VolGAN:
a generative model for arbitrage-free implied volatility surfaces

Milena Vuletić[*] and Rama Cont
Mathematical Institute, University of Oxford

2023.

Abstract

We introduce VolGAN, a generative model for arbitrage-free implied volatility surfaces. The model is trained on time series of implied volatility surfaces and underlying prices and is capable of generating realistic scenarios for joint dynamics of the implied volatility surface and the underlying asset. We illustrate the performance of the model by training it on SPX implied volatility time series and show that it is able to learn the covariance structure of the co-movements in implied volatilities and generate realistic dynamics for the (VIX) volatility index. In particular, the generative model is capable of simulating scenarios with non-Gaussian distributions of increments for state variables as well as time-varying correlations.

JEL Classification: G13, G17, C15, C22, C45, C53, C63

[*] Corresponding author. Email: Milena.Vuletic@maths.ox.ac.uk

Contents

1 Introduction

Option prices are quoted in terms of their *implied volatilities*, which are obtained by inverting the Black-Scholes formula given the market prices of options. The implied volatility surface, which summarises the cross-section of option prices across strikes and maturities, gives a snapshot of the state of the options market. The dependence of implied volatility on moneyness and time-to-maturity, which is referred to as the *smile*, *skew* and *term structure* have inspired the development of alternative option pricing models [Gatheral, 2011, Heston, 1993, Cont and Tankov, 2004]. Any such option pricing model implies a model for the cross-sectional dependence of implied volatilities on strike and maturity, as well as their dynamics across time. However, this dynamics is typically intractable and there has been an interest from practitioners in directly modelling the dynamics of implied volatility as a state variable [Schönbucher, 1999, Babbar, 2001, Cont and da Fonseca, 2002, Cont et al., 2002, Durrleman, 2010, Cont and Vuletic, 2023, Avellaneda et al., 2020]. Such 'market models' of implied volatility should appropriately capture the co-movements of implied volatilities across moneyness and time-to-maturity, reproduce the empirically observed dynamics of implied volatilities [Cont and da Fonseca, 2002], be able to capture the smile, skew, and term structure, and satisfy arbitrage constraints [Davis and Hobson, 2007, Gerhold and Gülüm, 2020].

Given the high dimensionality of the volatility surface and the complexity of its dynamics, it is challenging to capture all these properties in a parametric model. It is therefore of interest to examine whether a data-driven approach can be used to overcome these modelling challenges.

In the present work we propose VolGAN, a *fully data-driven* generative model for the dynamic simulation of arbitrage-free implied volatility surfaces. Our model is trained on a time series of market-quoted implied volatilities and is capable of generating realistic dynamic scenarios for implied volatility surfaces. We illustrate the performance of the model by training it on SPX implied volatility time series and show that it is able to learn the covariance structure of co-movements in implied volatilities and generate realistic dynamics for the (VIX) volatility index [CBOE, 2022]. In particular, the generative model is capable of simulating scenarios with non-Gaussian distributions of increments for state variables as well as time-varying correlations.

Our model builds on a line of previous work on the use of generative adversarial networks (GANs) for scenario simulation in finance, starting with [Takahashi et al., 2019] and [Wiese et al., 2020] for price dynamics. More recently, GAN methods have been deployed for scenario simulation in options markets. [Wiese et al., 2019] and [Wiese et al., 2021] use a classical GAN approach. [Cuchiero et al., 2020] and [Cohen et al., 2022] use a "neural SDE" to parameterize volatility surface dynamics. [Cao et al., 2020] use a supervised learning approach to extract information from historical implied volatility dynamics, while [Ning et al., 2023] combines SDEs with Variational Autoencoders [Kingma et al., 2019].

In contrast with the aforementioned approaches which deploy the classical

GAN methodology of [Goodfellow et al., 2014] using binary cross-entropy (BCE) as a training objective, VolGAN uses a bespoke loss function adapted to the financial application at hand, as advocated in [Cont et al., 2022, Vuletić et al., 2023], combined with a scenario reweighting approach based on [Cont and Vuletic, 2023].

Outline. Section 2 summarizes properties of implied volatility surfaces and outlines some desirable requirements for a dynamic model of implied volatility. Section 3 describes VolGAN, our proposed generative model for implied volatility surfaces. Section 4 presents the results obtained by training VolGAN on SPX implied volatility data and discusses the model's ability to produce realistic scenarios for implied volatility co-movements and the VIX index.

2 Implied volatility surfaces: shape constraints and dynamics

Denoting the price of the underlying asset by S_t, the implied volatility may be parameterized in terms of moneyness $m = K/S_t$ and time to maturity $\tau = T - t$ of the option. The implied volatility associated with a call option with moneyness m and time-to-maturity τ on a non-dividend paying asset S is the unique value $\sigma_t(m, \tau)$ such that the Black-Scholes price $C_{BS}(S_t, K, \tau, \sigma_t(m, \tau))$ matches the market price $C_t(m, \tau)$ of the call:

$$C_t(m,\tau) = C_{BS}(S_t, K, \tau, \sigma_t(m,\tau)) \quad = \quad S_t N(d_1) - K e^{-r\tau} N(d_2)$$

$$d_1 = \frac{-\ln m + \tau(r + \frac{\sigma^2}{2})}{\sigma\sqrt{\tau}} \qquad d_2 = \frac{-\ln m + \tau(r - \frac{\sigma^2}{2})}{\sigma\sqrt{\tau}},$$

where N is the c.d.f of a $\mathcal{N}(0, I_d)$ random variable. The implied volatility surface $\sigma_t(m, \tau)$ at date t provides a snapshot of options prices in the market [Gatheral, 2011]: specifying the implied volatility surface is equivalent to specifying the prices of all European calls and puts available in the market, given the current term structure of interest rates and dividends.

2.1 Static arbitrage and shape constraints

It has been empirically observed that implied volatilities of call and put options in listed options markets exhibit a dependence on exercise price K and maturity date T [Cont and da Fonseca, 2002, Dumas et al., 1998, Dupire, 1994, Gatheral, 2011] (or, alternatively, on the moneyness $m = K/S_t$ and time-to-maturity $\tau = T - t$). However not every cross-sectional profile for the function $(m, \tau) \mapsto \sigma_t(m, \tau)$ is admissible, as the resulting call/put option prices should satisfy certain *static arbitrage constraints* [Davis and Hobson, 2007, Gerhold and Gülüm, 2020]. In particular call option prices should be:

- increasing in time to maturity: $\partial_\tau C_{BS}(S_t, K, \tau, \sigma_t(m, \tau)) \geq 0$,
- decreasing in moneyness: $\partial_m C_{BS}(S_t, K, \tau, \sigma_t(m, \tau)) \leq 0$,

- convex in moneyness: $\partial_m^2 C_{BS}(S_t, K, \tau, \sigma_t(m,\tau)) \geq 0$.

These constraints translate to nonlinear inequalities involving σ_t, $\partial_m \sigma_t$, $\partial_m^2 \sigma_t$, $\partial_\tau \sigma_t$ [Cont et al., 2002], which in turn impose constraints on the possible shapes of the implied volatility surface $\sigma_t(m,\tau)$.

Given a fixed grid in moneyness and time to maturity

$$(\boldsymbol{m}, \boldsymbol{\tau}) = (m_i, \tau_j)_{i=1,\ldots,N_m; j=1,\ldots N_\tau},$$

with $m_i < m_{i+1}$ and $\tau_j < \tau_{j+1}$, we define the relative call prices

$$c(m,\tau) := \frac{1}{S} C_{BS}(S, K, \tau, \sigma) = N(d_1) - m e^{-r\tau} N(d_2). \tag{1}$$

Static arbitrage constraints [Davis and Hobson, 2007] are then equivalent to

$$\Phi(\boldsymbol{\sigma}(\boldsymbol{m}, \boldsymbol{\tau})) = 0,$$

where $\Phi(\boldsymbol{\sigma}(\boldsymbol{m}, \boldsymbol{\tau}))$ is the *arbitrage penalty* associated with the (discretely sampled) volatility surface $\boldsymbol{\sigma}(\boldsymbol{m}, \boldsymbol{\tau})$ [Cont and Vuletic, 2023]:

$$\Phi\left(\boldsymbol{\sigma}(\boldsymbol{m}, \boldsymbol{\tau})\right) = p_1(\boldsymbol{\sigma}(\boldsymbol{m}, \boldsymbol{\tau})) + p_2(\boldsymbol{\sigma}(\boldsymbol{m}, \boldsymbol{\tau})) + p_3(\boldsymbol{\sigma}(\boldsymbol{m}, \boldsymbol{\tau})). \tag{2}$$

The quantities p_1, p_2, p_3 correspond to violations of calendar, call and butterfly arbitrage constraints, respectively:

$$p_1(\boldsymbol{\sigma}(\boldsymbol{m}, \boldsymbol{\tau})) = \sum_{i=1}^{N_m} \sum_{j=1}^{N_\tau} \left(\tau_j \frac{c(m_i, \tau_j) - c(m_i, \tau_{j+1})}{\tau_{j+1} - \tau_j} \right)^+, \tag{3}$$

$$p_2(\boldsymbol{\sigma}(\boldsymbol{m}, \boldsymbol{\tau}) = \sum_{i=1}^{N_m} \sum_{j=1}^{N_\tau} \left(\frac{c(m_{i+1}, \tau_j) - c(m_i, \tau_j)}{m_{i+1} - m_i} \right)^+, \tag{4}$$

$$p_3(\boldsymbol{\sigma}(\boldsymbol{m}, \boldsymbol{\tau})) = \sum_{i=1}^{N_m} \sum_{j=1}^{N_\tau} \left(\frac{c(m_i, \tau_j) - c(m_{i-1}, \tau_j)}{m_i - m_{i-1}} - \frac{c(m_{i+1}, \tau_j) - c(m_i, \tau_j)}{m_{i+1} - m_i} \right)^+. \tag{5}$$

2.2 Dynamics of implied volatility co-movements

Static arbitrage constraints on the shape of the implied volatility surface are a necessary but not sufficient requirement for a good model of implied volatility dynamics: one also needs the model to capture the statistical properties of implied volatility co-movements, a crucial point for any hedging and risk management task. Here we summarise some of the empirically observed statistical properties of implied volatilities on various exchange-traded indices [Cont and da Fonseca, 2002, Avellaneda et al., 2020, Cont and Vuletic, 2023]:

- The implied volatility has a non-flat cross-section, with dependence in strike and maturity.

