Deep self-consistent learning of local volatility

Zhe Wang Ameir Shaa*
CNRS@CREATE Ltd
#08-01 CREATE Tower
1 Create Way, Singapore 138602

Nicolas Privault Claude Guet
School of Physical and Mathematical Sciences
Nanyang Technological University
21 Nanyang Link, 637371 Singapore.

October 16, 2025

Abstract

We present an algorithm for the calibration of local volatility from market option prices through deep self-consistent learning, by approximating both market option prices and local volatility using deep neural networks. Our method uses the initial-boundary value problem of the underlying Dupire's partial differential equation solved by the parameterized option prices to provide corrections to the parameterization in a self-consistent way. By exploiting the differentiability of neural networks, we can evaluate Dupire's equation locally at each strike-maturity pair; while by exploiting their continuity, we sample strike-maturity pairs uniformly from a given domain, going beyond the discrete points where the options are quoted. Moreover, the absence of arbitrage opportunities is imposed by penalizing an associated loss function as a soft constraint. For comparison with existing approaches, the proposed method is tested on both synthetic and market option prices, and it shows an improved performance in terms of reduced interpolation and reprice errors, as well as the smoothness of the calibrated local volatility. An ablation study has been performed, asserting the robustness and significance of the proposed method.

Keywords: Local volatility; deep learning; option pricing; Dupire PDE; physics-informed; self-consistent.

*Starting 9 Dec 2024, the second author will be with School of Physical and Mathematical Sciences, Nanyang Technological University, 637371 Singapore.

1 Introduction

An option is a financial contract that gives options holders the right but not the obligation to buy (a call option) or to sell (a put option) an asset, e.g. stocks or commodity, for a predetermined price (strike price) on a predetermined date (maturity). To acquire today the right to buy or sell an asset in the future, a premium, namely the option price, must be paid to the option writers. The standard approach to model option prices is based on the Black-Scholes formula, which assumes a constant local volatility surface and therefore applies only to situations where the stock prices process follows a geometric Brownian motion. However, the market price data of underlying assets typically invalidate the assumptions of standard Brownian motion, hence a more sophisticated model than the Black-Scholes formula is required. Among alternatives, we consider an asset price process expressed in a local volatility model as

$$dS_t = rS_t dt + \sigma(S_t, t)S_t dB_t, \quad S_{t|t=0} = S_0. \tag{1}$$

where S_t denotes the stock price at the time $t \in \mathbb{R}_+$, and $(B_t)_{t\in\mathbb{R}_+}$ is a standard Brownian motion. For simplicity, we assume a zero dividend yield and a constant risk-free interest rate r. Rather than taking a constant value as in the canonical Black-Scholes equation, in Eq. (1), the local volatility $\sigma(S_t, t)$ is a deterministic function of the underlying asset price S_t and of time t, satisfying the usual Lipschitz conditions. Without introducing additional sources of randomness, the local volatility model is the only complete and consistent model that allows hedging based on solely the underlying asset, cf. Appendix A1 in Bennett (2014); and it is used for daily risk management in most investment bank production systems.

The practical implementation of such a stochastic volatility model in option pricing requires to solve the challenging problem of calibrating the local volatility function to the market data of option prices. Given $\pi(K, T)$ a family of option prices with strike prices $K > 0$, maturities $T > 0$, and underlying asset price $S_0 = x$, the Dupire formula Dupire (1994); Derman and Kani (1994)

$$\sigma(K, T) := \sqrt{\frac{2\frac{\partial\pi}{\partial T}(K, T) + 2rK\frac{\partial\pi}{\partial K}(K, T)}{K^2\frac{\partial^2\pi}{\partial K^2}(K, T)}}, \tag{2}$$

brings a solution to this problem by constructing an estimator of the local volatility as a function of strike price K and maturity T values, i.e. Eq. (2) matches the stock's volatility $\sigma(S_t, t)$ when the underlying stock price S_t is at the level K at time $T = t$.

The numerical estimation of local volatility using Eq. (2), involves evaluating partial derivatives, which has classically been achieved using the finite difference method. More efficient methods have been introduced using spline functions, see e.g. Chapter 8 in Achdou and Pironneau (2005), where local volatility is first approximated as the sum a piecewise affine function satisfying a boundary condition and a bi-cubic spline function. Then, the model parameters are determined by minimizing the discretized Dupire (1994) equation at selected collocation points via Tikhonov regularization. Thereafter, Tikhonov regularization has been applied to the calibration of local volatility in a trinomial model in Crépey (2002).

As an alternative to calibrating local volatility using parameterized functions, neural networks are able to approximate any arbitrary nonlinear function with a finite set of parameters Gorban and Wunsch (1998); Winkler and Le (2017); Lin and Jegelka (2018). By exploiting this property, Chataigner et al. Chataigner et al. (2020) advocated to parameterize the market option prices $\pi(K,T)$ using neural networks, which allow for the computation of the derivatives in Eq. (2) by automatic differentiation Baydin et al. (2018).

In addition, for Dupire's formula to be meaningful, one must ensure that the argument in the square root is positive and remains bounded in (2), which can be achieved under a no arbitrage condition for the quoted option prices. In Chataigner et al. (2020), additional constraints are imposed in order to ensure that the market option prices are well fitted, in which case an approximation of derivatives of the option price is then substituted into Eq. (2) to calibrate the local volatility.

In this case, the quality of the reconstructed local volatility surface depends on the resolution and quality of the quoted market option prices at given strike-maturity pairs. Hence, instead of using market option prices, Chataigner et al. Chataigner et al. (2021) proposed subsequently, to use the implied volatility surface for the calibration of local volatility, leading to an improved performance. It is, however, important to note that, when fitting option prices or implied volatilities using neural networks, one assumes that the data are noise free. Except for those that violate the arbitrage-free conditions, the proposed methods in Chataigner et al. (2020, 2021) are not able to filter out noises in the data.

In this work, we follow the approach of Wang and Guet (2022b), and calibrate the local volatility surface from the observed market option prices in a self-consistent manner. Going beyond merely fitting option prices using neural networks, we rely on recent studies on physics-informed machine learning Raissi et al. (2019); Rackauckas et al. (2020); Karniadakis et al. (2021); Wang and Guet (2022b) which have shown that the inclusion of the residual

of an unknown, underlying governing equation as a regularizer can not only calibrate the unknown terms in the governing equations, but also correct the data by filtering out noises that is not defined by the equation.

To do this, we approximate both the market option prices and the squared local volatility using deep neural networks. By requiring $\pi(K,T)$ to be a solution to the Dupire equation subjected to given initial and boundary conditions to be discussed in Section 2.1, we regularize $\pi(K,T)$ and determine the unknown $\sigma(K,T)$ self-consistently. Since the resulting $\pi(K,T)$ and $\sigma(K,T)$ are continuous and differentiable functions of K and T, we can evaluate Dupire's equation locally on a uniformly sampled (K,T) from the domain of Dupire's equation. Unlike previous approaches, where constraints were imposed on fixed strike-maturity pairs, a successive resamplings enable us to regularize the entire domain. In addition, the positiveness of $\sigma(K,T)$ is ensured by a properly selected output activation function of the neural network, hence it is everywhere well defined. Finally, we mitigate the risk of arbitrage opportunities by penalizing a loss as a soft constraint. In this way, local volatility is estimated by construction as a smooth surface instead of applying automatic differentiation to neural option prices. In addition, noise in market data is filtered out from the constraint imposed by Dupire's equation. PyTorch and TensorFlow implementations of our algorithm are available from the "Local Volatility Modeling with Neural Networks" Github repository `https://github.com/ameirtheshaa/LocalVolatility`.

