Beyond Surrogate Modeling: Learning the Local Volatility Via Shape Constraints[*]

Marc Chataigner[†], Areski Cousin [‡], Stéphane Crépey[§], Matthew Dixon[¶], and Djibril Gueye[‡]

Abstract. We explore the abilities of two machine learning approaches for no-arbitrage interpolation of European vanilla option prices, which jointly yield the corresponding local volatility surface: a finite dimensional Gaussian process (GP) regression approach under no-arbitrage constraints based on prices, and a neural net (NN) approach with penalization of arbitrages based on implied volatilities. We demonstrate the performance of these approaches relative to the SSVI industry standard. The GP approach is proven arbitrage-free, whereas arbitrages are only penalized under the SSVI and NN approaches. The GP approach obtains the best out-of-sample calibration error and provides uncertainty quantification. The NN approach yields a smoother local volatility and a better backtesting performance, as its training criterion incorporates a local volatility regularization term.

Key words. Gaussian Processes; Local Volatility; Option pricing; Neural Networks; No-arbitrage.

1. Introduction.

There have been recent surges of literature about the learning of derivative pricing functions by machine learning surrogate models, i.e. neural nets and Gaussian processes that are respectively surveyed in [11] and [4, Section 1]. There has, however, been relatively little coverage of no-arbitrage constraints when interpolating prices, and of the ensuing question of extracting the corresponding local volatility surface.

Tegnér & Roberts [12, see their Eq. (10)] first attempt the use of GPs for local volatility modeling by placing a Gaussian prior directly on the local volatility surface. Such an approach leads to a nonlinear least squares training loss function, which is not obviously amenable to gradient descent (stochastic or not), so the authors resort to a MCMC optimization. Zheng et al. [13] introduce shape constraint penalization via a multi-model gated neural network, which uses an auxiliary network to fit the parameters. The gated network is interpretable and lightweight, but the training is

Acknowledgements: The authors are thankful to Antoine Jacquier and Tahar Ferhati for useful hints regarding the SSVI method, and to an anonymous referee for stimulating comments.

[*] Single-file demos Master.html (with the results of the paper) and Master.ipynb (for dynamic execution of all scripts) are available on [https://github.com/mChataign/Beyond-Surrogate-Modeling-Learning-the-Local-Volatility-Via-Shape-Constraints]. Note that, due to github size limitations, the file Master.html file must be downloaded locally (and then opened with a browser) to be displayed.

[†] LaMME, Université d'Evry, CNRS, Université Paris-Saclay; marc.chataigner@univ-evry.fr. The PhD thesis of Marc Chataigner is co-funded by the Research Initiative "Modélisation des marchés actions, obligations et dérivés", financed by HSBC France under the aegis of the Europlace Institute of Finance, and by the public grant ANR-11-LABX-0056-LLH LabEx LMH.

[‡] Institut de Recherche en Mathématique Avancée, Université de Strasbourg, 7 rue René Descartes, 67084 Strasbourg, cedex; a.cousin@unistra.fr

[§] LPSM, Université de Paris; Stephane.Crepey@lpsm.paris. The research of S. Crépey benefited from the support of the Chair Stress Test, RISK Management and Financial Steering, led by the French Ecole polytechnique and its Foundation and sponsored by BNP Paribas.

[¶] Department of Applied Mathematics, Illinois Institute of Technology, Chicago; matthew.dixon@iit.edu.

expensive and there is no guarantee of no-arbitrage. They do not consider the local volatility and the associated regularization terms, nor do they assess the extent to which no-arbitrage is violated in a test set.

Maatouk & Bay [9] introduce finite dimensional approximation of Gaussian processes (GP) for which shape constraints are straightforward to impose and verify. Cousin et al. [3] apply this technique to ensure arbitrage-free and error-controlled yield-curve and CDS curve interpolation.

In this paper, we propose an arbitrage-free GP option price interpolation, which jointly yields the corresponding local volatility surface, with uncertainty quantification. Another contribution of the paper is to introduce a neural network approximation of the implied volatility surface, penalizing arbitrages on the basis of the Dupire formula, which is also used for extracting the corresponding local volatility surface. This is all evidenced on an SPX option dataset.

Throughout the paper we consider European puts on a stock (or index) S with dividend yield q, in an economy with interest rate term r, with q and r constant in the mathematical description and deterministic in the numerics.

Given any rectangular domain of interest in time and space, we tacitly rescale the inputs so that the domain becomes $\Omega = [0,1]^2$. This rescaling avoids any one independent variable dominating over another during any fitting of the market prices.

2. Gaussian process regression for learning arbitrage-free price surfaces.

We denote by $P_*(T,K)$ the time-0 market price of the put with maturity T and strike K on S, observed for a finite number of pairs (T,K). Our first goal is to construct, by Gaussian process regression, an arbitrage-free and continuous put price surface $P:\mathbb{R}_+\times\mathbb{R}_+\to\mathbb{R}_+$, interpolating P_* up to some error term, and to retrieve the corresponding local volatility surface $\sigma(\cdot,\cdot)$ by the Dupire formula.