- Implied volatilities display high positive autocorrelation and mean-reverting behavior.

- Daily variations in the implied volatilities can be satisfactorily explained with a small number of principal components.

- The first principal component corresponds to a *level*, whereas the second principal component corresponds to a *skew* factor.

- The returns of the underlying are negatively correlated with the projections of log-increments of implied vol on the *level* and *skew* principal components, which is a more precise formulation of the so-called 'leverage effect'.

We now describe a data-driven approach for the simulation of implied volatility dynamics designed to account for the above properties.

3 A generative model for implied volatility surfaces

VolGAN is a customised conditional generative adversarial network with a smoothness penalty incorporated into the generator's loss function, combined with scenario re-weighting applied to the output scenarios [Cont and Vuletic, 2023].

VolGAN receives as input

- the implied vol surface at the previous date,

- the two previous underlying returns,

- the realized volatility from the previous period,

and outputs (joint) scenarios for

- the return of the underlying asset and

- the implied volatility surface

for the next period, along with a set of weights (probabilities) associated with these scenarios. We now discuss the methodology in more detail.

3.1 Architecture

We design a Conditional GAN [Mirza and Osindero, 2014], composed of two neural networks, a *generator* and a *discriminator*. Suppose we have observations at times $t = 1, \dots, T$, with S_t the price of the underlying, and $\sigma_t(\boldsymbol{m}, \boldsymbol{\tau})$ the implied volatility surface on the grid $(\boldsymbol{m}, \boldsymbol{\tau})$ at time t. Denote by $g_t(\boldsymbol{m}, \boldsymbol{\tau})$ the log-implied vol surface at time t:

$$g_t(\boldsymbol{m}, \boldsymbol{\tau}) = \log \sigma_t(\boldsymbol{m}, \boldsymbol{\tau}), \quad \Delta g_t(\boldsymbol{m}, \boldsymbol{\tau}) = g_{t+1}(\boldsymbol{m}, \boldsymbol{\tau}) - g_t(\boldsymbol{m}, \boldsymbol{\tau}). \tag{6}$$

Let r_t be the log-return of the underlying:

$$r_t = \log\left(\frac{S_{t+1}}{S_t}\right), \tag{7}$$

and denote by γ_t the one-month realized volatility:

$$\gamma_t = \sqrt{\frac{21}{252}\sum_{i=0}^{20} r_{t-i}^2}. \tag{8}$$

We aggregate $r_{t-1}, t_{t-2}, \gamma_{t-1}, g_t(\boldsymbol{m}, \boldsymbol{\tau})$ into a *condition/input* vector a_t:

$$a_t = (r_{t-1}, t_{t-2}, \gamma_{t-1}, g_t(\boldsymbol{m}, \boldsymbol{\tau})). \tag{9}$$

The generator G takes as input this condition a_t and i.i.d. noise $z_t \sim \mathcal{N}(0, I_d)$ and outputs simulated values $\hat{r}_t(z), \Delta\hat{g}_t(\boldsymbol{m}, \boldsymbol{\tau})$ for the return and implied volatility (log-)increments:

$$G(a_t, z_t) = (\hat{r}_t(z_t), \Delta\hat{g}_t(\boldsymbol{m}, \boldsymbol{\tau})(z_t)). \tag{10}$$

We denote by $G(a_t, z)|_{2:} = \Delta\hat{g}_t(\boldsymbol{m}, \boldsymbol{\tau})(z)$ the second component of the generator's output which corresponds to the simulated log implied vol increment.

The discriminator is a classifier, taking as input a value $(r, \Delta g)$ representing either the output of the generator or the corresponding data realization, together with a condition vector a_t as in (9).

It outputs a value $D(a_t, (r, \Delta g))$ between 0 and 1, interpreted as the probability that the input is compatible with the condition i.e. the probability that $(r, \Delta g)$ it is drawn from the conditional distribution of $(r_t, \Delta g_t)$ given a_t originating from the data rather than from the generator.

The generator G and the discriminator D are feed-forward neural networks, whose respective parameters (weights) we denote by θ_g and θ_d. The architecture of the generator is displayed in Figure 1, and the architecture of the discriminator is shown in Figure 2.

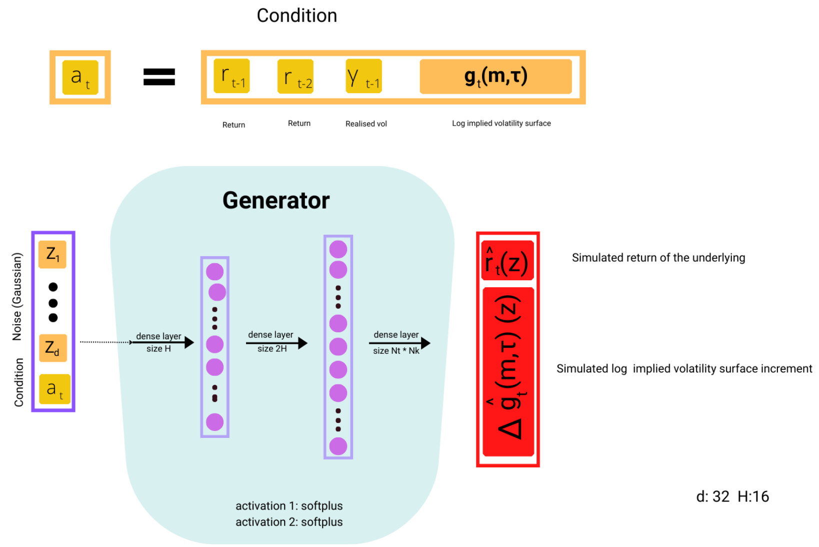


Figure 1: VolGAN generator architecture.

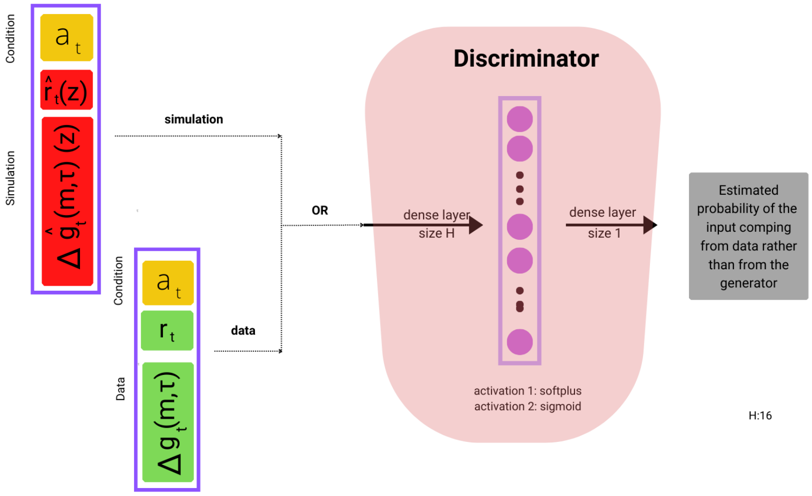


Figure 2: VolGAN discriminator architecture.

3.2 Training objective

The core component of VolGAN is a customised loss function catering to the desired properties of the output volatility surface, as advocated in [Cont et al., 2022, Vuletić et al., 2023]. A classical GAN trained using binary cross-entropy (BCE) loss [Goodfellow et al., 2014] would result in irregular surfaces. In order to generate smooth surfaces, we use smoothness penalty terms (discrete Sobolev semi-norm) in m and τ (on the grid $(\boldsymbol{m}, \boldsymbol{\tau})$), defined as

$$L_m(\boldsymbol{g}) = \sum_{i,j} \frac{\left(\boldsymbol{g}(m_{i+1}, \tau_j) - \boldsymbol{g}(m_i, \tau_j)\right)^2}{|m_{i+1} - m_i|^2} \simeq \|\partial_m g\|^2_{L^2}, \tag{11}$$

$$L_\tau(\boldsymbol{g}) = \sum_{i,j} \frac{\left(\boldsymbol{g}(m_i, \tau_{j+1}) - \boldsymbol{g}(m_i, \tau_j)\right)^2}{|\tau_{j+1} - \tau_j|^2} \simeq \|\partial_\tau g\|^2_{L^2}. \tag{12}$$

These terms are included in the training objective $J^{(G)}(\theta_d, \theta_g)$ for the generator:

$$\begin{aligned} J^{(G)}(\theta_d, \theta_g) = -\frac{1}{2}\mathbb{E}\big[& \log\left(D\left(a_t, G(a_t, z; \theta_g); \theta_d\right)\right) \\ & + \alpha_m L_m\left(g_t(\boldsymbol{m}, \boldsymbol{\tau}) + G(a_t, z_t; \theta_g)|_{2:}\right) \\ & + \beta_\tau L_\tau\left(g_t(\boldsymbol{m}, \boldsymbol{\tau}) + G(a_t, z_t; \theta_g)|_{2:}\right)\big], \end{aligned} \tag{13}$$

where $a_t = (r_{t-1}, t_{t-2}, \gamma_{t-1}, g_t(\boldsymbol{m}, \boldsymbol{\tau}))$, as defined in Equation (9). The first term is a binary cross-entropy for the output of the discriminator. $\alpha_m > 0$ and $\alpha_\tau > 0$ are regularisation parameters; iid Gaussian noise input $\boldsymbol{z}_t \sim N(0, I_d)$, a_t is the input "condition" (Eq. (9)); θ_g and θ_d are respectively the parameters (weights) of the generator and the discriminator networks. The smoothness penalties L_m and L_τ are applied to the simulated log-implied volatility surfaces:

$$g_t(\boldsymbol{m}, \boldsymbol{\tau}) + G(a_t, z_t; \theta_g)|_{2:} = g_t(\boldsymbol{m}, \boldsymbol{\tau}) + \Delta\hat{g}_t(\boldsymbol{m}, \boldsymbol{\tau})(z_t) = \hat{g}_t(\boldsymbol{m}, \boldsymbol{\tau})(z_t).$$

The discriminator is trained to minimise the binary cross-entropy loss:

$$\begin{aligned} J^{(D)}(\theta_d, \theta_g) = & -\frac{1}{2}\mathbb{E}\left[\log\left(D(a_t, (r_t, \Delta g_t(\boldsymbol{m}, \boldsymbol{\tau})); \theta_d\right)\right] \\ & -\frac{1}{2}\mathbb{E}\left[\log\left(1 - D(a_t, G(a_t, z_t; \theta_g); \theta_d\right)\right], \end{aligned} \tag{14}$$

where a_t is the input condition (Eq. (9)), r_t and $\Delta g_t(\boldsymbol{m}, \boldsymbol{\tau})$ are the corresponding data observations.

