able to correct the Calendar spread violation for most of the days; the only outlier is during the beginning of the COVID-19 period. Still, the magnitude of the violation is lower after the arbitrage correction than without it. Some of the arbitrage violations that are generated by the non-parametric correction, such as the one at the onset of the COVID-19 crisis, might be due to the fact that this arbitrage was present in the market data. It is important to mention that the magnitude of the arbitrage violations for the non-parametric correction is considerably higher than for the market data and appears where market data has no arbitrage violations. Thus, training the NN on market data that would be completely arbitrage-free would still result in the non-parametric correction generating IVS with multiple arbitrage violations (see Calendar spread in Figure 7 and Figure 9).

## 5.2 Option Panel

The results for the options panel setting are presented in Table 7. Since the non-parametric correction is performing similarly across different parametric models, we only present results for Black Scholes that correspond to fitting the IVS directly. For the weekly options, we notice that fitting the options panel using only moneyness and $\tau$ as predictors does not lead to considerable improvement of the RMSE compared to simple constant Black Scholes volatility. However, by adding the time-varying variables such as VIX, VIX9, Spot Volatility, and Left Tail Risk measure the RMSE is more than halved. From the arbitrage point of view, interestingly, when adding the time-varying variables to the feature mix, the Calendar and Butterfly spread greatly increase, while the Call spread is eliminated. Nonetheless, both of the arbitrage corrections are able to remove most of the arbitrage and also reduce the root-mean-squared error.

| | BS | BS + NN3 | BS + NN3F | BS + NN3 + LP | BS + NN3 + LF | BS + NN3F + LP | BS + NN3F + LF |
|---|---|---|---|---|---|---|---|
| RMSE | 10.34 | 7.31 | 3.03 | 6.85 | 6.64 | 2.83 | 2.81 |
| Calendar | – | 2.89 | 12.85 | $4.9 \times 10^{-17}$ | 0.47 | $2.3 \times 10^{-4}$ | 0.01 |
| Call | – | 4.40 | 0.00 | $1.1 \times 10^{-15}$ | 0.18 | 0.00 | 0.00 |
| Butterfly | – | 3.97 | 16.78 | $6.8 \times 10^{-8}$ | 0.75 | $9.8 \times 10^{-8}$ | 0.11 |

Table 7: Results for Panel Data Setting. Model + NN3F has extended feature space of time-varying variables: VIX, VIX9, Spot Volatility, and Left Tail Risk measure. LP denotes the Linear Programming correction and LF the customized loss approach to removing arbitrage. Train set is 1st January 2018 to 18th October 2021; test set is ending on 28th February 2023

The RMSE is reduced by more than 5 %, when removing arbitrage. Hence, the arbitrage correction is usually done in the right direction of the observed implied volatilities. The best-performing model is the non-parametric correction with time-varying factors trained using the customized loss function. The RMSE is 2.81 while there is almost no arbitrage in the Calendar and Call spread and minimal in the Butterfly spread. Therefore, we witness similar behavior of the two arbitrage corrections as in the cross-section setting. The LP is more efficient in removing arbitrage, however, in the options panel the difference for $p_3$ is not so different, while the LF approach decreases RMSE of the non-parametric correction.

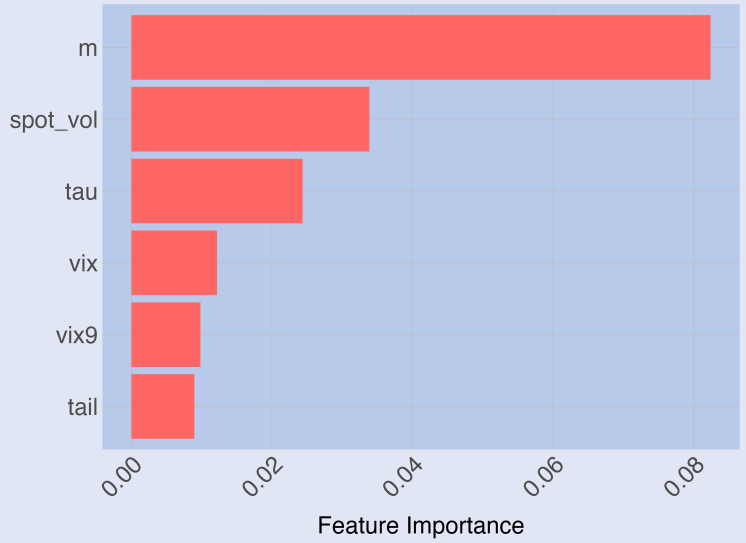


Figure 5: Feature Importance. Most important variables for the non-parametric correction of Black Scholes including time-varying index values of Spot Volatility, Jump Risk measure, VIX and VIX9 (BS + NN3F)

In Figure 5 we plot the most important variables in predicting the IVS of the weekly options based on the Shapley Values. We display it for the BS + NN3F model such that we can compare it to the results of the monthly options in Almeida *et al.* (2023). It is not surprising that the moneyness variable is the most important – considerably more than other variables. The Spot Volatility measure, which is one of the state of the economy variables, is the second most important variable, and tau, which has a limited range, is the third. By comparing the most important features of weekly options with the most important variables for options with longer maturities used in Almeida *et al.* (2023), the main difference is in the influence of time-to-maturity. For the longer options, the time to maturity plays a pivotal role, while weekly options are mostly driven by moneyness and spot volatility. Also, the VIX, which measures the expected volatility over the next 30 days, is of relatively low importance to the options with a time to maturity of less than 7 days. While the Spot Volatility Index, constructed from the out-of-money short-dated options to minimize the impact of price jumps on the estimation of S&P 500 spot volatility, plays a much more important role in modeling the IVS of weekly options.