In terms of the reduced prices $p(T,k)=e^{qT}P(T,K)$, where $k=Ke^{-(r-q)T}$, the Dupire formula [5] reads (assuming p of class $\mathcal{C}^{1,2}$ on $\{T>0\}$):

$$\frac{\sigma^2(T,K)}{2}=\frac{\partial_T p(T,k)}{k^2\partial^2_{k^2}p(T,k)}=:\mathrm{dup}(T,k). \tag{2.1}$$

Obviously, for this formula to be meaningful, its output must be nonnegative, which holds if the interpolating map p exhibits nonnegative derivatives w.r.t. T and second derivative w.r.t. k, i.e.

$$\partial_T p(T,k)\geq 0\,,\ \partial^2_{k^2}p(T,k)\geq 0, \tag{2.2}$$

In this section, we consider a zero-mean Gaussian process prior on the mapping $p=p(x)_{x\in\Omega}$ with correlation function c given, for any $x=(T,k), x'=(T',k')\in\Omega$, by

$$c(x,x')=\sigma^2\gamma_T(T-T',\theta_T)\gamma_k(k-k',\theta_k). \tag{2.3}$$

Here $(\theta_T,\theta_k)=\theta$ and σ^2 correspond to length scale and variance hyper-parameters of the kernel function c, whereas the functions γ_T and γ_k are kernel correlation functions.

Without consideration of the conditions (2.2), (unconstrained) prediction and uncertainty quantification are made using the conditional distribution $p \mid p(\mathbf{x}) + \boldsymbol{\varepsilon} = \boldsymbol{y}$, where $\boldsymbol{y} = [y_1, \dots, y_n]^\top$ are n noisy observations of the function p at input points $\mathbf{x} = [x_1, \dots, x_n]^\top$, corresponding to observed maturities and strikes $x_i = (T_i, k_i)$; the additive noise term $\boldsymbol{\varepsilon} = [\varepsilon_1, \dots, \varepsilon_n]^\top$ is assumed to be a zero-mean Gaussian vector, independent from $p(\mathbf{x})$, and with an homoscedastic covariance matrix given as $\varsigma^2 I_n$, where I_n is the identity matrix of dimension n. Note that bid and ask prices are considered here as (noisy) replications at the same input location.

2.1. Imposing the no-arbitrage conditions. To deal with the constraints (2.2), we adopt the solution of Cousin et al. [3] that consists in constructing a finite dimensional approximation p^h of the Gaussian prior p for which these constraints can be imposed in the entire domain Ω with a finite number of checks. One then recovers the (non Gaussian) constrained posterior distribution by sampling a truncated Gaussian process.

Remark 1. *Switching to a finite dimensional approximation can also be viewed as a form of regularization, which is also required to deal with the ill-posedness of the (numerical differentiation) Dupire formula.*

We first consider a discretized version of the (rescaled) input space $\Omega = [0,1]^2$ as a regular grid $(\imath h)_\imath$, where $\imath = (i,j)$, for a suitable mesh size h and indices i, j ranging from 0 to $1/h$ (taken in $\mathbb{N}^\star$). For each knot $\imath = (i,j)$, we introduce the hat basis functions $\phi_\imath$ with support $[(i-1)h, (i+1)h] \times [(j-1)h, (j+1)h]$ given, for $x = (T,k)$, by

$$\phi_\imath(x) = \max(1 - \frac{|T - ih|}{h}, 0) \max(1 - \frac{|k - jh|}{h}, 0).$$

We take $V = H^1(\Omega) = \{u \in L_2(\Omega) : D^\alpha u \in L_2(\Omega), |\alpha| \leq 1\}$, where $D^\alpha u$ is a weak derivative of order $|\alpha|$, as the space of (the realizations of) p. Let $V^h \subset V$ denote the finite dimensional linear subspace spanned by the M linearly independent basis functions $\phi_\imath$. The (random) surface p in V is projected onto V^h as

$$p^h(x) = \sum_\imath p(\imath h)\phi_\imath(x), \ \forall x \in \Omega. \tag{2.4}$$

If we denote $\varrho_\imath = p(\imath h)$, then $\boldsymbol{\varrho} = (\varrho_\imath)_\imath$ is a zero-mean Gaussian column vector (indexed by $\imath$) with $M \times M$ covariance matrix Γ^h such that $\Gamma^h_{\imath,\jmath} = c(\imath h, \jmath h)$, for any two grid nodes $\imath$ and $\jmath$. Let $\boldsymbol{\phi}(x)$ denote the vector of size M given by $\boldsymbol{\phi}(x) = (\phi_\imath(x))_\imath$. The equality (2.4) can be rewritten as $p^h(x) = \boldsymbol{\phi}(x) \cdot \boldsymbol{\varrho}$. Denoting by $p^h(\mathbf{x}) = [p^h(x_1), \dots, p^h(x_n)]^\top$ and by $\boldsymbol{\Phi}(\mathbf{x})$ the $n \times M$ matrix of basis functions where each row ℓ corresponds to the vector $\boldsymbol{\phi}(x_\ell)$, one has $p^h(\mathbf{x}) = \boldsymbol{\Phi}(\mathbf{x}) \cdot \boldsymbol{\varrho}$. By application of the results of [9]:

Proposition 2. (i) *The finite dimensional process p^h converges uniformly to p on Ω as $h \to 0$, almost surely,*
(ii) *$p^h(T,k)$ is a nondecreasing function of T if and only if $\varrho_{i+1,j} \geq \varrho_{i,j}, \forall (i,j)$,*
(iii) *$p^h(T,k)$ is a convex function of k if and only if $\varrho_{i,j+2} - \varrho_{i,j+1} \geq \varrho_{i,j+1} - \varrho_{i,j}, \forall (i,j)$.* ∎

In view of (i), denoting by $\mathcal{I}$ the set of 2d continuous positive functions which are nondecreasing in T and convex in k, we choose as constrained GP metamodel for the put price surface the law of p^h conditional on

$$\begin{cases} p^h(\mathbf{x}) + \boldsymbol{\varepsilon} = \boldsymbol{y} \\ p^h \in \mathcal{I}. \end{cases}$$