Here $\mathbb{P}$ represents the law of the process (r_t, g_t) generating the data. We assume this process to be ergodic, so given a long enough sample we can approximate the expected values above by sample averages:

$$\mathbb{E}[f(r_t, g_t)] \simeq \frac{1}{T}\sum_{t=1}^{T} f(r_t, g_t).$$

Remark 1 (Inclusion of the arbitrage penalty in the loss function)**.** *It is possible to incorporate the arbitrage penalty* (2) *into the loss function of the generator* (13)*. However, both the smoothness penalty* (11)*-*(12) *and the arbitrage penalty* (2) *impose restrictions on discrete derivatives. Our numerical experiments showed insignificant performance gain when incorporating the arbitrage penalty into the loss function, not justifying the additional computational cost.*

3.3 Scenario re-weighting

The outputs of the generator described above are not guaranteed to satisfy the static arbitrage constraints described in Section 2.1. To correct for this we apply the methodology described in [Cont and Vuletic, 2023] to re-weight the one-day-ahead scenarios generated by the GAN.

Let $\mathbb{P}_0$ be the law of the generator's output i.e. the joint dynamics of the underlying return and the implied volatility surface $(r_t, \sigma_t(\boldsymbol{m}, \boldsymbol{\tau}); t = 1, \ldots, T)$. To adjust for static arbitrage, [Cont and Vuletic, 2023] apply the change of measure:

$$\frac{d\mathbb{P}_\beta}{d\mathbb{P}_0}(\omega) = \frac{\exp\left(-\beta\Phi(\sigma(\boldsymbol{m}, \boldsymbol{\tau}; \omega))\right)}{Z(\beta)} \tag{15}$$

where $Z(\beta)$ is a normalization factor:

$$Z(\beta) = \mathbb{E}^{\mathbb{P}_0}\left[\exp\left(-\beta\Phi\left(\sigma(\boldsymbol{m}, \boldsymbol{\tau}; \omega)\right)\right)\right]. \tag{16}$$

VolGAN samples from this target distribution (15) using a Weighted Monte Carlo approach. Given N samples from the generator $(\hat{r}^i, \hat{\boldsymbol{\sigma}}^i)$, $i = 1, \ldots, N$, we compute the arbitrage penalty $\Phi(\hat{\boldsymbol{\sigma}}^i)$ corresponding to each output scenario $(\hat{r}^i, \hat{\boldsymbol{\sigma}}^i)$ using (2) and assign to this scenario a probability

$$w^i = \frac{\exp(-\beta\Phi(\hat{\boldsymbol{\sigma}}^i))}{\sum_{j=1}^N \exp(-\beta\Phi(\hat{\boldsymbol{\sigma}}^j))}. \tag{17}$$

VolGAN samples the scenarios $(\hat{r}^i, \hat{\boldsymbol{\sigma}}^i)$ with probability w^i. These weighted scenarios may then be used to compute expectations and quantiles of various quantities of interest under $\mathbb{P}_\beta$.

Let X be a function of the state variables, and let x_i be its value in scenario i. Denote by $F_{X,\beta}$ the law of X under $\mathbb{P}_\beta$ and by $\mathbb{E}_\beta[X]$ its expectation.

We can then estimate $\mathbb{E}_\beta[X]$ by

$$\widehat{\mathbb{E}_\beta[X]} = \sum_{i=1}^N w_i x_i, \tag{18}$$

while the quantiles of X are estimated as

$$\widehat{F_{X,\beta}^{-1}(q)} = x_{(k)}, \qquad \text{where} \quad k = \min\{j \in \{1, \ldots, N\} : \sum_{i=1}^j w_{(i)} \geq q\}, \tag{19}$$

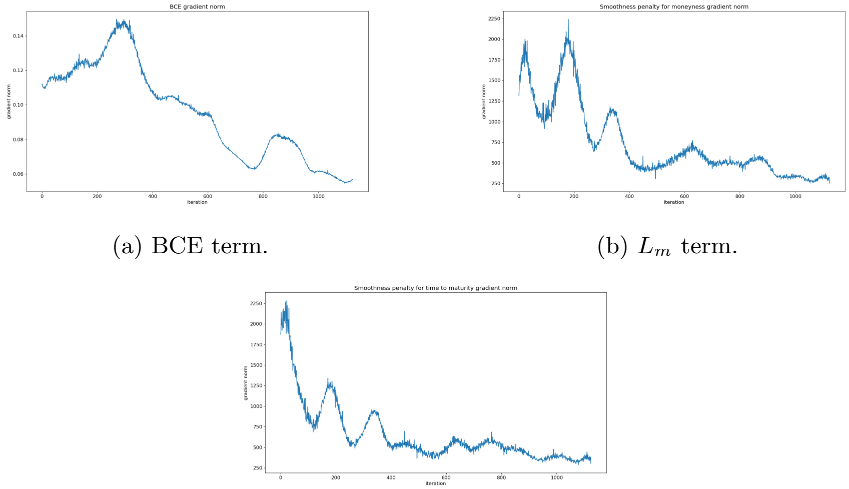
where $x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(N)}$ are the order statistics of $x_1, \ldots, x_N$. Quantile estimates allow us to estimate appropriate confidence intervals and tail behaviors, without making any distributional assumptions.

3.4 Numerical implementation

The generator G is a three-layer feedforward dense neural network, with the first two activations softplus, and the final layer an affine layer. The random input is (standard) i.i.d Gaussian noise with dimension $d = 32$. The first layer consists of $H = 16$ neurons, whereas the second layer contains $2H = 32$ neurons. Similarly, the discriminator D is a two-layer feedforward neural network, with softplus and sigmoidal activation functions and layer sizes of $H = 16$ and 1, respectively. The discriminator has a simpler architecture than the generator, as it is of the utmost importance to keep the two neural networks in balance. The architecture of the discriminator is shown in Figure 2, and the architecture of the generator is displayed in Figure 1.

The hyperparameters $\alpha_m, \alpha_\tau > 0$ are chosen by *gradient norm matching*. We first train VolGAN for $n_{grad} = 25$ epochs by performing optimisation via the binary cross-entropy loss only (classical GAN setting). At each update, we calculate the gradient norms of each of the three loss function terms in (13): BCE, L_m, $L_\tau$with respect to θ_g. We then set α_m and α_τ, to be the means of observed ratios of the gradient norms of the BCE term to the gradient norms of the L_m and L_τ, respectively. The gradient norms of the BCE, L_m, L_τ terms with respect to θ_g during this stage are shown in Figure 3. We note that all three gradients behave similarly, that they stabilise over time, and that there is no gradient explosion or vanishing gradient phenomena.

We then restart training VolGAN (from the same initialisation used for the start of the gradient norm matching procedure) with the loss function defined by Equation (13) for $n_{epochs} = 10000$ epochs, using an alternating direction method i.e. one discriminator update for each generator update. The optimiser used is RMSProp [Hinton et al., 2012], and the learning rates of both networks are set to 0.0001. We use $\beta = 1000$ and take $N = 10000$ raw samples from the generator. The mini-batch size is $n_{batch} = 100$.



(a) BCE term.

(b) L_m term.

(c) L_τ term.

Figure 3: Norm of gradient of the BCE term, L_m term, and L_τ term with respect to θ_g during the first stage of VolGAN training. Updates are performed via the BCE loss only.

Remark 2 (Data normalization)**.** *One should avoid data normalization when training VolGAN. It is not uncommon to do so if there are additional penalty terms placed in the loss function, for example in Wasserstein GANs with Gradient Penalty [Arjovsky et al., 2017, Gulrajani et al., 2017]. If the input condition is normalized, numerical experiments show that the generator converges to a collapsed market, i.e. it outputs implied vols and the underlying equal to zero. normalization places all information contained in the input condition on the same scale as the noise, rendering the problem of differentiating between different input components more challenging for the generator. However, mode collapse does not occur if the data is not normalized prior to training.*

4 Learning to simulate SPX implied volatility scenarios

In this section, we demonstrate VolGAN's ability to generate realistic scenarios for SPX implied volatility dynamics. We train VolGAN on the daily time series of market data *directly*, without making any modelling assumptions. The same approach might be applied to other equity options.

