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

Milena Vuletić and Rama Cont
Mathematical Institute
University of Oxford

Bayes Business School, 2nd April 2025

University of Oxford
Mathematical Institute

Centre for Doctoral Training
Mathematics of Random Systems

Implied volatility surfaces

- Implied volatility of a call option with moneyness $m = K/S_t$ and time-to-maturity $\tau = T - t$ 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:

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

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

- $\sigma_t : (m,\tau) \mapsto \sigma_t(m,\tau)$ is the implied volatility surface at time t.
- Here $m = K/S_t$ is the moneyness, $\tau = T - t$ is time to maturity
- Denote by $c(m,\tau) := \frac{1}{S} C_{BS}(S, K, \tau, \sigma)$ the relative call price.

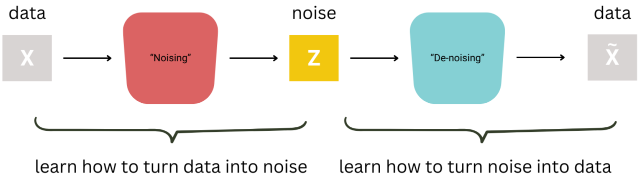
- The **implied volatility surface** has a nonlinear state space and complex dynamics.
- Absence of static arbitrage leads to nonlinear shape constraints on the implied volatility surface, difficult to manage in parametric models.
- On a fixed grid $(\boldsymbol{m}, \boldsymbol{\tau}) = (m_i, \tau_j)_{i=1,\ldots,N_m; j=1,\ldots N_\tau}$ in moneyness/ time to maturity, implied volatility $\sigma(m, \tau)$ and relative call price $c(m, \tau) = C^{BS}(m, \tau)/S_t$, there is no static arbitrage if and only if $\Phi\left(\boldsymbol{\sigma}(\boldsymbol{m}, \boldsymbol{\tau})\right) = 0$, where Φ is the *arbitrage penalty* (Cont and Vuletić 2022)
- Can a data-driven generative model overcome these challenges?

GenAI in a nutshell

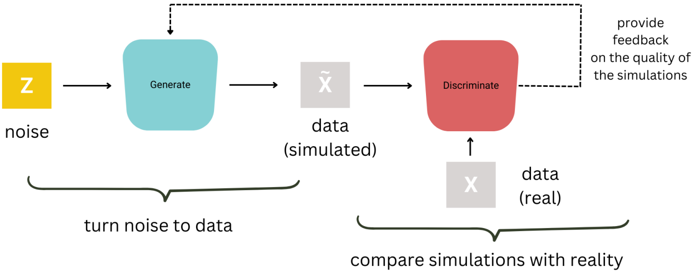
UNIVERSITY OF OXFORD
Mathematical Institute

- We associate generative models with LLMs and ChatGPT, but they are a much broader range of models than just that.
- ChatGPT: "A generative model is a type of probabilistic model used in statistics and machine learning with the aim of modeling the joint probability distribution $\mathbb{P}(X|Y)$ over observed data X and potential labels or latent variables Y. This type of model is capable of generating new samples from the estimated distribution, mimicking the properties of the real data."
- Generative models are essentially simulation tools.
- Classical modelling approaches are also generative!

- Map the observed data to noise, and then map the noise back to the data.
- Example: diffusion models, Variational Autoencoders...
- Simulate new data by resampling noise and applying the 'de-noising' map.



- Start from the noise and map it to data. Validate the quality of the output simulation → adversarial training.
- Simulate new data by resampling the input noise of the generator.



VolGAN: model details

- VolGAN is a customised **conditional** generative adversarial network (GAN) that simulates next-day scenarios for implied vol and underlying log-returns, given today's implied vols and spot. It is fully data-driven: it learns from market data directly, with no model assumptions.
- The inputs are the two previous log-returns of the underlying at steps $t - \Delta t$ and $t - 2\Delta t$, the one-month realised volatility and the implied volatility surface at time $t - \Delta t$.
- The generator outputs the return of the underlying and the implied volatilities for the next period (t).
- The discriminator checks the compatibility of the simulated scenario with the history.
- The two core components of VolGAN are the tailored loss function and arbitrage penalty scenario re-weighting.