## 6 Conclusion

Focusing on the arbitrage correction of Machine Learning predicted implied volatility surfaces, we apply the Linear Programming approach and adjustment of the Neural Network loss function to achieve arbitrage-free IVS generated by the Almeida *et al.* (2023) non-parametric correction of pricing errors of parametric models. We discover that the arbitrage violations of the ML predicted IVS are similar across correcting parametric models and fitting IVS directly, which corresponds to correcting errors of the Black-Scholes model. The arbitrage violations of the error correction of weekly options are more pronounced than for monthly options, making the weekly options the primal interest of this thesis. Most of the arbitrage opportunities arise for the Calendar spread, which can be connected to the limited range of time-to-maturity of the weekly options, making the NN perform poorly over the $\tau$ variable, which usually only takes 3 unique values in the range of 1 to 7 days. In terms of the arbitrage correction, the Linear Programming method is consistent in removing arbitrage across all three Calendar, Call, and Butterfly spreads. The adjustment of the loss function approach struggles to completely remove arbitrage in the Butterfly spread. Nonetheless, it is still able to remove around 95 % of the arbitrage generated by the non-parametric correction method. Moreover, this arbitrage method has better prediction accuracy compared to the parametric correction at 5% significance level based on the Diebold-Mariano test; the removal leads to a reduction in the RMSE of 8 % on average compared to the non-parametric correction for 1-day ahead and for 21-day ahead prediction the reduction is around 14 %. The reduction happens especially for the deep out-of-money puts, where the parametric as well as the non-parametric correction tends to perform the worst. However, this comes with the cost of fitting the out-of-money calls. We also fit the IVS on the options panel together with VIX, VIX9, Spot Volatility, and Left Tail Risk measure. After applying the arbitrage correction to the options panel with time-varying variables, it produces IVS with nearly no arbitrage and decreased RMSE. In the options panel setting, the most important variables in explaining the IVS of the weekly options are moneyness, SpotVol Index, and time-to-maturity. Comparing that to the results of monthly options in Almeida *et al.* (2023), we notice that the UTS options are not driven by the VIX index but rather by the SpotVol Index, which seems to be in line with the intuition of the options with short time-to-maturity.

# References

Ackerer, D., N. Tagasovska, & T. Vatter (2020): "Deep smoothing of the implied volatility surface." *Advances in Neural Information Processing Systems* **33**: pp. 11552–11563.

Alexiou, L., A. Goyal, A. Kostakis, & L. Rompolis (2021): "Pricing Event Risk: Evidence from Concave Implied Volatility Curves." *SSRN Electronic Journal* .

Almeida, C., J. Fan, G. Freire, & F. Tang (2023): "Can a Machine Correct Option Pricing Models?" *Journal of Business & Economic Statistics* **41(3)**: pp. 995–1009.

Andersen, T. G., N. Fusari, & V. Todorov (2017): "Short-Term Market Risks Implied by Weekly Options." *The Journal of Finance* **72(3)**: pp. 1335–1386.

Ait-Sahalia, Y. & J. Duarte (2003): "Nonparametric option pricing under shape restrictions." *Journal of Econometrics* **116(1-2)**: pp. 9–47.

Black, F. & M. Scholes (1973): "The Pricing of Options and Corporate Liabilities." *Journal of Political Economy* **81(3)**: pp. 637–654.

Bollerslev, T., V. Todorov, & L. Xu (2015): "Tail risk premia and return predictability." *Journal of Financial Economics* **118(1)**: pp. 113–134.

Carr, P. & D. B. Madan (2005): "A note on sufficient conditions for no arbitrage." *Finance Research Letters* **2(3)**: pp. 125–130.

Carr, P. & L. Wu (2005): "A tale of two indices." *SSRN Electronic Journal* .

Carr, P. & L. Wu (2016): "Analyzing volatility risk and risk premium in option contracts: A new theory." *Journal of Financial Economics* **120(1)**: pp. 1–20.

Chen, Y., M. Grith, & H. Lai (2023): "Neural Tangent Kernel in Implied Volatility Forecasting: A Nonlinear Functional Autoregression Approach." *SSRN Electronic Journal* .

Christensen, K., M. Siggaard, & B. Veliyev (2023): "A Machine Learning Approach to Volatility Forecasting." *Journal of Financial Econometrics* **21(5)**: pp. 1680–1727.

Cohen, S. N., C. Reisinger, & S. Wang (2020): "Detecting and repairing arbitrage in traded option prices." *Applied Mathematical Finance* **27(5)**: pp. 345–373. ArXiv:2008.09454.

Cont, R. & M. Vuletić (2023): "Simulation of Arbitrage-Free Implied Volatility Surfaces." *Applied Mathematical Finance* **30(2)**: pp. 94–121.

