# Deep self-consistent learning of local volatility

Zhe Wang *
Energy research institute @NTU
Nanyang Technological University
637141 Singapore
zhe.wang@ntu.edu.sg

Nicolas Privault
School of Physical and Mathematical Sciences
Nanyang Technological University
637371 Singapore
nprivault@ntu.edu.sg

Claude Guet
Energy research institute @NTU
Nanyang Technological University
637141 Singapore
and School of Physical and Mathematical Sciences
Nanyang Technological University
637371 Singapore
cguet@ntu.edu.sg

September 28, 2024

**Abstract**

We present an algorithm for the calibration of local volatility from market option prices through deep self-consistent learning, by approximating 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 bring corrections to the parameterization in a self-consistent way. By exploiting the differentiability of the neural networks, we can evaluate Dupire's equation locally at each maturity-strike pair; while by exploiting their continuity, we sample maturity-strike pairs uniformly from a given domain, going beyond the discrete points where the options are quoted. For comparison with existing approaches, the proposed method is tested on both synthetic and market option prices, which shows an improved performance in terms of repricing error, no violation of the no-arbitrage constraints, and smoothness of the calibrated local volatility.

*Keywords*: Deep self-consistent learning, local volatility, Dupire partial differential equation, Neural networks.

*Now at CNRS@CREATE LTD, 1 Create Way, #08-01 CREATE Tower, Singapore 138602.

# 1 Introduction

Consider an asset price process $(S_t)_{t\in\mathbb{R}_+}$ expressed in a local volatility model as

$$\frac{dS_t}{S_t} = rdt + \sigma(t, S_t)dB_t, \tag{1}$$

where $(B_t)_{t\in\mathbb{R}_+}$ is a standard Brownian motion, $r$ denotes constant risk-free interest rate, and $\sigma(t,x)$ is a deterministic function of time $t$ and of the underlying asset price $x$, satisfying the usual Lipschitz conditions.

The practical implementation of such a stochastic volatility model in option pricing requires to solve the challenging problem of calibrating the local volatility function $\sigma(t,x)$ to the market data of option prices. Given $(p(T,x,K))_{T,K>0}$ a family of call option prices with maturities $T$, underlying asset price $S_0 = x$ and strike prices $K$ given at time 0, the Dupire (1994) formula brings a solution to this problem by the construction of an estimator of $\sigma(t,x)$ as a function $\sigma(t,K)$ of strike price $K$ values, satisfying the following partial differential equation (PDE)

$$-\frac{\partial p}{\partial T}(T,K) = rK\frac{\partial p}{\partial K}(T,K) - \frac{1}{2}K^2\sigma^2(T,K)\frac{\partial^2 p}{\partial K^2}(T,K), \qquad K > 0, \tag{2}$$

see Dupire (1994), Derman and Kani (1994), which implies

$$\sigma(t,K) := \sqrt{\frac{2\dfrac{\partial p}{\partial t}(t,K) + 2rK\dfrac{\partial p}{\partial K}(t,K)}{K^2\dfrac{\partial^2 p}{\partial K^2}(t,y)}}, \tag{3}$$

where for simplicity we also assumed a zero dividend yield. In this setting, European option prices of the form

$$g(t,x,K) := e^{-(T-t)r}\mathbb{E}[(S_T - K)^+ \mid S_0 = x], \qquad 0 \le t \le T,$$

where $x^+ := \max(x,0)$, generated by the Black-Scholes type PDE

$$rg(t,x,K) = \frac{\partial g}{\partial t}(t,x,K) + rx\frac{\partial g}{\partial x}(t,x,K) + \frac{1}{2}x^2\sigma^2(t,x)\frac{\partial^2 g}{\partial x^2}(t,x,K), \tag{4}$$

with terminal condition $g(T,x,K) = (x-K)^+$, become compatible with the market option prices $p(T,x,K)$ in the sense that

$$p(T,x,K) = g(0,x,K) = e^{-rT}\mathbb{E}[(S_T - K)^+ \mid S_0 = x], \qquad K,T > 0. \tag{5}$$

The numerical estimation of local volatility $\sigma(t, y)$ using Eq. (3), involves evaluating partial derivatives, which is classically achieved using the finite difference method. More efficient methods have been introduced using spline functions, see e.g. Chapter 8 of Achdou and Pironneau (2005), where local volatility is approximated as the sum a piece-wise affine function satisfying a boundary condition and a bi-cubic spline function. In this setting, the model parameters are determined by minimizing the discretized Eq. (2) at selected collocation points with Tikhonov regularization. Tikhonov regularization has been applied to the calibration of local volatility in a trinomial model in Crépey (2002).

Recently, Chataigner et al. (2020) advocated parameterizing market option prices $p(T, K)$ with a deep neural network. As neural networks are continuous and differentiable functions, the derivatives in Eq. (3) are computed using automatic differentiation (Baydin et al. 2017). For Dupire's formula (3) to be meaningful, one must ensure that the argument in the square root is positive and remains bounded, leading additional constraints when fitting the option prices in Chataigner et al. (2020).

In all previous studies, a first approximation of derivatives of the option price is substituted into Eq. (3) to calibrate the local volatility. Hence, the quality of the reconstructed local volatility surface depends on the resolution and features of the observed market option prices at given maturity-strike pairs. Moreover, in Chataigner et al. (2020; 2021) the no-arbitrage conditions are imposed at discrete maturities strikes pairs where the option prices are quoted, hence do not ensure a zero violation throughout the domain of interest.

In this work, we adopt the idea of Wang and Guet (2021a) and calibrate the local volatility surface from the observed market option prices in a self-consistent manner. For this, we rely on recent studies on deep self-consistent learning which have shown that the inclusion of an unknown, underlying governing equation as a regularizer can not only filter out stochastic outliers in the data, but also mitigates overfitting.

We parameterize the observed market option prices by a continuous function $p(T, K)$ of $(T, K)$. By requiring that $p(T, K)$ be a solution to the Dupire equation (2), we regularize the parameterization $p(T, K)$ and determine the unknown $\sigma(T, K)$ self-consistently. By approximating both the market option prices and the squared local volatility using deep neural networks, the resulting $p(T, K)$ and $\sigma(T, K)$ are continuous and differentiable functions of $T$ and $K$. This allows us to evaluate Dupire's equation locally on a uniformly sampled $(T, K)$ from the support manifold $\Omega$ of Eq. (2). Unlike previous approaches, where constraints were imposed on fixed maturity-strike pairs, a successive re-sampling enables us to regularize the entire support $\Omega$. In addition, the positiveness of $\sigma(T, K)$ is ensured by a properly selected output activation function of the neural network, mitigating the risk of violating the no-arbitrage constraints by construction.

After rescaling and reparameterization of market option prices in Sections 2.1-2.2 and inclusion of a neural Dupire equation as a regularizer in Section 2.3, our deep self-consistent algorithm is given and discussed in Section 2.4. The proposed method is tested on synthetic datasets in Section 3.1, and applied to market option prices in Section 3.2, where a comparison with other methods (Chataigner et al. 2020, Crépey 2002) is performed. We check in Table 2 that the inclusion of Dupire's equation as a regularizer reduces significantly the repricing Root Mean Squared Error (RMSE) in comparison with other works using Tikhonov regularization (Crépey 2002) or regularization that impose positiveness and boundedness on local volatility (Chataigner et al. 2020). Moreover the repricing RMSEs are relatively insensitive to the regularization parameter $\lambda$, showing the robustness of our algorithm. As an ablation study, we compare our results with those obtained without including Dupire's equation as a regularizer. Finally, the advantages and limitations of our method are discussed in Section 4, where conclusions are drawn.