4.1 Data

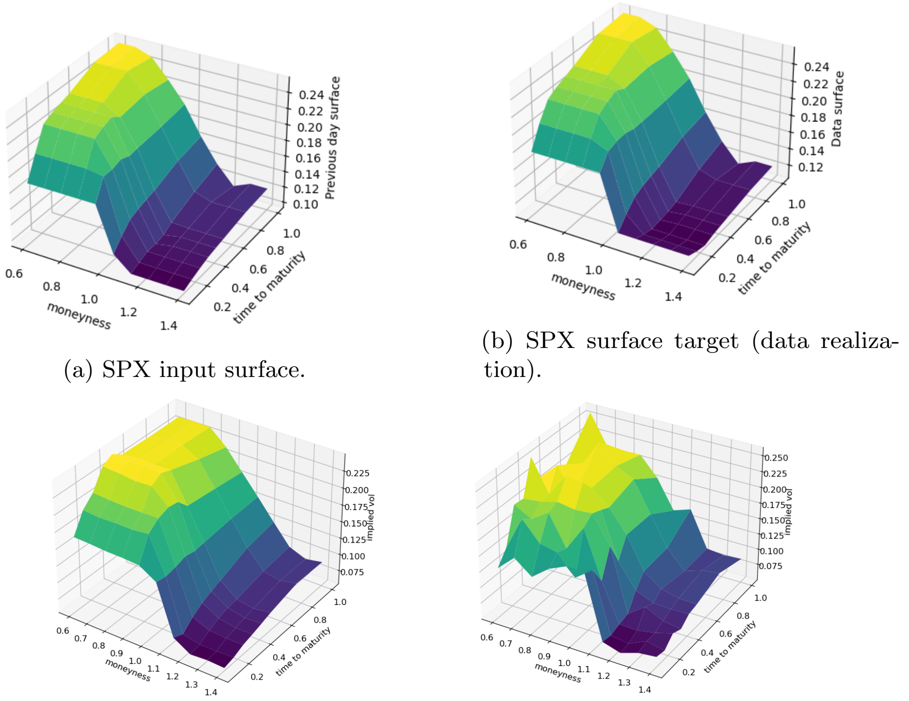
We consider the SPX implied volatility surface from the Implied Volatility Surface File from OptionMetrics on WRDS. The training period is from 2000-01-03 to 2017-08-08, and the testing period is from 2017-08-09 to 2019-10-21. We fix the grid $(\boldsymbol{m}, \boldsymbol{\tau})$ in moneyness and time to maturity such that the moneyness consists of 10 equispaced values between 0.6 and 1.4 and the times to expiry are 30, 60, 91, 122, 152, 182, 273, 365 calendar days. In order to reach the values on the $(\boldsymbol{m}, \boldsymbol{\tau})$ grid, we interpolate linearly first in moneyness, and then in times to maturity. Other methods of interpolation are also possible, for example, [Kahalé, 2004]. The Implied Volatility Surface File contains pre-processed data, which we interpolate further, resulting in non-zero arbitrage penalties in both the training and the testing set [Cont and Vuletic, 2023]. The historical VIX closing prices are available on the CBOE website. For simplicity, we set the interest rate $r = 0$ in the arbitrage penalty computation.

4.2 Out-of-sample performance

As discussed in Section 2, the main goal of an implied volatility model is to correctly capture the co-movements of implied volatilities, while satisfying static arbitrage constraints. We can measure the latter by considering the distance to arbitrage' using the arbitrage penalty (2). In order to measure how well VolGAN learns the dynamics and captures the co-movements of implied vols, we perform PCA on the generated increments, and compare them with the principal components of the data increments. Furthermore, we simulate the CBOE volatility index VIX [CBOE, 2022], which is a non-linear combination of tradable calls and puts. We compare the dynamics of the simulated and market data.

4.2.1 Smoothness and arbitrage constraints

Before discussing arbitrage in simulated scenarios, we note that training via the classical BCE loss [Goodfellow et al., 2014], using the same architecture, hyperparameters, and the same number of training epochs, would result in irregular surfaces. We compare the input, target (data realization), VolGAN and BCE GAN implied volatility sample surfaces in Figure 4. Incorporating the smoothness penalty (11)-(12) into the loss function (13) is crucial for generating smooth and realistic samples.



(a) SPX input surface.

(b) SPX surface target (data realization).

(c) Sample VolGAN output surface.

(d) Sample BCE GAN output surface.

Figure 4: Sample generated implied volatility surfaces.

As the input surfaces might admit static arbitrage, it is not realistic for the outputs to be completely arbitrage-free. What is plausible, however, is for the outputs to have lower arbitrage penalties than the inputs. In Table 1 we compare the out-of-sample arbitrage penalties in the input data and in the outputs of the BCE GAN, raw VolGAN (prior to re-weighting), and VolGAN. As training via the BCE loss results in irregular surfaces (Figure 4) it is unsurprising that the arbitrage penalty in the BCE GAN samples is high. There is a notable reduction in arbitrage penalty in the raw (before re-weighting) VolGAN outputs compared to the input data, and a significant decrease in arbitrage levels in the VolGAN samples (after scenario re-weighting). The mean arbitrage penalty pre- and post-scenario re-weighting, alongside the arbitrage penalty of the inputs, is displayed in Figure 5, where the reduction in arbitrage is clearly visualised. The mean, standard deviation, and median values from Table 1 correspond to the statistics of the time series displayed in Figure 5.

	Mean	**Std**	**Median**
SPX data	0.0059	0.0419	0
BCE GAN	0.2249	0.136	0.1893
Raw VolGAN (before reweighting)	0.0048	0.034	$1.13 \cdot 10^{-5}$
VolGAN	0.0022	0.020	$1.04 \cdot 10^{-6}$

Table 1: Arbitrage penalties in SPX implied volatility market data (test set) vs generated data via GANs trained using (i) BCE loss only (ii) VolGAN loss (iii) VolGAN re-weighted scenarios ($\beta = 1000$). Standard deviation and median for GAN outputs correspond to the standard deviation and the median of (re-weighted) average outputs given 10000 samples.

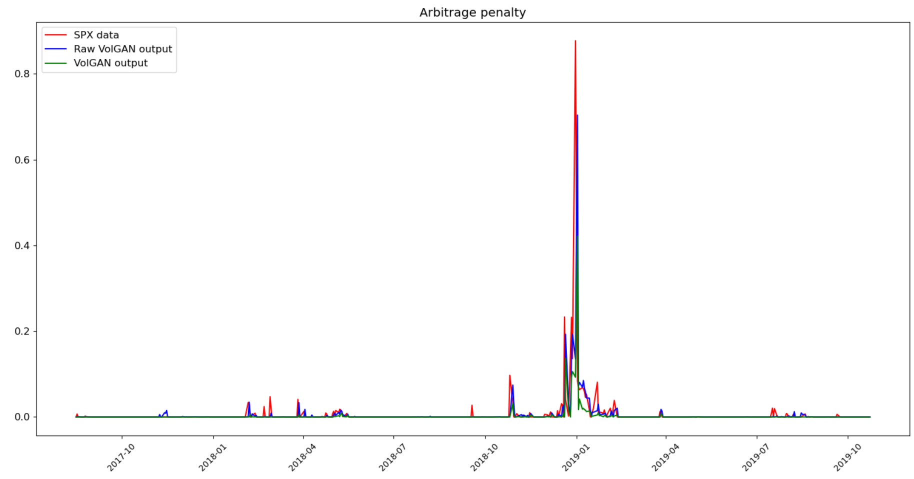


Figure 5: Distance to static arbitrage as measures by the arbitrage penalty (2) in SPX implied volatility data (red) vs. mean arbitrage penalty of surfaces generated via VolGAN, before (blue) and after (green) re-weighting with $\beta = 1000$.

4.2.2 Simulated vs realized values

We consider the daily simulated mean (VolGAN conditional expectation of the variable given the *history/condition*) and the central 95% confidence interval obtained by considering the 2.5% and 97.5% quantiles for the following quantities of interest:

- three-month at-the-money (ATM) implied vol $\sigma_t(1, 0.25)$,

- three-month out-of-the-money (OTM) implied vol $\sigma_t(1.25, 0.25)$,
- three-month in-the-money (ITM) implied vol $\sigma_t(0.75, 0.25)$,
- SPX price S_t,
- VIX price σ_t^{VIX}.

In Figure 6 we compare the realized and the simulated ATM implied vol, and note that the simulated confidence intervals capture the observed data realizations well. We find analogous results for the OTM implied vol and for the ITM implied vol, displayed in Figures 7 and 8, respectively. We note that as the value of the moneyness m increases, the confidence intervals become more narrow.

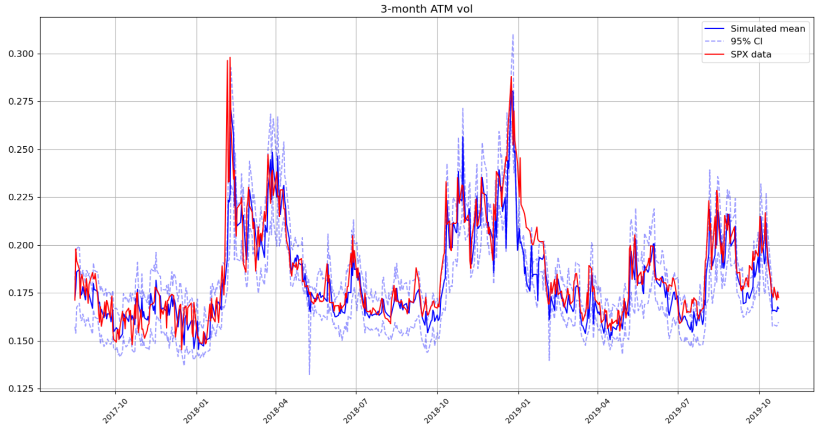


Figure 6: realized and simulated implied volatility ($m = 1, \tau = 0.25$) on the test set. Market data (red), simulated mean ($\mathbb{E}_\beta[\sigma_t(1, 0.25)|a_{t-1}]$) and the 95% confidence interval (blue). The confidence interval is calculated based on the 2.5% and 97.5% VolGAN quantiles.