The rest of the paper is organized as follows. In Section 2 we review background knowledge on Dupire's equation and arbitrage-free conditions, followed by a comment on market data. After rescaling and reparameterization of market option prices in Section 3.1 and a discussion of neural ansatz and loss functions in Section 3.2, our deep self-consistent algorithm is summarized and discussed in Section 3.3. The proposed method is first tested on synthetic datasets in Section 4.1, and then applied to market option prices in Section 4.2 where, as an ablation study, we compare our results with those obtained without including the residual of Dupire's equation as a regularizer. Finally, the advantages and limitations of our method are discussed in Section 5, where conclusions are drawn.

2 Background and objectives

2.1 Dupire's equation

Denoting by $\pi^{\mathrm{c}}(K,T)$ and $\pi^{\mathrm{p}}(K,T)$ the prices for European call and put options

$$\pi^{\mathrm{c}}(K,T) = e^{-rT}\mathbb{E}\big[(S_T-K)^+\big], \qquad \pi^{\mathrm{p}}(K,T) = e^{-rT}\mathbb{E}\big[(K-S_T)^+\big], \tag{3}$$

the relation between the market option price and the local volatility is established explicitly by Dupire's equation Dupire (1994)

$$\frac{\partial \pi}{\partial T}(K,T) + rK\frac{\partial \pi}{\partial K}(K,T) - \frac{1}{2}K^2\sigma^2(K,T)\frac{\partial^2 \pi}{\partial K^2}(K,T) = 0, \tag{4}$$

subject to the initial conditions

$$\pi^{\mathrm{c}}(K,0) = (S_0-K)^+, \qquad \pi^{\mathrm{p}}(K,0) = (K-S_0)^+,$$

and to the boundary conditions

$$\pi^{\mathrm{c}}(\infty,T) = 0, \qquad \pi^{\mathrm{p}}(0,T) = 0,$$

respectively for call and put options. We refer the reader to Chapter 1 in Itkin (2020) or Proposition 9.2 in Privault (2022) for derivations of Dupire's equation, and to Chapter 8 in Achdou and Pironneau (2005) for a discussion of initial and boundary conditions. From the definitions of European option prices in (3), we note that these are nonnegative, with the call option price being always less than the value of the underlying asset, and the price of the put options being always less than the present value of the strike price, leading to the lower and upper bounds:

$$0 \le \pi^{\mathrm{c}}(K,T) \le S_0, \qquad 0 \le \pi^{\mathrm{p}}(K,T) \le K. \tag{5}$$

2.2 Arbitrage-free surface

A static arbitrage is a trading strategy that has zero initial cost, and subsequently a non-negative value, representing a risk-free profitable investment. Under the assumption that economic agents are rational, arbitrage opportunities, if they ever exist, will be instantaneously exploited until the market is arbitrage free. Therefore, as a prerequisite, the absence of arbitrage opportunities is a fundamental principle underpinning the modern theory of financial asset pricing. An option price surface is free of static arbitrage if and only if

(i) each strike slice is free of calendar spread arbitrage; and

(ii) each maturity slice is free of butterfly arbitrage Ackerer et al. (2020); Chataigner et al. (2020, 2021).

Next, we consider both calendar spread and butterfly arbitrage, which can be created with put options as well as call options, cf. Sec. 9.2 in Hull (2003) for details. Calendar spread arbitrage is created by buying a long-maturity put option and selling a short-maturity put option , and the absence of arbitrage can be characterized in the continuous limit by the condition

$$\frac{\partial \pi}{\partial T}(K,T) \geq 0.$$

On the other hand, butterfly arbitrages can be created by buying two options at prices K_1 and K_3, with $K_3 > K_1$; and selling two options with a strike price, $K_2 = (K_1 + K_3)/2$. In the continuous limit, the absence of butterfly arbitrage is formulated as

$$\frac{\partial^2 \pi}{\partial K^2}(K,T) \geq 0.$$

2.3 Problem formulation

Market option prices are observed on N discrete pairs of strike prices and maturity values, leading to triplets (π_i, K_i, T_i) with $i = 1, ..., N$. In practice, to secure a transaction, two option prices are quoted in the market, a bid price and an ask price, and there is no guarantee that the mid-price π_i between the bid and ask prices is arbitrage-free Ackerer et al. (2020). Moreover, the observed bid and ask prices may not be updated timely, hence not actionable, leading to additive noise to the market data. Lastly, market option prices are not evenly quoted over a plane spanned by K and T: the market data are usually dense close to the money; whereas sparse away from the money. Canonical methods often lack the ability to agilely interpolate option prices and calibrate the associated local volatility without quotes, agilely. Therefore, a validation, a correction, and an interpolation to the market data are required.

Consequently, the main challenge when calibrating the local volatility is twofold. First, given limited observations of unevenly distributed option prices quoted on discrete pairs of strikes and maturities, the calibrated local volatility should span a continuous domain. Second, the corresponding option prices should preclude any arbitrage opportunities. Our

objective is, therefore, to construct from observed market option prices a self-consistent approximation for both $\sigma(K,T)$ and $\pi(K,T)$ in the sense that the calibrated local volatility will generate an arbitrage-free option price surface that is in line with the market data up to some physics-informed corrections.

3 Methodology

3.1 Rescaling Dupire's equation

In this section, a change of variable and a rescaling is applied to Dupire's equation (4) to ensure that the terms in the scaled equation, (7), are of order of unity, so that no term is dominating over the others during the minimization. For this, we use the change of variables

$$k = \frac{e^{-rT}K}{K_{\max}}, \quad t = \frac{T}{T_{\max}}, \tag{6}$$

by picking sufficiently large $K_{\max} = \max(K)$ and $T_{\max} = \max(T)$. Letting

$$\eta(k,t) := \frac{T_{\max}}{2}\sigma^2(K,T)$$

denote the squared local volatility, the Dupire equation (4) can be rewritten as

$$f_{\text{dup}}(k,t) := \frac{\partial \pi}{\partial t}(k,t) - \eta(k,t)k^2\frac{\partial^2 \pi}{\partial k^2}(k,t) = 0, \tag{7}$$

subject to the rescaled initial and boundary conditions

$$\pi^{\text{c}}(k,0) = \left(S_0 - K_{\max}k\right)^+, \quad \pi^{\text{c}}(\infty,t) = 0, \tag{8}$$

for call option prices, and

$$\pi^{\text{p}}(k,0) = \left(K_{\max}k - S_0\right)^+, \quad \pi^{\text{p}}(0,t) = 0, \tag{9}$$

for put option prices, as well as the inequality constraints (5). We note the conditions

$$\frac{\partial \pi}{\partial T}(K,T) \propto \frac{\partial \pi}{\partial t}(k,t) - rT_{\max}k\frac{\partial \pi}{\partial k}(k,t) \geq 0, \tag{10}$$

and

$$\frac{\partial^2 \pi}{\partial K^2}(K,T) \propto \frac{\partial^2 \pi}{\partial k^2}(k,t) \geq 0, \tag{11}$$

for the absence of strike and calendar arbitrage opportunities respectively, see, e.g. §2.2.2 of Bergomi (2016).

Lemma 3.1 *Conditions* (10) *and* (11) *for the absence of strike and calendar arbitrage can be simultaneously enforced under the single inequality*

$$f_{arb}(k,t) := \frac{\partial \pi}{\partial t}(k,t) - rT_{max}k\left(\frac{\partial \pi}{\partial k}(k,t)\right)^{+} \geq 0. \tag{12}$$

Proof. We consider two cases.