# 2 Problem formulation

## 2.1 Rescaling of Dupire's equation

In this section we apply a change of variable and rescaling to the Dupire equation, in order to ensure that the variables in Eq. (10) below are of order of unity, so that no variable is dominating over the others during the minimization. For this, we first introduce the change of variables

$$K = e^{rT} k, \quad \text{and} \quad \eta = \sigma^2, \tag{6}$$

under which we rewrite the Dupire PDE (2) as

$$\frac{\partial p}{\partial t}(t, k) = \frac{\eta}{2} k^2 \frac{\partial^2 p}{\partial k^2}(t, k). \tag{7}$$

In addition, the above equation can be made dimensionless by rescaling the time and space variables using their boundary, as

$$\tilde{t} = \frac{t}{t_{\max}}, \quad \tilde{k} = \frac{k - k_{\min}}{k_{\max} - k_{\min}}, \tag{8}$$

so that $(\tilde{t}, \tilde{k}) \in \Omega := [0,1]^2$. Together with the scaling

$$\tilde{p} = \frac{p}{S_0}, \quad \tilde{\eta} = \frac{t_{\max}\eta}{2}, \tag{9}$$

the dimensionless counterpart of Eq. (7) can be formulated as the PDE

$$\frac{\partial \tilde{p}}{\partial \tilde{t}} = \tilde{\eta}\left(\frac{k_{\min}}{k_{\max} - k_{\min}} + \tilde{k}\right)^2 \frac{\partial^2 \tilde{p}}{\partial \tilde{k}^2} + e(\tilde{t}, \tilde{k}), \tag{10}$$

subject to the initial condition

$$\tilde{p}'(0, \tilde{k}) = \left(1 - \frac{e^{r t_{\max} \tilde{t}}}{S_0}\big(k_{\min} + (k_{\max} - k_{\min})\tilde{k}\big)\right)^+ \tag{11}$$

and to the boundary condition $\tilde{p}'(\tilde{t}, 0) = 1$, where $e(\tilde{t}, \tilde{k})$ denotes an error. Note that the boundary condition is implied by Eqs. (10) and (11), hence need not be imposed, cf. Achdou and Pironneau (2005) for details.

## 2.2 Parameterization of market option prices

Market option prices are observed at discrete maturity-strike pairs, whereas local volatility is a continuous function. To obtain a continuous limit, we parameterize the scaled market option prices $\tilde{\boldsymbol{p}}' = (\tilde{p}'_1, \ldots, \tilde{p}'_N)$ observed at $N$ maturity-strike pairs $\big((\tilde{t}'_1, \tilde{k}'_1), \ldots, (\tilde{t}'_N, \tilde{k}'_N)\big)$ by a neural network

$$\tilde{p}(\tilde{t}, \tilde{k}) = \mathcal{N}_{\tilde{p}}(\tilde{t}, \tilde{k}; \boldsymbol{\theta}_p), \quad \mathcal{N}_{\tilde{p}} : \mathbb{R}_+ \times \mathbb{R}_+ \to \mathbb{R}_+. \tag{12}$$

Conventionally, the model parameters $\boldsymbol{\theta}_{\tilde{p}}$ of the neural network are determined by minimizing the deviation

$$L_{\text{fit}} := \frac{1}{N}\sum_{i=1}^{N}\big(\tilde{p}(\tilde{t}'_i, \tilde{k}'_i) - \tilde{p}'_i\big)^2 w\left(\frac{\text{mean}(\tilde{p}')}{\tilde{p}'_i}\right) + \frac{1}{M_1}\sum_{i=1}^{M_1}\big(\tilde{p}(0, \tilde{k}_i) - \tilde{p}'(0, \tilde{k}_i)\big)^2 \tag{13}$$

to the market option price and to the initial condition (11), where $\text{mean}(\tilde{p}') := \frac{1}{N}\sum_{i=1}^{N}\tilde{p}'_i$. A weight function

$$w(x) := 0.1 \times \mathbf{1}_{[0,0.1)}(x) + x\mathbf{1}_{[0.1,10]}(x) + 10 \times \mathbf{1}_{[10,\infty)}(x), \qquad x > 0, \tag{14}$$

with

$$\mathbf{1}_A(x) := \begin{cases} 1, & \text{if} \quad x \in A, \\ 0, & \text{if} \quad x \notin A. \end{cases}$$

is so selected that a larger weight is assigned to small option prices. To account for the initial condition, we generate $M_1$ reference option prices by substituting uniformly sampled $[\tilde{k}_1, \ldots, \tilde{k}_{M_1}] \in [0,1]$ into Eq. (11) at each iteration.

For a small dataset, directly minimizing $L_{\text{fit}}$ alone would lead to an overfitting in the sense that the obtained $\tilde{p}(\tilde{t}, \tilde{k})$ does not necessarily comply with Dupire's equation. Consequently, the argument of the square root in Eq. (3) can be negative, leading to numerical instability. In this work, we propose to get around this difficulty by including the scaled Dupire equation (10), with the local volatility unknown and approximated by another neural network, as a regularizer. Then, a joint minimization of

$$L = L_{\text{fit}} + \lambda L_{\text{pde}}, \tag{15}$$

where $L_{\text{pde}}$ denotes a loss function associated with the Dupire equation to be discussed in Section 2.3 and $\lambda$ a hyper-parameter, not only regularizes the parameterization for the market option prices but also determines the unknown local volatility, entailing self-consistency.

## 2.3 Neural Dupire equation

Instead of calibrating the local volatility using Dupire's formula, we propose to approximate the squared local volatility using a neural network

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

where $\boldsymbol{\theta}_{\tilde{\eta}}$ are the model parameters to be determined. Since both $\mathcal{N}_{\tilde{p}}(\tilde{t}, \tilde{k}; \boldsymbol{\theta}_{\tilde{p}})$ and $\mathcal{N}_{\tilde{\eta}}(\tilde{t}, \tilde{k}; \boldsymbol{\theta}_{\tilde{\eta}})$ are continuous and differentiable functions of $(\tilde{t}, \tilde{k})$, one can evaluate the deviation from the scaled Dupire equation (10), denoted by $e(\tilde{t}, \tilde{k})$, on a uniformly sampled set of $M_2$ collocation points $(\tilde{t}, \tilde{k})$ from $\Omega : [0,1]^2$, going beyond the $N$ measured maturity-strike pairs. Thus, the loss function associated with the scaled Dupire PDE reads

$$L_{\text{pde}} := \frac{1}{M_2} \sum_{i=1}^{M_2} \left( e(\tilde{t}_i, \tilde{k}_i) \right)^2 w \left( \frac{\text{mean}\left(|\partial_{\tilde{t}} \tilde{p}|\right)}{|\partial_{\tilde{t}} \tilde{p}(\tilde{t}_i, \tilde{k}_i|)} \right), \tag{17}$$

where we let $\text{mean}\left(|\partial_{\tilde{t}} \tilde{p}|\right) := \frac{1}{N} \sum_{i=1}^{N} |\partial_{\tilde{t}} \tilde{p}(\tilde{t}_i, \tilde{k}_i)|$, and the weight function is introduced to compensate terms with vanishing amplitudes. A successive re-sampling covers the entire support manifold $\Omega$ of Eq. (10), ensuring that Dupire's equation is everywhere satisfied. This not only avoids the need of designing an adaptive mesh for Dupire's equation as in Achdou and Pironneau (2005), but also mitigates any overfitting associated with a small dataset. It is important to emphasize that, during the minimization, the weight is treated as a scalar quantity, not a function of $\boldsymbol{\theta}_{\widetilde{p}}$. A minimization of $L_{\text{pde}}$ with respect to $\boldsymbol{\theta}_{\widetilde{\eta}}$ enables a non-parameterized calibration of local volatility.

## 2.4 Algorithm

Based on the self-consistent deep learning computational scheme discussed above we build an algorithm for option prices and local volatility which is outlined in Algorithm 1.