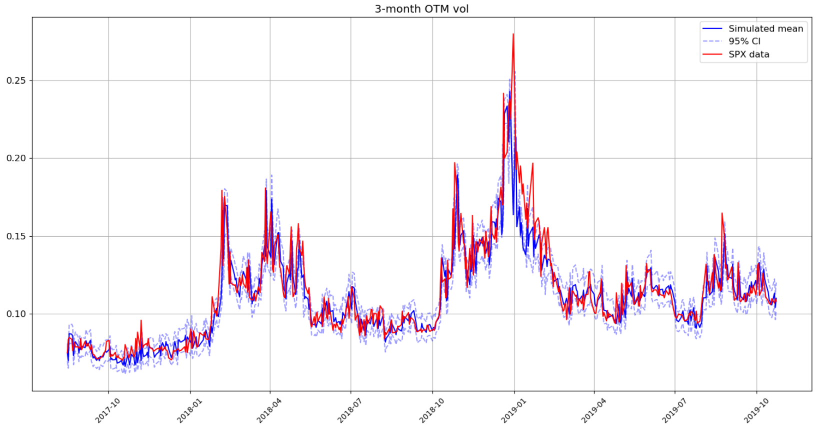


Figure 7: realized and simulated implied volatility ($m = 1.25, \tau = 0.25$) on the test set. Market data (red), simulated mean ($\mathbb{E}_{\beta}[\sigma_t(1.25, 0.25)|a_{t-1}]$) and the 95% confidence interval (blue). The confidence interval is calculated based on the 2.5% and 97.5% VolGAN quantiles.

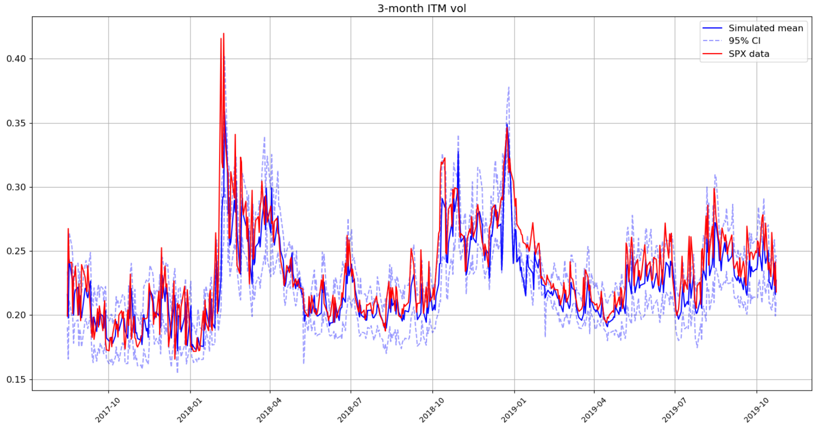


Figure 8: realized and simulated implied volatility ($m = 0.75, \tau = 0.25$) on the test set. Market data (red), simulated mean ($\mathbb{E}_{\beta}[\sigma_t(0.75, 0.25)|a_{t-1}]$) and the 95% confidence interval (blue). The confidence interval is calculated based on the 2.5% and 97.5% VolGAN quantiles.

Apart from the end of 2018 and early 2019, the VolGAN confidence intervals are able to appropriately capture the underlying. Figure 9 displays the simulated and real SPX prices. The time period during which VolGAN does not capture the data inside the confidence intervals coincides with a significant drop in SPX and a period of high arbitrage penalty (Figure 5). Given the input of VolGAN, it is unsurprising that the simulations revert to more *stable* values faster than the market does, as VolGAN is given only the recent history. Furthermore, the arbitrage penalty being high implies that a small number of simulations hold most of the weight, inducing very narrow confidence intervals. This behavior is visible not just in the simulations for the underlying, but for the ATM, OTM, and ITM vols (Figures 6, 7, 8 respectively). From Figure 10, we note that if arbitrage is not penalized ($\beta = 0$), the forecasts are more accurate, including for the early 2019 time period. However, choosing to use the raw generator induces static arbitrage.

Lastly, in Figure 11 we compare the simulated values of VIX with the historical closing prices on target days in the test set. Once again, VolGAN is able to capture properties of the time series of interest appropriately, apart from the end of 2017 when VIX was very low.

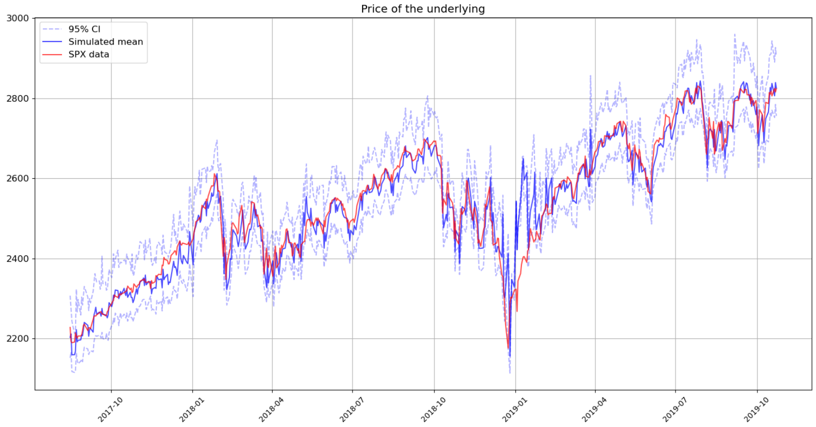


Figure 9: realized and simulated SPX price on the test set. Market data (red), simulated mean ($\mathbb{E}_\beta[S_t|a_{t-1}]$) and the 95% confidence interval (blue). The confidence interval is calculated based on the 2.5% and 97.5% VolGAN quantiles.

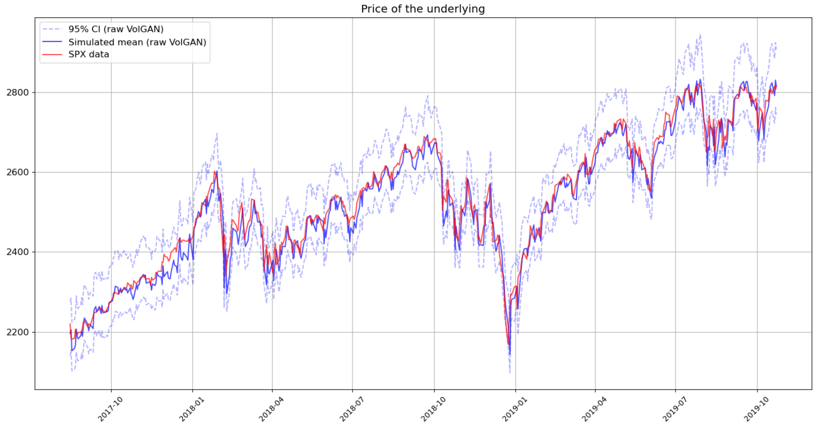


Figure 10: realized and simulated SPX price on the test set. Market data (red), simulated raw mean ($\mathbb{E}_0[S_t|a_{t-1}]$, with $\beta = 0$) and the 95% confidence interval (blue). The confidence interval is calculated based on the 2.5% and 97.5% quantiles.

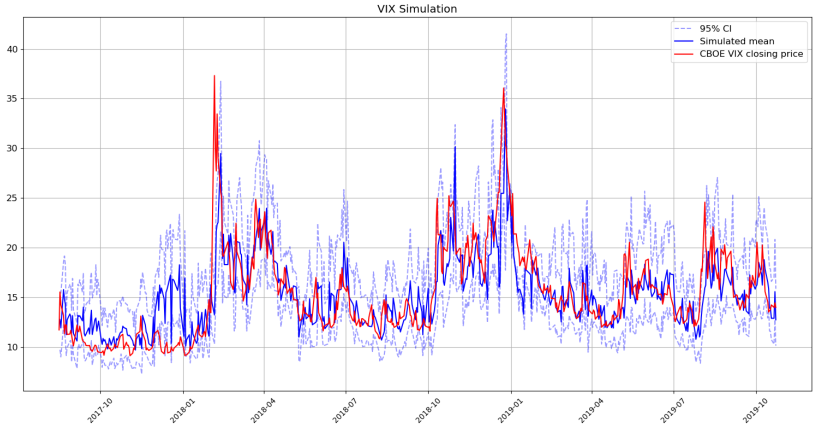


Figure 11: realized and simulated VIX on the test set. Market data (red), simulated mean ($\mathbb{E}_\beta\left[\sigma_t^{VIX}|a_{t-1}\right]$) and the 95% confidence interval (blue). The confidence interval is calculated based on the 2.5% and 97.5% VolGAN quantiles.

We further investigate the quality of simulated samples in Table 2 by considering the percentage of data realizations falling below the simulated 1%, 2.5%, 97.5%, and 99% quantiles. We note that the best fit of the right tail is for the underlying, whereas the best fit for the left tail is for the 3-month ATM and ITM vol. VolGAN underestimates the extremely high values of the implied vol and VIX, whereas for the out-of-the-money vol, VIX, and the underlying, the left-tail quantiles are too high. Given that the volatility index is a non-linear transformation of the state variables, it is not surprising that VolGAN underestimates and overestimates the right and the left tails of the distribution, respectively. The findings from Table 2 are in line with the previous observations: VolGAN underestimates the highest volatility values and overestimates the low SPX values, mainly stemming from the early 2019 and late 2018 period. It is important to note that the observed behavior is out-of-sample, two years after training.

Variable/Quantile	**0.01**	**0.025**	**0.975**	**0.99**
SPX	3.63%	4.36%	98.19%	99.09%
3-month ATM vol	0.91%	2.72%	81.49%	86.75%
3-month OTM vol	7.08%	9.80%	80.76%	84.94%
3-month ITM vol	1.45%	2.90%	78.58%	84.21%
VIX	13.61%	16.88%	74.95%	76.04%

Table 2: Percentage of data points below the simulated VolGAN quantiles on the test set (two years after training).

As already observed in Figure 10, there are instances (of market turbulence) where *not* correcting for the presence of static arbitrage (i.e. setting $\beta = 0$) actually *improves* forecasting performance. In Figure 12 we compare the simulated 95% confidence intervals of VolGAN with $\beta = 0$ and $\beta = 1000$ for SPX in late 2018 and early 2019.