Cousot, L. (2007): "Conditions on option prices for absence of arbitrage and exact calibration." *Journal of Banking & Finance* **31(11)**: pp. 3377–3397.

Davis, M. H. A. & D. G. Hobson (2007): "THE RANGE OF TRADED OPTION PRICES." *Mathematical Finance* **17(1)**: pp. 1–14.

Diebold, F. X. & R. S. Mariano (2002): "Comparing Predictive Accuracy." *Journal of Business & Economic Statistics* **20(1)**: pp. 134–144. Publisher: Taylor & Francis.

Dumas, B., J. Fleming, & R. E. Whaley (1998): "Implied Volatility Functions: Empirical Tests." *The Journal of Finance* **53(6)**: pp. 2059–2106.

Fengler, M. R. (2009): "Arbitrage-free smoothing of the implied volatility surface." *Quantitative Finance* **9(4)**: pp. 417–428.

Fengler, M. R. & L.-Y. Hin (2015): "Semi-nonparametric estimation of the call-option price surface under strike and time-to-expiry no-arbitrage constraints." *Journal of Econometrics* **184(2)**: pp. 242–261.

Gatheral, J. & A. Jacquier (2014): "Arbitrage-free SVI volatility surfaces." *Quantitative Finance* **14(1)**: pp. 59–71.

Harrison, J. & D. M. Kreps (1979): "Martingales and arbitrage in multiperiod securities markets." *Journal of Economic Theory* **20(3)**: pp. 381–408.

Heston, S. L. (1993): "A Closed-Form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options." *The Review of Financial Studies* **6(2)**: pp. 327–343.

Horvath, B., A. Muguruza, & M. Tomas (2021): "Deep learning volatility: a deep neural network perspective on pricing and calibration in (rough) volatility models." *Quantitative Finance* **21(1)**: pp. 11–27. Publisher: Routledge.

Johannes, M. S., A. Kaeck, & N. Seeger (2023): "FOMC Announcement Event Risk." *SSRN Electronic Journal* .

Kahale, N. (2004): "An arbitrage-free interpolation of volatilities." *Risk Magazine* **17**: pp. 102–106.

Masters, T. (1993): *Practical Neural Network Recipes in C++.* Morgan Kaufmann.

Ning, B. X., S. Jaimungal, X. Zhang, & M. Bergeron (2023): "Arbitrage-Free Implied Volatility Surface Generation with Variational Autoencoders." *SIAM Journal on Financial Mathematics* **14(4)**: pp. 1004–1027. Publisher: Society for Industrial and Applied Mathematics.

Orosi, G. (2015): "Arbitrage-free call option surface construction using regression splines." *Applied Stochastic Models in Business and Industry* **31(4)**: pp. 515–527.

Roper, M. (2010): "Arbitrage Free Implied Volatility Surfaces." .

Shapley, L. S. (1953): "Stochastic games*." *Proceedings of the National Academy of Sciences* **39(10)**: pp. 1095–1100.

Todorov, V. (2019): "Nonparametric spot volatility from options." *The Annals of Applied Probability* **29(6)**.

Todorov, V. (2022): "Nonparametric jump variation measures from options." *Journal of Econometrics* **230(2)**: pp. 255–280.

Turner, P. R. (1994): *Numerical Differentiation*, pp. 160–171. London: Macmillan Education UK.

Vrontos, S. D., J. Galakis, & I. D. Vrontos (2021): "Implied volatility directional forecasting: a machine learning approach." *Quantitative Finance* **21(10)**: pp. 1687–1706.

Vuletić, M. & R. Cont (2023): "VolGAN: a generative model for arbitrage-free implied volatility surfaces." *SSRN Electronic Journal* .

Wallmeier, M. (2024): "Quality issues of implied volatilities of index and stock options in the OptionMetrics IvyDB database." *Journal of Futures Markets* **44(5)**: pp. 854–875.

Wright, J. H. (2020): "Event-day Options." *National Bereau of Econoic Research* .

Yatchew, A. & W. Härdle (2006): "Nonparametric state price density estimation using constrained least squares and the bootstrap." *Journal of Econometrics* **133(2)**: pp. 579–599.

Zhang, C., Y. Zhang, M. Cucuringu, & Z. Qian (2024): "Volatility Forecasting with Machine Learning and Intraday Commonality." *Journal of Financial Econometrics* **22(2)**: pp. 492–530.

Zhang, W., L. Li, & G. Zhang (2023): "A two-step framework for arbitrage-free prediction of the implied volatility surface." *Quantitative Finance* **23(1)**: pp. 21–34.