(i) $\partial\pi/\partial k \geq 0$. In this case, (12) is clearly equivalent to (10). In addition, if (10) holds then (7) shows that (11) is satisfied, as

$$\eta(k,t)k^2\frac{\partial^2 \pi}{\partial k^2}(k,t) = \frac{\partial \pi}{\partial t}(k,t) \geq rT_{\max}k\frac{\partial \pi}{\partial k}(k,t) \geq 0.$$

(ii) $\partial\pi/\partial k < 0$. In this case, (12) becomes

$$\frac{\partial \pi}{\partial t}(k,t) \geq 0, \tag{13}$$

which by (7) is equivalent to (11). In addition, if (13) holds then (10) is satisfied. □

3.2 Model and loss function

Market option prices are observed at discrete strike-maturity pairs, whereas it is favorable to have an option price and a local volatility that span continuous surfaces over the extended regimes of strike price and maturity. To obtain a continuous limit, we propose to model both the option price and the local volatility using neural networks. The associated neural network ansatz and the corresponding loss functions are detailed below.

3.2.1 Neural network ansatz

As option prices may vary dramatically against strike prices, we model the option price surface as an exponential function where the exponent is approximated using a neural network, whose model parameters $\boldsymbol{\theta}$ are learned from the market data. We consider the neural ansätze

$$\pi_\theta^{\mathrm{c}}(k,t) = S_0\big(1 - e^{-(1-k)\mathcal{N}_c(k,t;\boldsymbol{\theta}_c)}\big), \tag{14}$$

and

$$\pi_\theta^{\mathrm{p}}(k,t) = K_{\max}e^{rT_{\max}t}k\big(1 - e^{-k\mathcal{N}_p(k,t;\boldsymbol{\theta}_p)}\big), \tag{15}$$

for call and put option prices, respectively. At the practical infinity $k = 1$ and at $k = 0$, the boundary conditions for call (8) and for put (9) options are satisfied by construction. Here, a

softplus function, $\text{softplus}(x) := \log(1 + e^x)$ is selected to be the activation function at the output layer of neural networks, so that the inequality constraints (5) are always satisfied.

Unlike for option prices, the variation of local volatility is relatively small, hence we model it using the neural network

$$\eta_\theta(k,t) = \mathcal{N}_\eta(k,t;\boldsymbol{\theta}_\eta), \quad \mathcal{N}_\eta : \mathbb{R}_+ \times \mathbb{R}_+ \to \mathbb{R}_+, \tag{16}$$

with a softplus activation function at the output layer, so that $\eta_\theta(k,t)$ is non-negative by construction. By modeling the squared local volatility using a neural network we also ensure the smoothness of local volatility, thus mitigating numerical instabilities.

Throughout the paper, we use the same neural network architecture for $\mathcal{N}_c$, $\mathcal{N}_\text{p}$ and $\mathcal{N}_\eta$. The network consists of three residual blocks He et al. (2016), where the residual connection is used around two sublayers formed of 64 neurons per layer and regularized with batch normalization Ioffe and Szegedy (2015) per layer. The model parameters of neural networks $\boldsymbol{\theta}_\text{c}$ (or $\boldsymbol{\theta}_\text{p}$) and $\boldsymbol{\theta}_\eta$ are determined by minimizing properly designed loss functions to be discussed in Section 3.2.2. Since evaluating loss functions associated with the arbitrage-free conditions and with the Dupire equations requires computing successive derivatives, the tanh activation function is selected for hidden network layers.

3.2.2 Loss functions

Given N market data triplets (π_i, K_i, T_i) with $i = 1, ..., N$, we transform K_i and T_i using Eq. (6), leading to (π_i, k_i, t_i), and consider the following loss functions.

i) The data loss arising from fitting market option prices using neural networks is defined as the squared error

$$L_\text{fit} = \frac{1}{N}\sum_{i=1}^{N} w(\pi_i)\big(\pi_\theta(k_i,t_i) - \pi_i\big)^2$$

with the weight function

$$w(\pi_i) := 1 + \frac{N g(\pi_i)}{\sum_{j=1}^{N} g(\pi_j)},$$

where

$$g(x_i) := \frac{1}{10}\mathbb{1}_{[0,0.1)}(x_i) + x_i\mathbb{1}_{[0.1,10]}(x_i) + 10\mathbb{1}_{[10,\infty)}(x_i),$$

and

$$x_i = \text{stop\_gradient}(\tilde{x}_i), \quad \text{with} \quad \tilde{x}_i = \frac{1}{N\pi_i^2}\sum_{j=1}^{N}\pi_j^2, \tag{17}$$

so that squared and relative squared errors are minimized. Here, indicator functions are used to mitigate numerical instabilities associated with $\pi_i = 0$, while leveraging contributions from large values of π_i. In addition, in (17) the operator stop_gradient$(\cdot)$ prevents the input variables of indicator functions from being taken into account when computing gradients, hence the weight function $w(\cdot)$ is treated as a scalar quantity during backpropagation. As the weight function w is designed to balance losses evaluated at each collation point, it will be included in all loss functions to be defined in the following.

ii) As the initial conditions in Eqs. (8) and (9) are piecewise linear and not differentiable, instead of imposing initial conditions as hard constraints into the neural ansatz we choose to minimize the loss function

$$L_{\text{ini}} = \frac{1}{M_1}\sum_{j=1}^{M_1} w\left(\pi(k_j,0)\right)\left(\pi_\theta(k_j,0) - \pi(k_j,0)\right)^2,$$

where the reference function $\pi(k_j, 0)$ is defined in Eq. (8) for call options, and in Eq. (9) for put options. To evaluate L_{ini}, M_1 synthetic collocation points are uniformly sampled from the interval $k_j \in [0, 1]$ at each iteration during the minimization.

iii) To ensure that the parameterized option price surface is arbitrage-free, on M_2 synthetic collocation points (k_j, t_j) uniformly sampled from $\Omega := [0, 1] \times [0, 1]$, we substitute $\pi_\theta(k_j, t_j)$ into the function $f_{\text{arb}}(k, t)$ defined in Eq. (12), and penalize the locations where $f_{\text{arb}}(k_j, t_j)$ is negative, which results into the loss function

$$L_{\text{arb}} = \frac{1}{M_2}\sum_{j=1}^{M_2} w\left(\frac{\partial \pi_\theta}{\partial t}(k_j,t_j)\right)\left((-f_{\text{arb}}(k_j,t_j))^+\right)^2$$

associated with the arbitrage-free conditions.

iv) Under the assumption that the price process of underlying assets of the option matches the local volatility model (1), $\pi_\theta(k, t)$ must be a solution to the rescaled Dupire's equation (7) up to some error terms. The self-consistency between the parameterized option price and the calibrated local volatility is therefore established by minimizing the residual

$f_{\text{dup}}(k,t)$ defined in Eq. (7), leading to

$$L_{\text{dup}} = \frac{1}{M_2}\sum_{j=1}^{M_2} w\left(\frac{\partial \pi_\theta}{\partial t}(k_j,t_j)\right)(f_{\text{dup}}(k_j,t_j))^2,$$

where, similar to L_{arb}, the loss function arising from the Dupire equation is evaluated from the ensemble of uniformly sampled M_2 collocation points.