Scenario penalization shifts the simulations towards higher returns of the underlying in the early 2019 period compared to the raw generator. When the arbitrage penalty is high, scenario re-weighting with $\beta = 1000$ very quickly concentrates the weights on the few simulated points with low arbitrage penalty, but away from the data realizations. However, we the raw generator ($\beta = 0$) produces very stable confidence intervals.

A similar effect is observed in the simulations of the 3-month ATM vol. In Figure 13 we compare the data realizations with the simulated confidence intervals for $\beta = 0$ and $\beta = 1000$ in the last quarter of 2018 and the first quarter of 2019. Once again, we note that when the arbitrage penalty is very low or zero, the penalization has negligible or no impact on the simulated confidence intervals. At the start of 2019, when the arbitrage penalty is high, the penalization favors smaller values of volatility compared to $\beta = 0$, an opposite shift to that observed in the simulations for the SPX returns (Figure 12). Once again, we note that the generator ($\beta = 0$) is very stable.

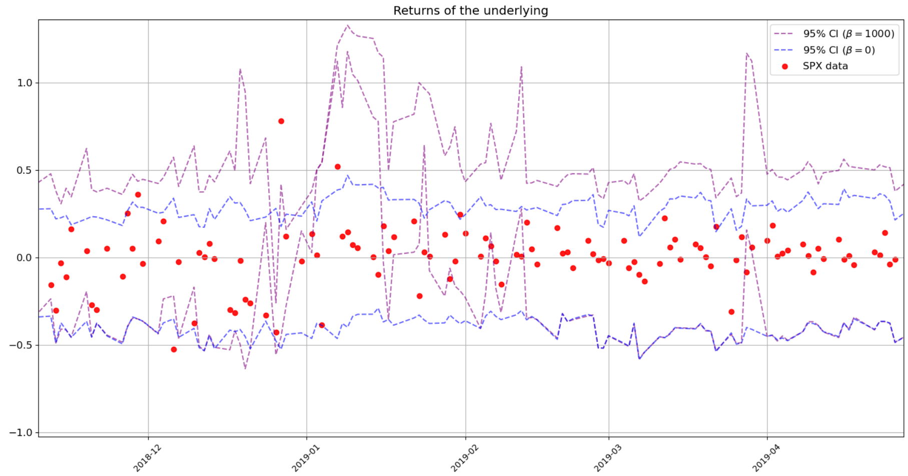


Figure 12: realized and simulated SPX returns in late 2018 and early 2019. Market data (red) with the 95% symmetric confidence intervals produced by the raw VolGAN generator ($\beta = 0$) (blue) and VolGAN ($\beta = 1000$) (purple). The confidence interval is calculated based on the 2.5% and 97.5% quantiles.

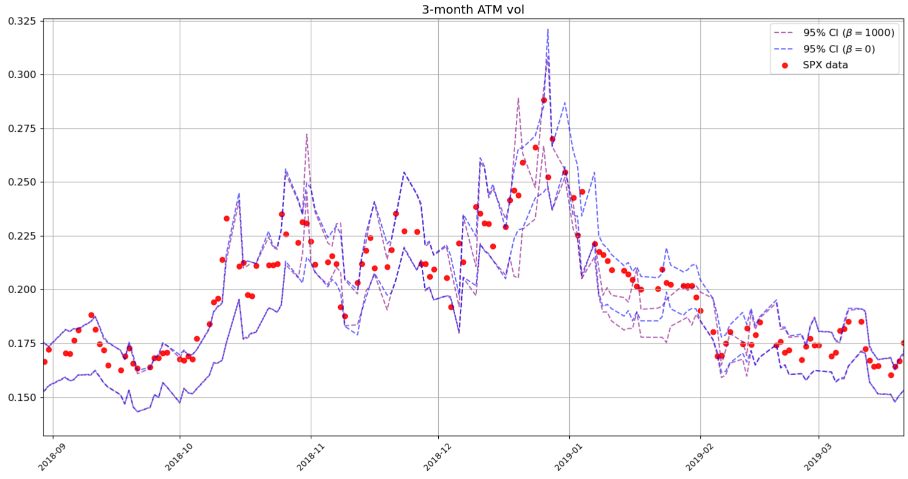


Figure 13: realized and simulated ATM vol on the test set. Market data (red) with the 95% symmetric confidence intervals produced by the raw VolGAN generator ($\beta = 0$) (blue) and VolGAN ($\beta = 1000$) (purple). The confidence interval is calculated based on the 2.5% and 97.5% quantiles.

To quantify the observations from Figures 12 and 13, we repeat the quantile analysis from Table 2 in the case of $\beta = 0$. Indeed, choosing to use the raw generator improves the forecasts. The most significant improvement is visible in the left-tail quantiles for the underlying, ATM vol, ITM vol, and the VIX. In the SPX and ATM vol case, the percentage of data realizations before the 1% and 2.5% quantiles are below those quantities, highlighting VolGAN's stability and that it does not require frequent re-calibration.

Variable/Quantile	**0.01**	**0.025**	**0.975**	**0.99**
SPX	0.36%	0.91%	98.19%	99.27%
3-month ATM vol	0.73%	2.18%	86.03%	92.38%
3-month OTM vol	8.34%	11.80%	84.57%	88.38%
3-month ITM vol	1.27%	2.90%	84.57%	89.84%
VIX	7.09%	7.63%	69.70%	71.87%

Table 3: Percentage of data points below the simulated raw generator ($\beta = 0$ quantiles on the test set (two years after training).

4.2.3 Distributions and correlations learned by the generator

Denote by ρ_t the instantaneous correlation between the 1-month ATM vol returns and the returns of the underlying at time t. We would like to explore whether or not VolGAN learns constant correlations. Therefore, we perform the following hypothesis test:

H_0**:** $\rho_t = \rho$ is constant, H_1**:** $\rho_t \neq \rho$ is time-varying.

Under H_0, the 95% confidence interval for ρ_t is given by $[\rho^L, \rho^U]$, where [Bonett and Wright, 2000]

$$\rho^U = \frac{\exp(2z_U) - 1}{\exp(2z_U) + 1}, \quad \rho^L = \frac{\exp(2z_L) - 1}{\exp(2z_L) + 1};$$

$$z_r = \frac{1}{2}\log\left[\frac{1+r}{1-r}\right], \quad z_U = z_\rho + \sqrt{\frac{1}{n-3}}z_{0.975}, \qquad z_U = z_\rho - \sqrt{\frac{1}{n-3}}z_{0.975},$$

where n is sample size. Estimating ρ by the sample mean of ρ_t on the test set, in Figure 14 we plot ρ_t and the 95% confidence interval $[\rho^L, \rho^U]$. We note that ρ_t is away from the confidence interval of H_0, indicating strong evidence against H_0. VolGAN learns time-varying instantaneous correlations which would be very difficult to capture with a parametric model.

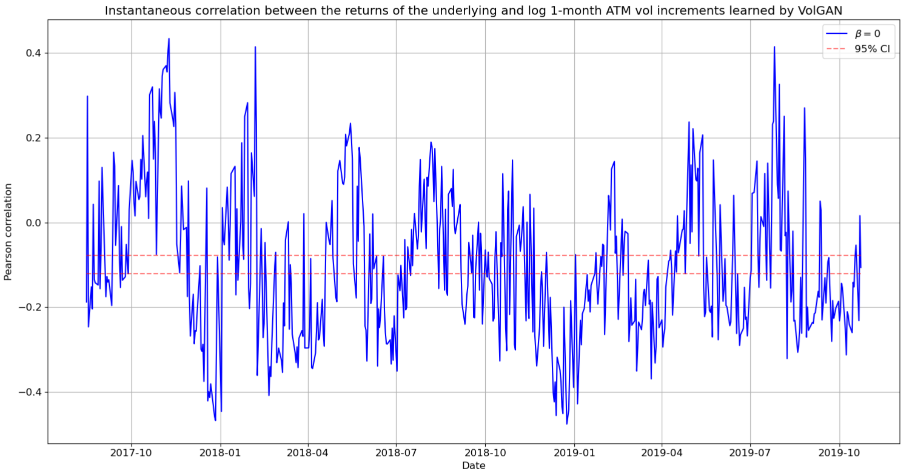


Figure 14: (Instantaneous) Pearson correlation on the test set between the simulated returns of the underlying and the 1-month ATM vol increments (blue), with symmetric 95% confidence interval of constant correlation (red). Raw generator ($\beta = 0$).

We explore the distributions of the simulated out-of-sample 1-month ATM vol returns and of the returns of the underlying. We compare both distributions with the distributions of the corresponding daily observations on the test set and with the Gaussian distributions with the appropriate mean and variance. From Figure 15, we note that VolGAN simulates returns of the underlying which have exponentially decaying tails, with high degrees of asymmetry, and a fat right tail. It is more conservative than the data, in the sense that it is able to produce values outside of the range of the test set data realizations. The same holds for the at-the-money vol returns in Figure 16, with the 1-month ATM vol returns having both left and right fat tails, and higher degrees of symmetry compared to the distribution of the simulated SPX returns. Such simulated distributions cannot be captured in a model with Brownian increments.