---

**Algorithm 1** Self-consistent learning for market option prices and local volatility.

---

**Input:** Mini-batched market option prices $\{\tilde{\boldsymbol{p}}'[1], \tilde{\boldsymbol{p}}'[2], \ldots, \tilde{\boldsymbol{p}}'[S]\}$ and the corresponding scaled maturity-strike pairs.
Guess initial parameters $\{\boldsymbol{\theta}_{\widetilde{p}}[1], \boldsymbol{\theta}_{\widetilde{p}}[2], \ldots, \boldsymbol{\theta}_{\widetilde{p}}[S]\}$ and $\boldsymbol{\theta}_{\widetilde{\eta}}$.

- **while** not converged **do**
  - **for** $s = 1, \ldots, S$ **do**
    - Uniformly draw $M_1$ points $\tilde{k} \in [0, 1]$ and $M_2$ scaled maturity-strike pairs $(\tilde{t}, \tilde{k}) \in \Omega : [0, 1]^2$.
    - Compute 2 loss functions: $L_{\text{fit}}$ and $L_{\text{pde}}$.
    - Optimize $\boldsymbol{\theta}_{\widetilde{p}}[s]$ using gradients $\partial_{\boldsymbol{\theta}_{\widetilde{p}}[s]}(L_{\text{fit}} + \lambda L_{\text{pde}})$.
    - Optimize $\boldsymbol{\theta}_{\widetilde{\eta}}$ using gradients $\partial_{\boldsymbol{\theta}_{\widetilde{\eta}}} L_{\text{pde}}$.
  - **end for**
- **end while**

**Output**: Optimized parameters $\{\boldsymbol{\theta}_{\widetilde{p}}[1], \boldsymbol{\theta}_{\widetilde{p}}[2], \ldots, \boldsymbol{\theta}_{\widetilde{p}}[S]\}$ and $\boldsymbol{\theta}_{\widetilde{\eta}}$.
Hyper-parameters used in this paper: $S = 1$, $M_1 = 100$, and $M_2 = 10,000$.

---

At variance with usual approaches where the derivatives of the option price are first approximated to calibrate the local volatility, using Dupire's formula, we approximate both the option price and the local volatility using neural networks. The training of neural networks is a non-convex optimization problem which does not guarantee convergence towards the global minimum, hence regularization is required. The differentiability of neural networks allows us to include the scaled Dupire equation as a regularizer. A joint minimization of $L_{\text{fit}}$ arising from the deviation of the parameterized option price to the market data and $L_{\text{pde}}$ associated with the scaled Dupire equation seeks for a self-consistent pair of approximations for the option price and for the underlying local volatility, excluding local minima which violate either of the constraints.

Moreover, in practice, the maturity-strike pairs with the quoted option prices are usually unevenly distributed (Chataigner et al. 2021). A direct minimization of loss functions evaluated on the measured maturity-strike pairs leads to large calibration errors of local volatility at locations where the measurement is scarce. Being continuous functions, neural networks enable a mesh-free discretization of the Dupire equation by uniformly sampling collocation points from $\Omega$ at each iteration, improving the calibration from scarce and unevenly distributed dataset. Lastly, since both option price and squared local volatility are approximated using neural networks, their positiveness are ensured by properly selected output activation functions. Therefore, there is no violation of the non-arbitrage constraints by construction nor numerical instabilities associated with calibrating local volatility using the Dupire formula occur.

For large datasets, algorithm 1 allows us to divide the observed market option prices and the corresponding maturity-strike pairs into $S$ batches. Each batch is parameterized separately, leading to $S$ parameterization functions $\{\tilde{p}(\tilde{t},\tilde{k})[1],\ldots,\tilde{p}(\tilde{t},\tilde{k})[S]\}$ to be determined. Since those $S$ parameterization networks for option prices are learned jointly with one calibration network for the squared local volatility, the computation time per epoch scales almost linearly with $S$. The scalability of our method allows for accelerating the calibration process by distributing the training on multiple GPUs.

# 3 Case studies

In this section, we test our self-consistent method first on synthetic option prices in Section 3.1, and second on real market data in Section 3.2. Synthetic option prices are generated as Monte Carlo estimates of

$$p'(T_i',K_i') = e^{-rT_i'}\frac{1}{M_s}\sum_{j=1}^{M_s}(S_t(T_i',x_j)-K_i')^+, \tag{18}$$

at given $(T_i',K_i')$. Here, the asset price paths are obtained by simulating a local volatility model (1), with $\sigma(t,x)$ given. The validity of local volatility calibration can then be assessed by comparison to closed-form option pricing formulas.

In the case of market option prices, since the exact local volatility is not available, the assessment is performed by computing the repricing error. More specifically, we replace $\sigma$ in Eq. (1) by the calibrated one and generate synthetic asset price paths via numerically integrating the neural local volatility model. The option is then repriced at quoted maturity-strike pairs using Eq. (18) and compared with the market option prices.

Throughout the paper, we use the same neural network architecture as in Wang and Guet (2021a) for both $\mathcal{N}_{\widetilde{p}}$ and $\mathcal{N}_{\widetilde{\eta}}$. The network consists of 5 residual blocks (He et al. 2016) with the residual connection being employed around two sub-layers made of 64 neurons and regularized with batch normalization (Ioffe and Szegedy 2015). Since both the option price and the local volatility are non-negative, we use the softplus function, $\text{softplus}(x) := \log(1+e^x)$, for the output layers of each neural networks, thereby mitigating the risk of static arbitrage by construction. Since the inclusion of Dupire's equation as a regularizer requires evaluating successive derivatives, the tanh activation function is selected for hidden network layers.

Moreover, we corrupt the input of the neural networks with Gaussian noise $N(0, 0.5^2)$ as in Vincent et al. (2008; 2010), which corresponds to performing local average of the learned model over $(\tilde{t}, \tilde{k})$. We start the training with an initial learning rate $10^{-3}$, which is divided by a decaying factor 1.1 every 2,000 iterations. The total number of iterations is capped at 50,000; on a laptop equipped with an Nvidia Quadro RTX 3000 GPU card, each iteration takes around 0.05 second for a dataset consisting of $100 \times 100$ quoted option prices.

## 3.1 Local volatility model

Without loss of generality, we take a constant $r = 0.02$ and

$$\sigma(t, x) = 0.2 + ye^{-y} \quad \text{with} \quad y = (t + 0.1)\sqrt{x + 0.1}. \tag{19}$$

The price trajectories are obtained by simulating Eq. (1) for $M_s = 10^6$ times from an initial condition $S_0 = 1$ for the period $t = [0, 1.5]$ with a time step $\mathrm{d}t = 0.01$. We consider a mesh grid consisting of evenly spaced 100 points for the $[0.5, 1.5]$ $T$-period and 100 points in the $[0.5, 2.5]$ $K$-interval, respectively. That is, the dataset consists of $100 \times 100$ option prices quoted at the corresponding maturity-strike pairs. The simulated price trajectories, the synthetic option price, and the exact local volatility are visualized in Figure 1.

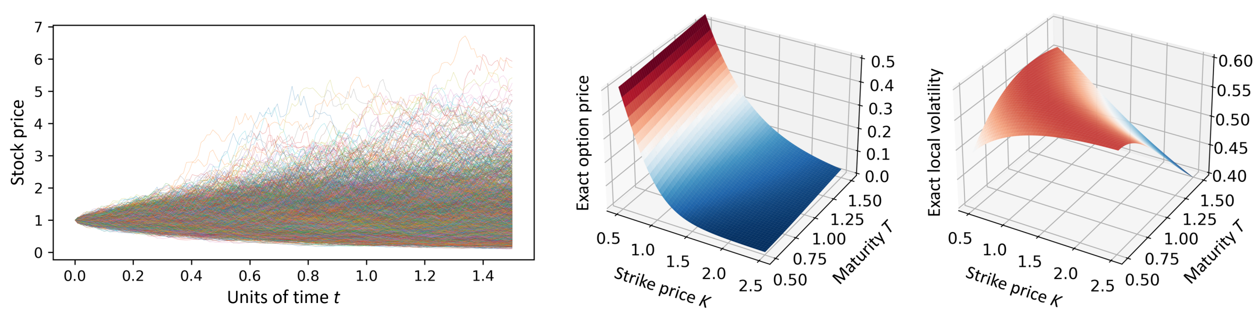


Figure 1: From left: simulated price trajectories by solving the local volatility model. cf. Eq. (1); middle: generated option price using Monte Carlo approximation, cf. Eq. (18); and right: exact local volatility, cf. Eq. (19).