# A Results for monthly options

| Monthly data | | | Panel A: Same-day | |
|---|---|---|---|---|
| | Parametric | Error correction | Arbitrage Correction (LP) | Arbitrage Correction (LF) |
| BS | 10.34 | 0.68 | 0.60 | na |
| AHBS | 2.62 | 0.61 | 0.61 | na |
| CW | 3.39 | 0.63 | 0.61 | na |
| | | | Panel A: 1-day ahead | |
| BS | 9.86 | 1.72 | 1.95 | 2.80 |
| AHBS | 3.5 | 1.96 | 1.98 | 2.64 |
| CW | 3.38 | 1.85 | 1.92 | 2.91 |

Table 8: RMSE (%) of prediction in the option cross-section. This table depicts the RMSE of prediction in the cross section of monthly options for Black-Scholes, Adhoc Black Scholes, and Carr and Wu models. The Parametric column denotes the RMSE error of parametric models, Error correction gives the non-parametric correction RMSE and Error + Arbitrage presents the RMSE of nonparametric correction that limit static arbitrage.

| Monthly data | | | | | Panel A: Same Day | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Error Correction | | | Arbitrage Correction (LP) | | | Arbitrage Correction (LF) | | |
| | Calendar | Call | Butterfly | Calendar | Call | Butterfly | Calendar | Call | Butterfly |
| BS | 0.02 | $8.0 \times 10^{-8}$ | 0.05 | $6.4 \times 10^{-4}$ | 0.0 | $2.2 \times 10^{-15}$ | na | na | na |
| AHBS | 0.06 | $1.0 \times 10^{-3}$ | 0.01 | $1.1 \times 10^{-4}$ | 0.0 | $1.0 \times 10^{-16}$ | na | na | na |
| CW | 0.05 | 0.09 | 0.08 | $7.5 \times 10^{-3}$ | 0.0 | $3.4 \times 10^{-17}$ | na | na | na |
| | | | | | Panel B: 1 day ahead | | | | |
| BS | 0.05 | 0.00 | 5.87 | 0.00 | 0.00 | $9.38 \times 10^{-9}$ | 0.47 | 0.0 | 2.01 |
| AHBS | 0.27 | 0.00 | 2.28 | 0.00 | 0.00 | $2.45 \times 10^{-8}$ | 0.43 | 0.0 | 2.44 |
| CW | 0.29 | 0.01 | 1.69 | 0.00 | 0.00 | $3.25 \times 10^{-8}$ | 0.93 | 0.00 | 1.53 |
| Market | 0.00 | 0.20 | 11.2 | 0.00 | 0.20 | 11.2 | 0.00 | 0.20 | 11.2 |

Table 9: Mean Arbitrage Errors. This table depicts the mean arbitrage errors of monthly options for the market price and for the error and arbitrage correction using linear programming. The average is taken over 1st January 2018 to 28th February 2023. The Error and Arbitrage penalizations are based on the one-day-ahead test predictions.

# B Market Data

| Market data | Median | 90th quantile | 95th quantile | 99th quantile |
|---|---|---|---|---|
| Total | 0.0 | 0.0 | 0.0 | 0.33 |
| Calendar spread | 0.0 | 0.0 | 0.0 | 0.33 |
| Call spread | 0.0 | 0.5 | 1.0 | 2.9 |
| Butterfly spread adjusted | 7.9 | 28 | 39 | 140 |
| Butterfly spread (Cont & Vuletić, 2023) | 39 | 108 | 144 | 409 |

Table 10: This table presents the values of arbitrage violations for market data at different quantiles.

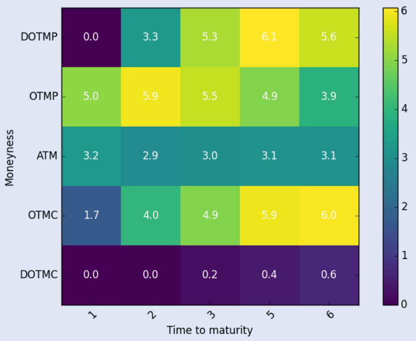
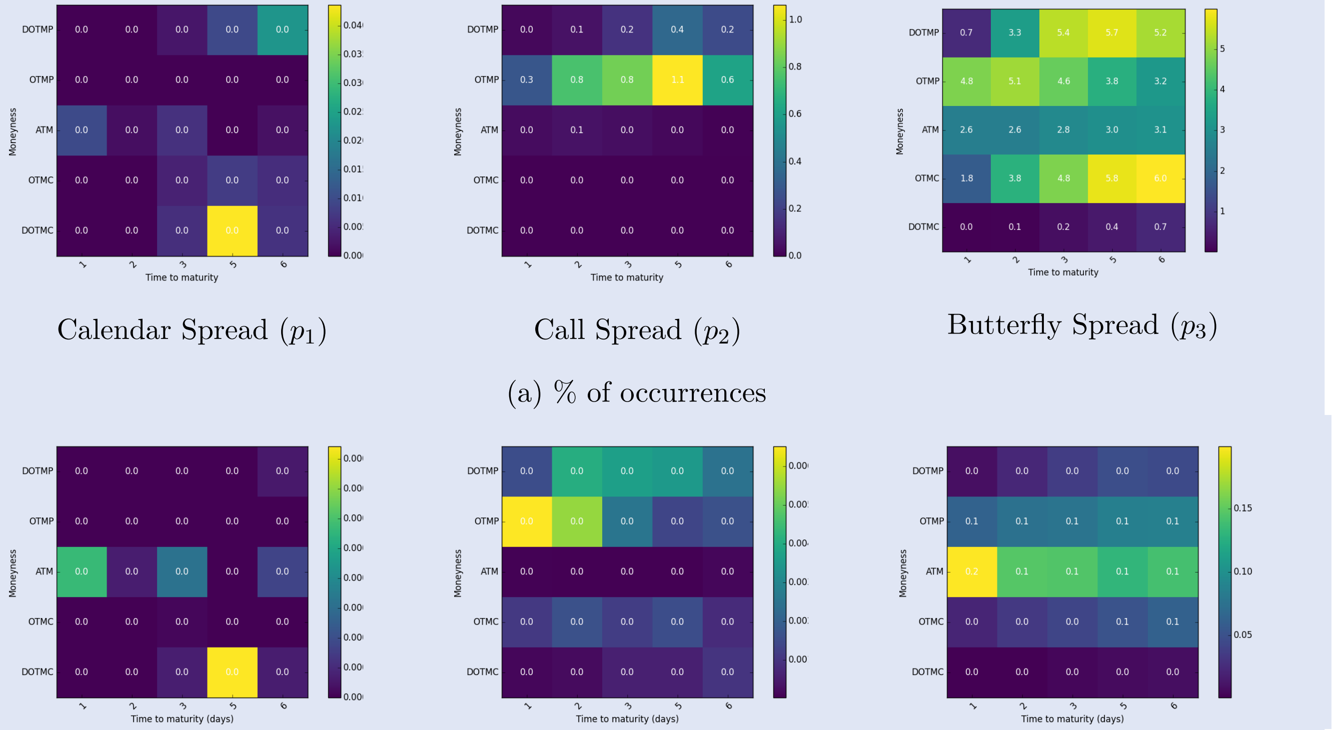