In view of (ii)-(iii), $p^h \in \mathcal{I}$ if and only if $\boldsymbol{\varrho} \in \mathcal{I}^h$, where $\mathcal{I}^h$ corresponds to the set of ($\imath$ indexed) vectors $\boldsymbol{\rho} = (\rho_\imath)_\imath$ such that $\rho_{i+1,j} \geq \rho_{i,j}$ and $\rho_{i,j+2} - \rho_{i,j+1} \geq \rho_{i,j+1} - \rho_{i,j}$ $\forall (i,j)$. Hence, our GP metamodel for the put price surface can be reformulated as the law of $\boldsymbol{\varrho}$ conditional on

$$\begin{cases} \boldsymbol{\Phi}(\mathbf{x}) \cdot \boldsymbol{\varrho} + \boldsymbol{\varepsilon} = \boldsymbol{y} \\ \boldsymbol{\varrho} \in \mathcal{I}^h. \end{cases} \tag{2.5}$$

2.2. Hyper-parameter learning.

Hyper-parameters consist in the length scales θ and the variance parameter σ^2 in (2.3), as well as the noise variance ς. Up to a constant, the so called marginal log likelihood of $\boldsymbol{\varrho}$ at $\lambda = [\theta, \sigma, \varsigma]^\top$ can be expressed as (see e.g. [10, Section 15.2.4, p. 523]):

$$\mathcal{L}(\lambda) = -\frac{1}{2}\boldsymbol{y}^\top \left(\boldsymbol{\Phi}(\mathbf{x})\Gamma^h \boldsymbol{\Phi}(\mathbf{x})^\top + \varsigma^2 I_n\right)^{-1} \boldsymbol{y} - \frac{1}{2}\log\Big(\det\left(\boldsymbol{\Phi}(\mathbf{x})\Gamma^h \boldsymbol{\Phi}(\mathbf{x})^\top + \varsigma^2 I_n\right)\Big).$$

We maximize $\mathcal{L}$ for learning the hyper-parameters λ (MLE estimation).

Remark 3. *The above expression does not take into account the inequality constraints in the estimation. However, Bachoc et al. [1, see e.g. their Eq. (2)] argue (and we observed empirically) that, unless the sample size is very small, conditioning by the constraints significantly increases the computational burden with negligible impact on the MLE.*

2.3. The most probable response surface and measurement noises.

We compute the joint MAP $(\hat{\boldsymbol{\rho}}, \hat{\boldsymbol{e}})$ of the truncated Gaussian vector $\boldsymbol{\varrho}$ and of the Gaussian noise vector $\boldsymbol{\varepsilon}$,

$$(\hat{\boldsymbol{\rho}}, \hat{\boldsymbol{e}}) = \underset{(\boldsymbol{\rho},\boldsymbol{e})}{\arg\max}\, \mathrm{Prob}\Big(\boldsymbol{\varrho} \in [\boldsymbol{\rho}, \boldsymbol{\rho} + \mathrm{d}\boldsymbol{\rho}], \boldsymbol{\varepsilon} \in [\boldsymbol{e}, \boldsymbol{e} + \mathrm{d}\boldsymbol{e}] \mid \boldsymbol{\Phi}(\mathbf{x}) \cdot \boldsymbol{\varrho} + \boldsymbol{\varepsilon} = \boldsymbol{y},\ \boldsymbol{\varrho} \in \mathcal{I}^{\mathrm{h}}\Big)$$

(for the probability measure Prob underlying the GP model). As $(\boldsymbol{\varrho}, \boldsymbol{\varepsilon})$ is Gaussian centered with block-diagonal covariance matrix with blocks Γ^h and $\varsigma^2 I_n$, this implies that the MAP $(\hat{\boldsymbol{\rho}}, \hat{\boldsymbol{e}})$ is a solution to the following quadratic problem :

$$\underset{\boldsymbol{\Phi}(\mathbf{x})\cdot\boldsymbol{\rho}+\boldsymbol{e}=\boldsymbol{y},\ \boldsymbol{\rho}\in\mathcal{I}^h}{\arg\min} \left(\boldsymbol{\rho}^\top (\Gamma^h)^{-1} \boldsymbol{\rho} + \boldsymbol{e}^\top (\varsigma^2 I_n)^{-1} \boldsymbol{e}\right). \tag{2.6}$$

We define the most probable measurement noise to be $\hat{\boldsymbol{e}}$ and the most probable response surface $\hat{p}^h(\mathbf{x}) = \boldsymbol{\Phi}(\mathbf{x}) \cdot \hat{\boldsymbol{\rho}}$. Distance to the data can be an effect of arbitrage opportunities within the data and/or misspecification / lack of expressiveness of the kernel.

2.4. Sampling finite dimensional Gaussian processes under shape constraints.

The conditional distribution of $\boldsymbol{\varrho} \mid \boldsymbol{\Phi}(\mathbf{x}) \cdot \boldsymbol{\varrho} + \boldsymbol{\varepsilon} = \boldsymbol{y}$ is multivariate Gaussian with mean $\boldsymbol{\eta}_{\mathbf{y}}(\mathbf{x})$ and covariance matrix $\mathbf{C}_{\mathbf{y}}(\mathbf{x})$ such that

$$\boldsymbol{\eta}_{\mathbf{y}}(\mathbf{x}) = \Gamma^h \boldsymbol{\Phi}(\mathbf{x})^\top (\boldsymbol{\Phi}(\mathbf{x}) \Gamma^h \boldsymbol{\Phi}(\mathbf{x})^\top + \varsigma^2 I_n)^{-1} \boldsymbol{y} \tag{2.7}$$

$$\mathbf{C}_{\mathbf{y}}(\mathbf{x}) = \Gamma^h \boldsymbol{\Phi}(\mathbf{x})^\top (\boldsymbol{\Phi}(\mathbf{x}) \Gamma^h \boldsymbol{\Phi}(\mathbf{x})^\top + \varsigma^2 I_n)^{-1} \boldsymbol{\Phi}(\mathbf{x}) \Gamma^h. \tag{2.8}$$

In view of (2.5), we thus face the problem of sampling from this truncated multivariate Gaussian distribution, which we do by Hamiltonian Monte Carlo, using the MAP $\hat{\boldsymbol{\varrho}}$ of $\boldsymbol{\varrho}$ as the initial vector (which must verify the constraints) in the algorithm.