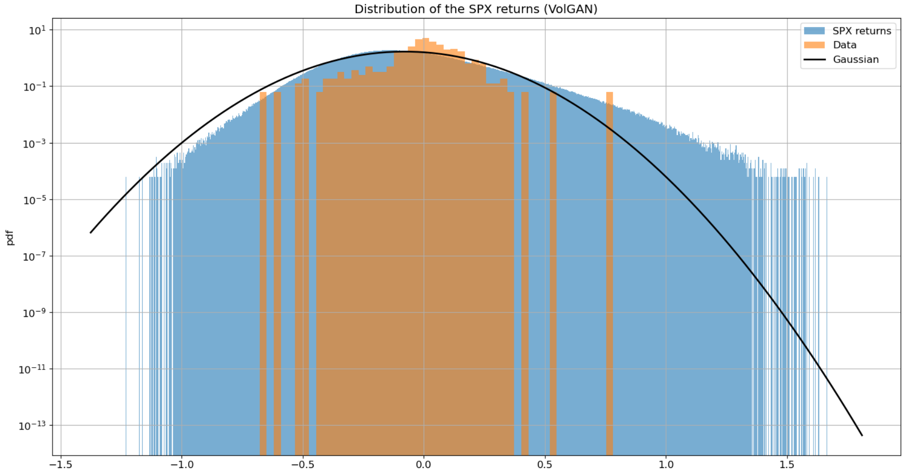


Figure 15: Simulated increments of the underlying (blue) on the test set compared to the Gaussian distribution with the same mean and variance (solid black line) and to the test set data (orange). Simulated returns have exponentially decaying tails, with high degrees of asymmetry. The range of simulated values captures the extreme values observed in the data. Raw generator ($\beta = 0$).

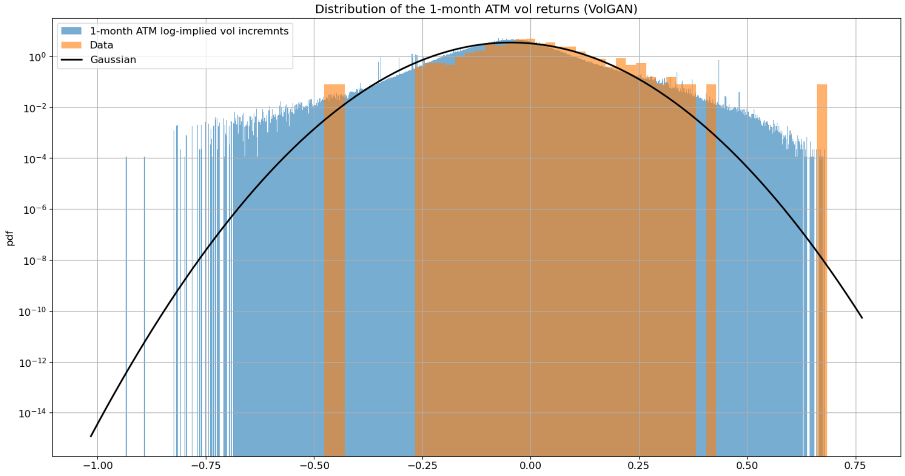


Figure 16: Simulated 3-month ATM vol increments (blue) on the test set compared to the Gaussian distribution with the same mean and variance (solid black line) and to the test set data (orange). Simulated returns have exponentially decaying fat tails. The range of simulated values captures the extreme values observed in the data. Raw generator ($\beta = 0$).

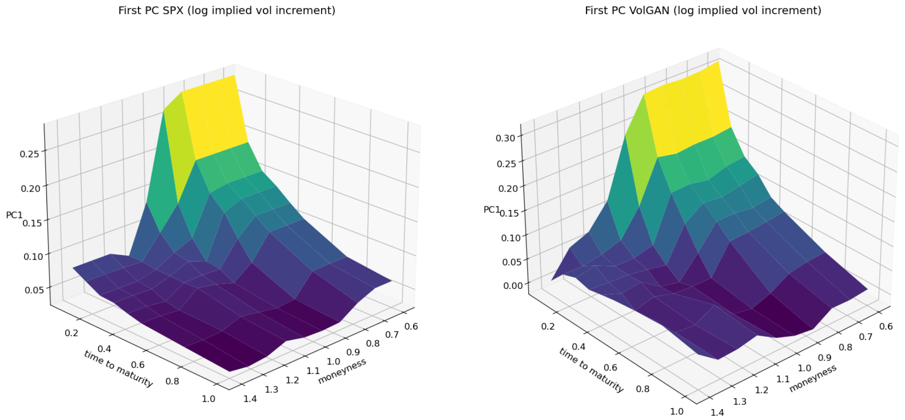
4.2.4 Principal component analysis

In order to investigate VolGAN's ability to appropriately capture the implied volatility co-movements, we perform out-of-sample principal component analysis on the simulated log-increments of implied vol. We compare the first three simulated principal components with the corresponding PCs of the data realizations. When performing PCA on two years of SPX implied vol data, the eigenvectors change depending on the period of observation, but nonetheless correspond to *level, skew* and *curvature*. In Table 4 we show variance explained by the first three eigenvectors in the testing data and in the VolGAN simulations. The significance of the first principal component is very similar in the test data and in VolGAN. The second principal component is more significant in the simulated data compared to the market data, and the third principal component explains more variance in simulations than it does in the SPX data.

Rank	Data	VolGAN
First	47.5%	$49.6 \pm 3.4\%$
Second	20.2%	$28.4 \pm 2.5\%$
Third	2.7%	$11.6 \pm 1.3\%$

Table 4: Out-of-sample (two years after training) percentage of variance explained by the top three principal components of the simulated and the data log implied vol increments. The VolGAN column contains the average $\pm 1.96\times$ standard deviation of the observed values, across 1000 VolGAN samples.

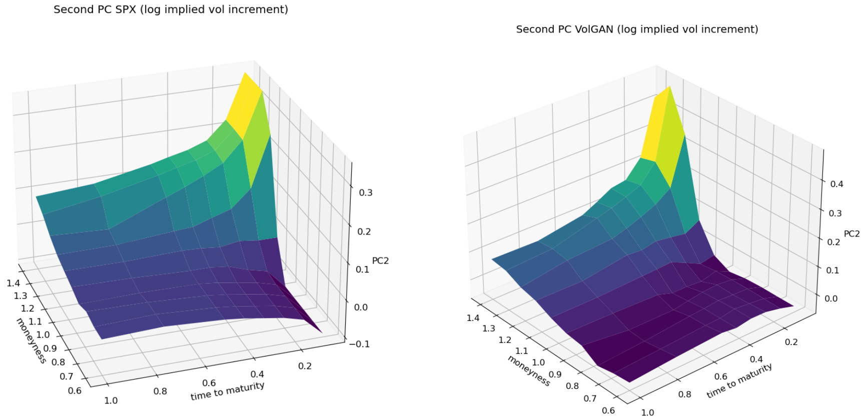
The first principal components of the sample VolGAN implied vol log-returns and of the corresponding SPX data are displayed in Figure 17. Both surfaces are consistently positive, indicating that they might have a *level* interpretation. They also exhibit *term-structure*. The second eigenvectors of both SPX data and of the simulated scenarios (Figure 18) can be interpreted as *skew*, while the third eigenvectors (Figure 19) can be interpreted as *curvature*. Figures 17, 18, 19 reflect on the clear resemblance between the principal components of the SPX market data and of the VolGAN simulations, showing that VolGAN is able to dynamically learn the covariance structure of the co-movements.



(a) SPX data.

(b) Sample VolGAN output.

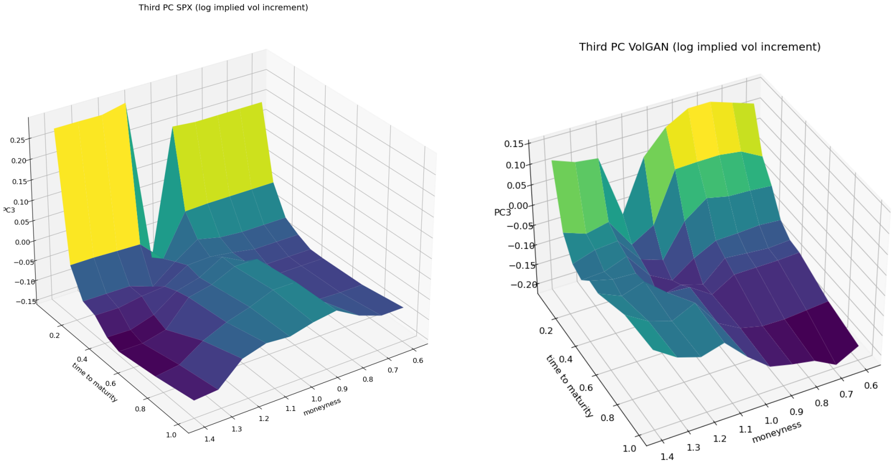
Figure 17: Out-of-sample (two years after training) first principal component of the daily log implied vol increments.



(a) SPX data.

(b) Sample VolGAN output.

Figure 18: Out-of-sample (two years after training) second principal component of the daily log implied vol increments.



(a) SPX data.

(b) Sample VolGAN output.

Figure 19: Out-of-sample (two years after training) third principal component of the daily log implied vol increments.

In order to quantify the similarity between the PCs of the simulated and the market data, we calculate the inner product between them (as vectors) over 1000 i.i.d. VolGAN samples. A value of one would indicate perfect alignment of the eigenvectors. From Table 5 we note that the first two inner products (PC1 with PC1, and PC2 with PC2) are very close to one, especially considering that the quantities are for the out-of-sample data. The inner product between the third eigenvectors of simulations and data realizations is lower than for the first two PCs, but it is nevertheless high. Therefore, VolGAN is able to learn the most important eigenvectors both qualitatively and quantitatively, showing the ability to learn the covariance structure of the SPX implied vol co-movements.

Rank	Mean	Standard deviation
First	0.943	0.015
Second	0.934	0.013
Third	0.714	0.015

Table 5: Out-of-sample inner products of eigenvectors of the covariance matrices of daily log-returns of SPX implied volatility and the corresponding eigenvectors of the covariance matrix of VolGAN implied vol increments.