Figure 6: The percentage number of violations of all spreads on the moneyness and time-to-maturity grid for Market Data



Calendar Spread ($p_1$)

Call Spread ($p_2$)

Butterfly Spread ($p_3$)

(a) % of occurrences

Calendar Spread ($p_1$)

Call Spread ($p_2$)

Butterfly Spread ($p_3$)

(b) Magnitude $\Phi(m, \tau)$

Figure 7: The percentage number of violations of the given spread on the moneyness and time-to-maturity grid for the Market Data

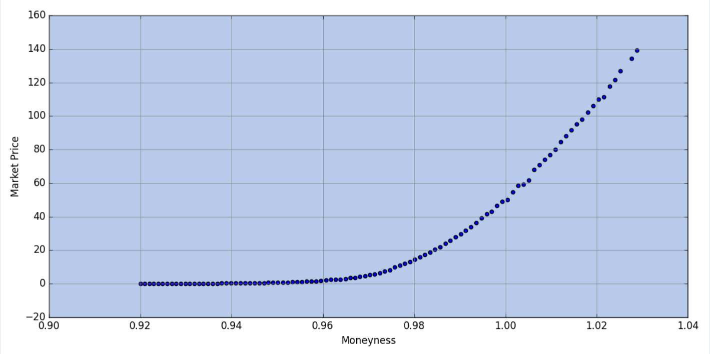
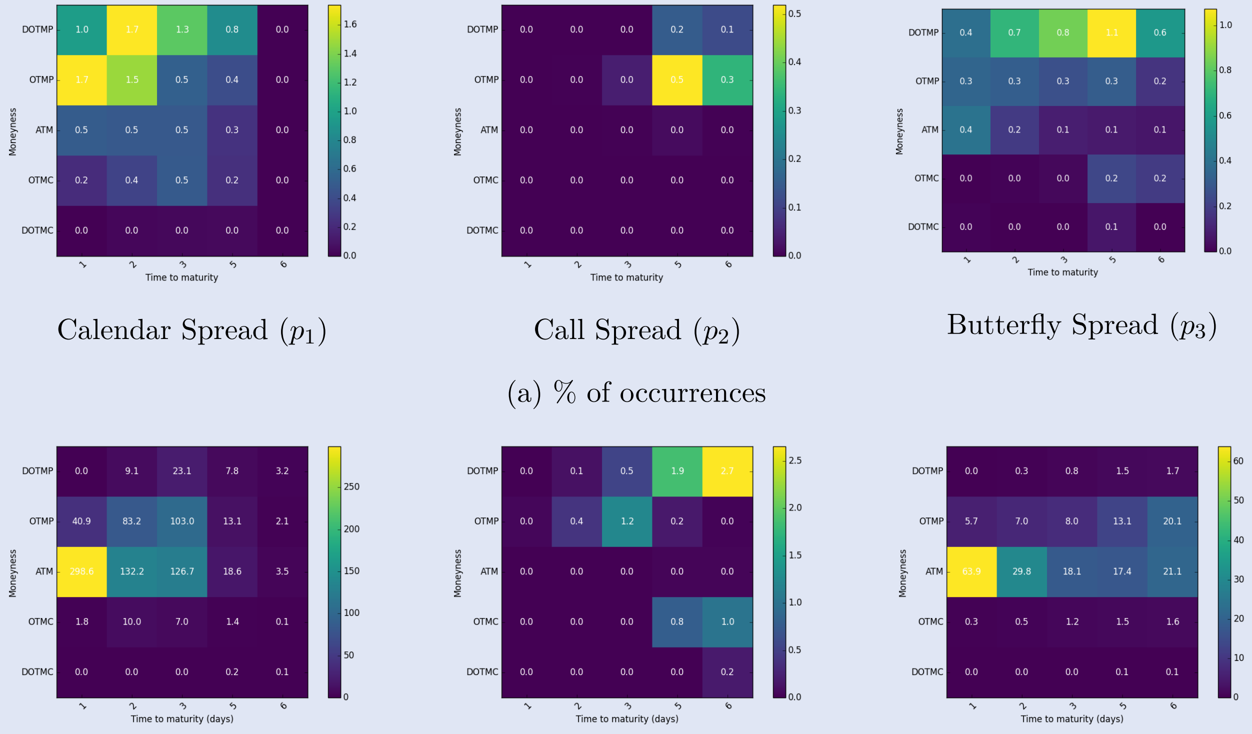


Figure 8: The Market Price on 26th January 2022. This plot shows the market price with respect to moneyness on x-axis.