2.5. Local volatility.

Due to the shape constraints and to the ensuing finite-dimensional approximation with basis functions of class $\mathcal{C}^0$ (for the sake of Proposition 2), p^h is not differentiable. Hence, exploiting GP derivatives analytics, as done for the mean in [4, cf. Eq. (10)] and also for the covariance in [8], is not possible for deriving the corresponding local volatility surface here. Computation of derivatives involved in the Dupire formula is implemented by finite differences with respect to a coarser grid (than the grid of basis functions). Another related solution would be to formulate a weak form of the Dupire equation and construct a local volatility surface approximation using a finite element method.

See Algorithm 2.1 for the main steps of the GP approach.

Algorithm 2.1 The GP algorithm for local volatility surface approximation.

```
Data: Put price training set $p_\star$
Result: $M$ realizations of the local volatility surface $\{\mathrm{dup}_i^h\}_{i=1}^M$
$\hat{\lambda}$ ← Maximize the marginal log-likelihood of the put price surface $p^h$ w.r.t. $\lambda$
    // Hyperparameter fitting
$(\hat{\boldsymbol{\rho}}, \hat{\boldsymbol{e}})$ ← Minimize quadratic problem (2.6) based on $\hat{\lambda}$ // Joint MAP estimate
$\hat{\boldsymbol{\rho}}$ → Initialize a Hamiltonian MC sampler
$p_1^h, \ldots, p_M^h$ ← Hamiltonian MC Sampler // Sampling price surfaces
$\mathrm{dup}_i^h$ ← Finite difference approximation using each $p_i^h$, $i := 1 \to M$
```

3. Neural networks implied volatility metamodeling.

Our second goal is to use neural nets (NN) to construct an implied volatility (IV) put surface $\Sigma : \mathbb{R}_+ \times \mathbb{R} \to \mathbb{R}_+$, interpolating implied volatility market quotes Σ_* up to some error term, both being stated in terms of a put option maturity T and log-(forward) moneyness $\kappa = \log(\frac{k}{S_0}) = \log\left(\frac{K}{S_0}\right) - (r-q)T$. The advantage of using implied volatilities rather than prices (as previously done in [2]), both being in bijection via the Black-Scholes put pricing formula as well known, is their lower variability, hence better performance as we will see.

The corresponding local volatility surface σ is given by the following local volatility implied variance formula, i.e. the Dupire formula stated in terms of the implied total variance[1] $\Theta(T,\kappa) = \Sigma^2(T,\kappa)T$ (assuming Θ of class $\mathcal{C}^{1,2}$ on $\{T > 0\}$):

$$\sigma^2(T,K) = \frac{\partial_T \Theta}{1 - \frac{\kappa}{\Theta}\partial_\kappa \Theta + \frac{1}{4}\left(-\frac{1}{4} - \frac{1}{\Theta} + \frac{\kappa^2}{\Theta^2}\right)(\partial_\kappa \Theta)^2 + \frac{1}{2}\partial_{\kappa^2}\Theta}(T,\kappa) =: \frac{\mathrm{cal}_T(\Theta)}{\mathrm{butt}_k(\Theta)}(T,\kappa). \tag{3.1}$$

We use a feedforward NN with weights $\mathbf{W}$, biases $\mathbf{b}$ and smooth activation functions for parameterizing the implied volatility and total variance, which we denote by

$$\Sigma = \Sigma_{\mathbf{W},\mathbf{b}}\,,\ \Theta = \Theta_{\mathbf{W},\mathbf{b}}.$$

The terms $\mathrm{cal}_T(\Theta_{\mathbf{W},\mathbf{b}})$ and $\mathrm{butt}_k(\Theta_{\mathbf{W},\mathbf{b}})$ are available analytically, by automatic differentiation, which we exploit below to penalize calendar spread arbitrages, i.e. negativity of $\mathrm{cal}_T(\Theta)$, and butterfly arbitrage, i.e. negativity of $\mathrm{butt}_k(\Theta)$.

The training of NNs is a non-convex optimization problem and hence does not guarantee convergence to a global optimum. We must therefore guide the NN optimizer towards a local optima that has desirable properties in terms of interpolation error and arbitrage constraints. This motivates the introduction of an arbitrage penalty function into the loss function to select the most appropriate local minima. An additional challenge is that maturity-log moneyness pairs with quoted option prices are unevenly distributed and the NN may favor fitting to a cluster of quotes to the detriment of fitting isolated points. To remedy this non-uniform data fitting problem, we re-weight the observations by the Euclidean distance between neighboring points. More precisely, given n observations $\chi_i = (T_i, \kappa_i)$ of maturity-log moneyness pairs and of the corresponding market implied volatilities $\Sigma_*(\chi_i)$, we construct the $n \times n$ distance matrix with general term $d(\chi_i, \chi_j) = \sqrt{(T_j - T_i)^2 + (\kappa_j - \kappa_i)^2}$. We then define the loss weighting w_i for each point χ_i as the distance $w_i = \min_{j, j \neq i} d(\chi_i, \chi_j)$. with the closest point. These modifications aim at reducing error for any isolated points. In addition, in order to avoid linear saturation of the neural network, we apply a further log-maturity change of variables (adapting the partial derivatives accordingly).