In addition, the initial condition provides reference option prices on collocation points sampled on the line $T = 0$ that are equivalent to the quoted market option prices. Successive resamplings then covers the entire support domain Ω, ensuring that both the arbitrage-free condition and the Dupire equation are satisfied everywhere. This not only avoids the need to design an adaptive mesh for Dupire's equation as in Achdou and Pironneau (2005), but also mitigates overfitting that can be due to the uneven distribution of scarce data. Computing loss functions L_{arb} and L_{dup} involves evaluating differential operators, which is conveniently achieved by automatic differentiation Baydin et al. (2018).

Finally, the joint minimization of

$$L_\pi = L_{\text{fit}} + \lambda_{\text{ini}} L_{\text{ini}} + \lambda_{\text{arb}} L_{\text{arb}} + \lambda_{\text{dup}} L_{\text{dup}}, \tag{18}$$

with respect to $\boldsymbol{\theta}_{\text{c}}$ (or $\boldsymbol{\theta}_{\text{p}}$) provides a correction to the observed market data by filtering out noises that may lead to arbitrage opportunities or violate the price dynamics underpinned by the scaled Dupire's equation. Without loss of generality, we take $\lambda_{\text{ini}} = \lambda_{\text{arb}} = 1$, reducing the number of hyper parameters to one: $\lambda_{\text{dup}} = \lambda \in \mathbb{R}_+$.

On the other hand, the minimization of L_{dup} with respect to $\boldsymbol{\theta}_\eta$ ensure consistency between the local volatility function estimate and the parameterized option price. In this sense, the proposed method is self-consistent. To assess the robustness of the proposed self-consistent method, various values of λ are considered in Section 4 as an ablation study.

3.3 Algorithm

Based on the deep self-consistent learning method discussed above, Algorithm 1 outlines a computational scheme for the calibration of local volatility. At variance with the usual approaches whereby the derivatives of the option price are first approximated to calibrate the local volatility using Dupire's formula (2), this algorithm approximates both the option price and the local volatility using neural networks.

Algorithm 1 Deep self-consistent learning for option prices and local volatility.

Input: Scaled market call (or put) option prices and transformed strike-maturity pairs (π_i, k_i, t_i) with $i = 1, ..., N$.
Guess initial parameters $\boldsymbol{\theta}_{\mathrm{c}}$ (or $\boldsymbol{\theta}_{\mathrm{p}}$) and $\boldsymbol{\theta}_{\eta}$.

1: **while** not converged **do**
2: Uniformly draw M_1 points $k \in [0, 1]$ and M_2 scaled strike-maturity pairs $(k, t) \in \Omega$.
3: Compute 4 loss functions: L_{fit}, L_{ini}, L_{arb}, L_{dup}, and their weighted sum L_{π}.
4: Optimize $\boldsymbol{\theta}_{\mathrm{c}}$ (or $\boldsymbol{\theta}_{\mathrm{p}}$) using gradients $\partial_{\boldsymbol{\theta}_{\pi}} L_{\pi}$.
5: Optimize $\boldsymbol{\theta}_{\eta}$ using gradients $\partial_{\boldsymbol{\theta}_{\eta}} L_{\mathrm{dup}}$.
6: **end while**

Output: Optimized parameters $\boldsymbol{\theta}_{\mathrm{c}}$ (or $\boldsymbol{\theta}_{\mathrm{p}}$) and $\boldsymbol{\theta}_{\eta}$.
Hyper-parameters used in this paper: $M_1 = 128$, $M_2 = 128^2$, $\lambda_{\mathrm{ini}} = \lambda_{\mathrm{arb}} = 1$ and $\lambda_{\mathrm{dup}} = \lambda$.

The training of neural networks is a non-convex optimization problem which does not guarantee convergence towards the global minimum, hence regularization is required and provided via the residual of the scaled Dupire equation in the term L_{dup}. The joint minimization of L_{fit} arising from the deviation of the parameterized option price to the market data, L_{ini} due to the initial conditions, L_{arb} linked with the arbitrage-free conditions, and L_{dup} seeks a self-consistent pair of approximations for the option price and for the underlying local volatility, and to exclude local minima that violate these constraints.

Moreover, in practice, strike-maturity pairs with quoted option prices are usually unevenly distributed Ackerer et al. (2020); Chataigner et al. (2021); hence, the direct minimization of loss functions evaluated on the quoted strike-maturity pairs may yield large calibration errors of local volatility at locations where the measurement is scarce. On the other hand, being continuous functions, neural networks allow for a mesh-free discretization of Dupire's equation by uniformly sampling collocation points from Ω at each iteration, thus improving calibration from scarce and unevenly distributed data. Lastly, since both option price and squared local volatility are approximated using neural networks, the smoothness of the option price and local volatility surface are guaranteed, while positiveness is ensured by a proper choice of output activation functions. As a result, the numerical instabilities associated with the calibration of local volatility using the Dupire formula (2) are less likely to occur.

4 Case studies

In this section, we test our self-consistent method, first on synthetic option prices in Section 4.1, and then on real market data in Section 4.2. We start the training with an initial learning

rate 10^{-3}, which is divided by a decaying factor 1.1 every $2,000$ iterations. On a workstation equipped with an Nvidia RTX 3090 GPU card, each iteration takes less than 0.01 second. The total number of iterations is capped at $30,000$ such that the overall computational time is compatible with that in Chataigner et al. (2020, 2021), enabling a fair comparison. Figure 1 presents a typical graph of loss functions L_{ini}, L_{dup} and L_{arb} observed in the following experiments as the number of iterations increases.

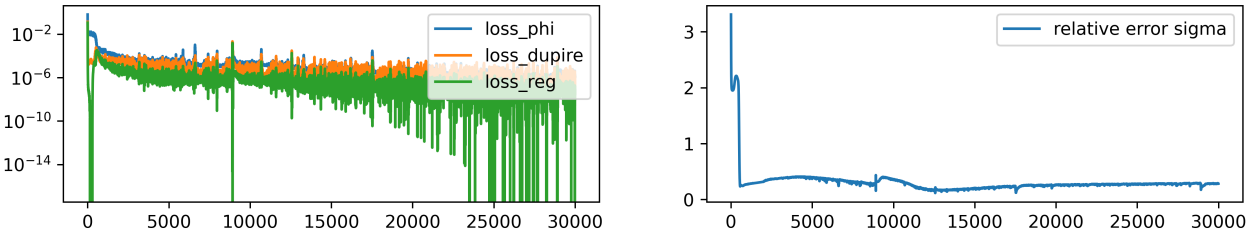

Figure 1: Left: violations of the non-arbitrage condition; and right: relative local volatility error.

Figure 2 presents a heatmap of the function f_{arb}, showing the no-arbitrage violations at various collocation points.

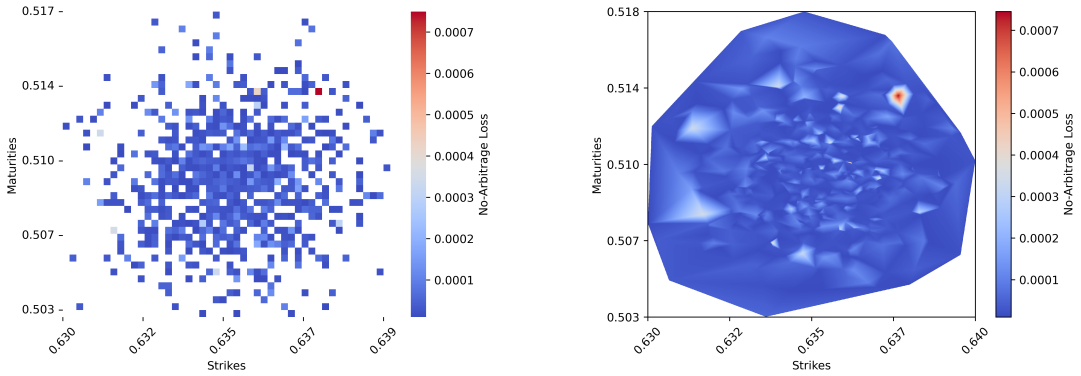

Figure 2: Left: heatmap of non-arbitrage condition violations; and right: linear interpolation.