The synthetic option price $p'(T_i', K_i')$ and the corresponding strike-maturity pairs $(T_i', K_i')$ are rescaled for training. As soon as $\boldsymbol{\theta}_{\widetilde{p}}$ and $\boldsymbol{\theta}_{\widetilde{\eta}}$ are determined, the parameterized option price and calibrated the local volatility are recovered by calling Eqs. (9). Moreover, 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 Eq. (18), completing a control loop for model assessment. With $\lambda = 1.0$, the recovered option price from the Monte Carlo simulation of the neural local volatility model, the calibrated local volatility, and their corresponding relative errors

$$\mathrm{error}_i = 1 - \frac{\mathrm{computed}_i}{\mathrm{exact}_i}, \tag{20}$$

are shown in Figure 2. Note that, for the calibrated and recovered option prices, the peak of relative errors is due to the normalization by exact option prices with vanishing amplitude at large strike price and small maturities. Therefore, in Figure 2 and subsequent figures where the relative pricing errors are plotted, we reverse both axes of strike price and of maturity for better visualization.

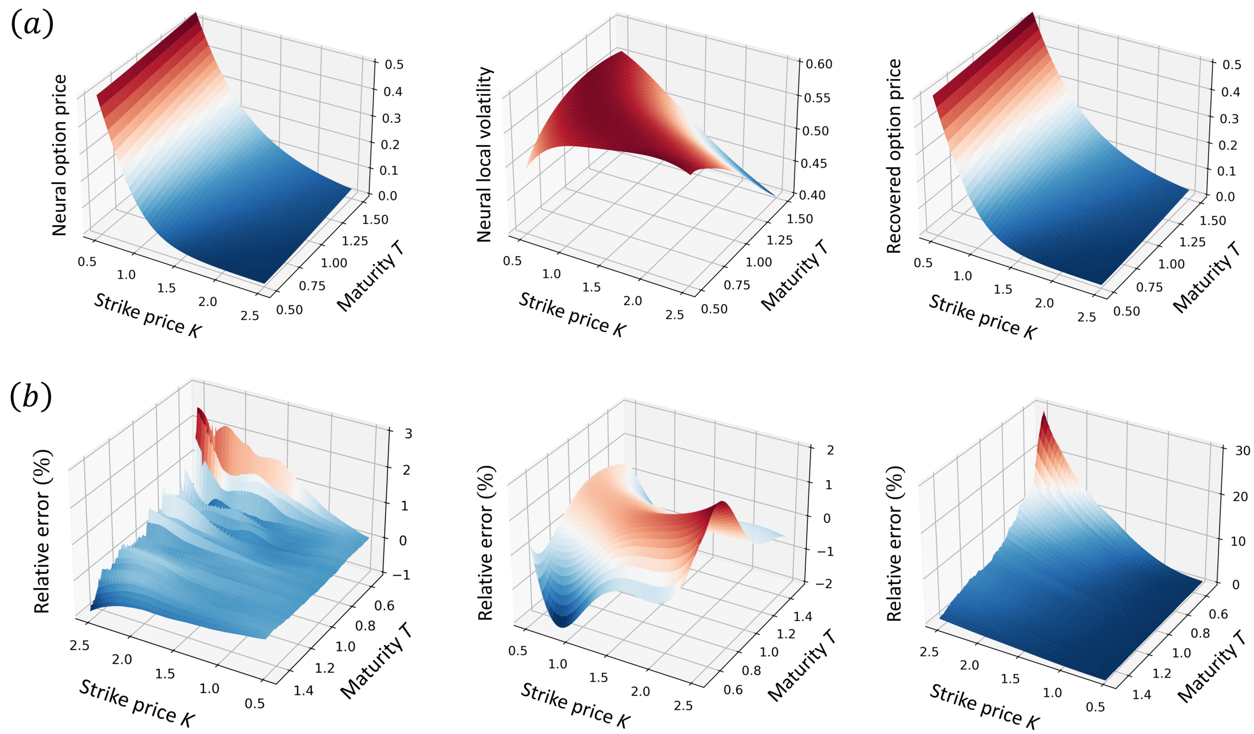


Figure 2: ($a$) Neural option price (left), the calibrated neural local volatility (middle), and the recovered option price from the neural local volatility model (right); and ($b$) their corresponding relative error, cf. Eq. (20). The models are trained on a dataset consists of $100 \times 100$ option prices and with the weight $\lambda = 1.0$. Note that, in this and following figures, we reverse both axes for the strike price and for the maturity in plotting the pricing errors.

To investigate the sensitivity of the proposed method with respect to the value of $\lambda$ and to the size of dataset, we consider in the following two additional dataset that consists of $100 \times 10$ and $100 \times 1000$ options prices evaluated at evenly space maturity-strike pairs and a sequence of values in the range $\lambda = [0, 4]$. For $\lambda = 0$, the parameterized option price is not regularized by the Dupire equation, reducing the self-consistent calibration of the local volatility to a one-way approach. Nevertheless, since the constraint $\sigma \geq 0$ is ensured by the softplus activation function of $\mathcal{N}_{\widetilde{\eta}}$, the numerical instability associated with the Dupire's formula (3) is absent. We assess the accuracy of our self-consistent method by the Root Mean Squared Error (RMSE)

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (\text{exact}_i - \text{computed}_i)^2}, \tag{21}$$

for the calibrated local volatility and for the recovered option price from the neural local volatility model, respectively. The results are summarized in Table 1. It is observed that, with fixed size of dataset and increasing $\lambda$ from 0, the relative error for the calibrated volatility first decreases then increases, indicative the existence of an optimal $\lambda$. The value of optimal $\lambda$ seems to be dependent on the dataset and its determination is out of the scope of this paper, hence left for future work. The decreased RMSEs for all $\lambda \neq 0$ support the inclusion of the Dupire's equation for regularizing the parameterized option price. With randomly initialized $\boldsymbol{\theta}_{\tilde{\eta}}$, the corresponding Dupire equation forms essentially a wrong a priori description for the measured data. During the training, a joint minimization of $L_{\text{fit}}$ and $L_{\text{pde}}$ yields a self-consistent pair of solutions where the parameterized option price provide an optimized description for both the measured data and the underlying Dupire equation with the calibrated local volatility. Assigning a too small weight leads to unregularized parameterization, whereas a large $\lambda$ slows down the minimization, resulting in an increased calibration error with fixed number of iterations.

| Size of data | RMSEs ($\times 10^{-2}$) | $\lambda$ | | | | | |
|---|---|---|---|---|---|---|---|
| | | 0.0 | 0.25 | 0.5 | 1.0 | 2.0 | 4.0 |
| $100 \times 10$ | Calibrated volatility | 18.21 | 0.81 | 0.56 | 0.68 | 0.87 | 1.13 |
| | Recovered option price | 0.89 | 0.12 | 0.08 | 0.10 | 0.07 | 0.11 |
| $100 \times 100$ | Calibrated volatility | 2.08 | 0.52 | 0.54 | 0.46 | 0.82 | 1.18 |
| | Recovered option price | 0.17 | 0.09 | 0.13 | 0.11 | 0.12 | 0.15 |
| $100 \times 1000$ | Calibrated volatility | 1.64 | 0.57 | 0.47 | 0.51 | 0.44 | 1.08 |
| | Recovered option price | 0.12 | 0.11 | 0.10 | 0.11 | 0.13 | 0.14 |

Table 1: Root mean squared errors (RMSEs).