## C Supplementary Results



Calendar Spread ($p_1$) Call Spread ($p_2$) Butterfly Spread ($p_3$)

(a) % of occurrences

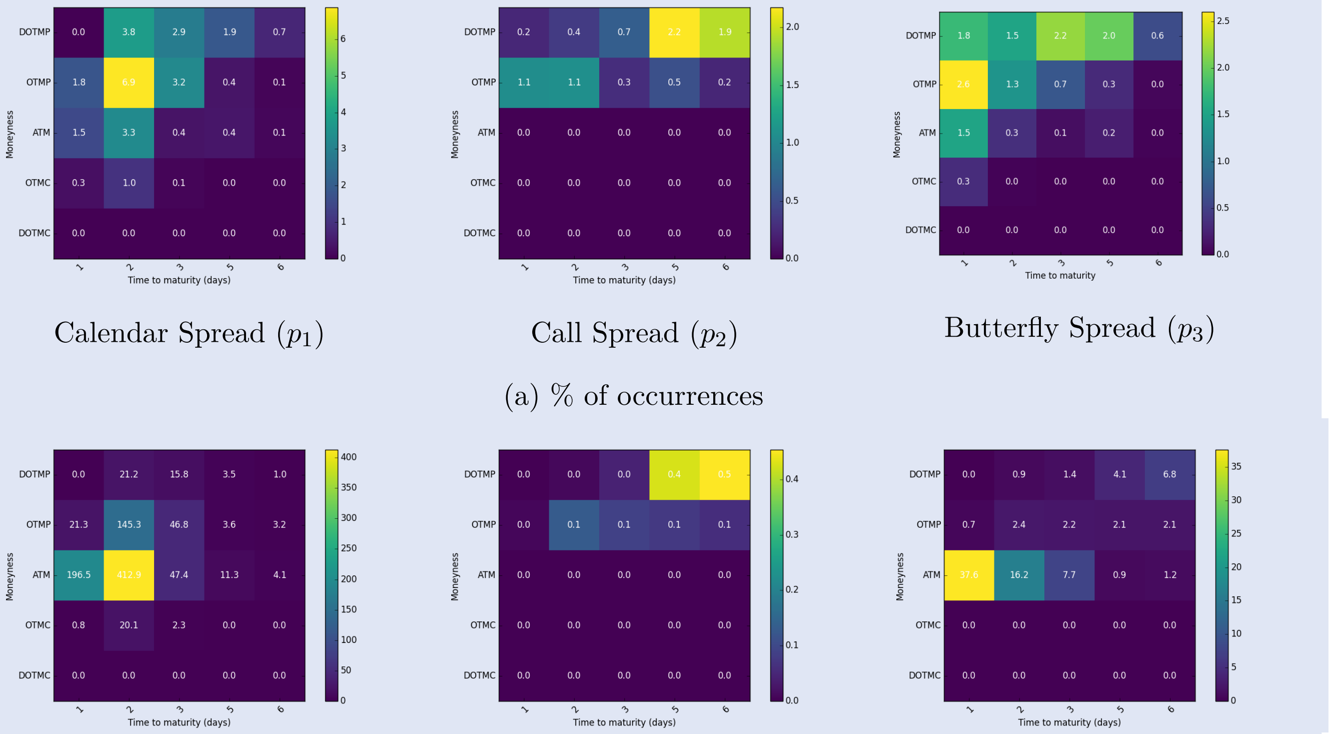
Calendar Spread ($p_1$) Call Spread ($p_2$) Butterfly Spread ($p_3$)

(b) Magnitude $\Phi(m, \tau)$

Figure 9: DECOMPOSITION OF FIGURE 3. The percentage number of violations of the given spread on the moneyness and time-to-maturity grid for the 1-day ahead Non-Parametric Correction (BS + NN)

| 1 Day Ahead | | | |
|---|---|---|---|
| | Parametric Models | | |
| | Calendar | Call | Butterfly |
| AHBS | 11.4 | 11.8 | 18.1 |
| CW | 0.18 | 0.32 | 0.65 |
| SSVI | 0.50 | 1.78 | 1.24 |

Table 11: Mean Arbitrage Errors of Parametric Mode. This table presents the arbitrage violations of the parametric models for 1-day ahead prediction. The BS was excluded as its constant volatility implies 0 arbitrage violations.