4.1 Synthetic data for European call options

Synthetic option prices are generated as Monte Carlo estimates of Eq. (3) for European call options at given strike-maturity pairs (K_i, T_i). Here, the asset price paths S_t are obtained by simulating the local volatility model (1) with $r = 0.04$ and

$$\sigma(x,t) = 0.3 + ye^{-y} \quad \text{with} \quad y = (t+0.1)\sqrt{x+0.1}, \tag{19}$$

making it possible to assess the validity of the calibrated local volatility by comparison with the closed-form expression of $\sigma(x,t)$. The price trajectories are obtained by simulating Eq. (1)

for 10^6 times from a single initial condition $S_0 = 1,000$ for the period $t = [0, 1.5]$ and with a time step $\mathrm{d}t = 0.01$.

Large data set

To begin with, we price European call options on a mesh grid consisting of 10 evenly spaced points in $[0.3, 1.5]$ for period T and 20 evenly spaced points K in the interval $[500, 3000]$, respectively. That is, the dataset consists of 10×20 option prices quoted at the corresponding strike-maturity pairs, leading to the triplet $(\pi_i^{\mathrm{c}}, K_i, T_i)$ with $i = 1, ..., 200$. The exact local volatility surface, the simulated price trajectories by numerically integrating the local volatility model, and the synthetic call option price surface are visualized in Figure 3.

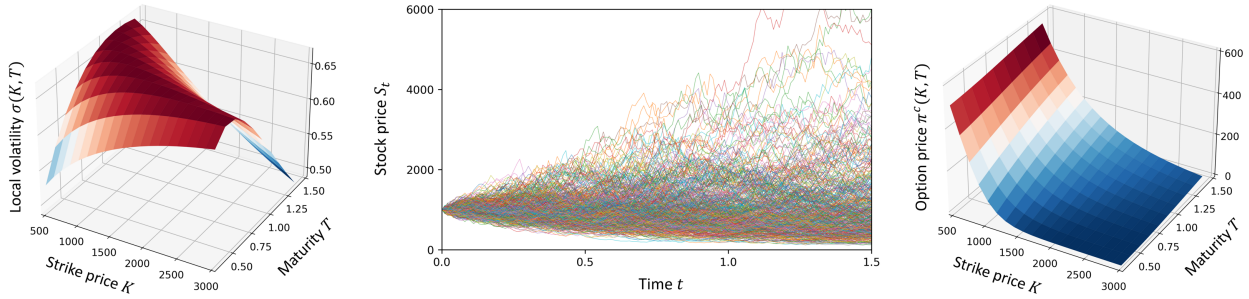

Figure 3: Left: exact local volatility surface, cf. Eq. (19); middle: simulated price trajectories by solving the local volatility model. cf. Eq. (1); and right: generated call option price using Monte Carlo simulation of Eq. (3) with 10^6 sampled price trajectories. The local volatility and the option price surfaces are visualized on an evenly spaced 10×20 grids and, for a better visualization, only the first 1024 price trajectories are shown in the middle pane.

After $\boldsymbol{\theta}_c$ and $\boldsymbol{\theta}_\eta$ are determined, the parameterized option price and calibrated local volatility are recovered from Eqs. (14) and (16). Thereafter, we substitute the calibrated neural local volatility into Eq. (1), forming a neural local volatility model, whose solution, in turn, gives the synthetic stock prices, from which the option price can then be recovered using Monte Carlo estimates of Eq. (3), completing a control loop for model assessment. The calibrated local volatility, the simulated price trajectory by numerically integrating the neural local volatility model with a pre-sampled sequence of the Brownian motion (B_t), and the relative errors of calibrated local volatility with respect to the exact expression (19) are shown in Figure 4.

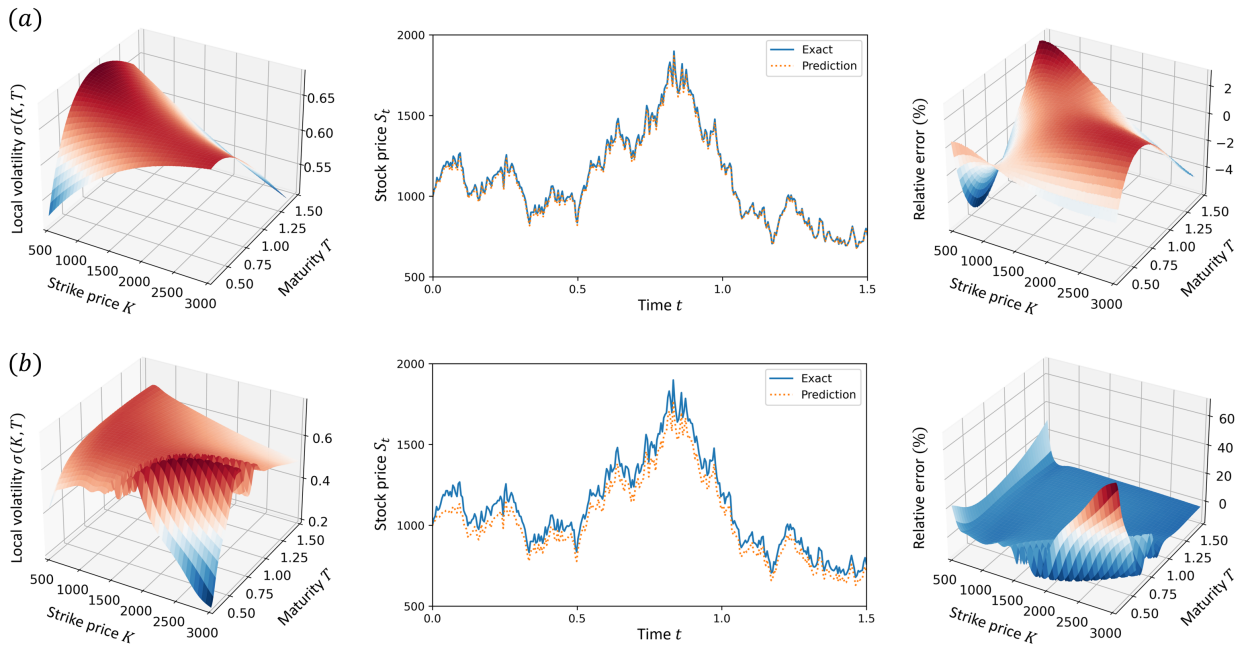

Figure 4: Visualization of calibrated neural local volatility (left), simulated price trajectory with a pre-sampled Brownian motion (B_t) (middle), and the relative error of the calibrated local volatility with respect to the exact expression (19) (a) with ($\lambda_{\text{dup}} = 1$); and (b) without ($\lambda_{\text{dup}} = 0$) the inclusion of L_{dup} as regularizer. The models shown in are trained on a dataset consists of 10×20 option prices, while the visualization and relative errors are computed on a different test grid with linearly spaced 256×256 collocation points.