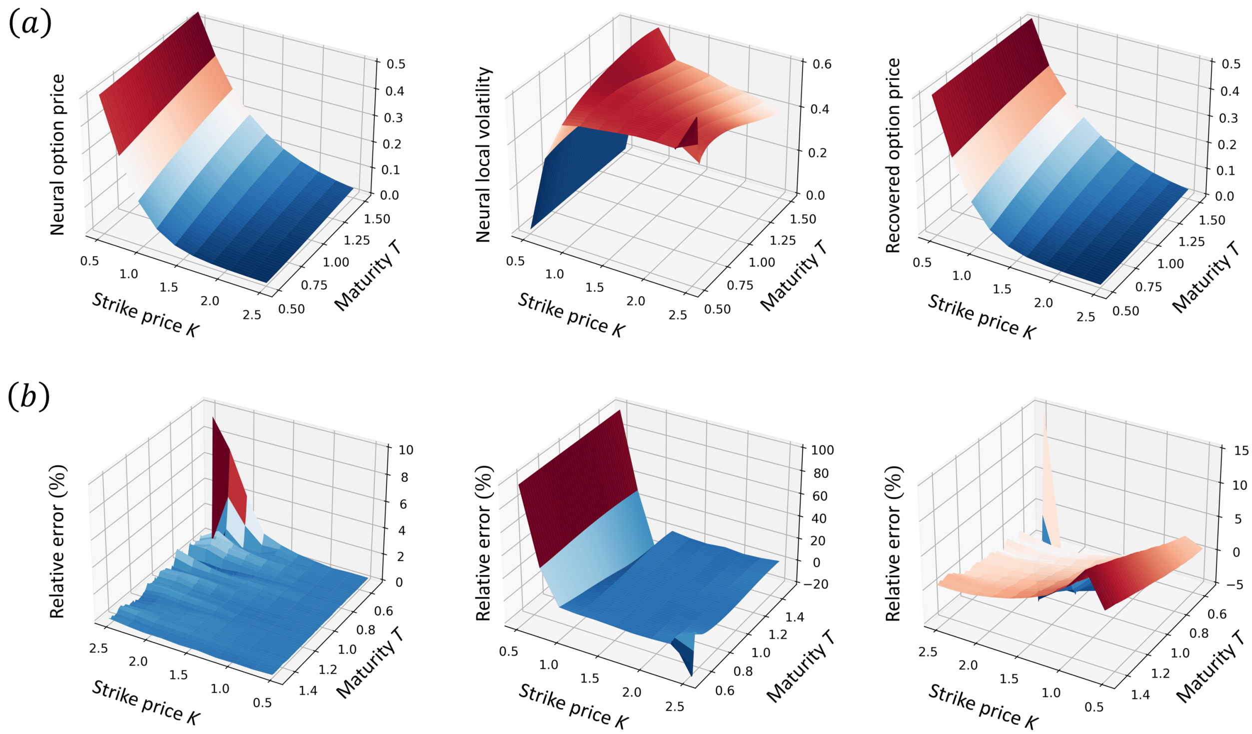


Figure 3: Same as Figure 2, except that the model is trained with option prices defined on a $100 \times 10$ with $\lambda = 0$.

Moreover, in the absence of regularization, i.e. when $\lambda = 0$, the calibration error depends sensitively on the size of the dataset. In principle, one can improve the quality of the volatility calibration by increasing the size of dataset. This, however, is hardly feasible in practice. Alternatively, for any reasonably chosen $\lambda \neq 0$, the inclusion of the Dupire regularization mitigates overfitting and leverages the calibrated volatility from scarce data substantially, see Figure 3 and 4 for a visual evidence. Although in both cases, i.e. $\lambda = 0$ and $\lambda = 0.25$, the relative error associated with the option price parameterization remains small, the qualitative difference observed in surfaces of the calibrated local volatility and of the recovered option price implies that the neural network overfits to the small dataset for $\lambda = 0$.

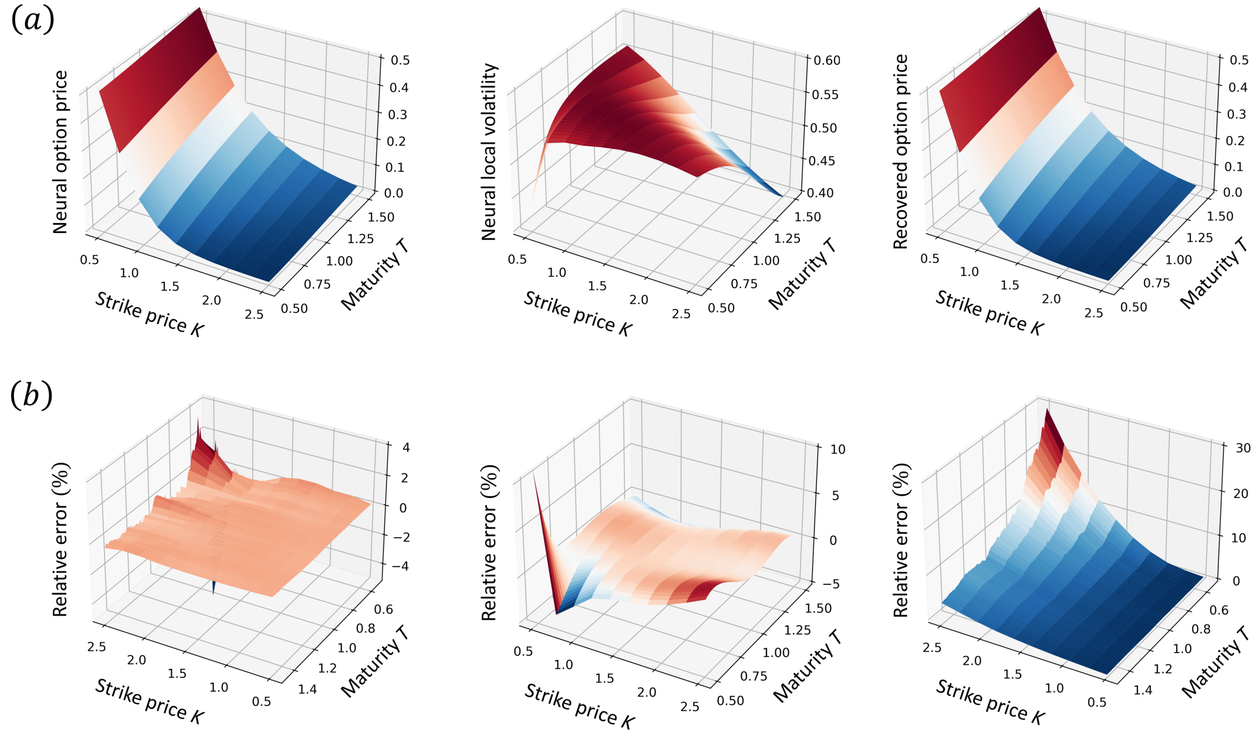


Figure 4: Same as Figure 3, except for $\lambda = 0.25$.