Learning the weights $\mathbf{W}$ and biases $\mathbf{b}$ to the data subject to no arbitrage soft constraints (i.e. with penalization of arbitrages) then takes the form of the following (nonconvex) loss minimization problem:

$$\underset{\mathbf{W},\mathbf{b}}{\arg\min} \sqrt{\frac{1}{n}\sum_i \left(w_i \frac{\Sigma_{\mathbf{W},\mathbf{b}}(\chi_i) - \Sigma_*(\chi_i)}{\Sigma_*(\chi_i)} \right)^2} + \frac{\mu_w}{m} \sum_{\xi \in \Omega_h} \lambda^\top \mathcal{R}(\Theta_{\mathbf{W},\mathbf{b}})(\xi), \tag{3.2}$$

where $\lambda = [\lambda_1, \lambda_2, \lambda_3]^\top \in \mathbb{R}^3_+$ and

$$\mathcal{R}(\Theta) = [\mathrm{cal}_T^-(\Theta), \mathrm{butt}_k^-(\Theta), \big(\frac{\mathrm{cal}_T}{\mathrm{butt}_k}(\Theta) - \overline{a}\big)^+ + \big(\frac{\mathrm{cal}_T}{\mathrm{butt}_k}(\Theta) - \underline{a}\big)^-]^\top$$

is a regularization penalty vector evaluated over a penalty grid Ω_h with m nodes as detailed below. The error criterion is calculated as a root mean square error on relative difference, so that it does not discriminate high or low implied volatilities.

[1] This follows from the Dupire formula by simple transforms detailed in [6, p.13].

The first two elements in the penalty vector $\mathcal{R}(\Theta)$ favor the no-arbitrage conditions (2.2) and the third element favors desired lower and upper bounds $0 < \underline{a} < \overline{a}$ (constants or functions of T) on the estimated local variance $\sigma^2(T, K)$. In order to adjust the weight of penalization, we multiply our penalties by the weighting mean $\mu_w := \frac{1}{m}\sum_i w_i$. Suitable values of the "Lagrange multipliers" λ, ensuring the right balance between fit to the market implied volatilities and the constraints, is then obtained by grid search. Of course a soft constraint (penalization) approach does not fully prevent arbitrages. However, for large λ, arbitrages are extremely unlikely to occur, except perhaps very far from Ω. With this in mind, we use a penalty grid Ω_h that extends well beyond the domain of the IV interpolation. This is intended so that the penalty term penalizes arbitrages outside of the domain used for IV Interpolation.

See Algorithm 3.1 for the pseudo-code of the NN approach.

Algorithm 3.1 The NN-IV algorithm for local volatility surface approximation.

```
Data: Market implied volatility surface $\Sigma_*$
Result: The local volatility surface $\sqrt{\frac{\text{cal}_T}{\text{butt}_k}}(\Theta_{\hat{\mathbf{W}},\hat{\mathbf{b}}})$
$(\hat{\mathbf{W}}, \hat{\mathbf{b}}) \leftarrow$ Minimize the penalized training loss (3.2) w.r.t. $(\mathbf{W}, \mathbf{b})$;
$\sqrt{\frac{\text{cal}_T}{\text{butt}_k}}(\Theta_{\hat{\mathbf{W}},\hat{\mathbf{b}}}) \leftarrow$ AAD differentiation of the trained NN implied vol. surface
```

4. Numerical results.

4.1. Experimental design.

Our training set is prepared using SPX European puts with different available strikes and maturities ranging from 0.005 to 2.5 years, listed on 18th May 2019, with $S_0 = \$2859.53$. Each contract is listed with a bid/ask price and an implied volatility corresponding to the mid-price. The associated interest rate is constructed from US treasury yield curve and dividend yield curve rates are then obtained from call/put parity applied to the option market prices and forward prices. We preprocess the data by removing the shortest maturity options, with $T < 0.055$, and the numerically inconsistent observations for which the gap between the listed implied volatility and the implied volatility calibrated from mid-price with our interest/dividend curves exceeds 5% of the listed implied volatility. But we do not remove arbitrable observations. The preprocessed training set is composed of 1720 market put prices. The testing set consists of a disjoint set of 1725 put prices.

All results for the GP method are based on using Matern $\nu = 5/2$ kernels over a $[0, 1]^2$ domain with fitted kernel standard-deviation hyper-parameter $\hat{\sigma} = 185.7611$, length-scale hyper-parameters $\hat{\theta}_k = 0.3282$ and $\hat{\theta}_T = 0.2211$, and homoscedastic noise standard deviation, $\hat{\varsigma} = 0.6876$.[2] The grid of basis functions for constructing the finite-dimensional process p^h has 100 nodes in the modified strike direction and 25 nodes in the maturity direction. The Matlab interior point convex algorithm `quadprog` is used to solve the MAP quadratic program (2.6).

[2] When re-scaled back to the original input domain, the fitted length scale parameters of the 2D Matern $\nu = 5/2$ are $\hat{\theta}_k = 973.1901$ and $\hat{\theta}_T = 0.5594$.

Regarding the NN approach, we use a three layer architecture similar to the one based on prices (instead of implied volatilities in Section 3) in [2], to which we refer the reader for implementation details. We use a penalty grid Ω_h with $m = 50 \times 100$ nodes. In the moneyness and maturity coordinates, the domain of the penalty grid is $[0.005, 10] \times [0.5, 2]$.