- Suppose that we have data observations at times $t = 1, \dots, T$. Denote by $g_t(\boldsymbol{m}, \boldsymbol{\tau})$ and by $\Delta g_t(\boldsymbol{m}, \boldsymbol{\tau})$ the implied volatility surface at time t, and the corresponding increment:

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

- Let r_t be the (log) returns of the underlying, and γ_t the one-month realised volatility:

$$R_t = \log\left(\frac{S_{t+1}}{S_t}\right), \quad \gamma_t = \sqrt{\frac{21}{252}\sum_{i=0}^{20} R_{t-i}^2}.$$

- Aggregate $R_{t-1}, R_{t-2}, \gamma_{t-1}, g_t(\boldsymbol{m}, \boldsymbol{\tau})$ into a single input condition vector a_t:

$$a_t = (R_{t-1}, R_{t-2}, \gamma_{t-1}, g_t(\boldsymbol{m}, \boldsymbol{\tau})).$$

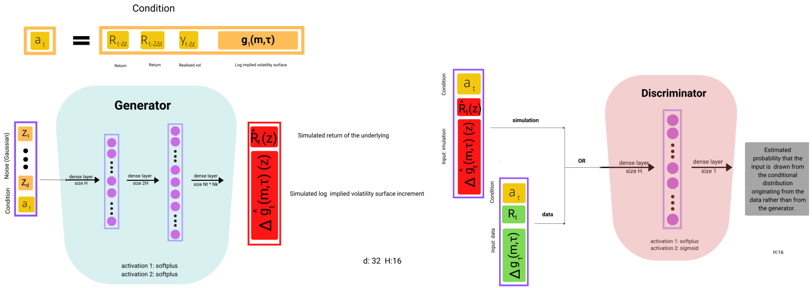
- The generator G takes as input this condition a_t and i.i.d. noise $z_t \sim \mathcal{N}(0, I_d)$ and outputs simulated values $\hat{R}_t(z), \Delta\hat{g}_t(\boldsymbol{m}, \boldsymbol{\tau})$ for the return and implied volatility (log-)increments:

$$G(a_t, z_t) = (\hat{R}_t(z_t), \Delta\hat{g}_t(\boldsymbol{m}, \boldsymbol{\tau})(z_t)),$$

$$G(a_t, z)|_{2:} = \Delta\hat{g}_t(\boldsymbol{m}, \boldsymbol{\tau})(z).$$

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

Architectures: illustrated



UNIVERSITY OF OXFORD
Mathematical Institute

Training via the (classical) BCE loss leads to irregular surfaces.
To generate smooth surfaces, we include a smoothness norm (discrete Sobolev semi-norm) in the objective for the generator:

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

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

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

where a_t is the input condition, $g_t(m,\tau)$ log implied volatility surface on day t, and $G(a_t,z_t;\theta_g)|_{2:}$ is the corresponding simulated implied volatility increment.
The discriminator is a classifier trained classically, via the BCE loss.

Penalisation via scenario reweighting

Cont and Vuletić (2022)

- 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 $(R_t, \sigma_t(\boldsymbol{m}, \boldsymbol{\tau}); t = 1, \dots, T)$. To adjust for static arbitrage, we apply the change of measure

$$\frac{d\mathbb{P}_\beta}{d\mathbb{P}_0}(\omega) = \frac{\exp\left(-\beta\Phi(\sigma(\boldsymbol{m}, \boldsymbol{\tau}; \omega))\right)}{Z(\beta)}, \tag{1}$$

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

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

$$w^i = \frac{\exp(-\beta\Phi(\hat{\boldsymbol{\sigma}}^i))}{\sum_{j=1}^N \exp(-\beta\Phi(\hat{\boldsymbol{\sigma}}^j))}. \tag{2}$$

Learning to simulate SPX implied volatility scenarios

- Consider the SPX implied volatility surface calculated by applying a Nadaraya-Watson kernel smoothing to the implied volatility data from OptionMetrics.
- Training: 2000-01-03 to 2018-07-16, testing: 2018-07-18 to 2023-02-28.
- Our grid $(\boldsymbol{m}, \boldsymbol{\tau})$ consists of

$$m \in \{0.6, 0.7, 0.8, 0.9, 0.95, 1, 1.05, 1.1, 1.2, 1.3, 1.4\},$$

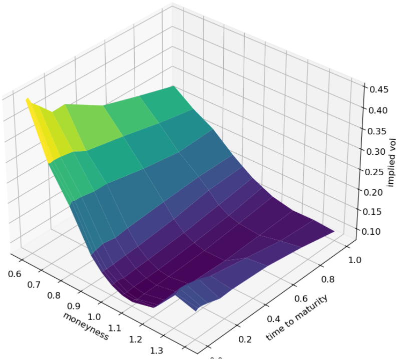
$$\tau \in \{\frac{1}{252}, \frac{1}{52}, \frac{2}{52}, \frac{1}{12}, \frac{1}{6}, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\}.$$

- When using black-box models which are trained on market data, it is *crucial* to properly assess model performance and understand what the model learns.