## 3.2 Application to market data

### DAX index European vanilla call options

To perform a comparison with previous works (Chataigner et al. 2020, Crépey 2002), we take the daily dataset of DAX index European vanilla options listed on the 7-th, 8-th, and 9-th, August 2001. The corresponding underlying assets are $S_0 = 5752.51, 5614.51, 5512.28$, respectively, with 217, 217, and 218 call options quoted at differential maturity-strike pairs. For simplicity, we take $r = 0.04$ constant. The calibrated local volatility, the recovered option price from the local volatility model (1), and the relative error of the recovered option price to the parameterized one are shown in Figures 5-7. Consistent with previous benchmarking on synthetic data, the inclusion of Dupire's equation as a regularizer smooths out the calibrated local volatility substantially. Comparing Figures 6 and 7 reveals that the calibrated local volatility is largely independent of the selection $\lambda$ except for regions with large strike prices and small maturities. In those regions, vanishing option prices render the loss associated with Dupire's equation, i.e. $L_{\text{pde}}$, negligible. Hence, the calibrated local volatility is essentially unregularized at large strikes and small maturities, leading to variations.

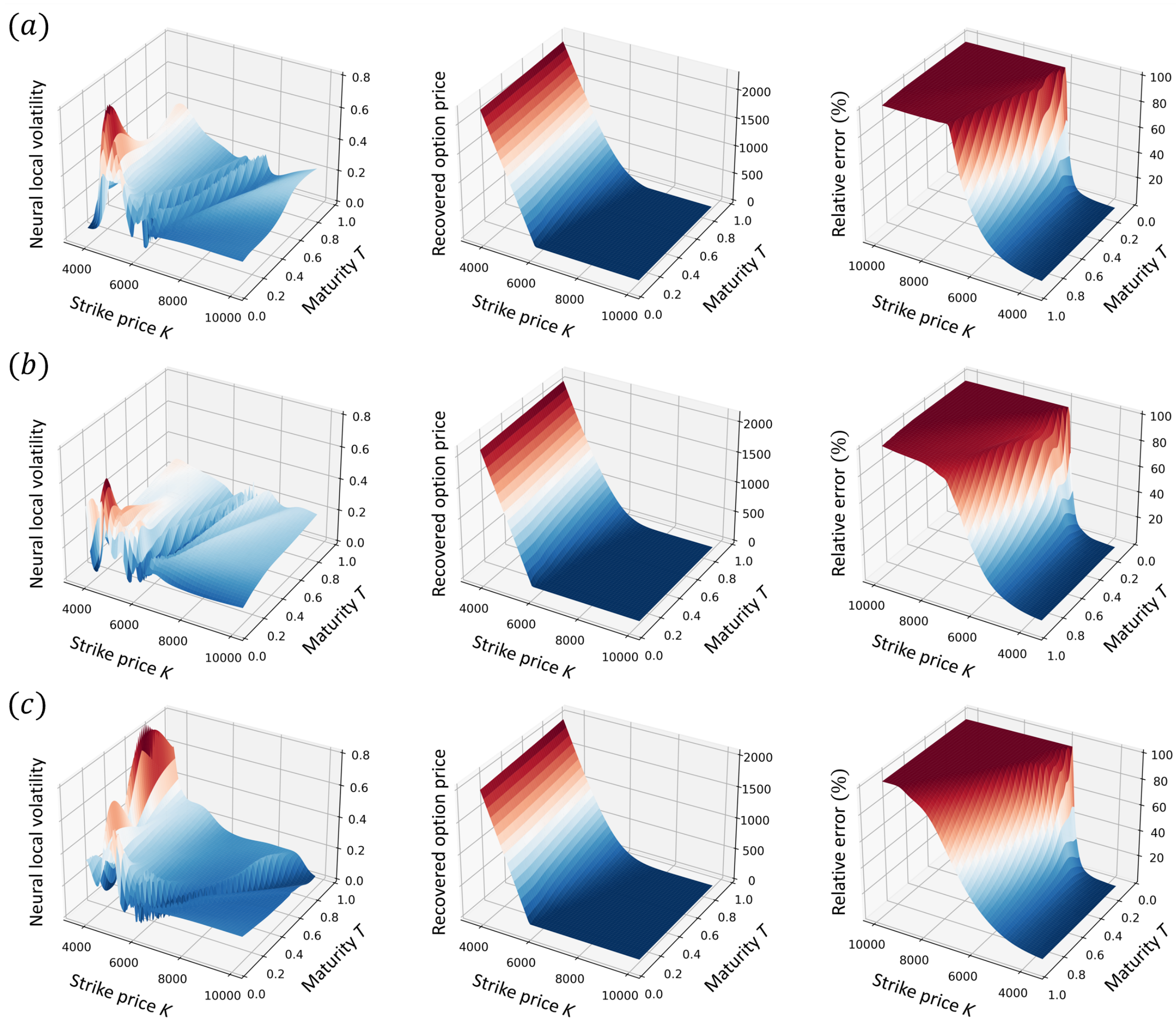


Figure 5: From left to right: calibrated local volatility, recovered option price from the local volatility model, and relative error against the parameterized option price trained with the DAX call options listed on $(a)$ 7-th; $(b)$ 8-th; and $(c)$ 9-th August 2001. Here, the weight $\lambda = 0$.

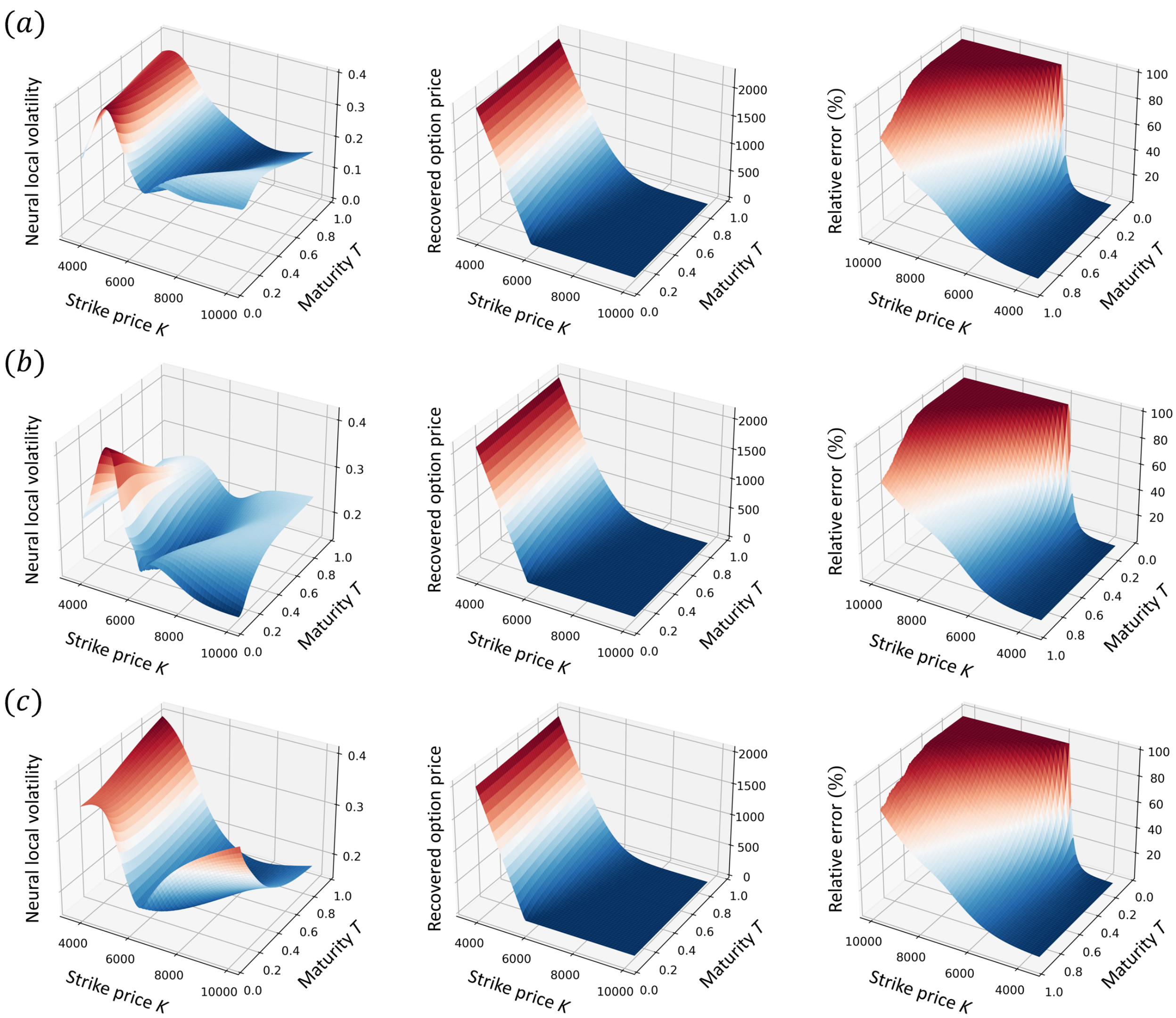


Figure 6: Same as Figure 6 but with $\lambda = 0.5$.