4.2. Arbitrage-free SVI. We benchmark the machine learning results with the industry standard provided by the arbitrage free stochastic volatility inspired (SVI) model of [7]. Under the "natural parameterization" $\mathrm{SVI} = (\Delta, \mu, \rho, \omega, \zeta)$, the implied total variance is given, for any fixed T, by

$$\Theta_{\mathrm{SVI}}(\kappa) = \Delta + \frac{\omega}{2}\left(1 + \rho(\kappa - \mu)\zeta + \sqrt{(\zeta(\kappa - \mu) + \rho)^2 + (1 - \rho^2)}\right). \tag{4.1}$$

Our SSVI parameterization of a surface corresponds to $\mathrm{SVI}_T = (0, 0, \rho, \Theta_T, \phi(\Theta_T))$ for each T, where Θ_T is the at-the-money total implied variance and we use for ϕ a power law function $\phi(\vartheta) = \frac{\eta}{\vartheta^{\gamma}(1+\vartheta)^{1-\gamma}}$. [7, Remark 4.4] provides sufficient conditions on SSVI parameters ($\eta(1 + |\rho|) \leq 2$ with $\gamma = 0.5$) that rule out butterfly arbitrage, whereas SSVI is free of calendar arbitrage when Θ_T is nondecreasing.

We calibrate the model as in [7]:[3] First, we fit the SSVI model; Second, for each maturity in the training grid, the five SVI parameters are calibrated, (starting in each case from the SSVI calibrated values. The implied volatility is obtained for new maturities by a weighted average of the parameters associated with the two closest maturities in the training grid, T and U, say, with weights determined by Θ_T and Θ_U. The corresponding local volatility is extracted by finite difference approximation of (3.1).

As, in practice, no arbitrage constraints are implemented for SSVI by penalization (see [7, Section 5.2]), in the end the SSVI approach is in fact only practically arbitrage-free, much like our NN approach, whereas it is only the GP approach that is proven arbitrage-free.

4.3. Calibration results. Training times for SSVI, GP, and NNs are reported in the last row of Table 1 which, for completeness, also includes numerical results obtained by NN interpolation of the prices as per [2]. Because price based NN results are outperformed by IV based NN results we only focus on the IV based NN in the figures that follow, referring to [2] for every detail on the price based NN approach. We recall that, in contrast to the SSVI and NNs which fit to mid-quotes, GPs fit to the bid-ask prices.

The GP implementation is in Matlab whereas the SSVI and NN approaches are implemented in Python. On our (large) dataset, the constrained GP has the longest training time. Training is longer for constrained SSVI than for unconstrained SSVI because of the ensuing amendments to the optimization routine. There are no arbitrage violations observed for any of the constrained methods in neither the training or the testing grid. Unconstrained methods yield 18 violations with NN and 177 with SSVI on the testing set, out of a total of 1725 testing points, i.e. violations in 1.04% and 10.26% of the test nodes. The unconstrained GP approach yields constraint violations on 12.5% of the basis function nodes $\imath h$. The NN penalizations

[3] Building on https://www.mathworks.com/matlabcentral/profile/authors/4439546.

IV RMSE (Price RMSE)	SSVI	GP	IV based NN	Price based NN	SSVI Unconstr.	GP Unconstr.	IV based NN Unconstr.	Price based NN Unconstr.
Calibr. fit on the training set	1.37% (2.574)	0.58% (0.338)	1.23% (2.897)	13.70% (9.851)	1.04% (2.691)	0.60% (0.321)	0.84% (2.163)	5.65 % (2.456)
Calibr. fit on the testing set	1.52% (2.892)	0.57% (0.355)	1.29% (2.966)	14.27% (10.347)	1.09% (2.791)	0.57% (0.477)	0.86% (2.045)	6.14% (2.888)
MC backtest	8.69% (22.826)	19.76% (74.017)	2.95% (4.989)	6.37% (11.764)	N/A	N/A	N/A	N/A
CN backtest	6.88% (33.545)	7.86% (35.270)	3.43% (11.976)	5.56% (26.785)	N/A	N/A	N/A	N/A
Comput. time (seconds)	33	856	191	185	1	16	76	229

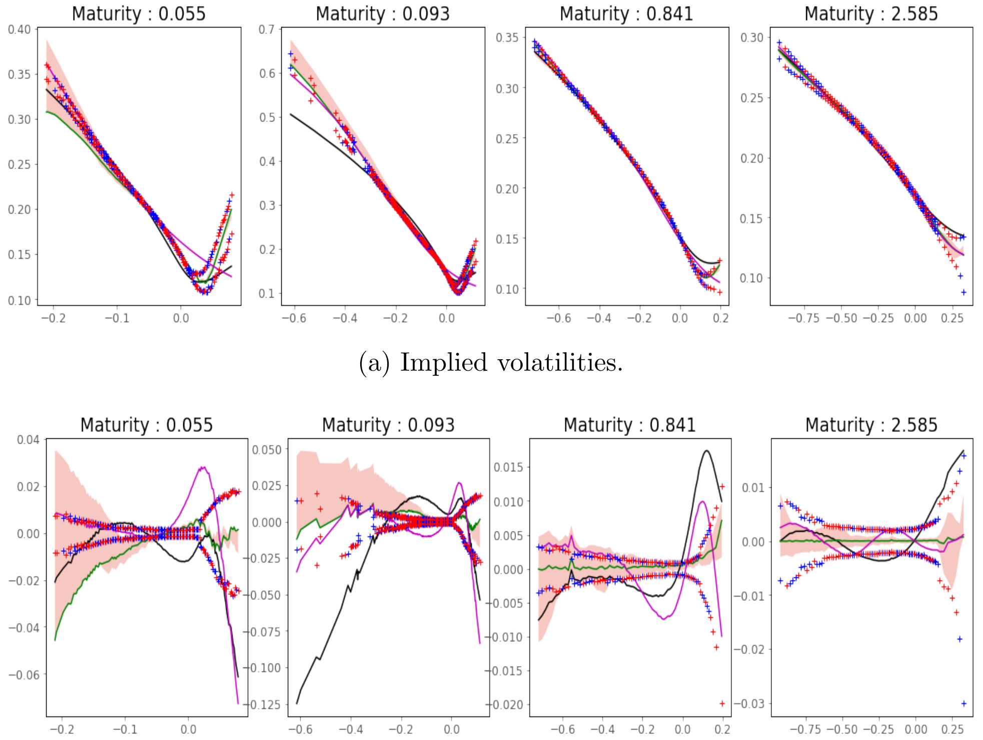
Table 1: The IV and price RMSEs of the SSVI, GP and NN approaches. Last row: computation times (in seconds).