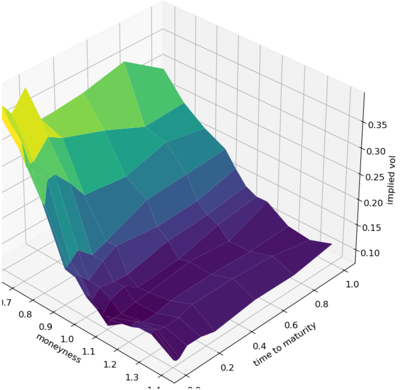
- How does the arbitrage penalty of the outputs compare to the arbitrage penalty of the inputs?
- Are the realised values in the range of the simulated values?
- Does the simulated dynamics produce the same principal components as the market data?
- Do the simulated co-movements of implied vol and the underlying exhibit statistical properties similar to the market data?
- Does the model output realistic joint dynamics with the VIX?
- What do the simulated distributions look like?

Designing a custom model: does the output look realistic?

We know when it clearly doesn't



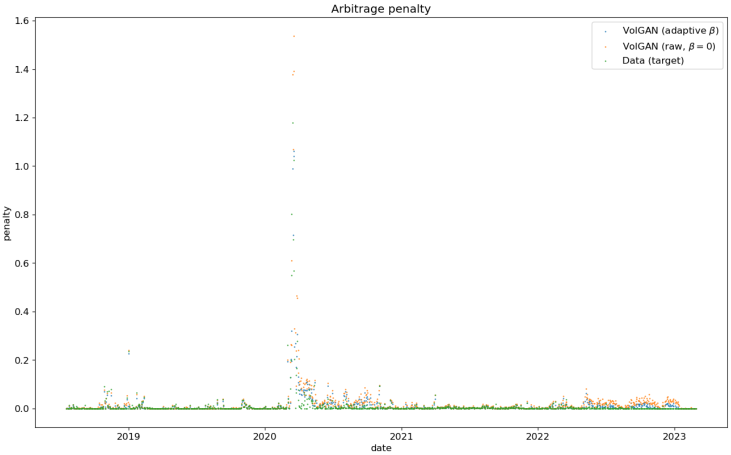
VolGAN SPX implied volatility sample



Sample SPX implied volatility surface from a BCE (vanilla) GAN.

VolGAN performance on SPX data

Arbitrage penalty



Distance to static arbitrage as measures by the arbitrage penalty.

Mathematical Institute

	Mean	Std	Median
SPX data	0.0096	0.0628	0.0005
BCE GAN	2.4635	0.9086	2.3164
Raw VolGAN (before weighting)	0.0199	0.088	0.003
VolGAN (after re-weighting)	0.0127	0.0620	0.0014

Table: 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. Standard deviation and median for GAN outputs correspond to the standard deviation and the median of (re-weighted) average outputs given 10000 samples.

SPX returns

Raw generator produces more stable confidence intervals

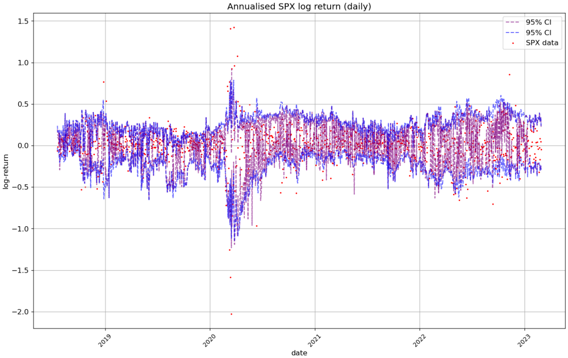
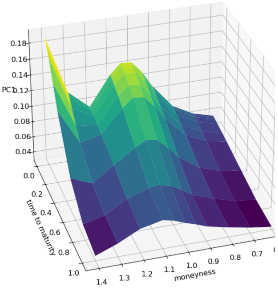


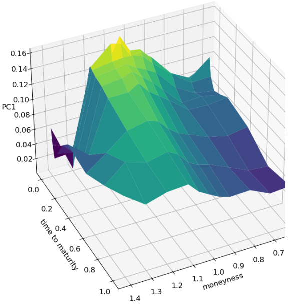
Figure: Realized and simulated SPX price on the test set. SPX data (red), next-day forecast ($\mathbb{E}_\beta[S_t|a_{t-\Delta t}]$) and the 95% confidence interval (blue: without re-weighting, purple: with re-weighting). The confidence interval is calculated based on the 2.5% and 97.5% VolGAN quantiles before and after re-weighting.