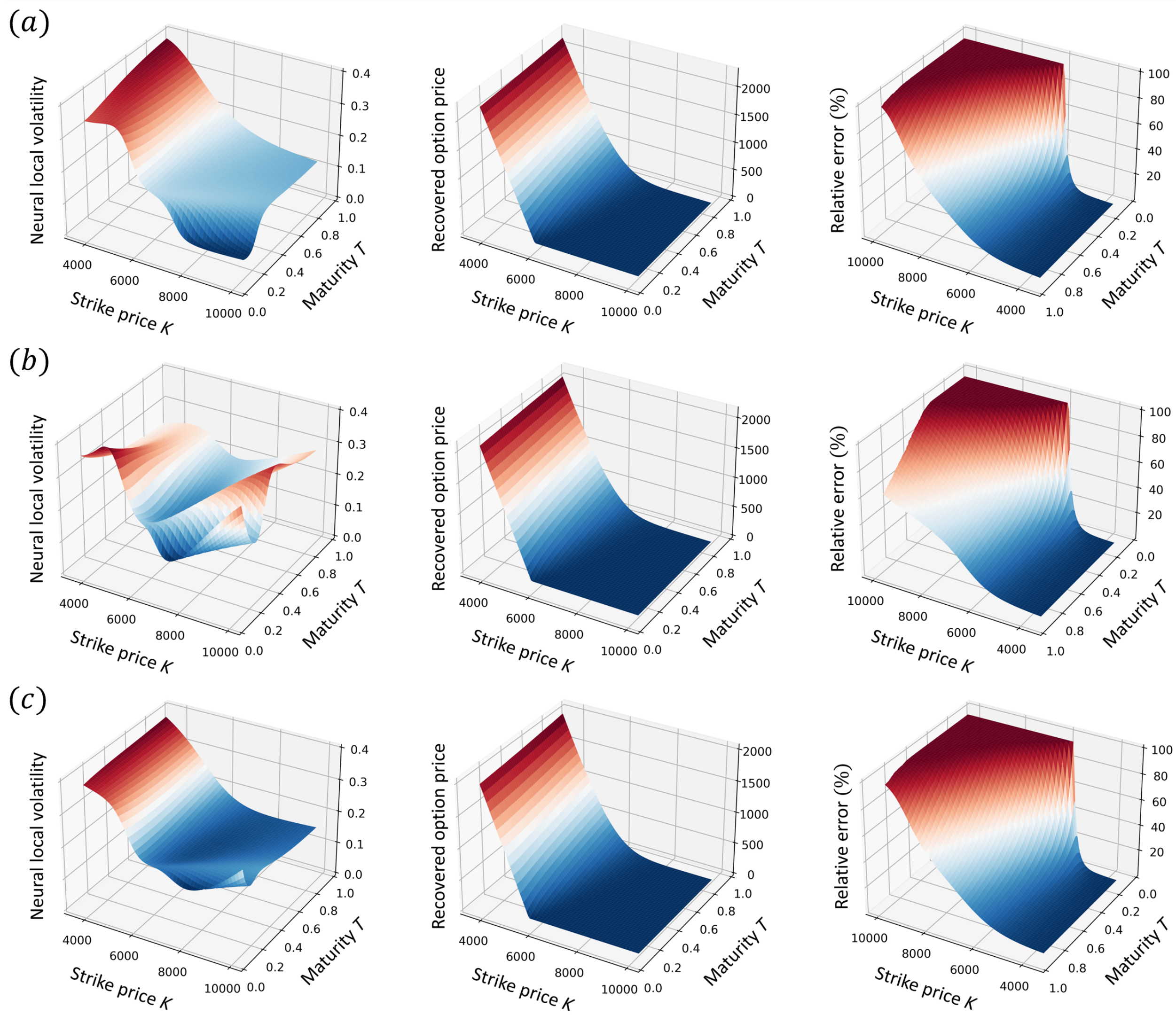


Figure 7: Same as Figure 5 but with $\lambda = 2$.

As in Chataigner et al. (2020), we benchmark the calibrated local volatility by computing the repricing RMSEs and the results are summarized in Table 2. The unregularized calibration, i.e. $\lambda = 0$, leads to the largest repricing RMSEs, as expected. The inclusion of Dupire's equation as a regularizer reduces significantly the repricing RMSEs, leading to the best results compared with previous works using Tikhonov regularization (Crépey 2002) and with regularizers that impose positiveness and boundedness of the local volatility (Chataigner et al. 2020). The proposed method exploits self-consistency between the option price and the local volatility entailed in the Dupire's equation, going beyond classical inverse problems with regularization. Note that, Table 2 evidences that the repricing RMSEs are relatively insensitive to the regularization parameter $\lambda$, signifying the robustness of our approach.

| Repricing RMSE | Ours | | | Tikhonov volatility (Crépey 2002) | Neural Dupire formula (Chataigner et al. 2020) |
|---|---|---|---|---|---|
| | $\lambda = 0$ | $\lambda = 0.5$ | $\lambda = 2$ | | |
| 7 August 2001 | 45.65 | 2.91 | 3.19 | 5.42 | 10.18 |
| 8 August 2001 | 43.84 | 3.83 | 3.93 | 5.55 | 7.44 |
| 9 August 2001 | 34.17 | 2.64 | 2.99 | 4.60 | 8.18 |

Table 2: Comparison of repricing RMSEs with previously published methods.

**Call options on the S&P 500**

To assess the dependence on the size of dataset, we apply in this section the proposed self-consistent method to calibrate the local volatility from a daily dataset of S&P 500 call options listed on 27-th September 2021. The dataset consists of 1,076 call option prices available at different maturity-strike pairs, 5 times larger than the ones used in the previous section. The underlying asset is $S_0 = 437.86$ and we assume a constant interest rate $r = 0.02$. As in previous cases, we calibrate the local volatility without and with the inclusion of the Dupire's equation as a regularizer. For $\lambda = 0$ and $\lambda = 0.5$, the repricing RMSEs are 5.60 and 2.62, respectively, and the results are shown in Figures 8 and 9. It is observed that, although the lowest repricing RMSE is achieved with nonzero $\lambda$, the increasing size of dataset improves the calibration for the unregularized case. This observation is consistent with previous benchmarking on synthetic dataset in Section 3.1.