4.2.5 Correlation structure of variables

We further investigate VolGAN's ability to simulate realistic scenarios by considering Pearson correlation between several variables of interest. First, we consider the relationship between the projections of the log-implied vol increments onto the first three principal components and the log-returns of the underlying. In Table 6 we compare the correlations in SPX data with those in scenarios simulated by VolGAN. The correlation between the first projection process and the simulated log-returns of SPX is close to that of market data, whereas the projections on the second and the third principal component have stronger correlations with the returns of the underlying in VolGAN than they do in the market data. We note that the observed data correlations are different on the test set and the training set (performing PCA on the entire training set). [Cont and Vuletic, 2023] show that the corresponding SPX data correlations over a 2-year window vary significantly, but do not change the sign. VolGAN is able to reproduce the correct relationships between the projection processes and the returns of the underlying: the correlations between the returns of the underlying and the projections of the log implied vol increments onto the level and skew principal component are negative, whereas the correlation with the projection onto the curvature principal component is positive.

PC rank	Data (test)	VolGAN (test)	Data (train)
First	-0.63	-0.56 ± 0.088	-0.34
Second	-0.15	-0.53 ± 0.079	-0.32
Third	0.26	0.53 ± 0.055	0.28

Table 6: Pearson correlation between (simulated) SPX log-returns and the projections of the (simulated) log-implied vol increments on the principal components. The VolGAN column contains the mean ±1.96× standard deviation of the observed Pearson correlations across 1000 samples.

In order to correctly capture joint dynamics of implied volatilities and the underlying index, we are interested in the relationship between the log increments of the index ($\Delta \log S_t$), the projection of the log-implied vol increments onto the first principal component (ΔX_t^1), the log increments of the 1-month at-the-money implied volatility ($\Delta \log \sigma_t^{ATM}$), and the log increments of VIX ($\Delta \log v_t$). Table 7 contains average Pearson correlations for VolGAN simulations (blue) vs the market data (red) on the test set. VolGAN simulations exhibit similar correlations between the volatility variables (ΔX_t^1, $\Delta \log \sigma_t^{ATM}$, $\Delta \log v_t$) and as previously discussed, between $\Delta \log S_t$ and ΔX_t^1. The correlations between $\Delta \log S_t$ and the log increments of at-the-money vol and VIX are lower in VolGAN scenarios than in the data observation on the test set. They are of the correct sign, and as we note once again from [Cont and Vuletic, 2023], the correlation between $\Delta \log S_t$ and $\Delta \log \sigma_t^{ATM}$ became significantly higher in magnitude in the period used for testing compared to the period used for training. Correlation between $\Delta \log S_t$ and $\Delta \log v_t$ in VolGAN simulations is similar to that produced by the PCA-based four-factor model with fixed correlations in [Cont and Vuletic, 2023].

	$\Delta \log S_t$	ΔX_t^1	$\Delta \log \sigma_t^{ATM}$	$\Delta \log v_t$
$\Delta \log S_t$	1.00	-0.56 -0.63	-0.18 -0.58	-0.35 -0.81
ΔX_t^1	-0.56 -0.62	1.00	0.87 0.94	0.68 0.74
$\Delta \log \sigma_t^{ATM}$	-0.18 -0.57	0.87 0.94	1.00	0.53 0.70
$\Delta \log v_t$	-0.35 -0.81	0.68 0.74	0.53 0.70	1.00

Table 7: Out-of-sample (2 years after training) average Pearson correlation for simulated vs real values of log-returns of SPX ($\Delta \log S_t$), implied vol level factor (ΔX_t^1), 1-month ATM vol ($\Delta \log \sigma_t^{ATM}$) and VIX ($\Delta \log v_t$). Average VolGAN outcome (blue) and data (red).

Our results demonstrate that VolGAN is able to simulate realistic co-movements for implied volatilities across a range of moneyness and maturities, as well as the underlying index and VIX: in particular we are able to reproduce time-varying correlations between increments of these variables.

Code availability

VolGAN code is available on GitHub: https://github.com/milenavuletic/VolGAN/.

Data availability

SPX options ata is available from OptionMetrics. VIX times series is available from CBOE (`www.cboe.com/`).

Funding

Milena Vuletić's research is supported by BNP Paribas through the *EPSRC Centre for Doctoral Training in Mathematics of Random Systems: Analysis, Modelling and Simulation* (ESPRC Grant EP/S023925/1).

Acknowledgements

We thank Katia Babbar, Andrey Chirikhin, Samuel N. Cohen, Mihai Cucuringu, Bruno Dupire, Blanka Horvath, Botao Li, Terry Lyons, Fabio Mercurio, Christoph Reisinger, Justin Sirignano and seminar participants at BNP Paribas PhD Days 2023, QuantMinds 2023 and Quant Summit Europe 2023 for helpful comments and remarks.

References

[Arjovsky et al., 2017] Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein Generative Adversarial Networks. In Precup, D. and Teh, Y. W., editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223. PMLR.

[Avellaneda et al., 2020] Avellaneda, M., Healy, B., Papanicolaou, A., and Papanicolaou, G. (2020). PCA for Implied Volatility Surfaces. *The Journal of Financial Data Science*, 2(2):85–109.

[Babbar, 2001] Babbar, K. A. (2001). *Aspects of stochastic implied volatility in financial markets*. PhD thesis, Imperial College London.

[Bonett and Wright, 2000] Bonett, D. G. and Wright, T. A. (2000). Sample size requirements for estimating pearson, kendall and spearman correlations. *Psychometrika*, 65:23–28.

[Cao et al., 2020] Cao, J., Chen, J., and Hull, J. (2020). A neural network approach to understanding implied volatility movements. *Quantitative Finance*, 20(9):1405–1413.

[CBOE, 2022] CBOE (2022). Volatility Index Methodology: Cboe Volatility Index. `https://cdn.cboe.com/api/global/us_indices/governance/VIX_Methodology.pdf`. [Online; accessed 8-May-2023].

[Cohen et al., 2022] Cohen, S., Reisinger, C., and Wang, S. (2022). Estimating risks of european option books using neural-sde market models. *Journal of Computational Finance*, 6(3):33–72.

[Cont et al., 2022] Cont, R., Cucuringu, M., Xu, R., and Zhang, C. (2022). Tail-GAN: Nonparametric Scenario Generation for Tail Risk Estimation. *arXiv preprint arXiv:2203.01664*.

[Cont and da Fonseca, 2002] Cont, R. and da Fonseca, J. (2002). Dynamics of implied volatility surfaces. *Quant. Finance*, 2(1):45–60.

[Cont et al., 2002] Cont, R., Fonseca, J. d., and Durrleman, V. (2002). Stochastic models of implied volatility surfaces. *Economic Notes*, 31(2):361–377.

[Cont and Tankov, 2004] Cont, R. and Tankov, P. (2004). *Financial modelling with jump processes*. CRC Press.

[Cont and Vuletic, 2023] Cont, R. and Vuletic, M. (2023). Simulation of arbitrage-free implied volatility surfaces. *Applied Mathematical Finance*, 30.

[Cuchiero et al., 2020] Cuchiero, C., Khosrawi, W., and Teichmann, J. (2020). A generative adversarial network approach to calibration of local stochastic volatility models. *Risks*, 8(4):101.

[Davis and Hobson, 2007] Davis, M. H. and Hobson, D. G. (2007). The range of traded option prices. *Mathematical Finance*, 17(1):1–14.

[Dumas et al., 1998] Dumas, B., Fleming, J., and Whaley, R. E. (1998). Implied volatility functions: Empirical tests. *The Journal of Finance*, 53(6):2059–2106.

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

[Durrleman, 2010] Durrleman, V. (2010). From implied to spot volatilities. *Finance and Stochastics*, 14:157–177.

[Gatheral, 2011] Gatheral, J. (2011). *The volatility surface: a practitioner's guide*. John Wiley & Sons.

[Gerhold and Gülüm, 2020] Gerhold, S. and Gülüm, I. C. (2020). Consistency of option prices under bid–ask spreads. *Mathematical Finance*, 30(2):377–402.

[Goodfellow et al., 2014] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

[Gulrajani et al., 2017] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. (2017). Improved Training of Wasserstein GANs. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30.

[Heston, 1993] Heston, S. L. (1993). A closed-form solution for options with stochastic volatility with applications to bond and currency options. *The review of financial studies*, 6(2):327–343.

[Hinton et al., 2012] Hinton, G., Srivastava, N., and Swersky, K. (2012). Neural Networks for Machine Learning, Lecture 6. Coursera.

[Kahalé, 2004] Kahalé, N. (2004). An arbitrage-free interpolation of volatilities. *Risk*, 17(5):102–106.

[Kingma et al., 2019] Kingma, D. P., Welling, M., et al. (2019). An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392.

[Mirza and Osindero, 2014] Mirza, M. and Osindero, S. (2014). Conditional Generative Adversarial Nets.

[Ning et al., 2023] Ning, B. X., Jaimungal, S., Zhang, X., and Bergeron, M. (2023). Arbitrage-free implied volatility surface generation with variational autoencoders. *SIAM Journal on Financial Mathematics*, 14(4):1004–1027.

[Schönbucher, 1999] Schönbucher, P. J. (1999). A market model for stochastic implied volatility. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 357(1758):2071–2092.

[Takahashi et al., 2019] Takahashi, S., Chen, Y., and Tanaka-Ishii, K. (2019). Modeling financial time-series with generative adversarial networks. *Physica A: Statistical Mechanics and its Applications*, 527:121261.

[Vuletić et al., 2023] Vuletić, M., Prenzel, F., and Cucuringu, M. (2023). Fin-gan: Forecasting and classifying financial time series via generative adversarial networks. *Quantitative Finance*, to appear.

[Wiese et al., 2019] Wiese, M., Bai, L., Wood, B., and Buehler, H. (2019). Deep hedging: learning to simulate equity option markets. *arXiv preprint arXiv:1911.01700*.

[Wiese et al., 2020] Wiese, M., Knobloch, R., Korn, R., and Kretschmer, P. (2020). Quant GANs: deep generation of financial time series. *Quantitative Finance*, 20(9):1419–1440.

[Wiese et al., 2021] Wiese, M., Wood, B., Pachoud, A., Korn, R., Buehler, H., Murray, P., and Bai, L. (2021). Multi-Asset Spot and Option Market Simulation.