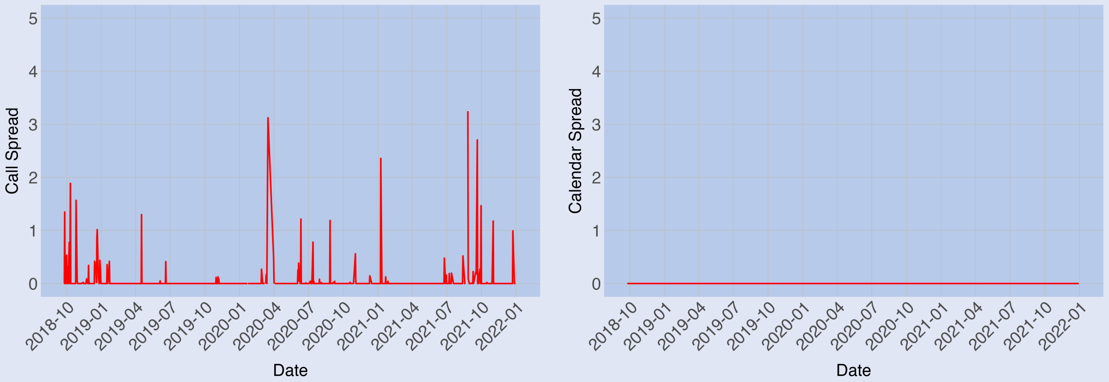
Calendar Spread ($p_1$) Call Spread ($p_2$) Butterfly Spread ($p_3$)

(a) % of occurrences

Calendar Spread ($p_1$) Call Spread ($p_2$) Butterfly Spread ($p_3$)

(b) Magnitude $\Phi(m, \tau)$

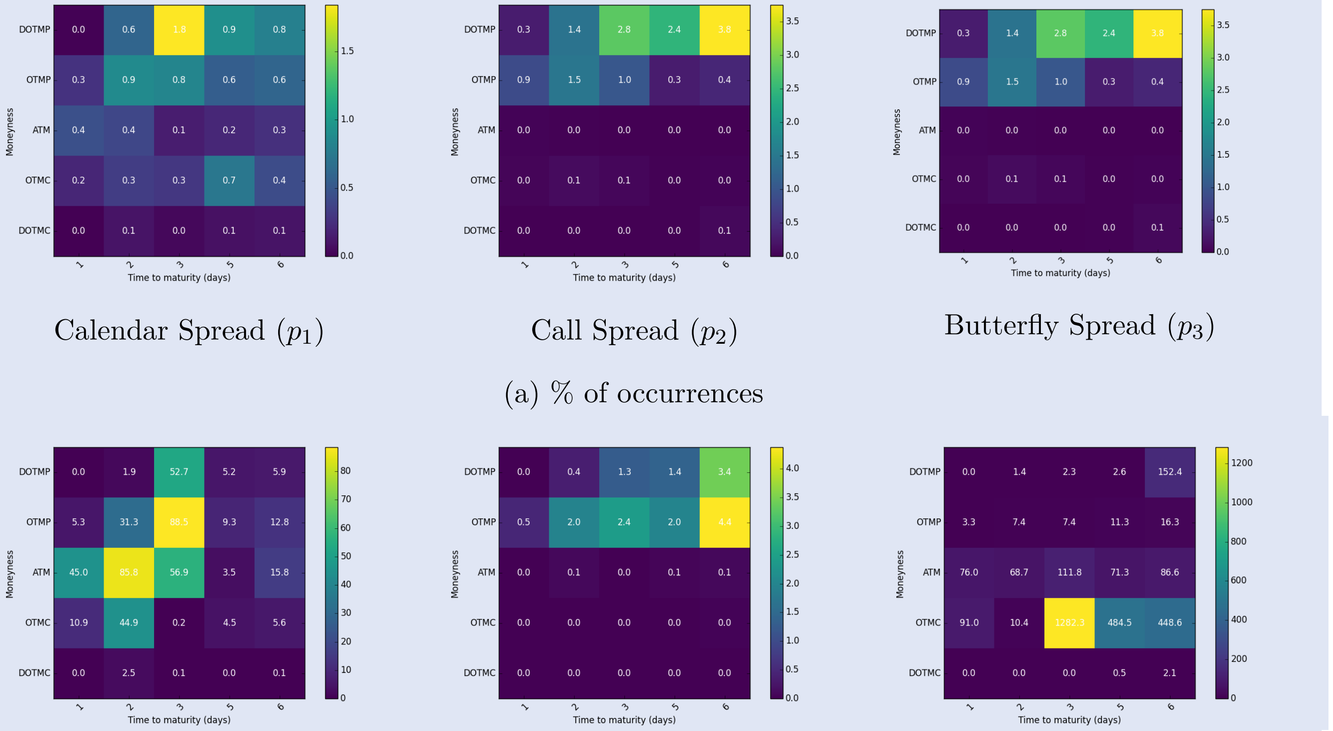
Figure 10: The percentage number of violations of the given spread on the moneyness and time-to-maturity grid for the 5-day ahead Non-Parametric Correction (BS + NN)



(a) Without Arbitrage Correction

(b) With Arbitrage Correction (LP)

Figure 11: Calendar Spread Violations of Error Corrected Black & Scholes with and without Linear Programming Arbitrage Correction for 1-Day Ahead Prediction