For $\lambda_{\text{dup}} = 0$, the parameterized option price is not regularized by a priori physics information encoded in Dupire's equation, reducing the self-consistent calibration of local volatility to a one-way approach. In both cases (a)-(b) the parameterized option prices are regularized by the arbitrage-free conditions as in the works by Ackerer et al. (2020); Chataigner et al. (2020, 2021). On the other hand, exploiting self-consistency by the inclusion of L_{dup} as a regularizer improves significantly the calibrated local volatility surface.

Small data set

To explore the small data limit of the proposed method, we present in Figure 5 the calibrated local volatility surface from a scarce dataset consisting of 3×6 option prices quoted on a linearly spaced grid $(K, T) \in [500, 3000] \times [0.3, 1.5]$.

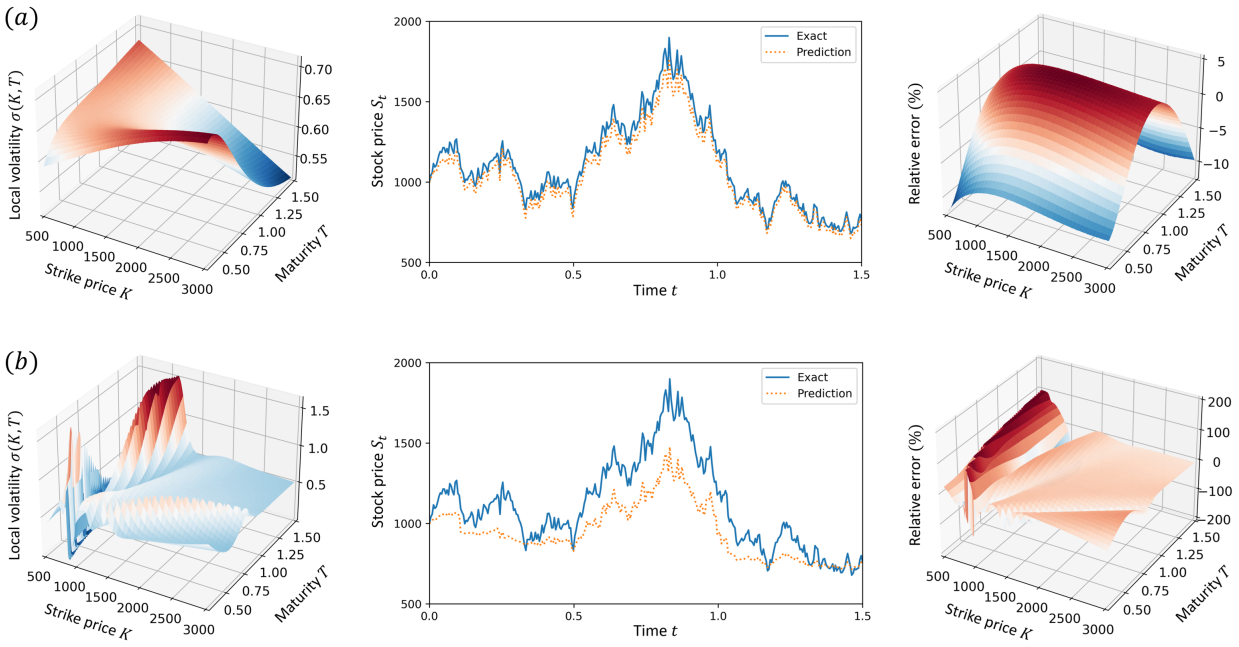

Figure 5: Same as in Figure 4, but with the models trained on a scarce dataset consisting of 3×6 option prices quoted on a linearly spaced grid $(K, T) \in [500, 3000] \times [0.3, 1.5]$.

We can observe that the use of self-consistency effectively compensates for the lack of data, validating the proposed method in the small data limit. Indeed, neural networks are essentially nonlinear functions. Given discrete pairs of inputs and outputs, such a mapping function is not uniquely determined. To ensure that the obtained neural networks can be generalized, we need to go through the entire domain of definition of neural networks, calling for big data. Alternatively, incorporating a priori physics information, completely or partially known, into the learning scheme can regularize neural networks at places where there is no data, allowing one to benefit from recent advances in deep learning without a large data set.

Sensitivity with respect to λ_{dup}

To investigate the sensitivity of the proposed method with respect to λ_{dup}, we consider a sequence of values in the range $\lambda_{\text{dup}} = [0, 4]$. We quantify the accuracy of our self-consistent method by the root mean squared error (RMSE) for the parameterized option price and the calibrated local volatility, which is less relevant due to the non-uniqueness of the solution to such calibration problems. Here, independent of the training dataset, RMSEs are computed on a different testing grid consisting of a linearly spaced 256×256 collocation points in $[K, T] \in [500, 3000] \times [0.3, 1.5]$. We perform three independent runs for each case and the

averaged results are summarized in Table 1.

Dataset	RMSEs	λ_{dup}				
		0.0	0.5	1.0	1.5	2.0
3×6	Option price	10.53	2.27	1.95	1.62	1.76
	Local volatility	0.25	0.03	0.03	0.02	0.03
	Reprice error	64.54	1.84	1.88	1.81	1.71
10×20	Option price	1.07	1.76	0.94	1.10	1.22
	Local volatility	0.04	0.01	0.01	0.01	0.02
	Reprice error	5.99	1.23	1.16	1.08	1.34

Table 1: Test of proposed self-consistent method on synthetic dataset.

We can observe that, when we increase λ_{dup} from zero, the RMSEs for the calibrated volatility and the repriced option price first decrease then increase, which is indicative of the existence of an optimal λ_{dup}. Its value seems to be dependent on the size of dataset, but its determination is beyond the scope of this paper and thus left for future work. The decreased RMSEs for any reasonably chosen $\lambda_{\text{dup}} \neq 0$ support the inclusion of L_{dup} for regularizing the parameterized option price. With randomly initialized $\boldsymbol{\theta}_\eta$, the corresponding Dupire equation essentially provides the wrong a priori for the measured data. During the training, a joint minimization of L_π with respect to $\boldsymbol{\theta}_{\text{c}}$ and of L_{dup} with respect to $\boldsymbol{\theta}_\eta$ yields a self-consistent pair of parameterized option price that provides an optimized description of the measured data, the initial and boundary conditions, the arbitrage-free conditions and the underlying Dupire's equation, as well as for a calibrated local volatility that matches, upon a physics-informed correction, the price dynamics of the observed data. Assigning too small a weight leads to unregularized parameterization, whereas a large λ_{dup} slows down the minimization, resulting in a larger calibration error after a fixed number of iterations.

4.2 Application to market data

In the case of market option prices, since the exact local volatility is not available, the assessment is performed by computing the reprice error. More specifically, we replace $\sigma(x, t)$

in Eq. (1) by the calibrated one

$$\sigma_\theta(x,t) = \sqrt{2\eta_\theta(x,t)/T_{\max}} \tag{20}$$

and generate synthetic asset price paths via numerically integrating the neural local volatility model. The option is then repriced at given strike-maturity pairs using Monte Carlo estimates of Eq. (3) and, eventually, compared with the market call (or put) option prices.

4.2.1 European call options on the DAX index

To perform a comparison with the literature (see, for example Crépey (2002); Chataigner et al. (2020)), we take the daily dataset of DAX index European call options listed on the 7-th, 8-th, and 9-th, August 2001. The spot prices for the underlying assets are $S_0 = 5752.51, 5614.51, 5512.28$, respectively, with 217, 217, and 218 call options quoted at differential strike-maturity pairs and $r = 0.04$. The calibrated local volatility surfaces with and without the inclusion of L_{dup} as a regularizer are shown in Figure 6.