$(\mathrm{cal}_T)^-$ and $(\mathrm{butt}_k)^-$ vanish identically on the penalty grid Ω_h in the constrained case, whereas in the unconstrained case their averages across grid nodes in Ω_h are $(\mathrm{cal}_T)^- = 3.91 \times 10^{-6}$ and $(\mathrm{butt}_k)^- = 1.60 \times 10^{-2}$ with the IV based NN.

Fig. 1(a-b) respectively compare the fitted IV surfaces and their errors with respect to the market mid-implied volatilities, among the constrained methods. The surface is sliced at various maturities (more slices are available in the github) and the IVs corresponding to the bid-ask price quotes are also shown – the blue and red points respectively denote training and test observations.

We generally observe good correspondence between the models and that each curve typically falls within the bid-ask spread, except for the shortest maturity contracts where there is some departure from the bid-ask spreads for observations with the lowest log-moneyness values. We see on Fig. 1(b) that the GP IV errors are small and mostly less than 5 volatility points, whereas NN and SSVI exhibit IV error that may exceed 15 volatility points. The green line and the red shaded envelopes respectively denote the GP MAP estimates and the posterior uncertainty bands under 100 samples per observation. The support of the posterior GP process assessed on the basis of 100 simulated paths of the GP captures the majority of bid-ask quotes. The GP MAP estimate occasionally corresponds to the boundary of the support of the posterior simulation. This indicates that the posterior truncated Gaussian distribution is heavily skewed for some points, and that the MAP estimate consequently saturates the arbitrage constraints. This indicates a tension between these constraints and the calibration requirement, which cannot be fully reconciled, most likely because some of the (short maturity) data are arbitrable (they are at least illiquid and hence noisy). See notebook for location of arbitrages in the unconstrained approach.

Fig. 1(a-b) suggest that the data may exhibit arbitrage at the lowest maturities where the methods depart from the bid-ask spreads. This is further supported in Fig. 2(a-b) which shows the corresponding methods without the no-arbitrage constraints. In Fig. 2(a-b) we observe that the estimated IVs now fall within close proximity of the bid-ask spreads–all methods exhibit an error typically less than 5 volatility points. Note that the y-axis has been scaled for each plot in Fig. 2(b) to accommodate the wide uncertainty band of the posterior for the unconstrained GP. Whereas the uncertainty band of the constrained GP spanned at most 10 volatility points, the uncertainty band of the unconstrained GP is an order of magnitude larger, sometimes spanning more than 100 volatility points.



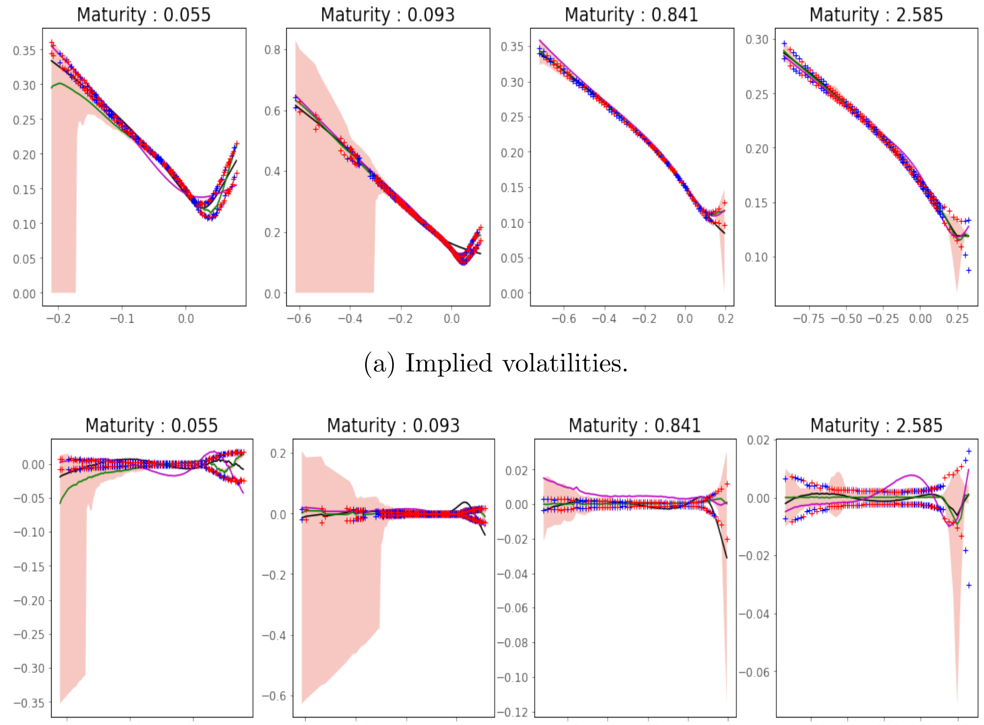
(a) Implied volatilities.

(b) Fitted IV errors with respect to mid-price IVs.