VolGAN learns principal components out-of-sample

First principal component corresponds to level



(a) Computed using SPX implied volatility data.



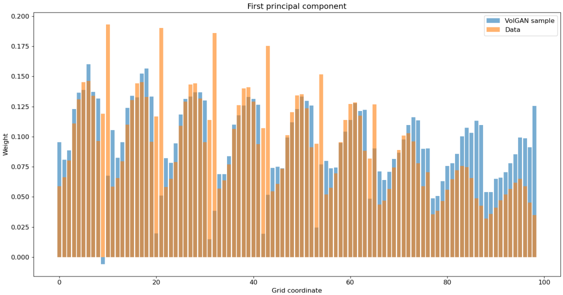
(b) Computed using a sample VolGAN output.

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

VolGAN learns principal components out-of-sample

First principal component corresponds to level

UNIVERSITY OF OXFORD
Mathematical Institute



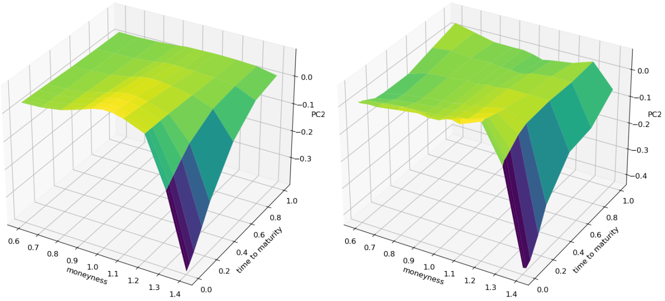
(a) Comparison of the first principal component in the data and in a sample simulation as vectors.

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

VolGAN tracks principal components out-of-sample

Second PC corresponds to skew

University of Oxford Mathematical Institute



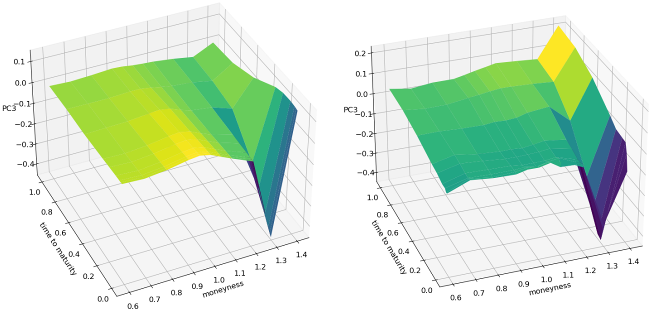
(a) Computed using SPX implied volatility data.

(b) Computed using a sample VolGAN output.

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

VolGAN learns principal components out-of-sample

Third PC corresponds to curvature



(a) Computed using SPX implied volatility data.

(b) Computed using a sample VolGAN output.

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

Principal component analysis: VolGAN vs SPX implied vol

Inner products of principal components for daily log-returns of SPX implied volatility vs VolGAN implied volatility increments:

Rank	Mean	Median	Standard deviation
First	0.921	0.922	0.009
Second	0.921	0.922	0.011
Third	0.798	0.798	0.011

Percentage of variance explained by the top three principal components of the simulated and the data log implied volatility increments:

Rank	Data	VolGAN
First	51.25%	$45.31 \pm 1.84\%$
Second	34.00%	$25.69 \pm 0.88\%$
Third	5.01%	$12.76 \pm 0.55\%$

Pearson correlation between (simulated) SPX log-returns and the projections of the (simulated) log-implied volatility increments on the principal components:

PC rank	Data (test)	VolGAN (test)	Data (train)
First	-0.76	-0.84 ± 0.024	-0.34
Second	-0.29	-0.38 ± 0.055	-0.32
Third	0.06	0.16 ± 0.020	0.28