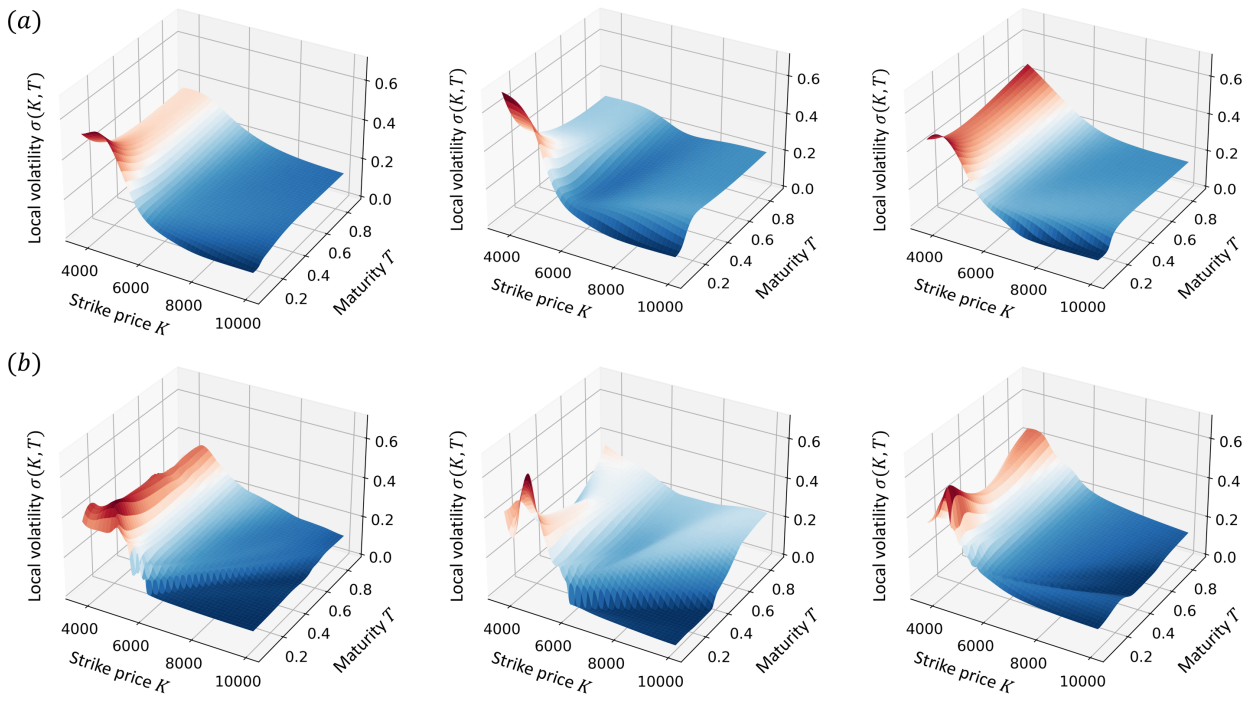

Figure 6: Visualization for the neural local volatility surface calibrated (a) with ($\lambda_{\text{dup}} = 1.0$); and ($b$) without ($\lambda_{\text{dup}} = 0.0$) the inclusion of the residual of Dupire's equation as a regularizer. The obtained results are trained with DAX call option prices listed on 7-th (left), 8-th (middle), and 9-th (right) August 2001.

It is observed that, comparing with the case $\lambda_{\text{dup}} = 0$, the inclusion of L_{dup} provides a greater stability of the local volatility surfaces over successive days. However, to determine whether the variations that we observe in Figure 6 (b) are linked with overfitting, we resort to the quantitative analysis below.

As in Chataigner et al. (2020), we assess the calibration of local volatility by computing the reprice RMSEs and the results are summarized in Table 2.

Reprice RMSE	Ours	Tikhonov Crépey (2002)	Ours	Price-based neural network Chataigner et al. (2020)
	$\lambda_{\text{dup}} = 1$		$\lambda_{\text{dup}} = 0$	
7 August 2001	2.85	5.42	9.44	10.18
8 August 2001	3.27	5.55	6.16	7.44
9 August 2001	2.67	4.60	8.93	8.18

Table 2: Comparison of reprice RMSEs on DAX call options.

Our numerical results are averaged over three independent runs. The unregularized calibration, i.e. $\lambda_{\text{dup}} = 0$, leads to relatively large reprice RMSEs, evidencing that the observed variations of local volatility surface over successive days in Figure 6 are indeed due to overfitting. On the other hand, compared with previous works using Tikhonov regularization Crépey (2002) and with regularizer that impose positiveness and boundedness of the local volatility Chataigner et al. (2020), the inclusion of L_{dup} as a regularizer reduces significantly the reprice RMSEs, leading to the best results. This, therefore, asserts quantitatively the effectiveness of the proposed self-consistent approach.

4.2.2 Put options on the SPX index

In our second application to market data, we use the same dataset, i.e. SPX European put options listed on 18th May 2019, as in Chataigner et al. (2021). The put options are quoted on maturities $T \in [0.055, 2.5]$ and with various strike prices $K \in [1150, 4000]$. The spot price of the underlying is $S_0 = 2859.53$, and $r = 0.023$. Following Chataigner et al. (2021), the dataset is split into a training dataset and a testing dataset consisting of 1720 and 1725 market put option prices, respectively, enabling a comparison. The calibrated local volatility surfaces for the cases $\lambda_{\text{dup}} = 0$ and $\lambda_{\text{dup}} = 1$, as well as their relative difference are shown in Figure 7. It is observed that a numerical singularity developed around $K = 3000$ for the case $\lambda_{\text{dup}} = 0$ is suppressed with the inclusion of L_{dup} as a regularizer.

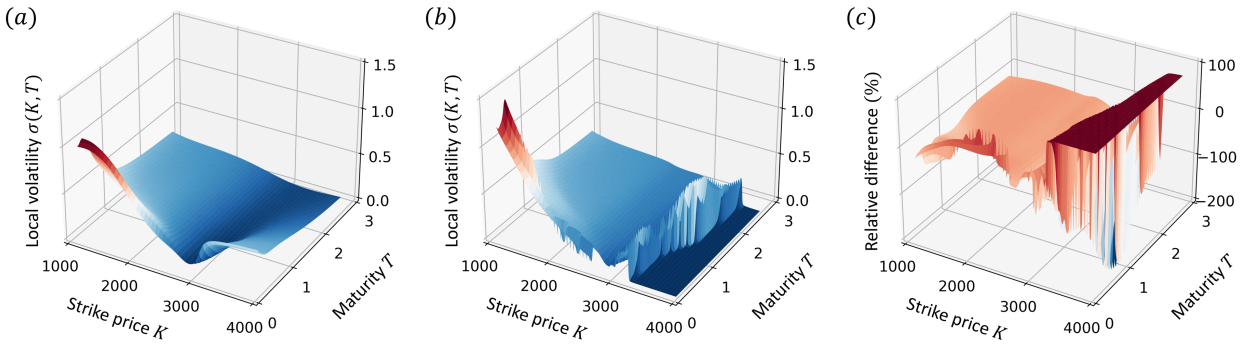

Figure 7: Visualization for the neural local volatility surface calibrated (a) with $(\lambda_{\text{dup}} = 1)$; (b) without $(\lambda_{\text{dup}} = 0)$ the inclusion of the residual of Dupire's equation as a regularizer; and the relative difference of the local volatility surface obtained in (b) computed with respect to that in (a). The obtained results are trained with SPX call option prices listed on 18th May 2019. For better visualization, the relative difference in (c) is truncated at 100%.