Figure 1: *Slices of constrained GP (green), NN (purple), and SSVI (black) models of SPX puts with training bid-asks IVs (+) and testing bid-asks IVs as a function of log forward moneyness (+)(the bid-ask IVs are reconstructed numerically from the corresponding bid-ask market prices). The shaded envelopes show 100 paths of the constrained GP's posterior.*

Fig. 3 shows the local volatility surfaces that stem from the three constrained approaches. Fig. 3(a) shows the spiky local volatility surface generated by SSVI, capped at the 200% level for scaling convenience. Fig. 3(b) shows the capped local volatility surface constructed from the GP MAP price estimate. Fig. 3(c) shows the (complete) NN local volatility surface.

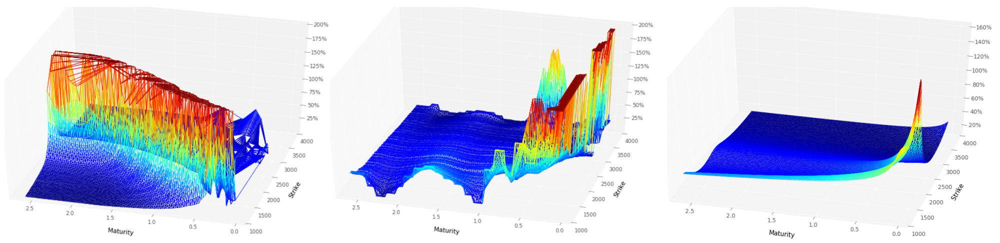
4.4. In-sample and out-of-sample calibration errors. The error between the prices of the calibrated models and the market data are evaluated on both the training and the out-of-sample data set. The first two rows of Table 1 compare the in-sample and out-of-sample RMSEs of the prices and implied volatilities across the different approaches. The differences between the training and testing RMSEs are small, suggesting that all approaches are not over-fitting the training set. The GP



(a) Implied volatilities.

(b) Fitted IV errors with respect to mid-price IVs.

Figure 2: *Same as Figure 1 but for unconstrained GP, NN and SSVI.*



(a) The local volatility surface generated by SSVI with finite differences, capped at the 200% level.

(b) The MAP estimate of the GP local volatility surface, capped at the 200% level.

(c) The implied volatility based NN local volatility surface (with the local volatility penalization).

Figure 3: *The GP, SSVI, and NN local volatility estimate.*

exhibits the lowest price RMSEs.

4.5. Backtesting results. The first repricing backtest estimates the prices of the European options corresponding to the testing set, by Monte Carlo sampling in each calibrated local volatility model (same methodology as in [2, Section 7.2]). The second approach uses finite differences to price the options with the calibrated local volatility surfaces. The pricing PDEs with local volatility are discretized using a

Crank-Nicolson (CN) scheme implemented on a 100×100 backtesting grid. The last two rows in Table 1 compare the resulting price backtest RMSEs across the different approaches. The NN fitted to implied volatilities exhibit significantly lower errors in the backtests, followed by NN based on prices, SSVI and GP. To quantify discretization error in these backtesting results (as opposed to the part of the error stemming from a wrong local volatility), we ran the same backtests in a Black-Scholes model with 20% volatility and the associated prices. The corresponding Monte Carlo and Crank-Nicholson backtesting IV(price) RMSEs are 2.90%(1.56) and 0.846%(4.10), confirming the significance of the above results.

Conclusion. We approach the option quote fitting problem from two perspectives: (i) the GP approach assumes noisy data and hence the existence of a latent function. The mid-prices are not considered, rather the GP calibrates to bid-ask quotes; and (ii) the NN and SSVI approaches fit to the mid-prices under a noise-free assumption. While these two approaches are important to distinguish on theoretical grounds, in practice there are other factors which are more important for, in particular, local volatility modeling. In line with classical inverse problems theory, we find that regularization of the local volatility is critical for backtesting performance.

References.

[1] François Bachoc, Agnes Lagnoux, Andrés F López-Lopera, et al. Maximum likelihood estimation for Gaussian processes under inequality constraints. *Electronic Journal of Statistics*, 13(2):2921–2969, 2019.

[2] Marc Chataigner, Stéphane Crépey, and Matthew Dixon. Deep local volatility. *Risks*, 8(3):82, 2020.

[3] Areski Cousin, Hassan Maatouk, and Didier Rullière. Kriging of financial term-structures. *European J. Oper. Res.*, 255(2):631–648, 2016.

[4] Stéphane Crépey and Matthew Dixon. Gaussian process regression for derivative portfolio modeling and application to CVA computations. *Journal of Computational Finance*, 24(1):47–81, 2020.

[5] Bruno Dupire. Pricing with a smile. *Risk*, 7:18–20, 1994.

[6] Jim Gatheral. *The volatility surface: a practitioner's guide*. Wiley, 2011.

[7] Jim Gatheral and Antoine Jacquier. Arbitrage-free SVI volatility surfaces. *Quantitative Finance*, 14(1):59–71, 2014.

[8] Mike Ludkovski and Yuri Saporito. Krighedge: Gaussian process surrogates for delta hedging, 2020. arXiv:2010.08407.

[9] Hassan Maatouk and Xavier Bay. Gaussian process emulators for computer experiments with inequality constraints. *Math. Geosci.*, 49(5):557–582, 2017.

[10] K. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

[11] Johannes Ruf and Weiguan Wang. Neural networks for option pricing and hedging: a literature review. *Journal of Computational Finance*, 24(1), 2020.

[12] Martin Tegnér and Stephen Roberts. A probabilistic approach to nonparametric local volatility. *arXiv preprint arXiv:1901.06021*, 2019.

[13] Yu Zheng, Yongxin Yang, and Bowei Chen. Gated neural networks for implied volatility surfaces, 2020. arXiv:1904.12834.