Calendar Spread ($p_1$)

Call Spread ($p_2$)

Butterfly Spread ($p_3$)

(a) % of occurrences

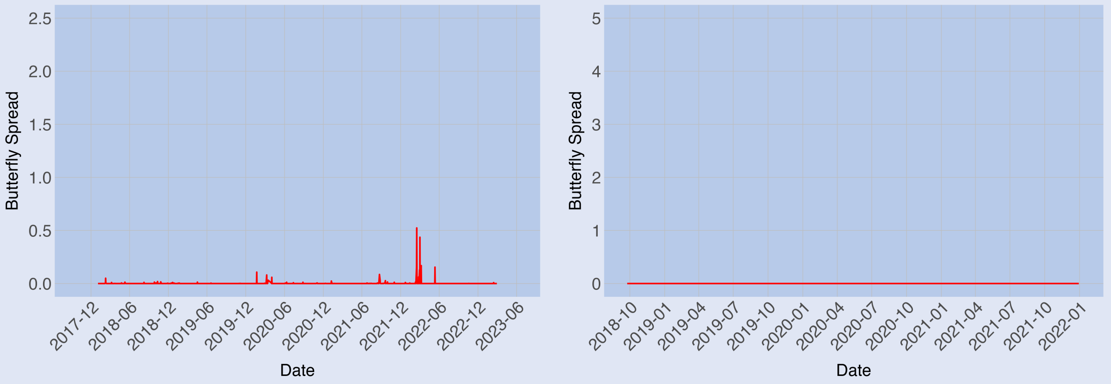
Calendar Spread ($p_1$)

Call Spread ($p_2$)

Butterfly Spread ($p_3$)

(b) Magnitude $\Phi(m, \tau)$

Figure 12: The percentage number of violations of the given spread on the moneyness and time-to-maturity grid for the 21-day ahead Non-Parametric Correction (BS + NN)



(a) Without Arbitrage Correction

(b) With Arbitrage Correction (LP)

Figure 13: Calendar Spread Violations of Error Corrected Black & Scholes with and without Linear Programming Arbitrage Correction for 1-Day Ahead Prediction

| Panel A: 5 days ahead | | | | | |
|---|---|---|---|---|---|
| | DOTMC | OTMC | ATM | OTMP | DOTMP |
| BS | 15.34 | 13.12 | 12.86 | **11.21** | 18.45 |
| BS + NN | 10.74 | 5.82 | **5.24** | 6.95 | 7.52 |
| BS + NN + LF | 10.05 | 6.23 | **4.91** | 5.11 | 5.22 |
| AdHoc BS | 10.23 | 8.56 | **5.98** | 9.23 | 13.12 |
| AdHoc BS + NN | 6.95 | 6.02 | **5.18** | 5.61 | 6.45 |
| AdHoc BS + NN + LF | 9.72 | 8.83 | **5.04** | **5.11** | 6.19 |
| Carr Wu | 18.15 | 10.23 | **5.65** | 8.95 | 14.03 |
| Carr Wu + NN | 9.34 | 5.32 | **4.95** | 6.51 | 8.24 |
| Carr Wu + NN + LF | 8.89 | 7.11 | **4.85** | **4.93** | **4.91** |
| SSVI | 17.59 | 8.75 | **5.14** | 8.69 | 12.45 |
| SSVI + NN | 9.51 | 4.74 | **4.36** | 5.89 | 9.95 |
| SSVI + NN + LF | 8.68 | 6.85 | **4.54** | **4.61** | 5.03 |
| Panel B: 21 days ahead | | | | | |
| BS | 18.92 | 13.78 | 12.12 | **11.55** | 21.40 |
| BS + NN | 12.65 | 7.25 | **6.62** | 8.43 | 11.56 |
| BS + NN + LF | 11.52 | 7.32 | 5.95 | **5.45** | 6.23 |
| AdHoc BS | 21.02 | 10.60 | **7.10** | 10.99 | 15.35 |
| AdHoc BS + NN | 8.02 | 7.25 | 6.91 | **6.86** | 7.90 |
| AdHoc BS + NN + LF | 8.93 | 7.87 | 6.43 | **6.20** | **5.81** |
| Carr Wu | 22.47 | 12.03 | **6.91** | 10.91 | 17.62 |
| Carr Wu + NN | 11.05 | 6.35 | **5.94** | 7.88 | 10.12 |
| Carr Wu + NN + LF | 10.49 | 8.10 | **5.82** | **5.89** | **5.84** |
| SSVI | 21.95 | 10.48 | **6.43** | 10.47 | 15.03 |
| SSVI + NN | 11.92 | 5.80 | **5.23** | 7.18 | 12.15 |
| SSVI + NN + LF | 10.95 | 8.46 | 5.45 | **5.55** | 5.76 |
| Number of observations | 7 066 | 83 910 | 347 511 | 199 783 | 91 710 |

Table 12: Decomposition of RMSE for 5 & 21 days ahead. This table depicts decomposition of RMSE into (deep) out-of-money calls: (D)OTMC, at-the-money options: ATM and (deep) out-of-money puts: (D)OTMP. The RMSE for LP approach are omitted due to being very similar to RMSE of BS + NN. The number in model denotes the lowest RMSE for given models.