Next, on the testing dataset, we evaluate the interpolation and reprice RMSEs of the option prices with and without the inclusion of L_{dup} as a regularizer. We then compare our results with published benchmarks obtained using the surface stochastic volatility inspired (SSVI) model, the Gaussian process (GP), as well as the implied volatility (IV) based and price based neural network methods, see Chataigner et al. (2021) for details. In Table 3, the RMSEs are evaluated on the testing grid and our results shown are averaged over three independent runs.

RMSEs	Ours		SSVI	GP	Neural networks	
	$\lambda_{\text{dup}} = 0$	$\lambda_{\text{dup}} = 1$			IV	Price
Interpolation	0.97	1.92	2.89	0.26	2.97	10.35
Reprice	5.24	3.26	22.83	74.02	4.99	11.76

Table 3: Comparison of interpolation and reprice RMSEs with Chataigner et al. (2021) on SPX put options.

With $\lambda_{\text{dup}} = 1$, our proposed method achieves the lowest reprice RMSEs on the testing dataset, confirming the significance of the proposed self-consistent approach. We note that the Gaussian process, which achieves the lowest interpolation error among all methods, results in the highest reprice RMSEs. This can indicate that the market data of the put option prices could contain a significant amount of noise, which may be linked to the singular behavior observed in Figure 7(b), leading eventually to the high reprice RMSEs in Table 3. Consequently, compared with the case $\lambda_{\text{dup}} = 0$, a simultaneous increase in interpolation

RMSE and decrease in reprice RMSEs in the case $\lambda_{\text{dup}} = 1$ may imply a physics-informed correction has been made to the market data via self-consistency.

5 Discussion and conclusions

In this work, we introduce a deep learning method that yields the parameterized option price and the calibrated local volatility in a self-consistent manner. More specifically, we approximate both the option price and the local volatility using deep neural networks. Self-consistency is established through Dupire's equation in the sense that the parameterized option price from the market data is required to be a solution to the underlying Dupire's equation with the calibrated local volatility. Consequently, by exploiting self-consistency, we are able to not only calibrates the local volatility surface from the market option prices but also to filter out noise in the data that violate Dupire's equation with the calibrated local volatility, going beyond classical inverse problems with regularization.

The proposed method has been tested on both synthetic and market option prices. In all cases, the proposed self-consistent method results in a smooth surface for the calibrated local volatility, with the reprice RMSEs are lower than that obtained either by the canonical methods or the deep learning approaches Crépey (2002); Achdou and Pironneau (2005); Chataigner et al. (2020, 2021). Moreover, the reprice RMSEs are relatively insensitive to the regularization parameter $\lambda_{\text{dup}} > 0$, demonstrating the robustness of our algorithm. Being continuous functions, the neural networks provide full surfaces for the parameterized option price and for the calibrated local volatility, at variance with discrete nodes using the canonical methods. However, incorporating the residual of Dupire's equation as a regularizer requires one to solve a two-dimensional partial differential equation at each iteration, leading to increased computation time. This drawback, however, can be mitigated by distributing the training task on multiple GPUs.

Although option prices may vary dramatically due to the price dynamics of the underlying assets, it is observed from Figure 6 that the variation of the local volatility surface over successive days remains small. Therefore, instead of starting from random initial parameters, initiating the training from converged solutions of the previous day can lead to a substantial reduction in computation time, see Wang and Guet (2022a) for a case study.

Acknowledgment

This research was partially carried within the DesCartes programme supported by the National Research Foundation, Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme. Zhe Wang would like to thank supports from Energy Research Institute@NTU, Nanyang Technological University, where part of the research underlying this paper was performed.

References

Y. Achdou and O. Pironneau. *Computational Methods for Option Pricing*. Society for Industrial and Applied Mathematics, 2005. doi:10.1137/1.9780898717495.

D. Ackerer, N. Tagasovska, and T. Vatter. Deep smoothing of the implied volatility surface. In *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS)*, pages 11552–11563, 2020.

A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind. Automatic differentiation in machine learning: a survey. *Journal of Machine Learning Research*, 18(153):1–43, 2018. URL `https://jmlr.org/papers/volume18/17-468/17-468.pdf`.

C. Bennett. *Trading Volatility, Correlation, Term Structure and Skew*. CreateSpace, 2014.

L. Bergomi. *Stochastic Volatility Modeling*. Financial Mathematics Series. Chapman & Hall/CRC, 2016.

M. Chataigner, S. Crépey, and M. Dixon. Deep local volatility. *Risks*, 8(3):82, 2020. doi:10.3390/risks8030082.

M. Chataigner, A. Cousin, S. Crépey, M. Dixon, and D. Gueye. Beyond surrogate modeling: Learning the local volatility via shape constraints. *SIAM Journal on Financial Mathematics*, 12(3):SC58–SC69, 2021. doi:10.1137/20M1381538.

S. Crépey. Calibration of the local volatility in a trinomial tree using Tikhonov regularization. *Inverse Problems*, 19(1):91, 2002. doi:10.1088/0266-5611/19/1/306.

E. Derman and I. Kani. Riding on a smile. *Risk Magazine*, 7(2):139–145, 1994.

B. Dupire. Pricing with a smile. *Risk Magazine*, 7(1):18–20, 1994.

A. N. Gorban and D. C. Wunsch. The general approximation theorem. In *1998 IEEE International Joint Conference on Neural Networks Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98CH36227)*, volume 2, pages 1271–1274 vol.2, 1998. doi:10.1109/IJCNN.1998.685957.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. doi:10.1109/CVPR.2016.90.

J. C. Hull. *Options, Futures, and Other Derivatives*. Prentice Hall, Upper Saddle River, NJ, fifth edition, 2003.

S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 448–456, 2015.

A. Itkin. *Fitting Local Volatility*. World Scientific, 2020.

G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang. Physics- informed machine learning. *Nature Reviews Physics*, 3:422–440, 2021. doi:10.1038/s42254-021-00314-5.

H. Lin and S. Jegelka. ResNet with one-neuron hidden layers is a universal approximator. In *Advances in Neural Information Processing Systems*, 2018. doi:10.5555/3454287.3454569.

N. Privault. *Introduction to Stochastic Finance with Market Examples (2nd edition)*. Financial Mathematics Series. Chapman & Hall/CRC, 2022.

C. Rackauckas, Y. Ma, J. Martensen, C. Warner, K. Zubov, R. Supekar, D. Skinner, A. Ramadhan, and A. Edelman. Universal differential equations for scientific machine learning. *Preprint arXiv:2001.04385*, 2020. doi:10.48550/arXiv.2001.04385.

M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019. doi:10.1016/j.jcp.2018.10.045.

Z. Wang and C. Guet. Deep learning in physics: A study of dielectric quasi-cubic particles in a uniform electric field. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 6(3):429–438, 2022a. doi:10.1109/TETCI.2021.3086237.

Z. Wang and C. Guet. Self-consistent learning of neural dynamical systems from noisy time series. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 6(5):1103–1112, 2022b. doi:10.1109/TETCI.2022.3146332.

D. A. Winkler and T. C. Le. Performance of deep and shallow neural networks, the universal approximation theorem, activity cliffs, and QSAR. *Molecular Informatics*, 36(1-2):1600118, 2017. doi:10.1002/minf.201600118.