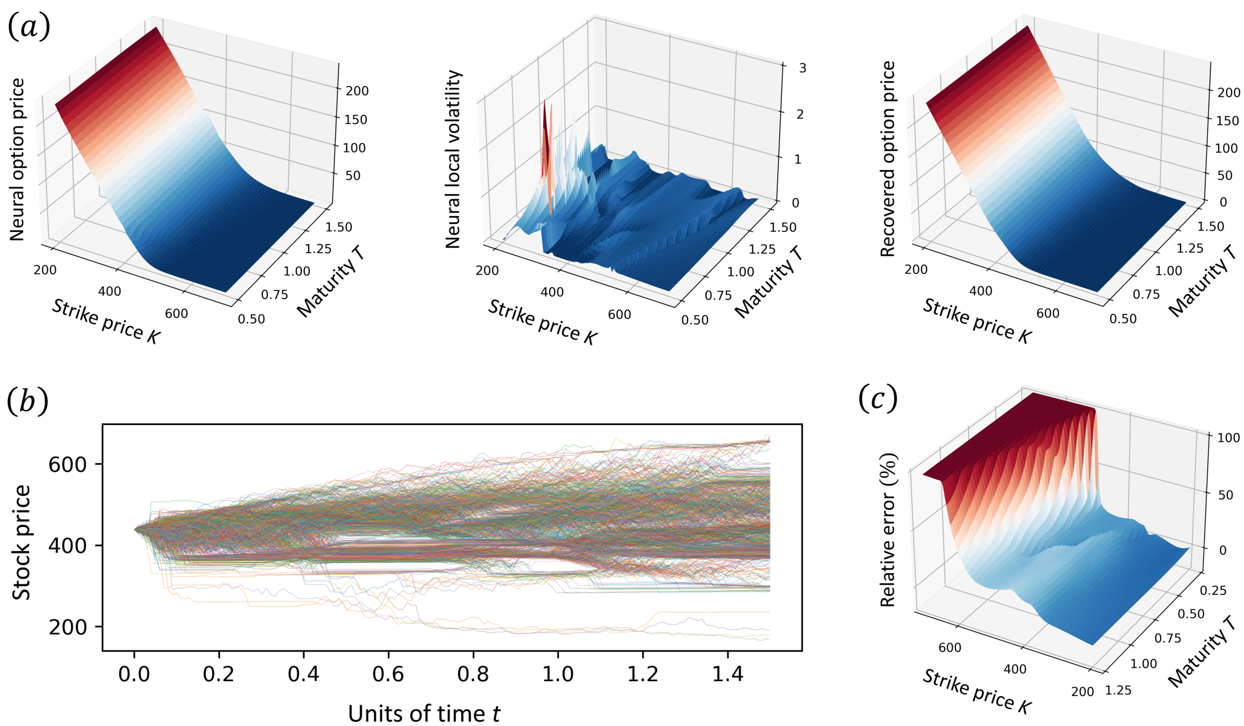


Figure 8: From left to right: $(a)$ parameterized option price, calibrated local volatility, and recovered option price from the neural local volatility model; $(b)$ price trajectories generated by simulating the neural local volatility model; and $(c)$ relative error between the recovered and the parameterized option prices. The models are trained with S&P 500 options listed on 27-th September 2021 with $\lambda = 0.0$. The dataset consists of $1,076$ option prices given at corresponding maturity-strike pairs. The repricing RMSE on training grid against market prices is: 5.60.

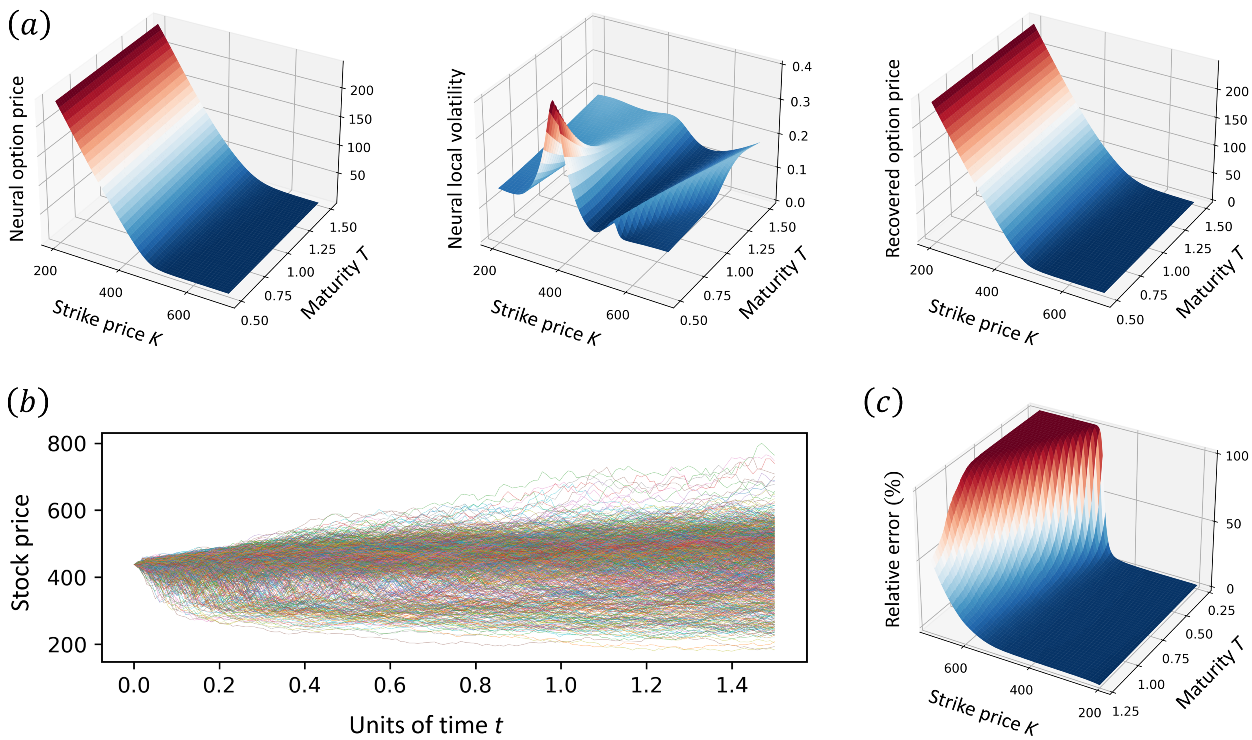


Figure 9: Same as Figure 9 but with $\lambda = 0.5$. The repricing RMSE on training grid against market prices is: 2.62.

# 4 Discussion and conclusions

In this work, we introduce a deep learning method that yields parameterized option price and 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.

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 repricing RMSEs are lower than obtained either by the canonical Tikhonov method (Achdou and Pironneau 2005, Crépey 2002) or a recent deep learning approach (Chataigner et al. 2020).

Being continuous functions, the neural networks provide full surfaces for the parameterized option price and for the calibrated local volatility, in contrast with discrete nodes seen using the canonical Tikhonov method. Correspondingly, by incorporating Dupire's equation as a regularizer, one needs to solve a two-dimensional partial differential equation at each iteration, leading to increased computation time. This drawback, however, can be leveraged by distributing the training task on multiple GPUs. Moreover, we observed from Figures 5-7 that both the market option price and the local volatility do not vary drastically over successive days. Instead of starting from random initial parameters, initiating the training from converged solutions of the previous day can lead to a substantially reduction in computation time, cf. Wang and Guet (2021b) for a case study.

# Declarations of Interest

The authors report no conflicts of interest. The authors alone are responsible for the content and writing of the paper

# Acknowledgment

The authors would like to thank Cherif Assaf for providing computational resources and Subodh Mhaisalkar for support.

# References


Y. Achdou and O. Pironneau. *Computational methods for option pricing*, volume 30 of *Frontiers in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2005.

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(1):5595–5637, 2017.

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

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

S. Crépey. Calibration of the local volatility in a trinomial tree using Tikhonov regularization. *Inverse Problems*, 19(1):91, 2002.

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.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, 2015.

P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008.

P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, and L. Bottou. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(12):3372–2408, 2010.

Z. Wang and C. Guet. Reconstructing a dynamical system and forecasting time series by self-consistent deep learning. *arXiv preprint arXiv:2108.01862*, 2021a.

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*, 2021b.