Correlation structure of variables

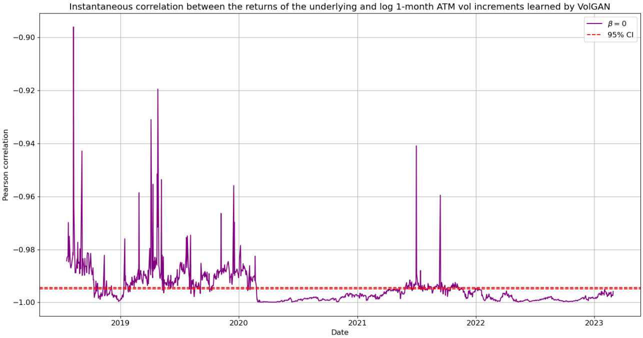
Out-of-sample (4.5 years after training including Covid) average Pearson correlation for simulated vs real values of log-returns of SPX ($\Delta \log S_t$), implied volatility level factor (ΔX_t^1), 1-month ATM volatility ($\Delta \log \sigma_t^{ATM}$) and VIX ($\Delta \log v_t$). Average VolGAN outcome (blue) and data (red).

	$\Delta \log S_t$	ΔX_t^1	$\Delta \log \sigma_t^{ATM}$	$\Delta \log v_t$
$\Delta \log S_t$	1.00	−0.84 −0.76	−0.86 −0.77	−0.55 −0.71
ΔX_t^1	−0.84 −0.76	1.00	0.95 0.89	0.66 0.84
$\Delta \log \sigma_t^{ATM}$	−0.86 −0.77	0.95 0.89	1.00	0.72 0.96
$\Delta \log v_t$	−0.55 −0.71	0.66 0.84	0.72 0.96	1.00

VolGAN learns time-varying correlations

It would be difficult to replicate such property in a parametric model

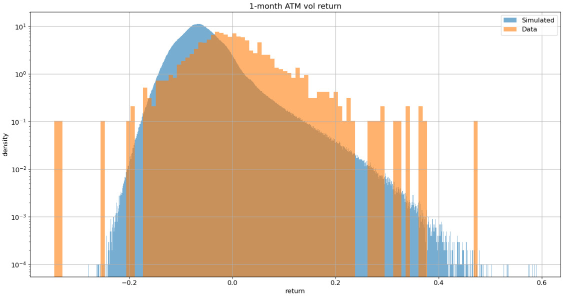
Pearson correlation between simulated index returns and 1-month ATM volatility increments (blue), with symmetric 95% confidence interval of constant correlation (red). VolGAN with $\beta = 0$.



Distribution of simulated ATM implied vol increments

University of Oxford
Mathematical Institute

Simulated 1-month ATM volatility increments (blue) exhibit asymmmetric, exponentially decaying tails. VolGAN with $\beta = 0$.



Bonus: discriminator can be used for anomaly detection

University of Oxford
Mathematical Institute

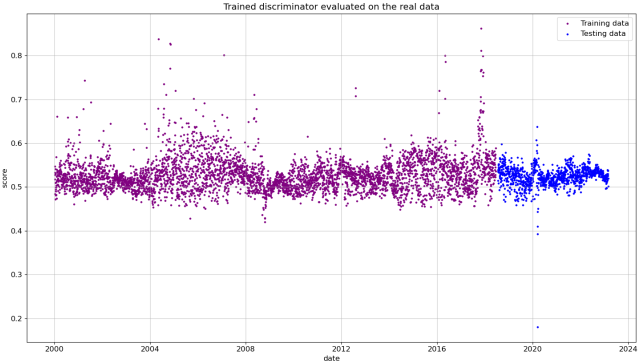


Figure: An advantage of adversarial training is that the discriminator may be used for anomaly detection. Output score of the trained discriminator on SPX data. Purple: seen by the discriminator, blue: out-of-sample.

Summary

- VolGAN is a custom data-driven generative model for simulating arbitrage-free implied volatility surfaces.
- It learns to simulate the next day's log-returns of implied vol and of the underlying, given today's data.
- VolGAN learns non-Gaussian distributions with time-varying correlations.
- VolGAN captures the co-dynamics of the implied volatility surface, the underlying, and the VIX effectively, even during volatile periods.
- It can be used for data-driven hedging of option portfolios.

Full paper and code available online

Mathematical Institute

Vuletić, M. and Cont, R. (2025) 'VolGAN: A Generative Model for Arbitrage-Free Implied Volatility Surfaces', Applied Mathematical Finance, pp. 1–36. doi: 10.1080/1350486X.2025.2471317.

Code available on GitHub:

https://github.com/milenavuletic/VolGAN/