# Generative Implied Volatility Modelling in the Era of 0DTE Options

Master Thesis
MSc ETH Mathematics

*written by*
John Skelton

*supervised by*
Prof. Dr. Josef Teichmann

ETH Zürich
Spring/Autumn 2024

## Contents

## Abstract

We develop a diffusion model for arbitrage-free implied volatility surface generation, capable of conditioning on arbitrary alpha signals. By decomposing the dynamics of the volatility surface into an endogenous part driven by the maturation of the surface, and an exogenous part driven by a conditional diffusion model, *VolaDiff* is able to accurately capture the dynamics of the SPX implied volatility surface at a resolution of 32 strikes and 16 DTEs. In particular, the model matches the first five principal components of daily variation and learns a negative spot-vol correlation while generating realistic volatility surfaces. VolaDiff leverages a novel conditioning mechanism, allowing it to condition on arbitrary alpha signals. It is capable of scenario forecasting, and responds correctly to bearish and bullish signals. Code is available at github.com/JPNotleks/VolaDiff.

# Acknowledgments

It is an honour to once again be supervised by Prof. Dr. Josef Teichmann, who remains the most influential and inspirational character in my journey as a mathematician. From the topic of implied volatility modelling to specific design choices, I owe a great majority of my ideas and ways of thinking to Prof. Teichmann's teachings over the years.

Special thanks also to Sayak Paul for advice on U-Net architectures and diffusion models.

# 1 Introduction

Since 2022, the SPX options market lists expiries for *every* day of the next month. Consequently, there are always options expiring at market close - these are known as 0DTE (zero days-to-expiration) options. The popularity of these options has skyrocketed since their introduction: SPX option volume has grown twice as fast as the rest of the option market, 0DTEs make up almost half of daily SPX option volume, and the notational exposure traded through SPX options (>1 trillion USD/day) now makes up more than half of all option exposure globally [Zai23]. This availability of daily expiries in conjunction with almost 200 different strikes per expiry empowers traders with surgical precision on the SPX, but is the bane of every researcher: recent at-the-money (ATM) bid-ask spreads of 1-2% make arbitrage-free volatility fitting on such liquid short-maturity option expiries incredibly challenging. Classical low-dimensional parametric models struggle to fit the W-shaped implied volatilities frequently observed in <5DTE, and non-parametric models which are capable of perfect fits are often too slow to use and cannot scale to the sheer number of contracts available on the SPX. While volatilities still react with a strong negative correlation to stock returns, and display significant path-dependence and clustering, the SPX option surface can become highly polarised prior to macro-economic events. This begs the question how to model and disentangle the evolution of the option surface as these events transpire - all with highly expressive models.

Previous literature on generative implied volatility modelling [VC23; Che+23; Ber+21; Nin+22; CD02; CGL23] does not consider the complexities of the 0DTE regime since 2022. In this thesis, inspired by Guyon's path-dependent volatility [GL22], we decompose the dynamics of the volatility surface into an endogenous part - the maturation of the surface as if no volatilities changed - and an exogenous part - volatility shocks to the surface caused by price movements and news. The conditional distribution of this volatility shock is modelled by a diffusion model due to their excellent performance in image generation. Predictions of tomorrow's volatility surface are then given by the sum of the matured surface and the output of the diffusion model, followed by model-free arbitrage-removal if desired.

We begin in sections 2 and 3 by framing the problem of implied volatility modelling. This is a remarkably difficult topic in mathematical finance - as we shall see, it amounts to robust regression under the constraint of a non-linear, second order partial differential inequality! In illiquid markets, it is possible to obtain tradeable fits to the market using parametric models which satisfy this PDI by construction (martingale models like Heston, SABR, PDV, etc). The more liquid the option market is, the worse these fits become due to their limited capacity. We pursue a model-free solution to this fitting problem which *improves* as option markets become more liquid, by alternating between model-free arbitrage removal and spline fitting similar to Fengler's method [Fen09]. We find a single iteration to be sufficient to generate smooth interpolations that remain within the bid-ask spread with guaranteed arbitrage-freeness.

In section 4, we decompose the dynamics of the volatility surface into an endogenous and exogenous part. The resulting residual to be modelled is far smoother than the raw surface and hence a convenient modelling objective.

Section 5 introduces diffusion models and how to build competitive architectures for image generation. We fine-tune our model - VolaDiff - to generate viable volatility shocks and test the realism of the generated surfaces by conditioning on tomorrow's price change. The

model correctly learns a negative correlation between price changes and volatility shocks, but also learns complex surface relationships reflecting changes in skewness and kurtosis. VolaDiff is capable of conditioning on arbitrary alpha signals by leveraging a novel conditioning mechanism, yielding great flexibility and rendering it a viable tool for quantitative research from deep-learning-based alpha generation to scenario forecasting and risk management.

## 2 Implied Volatility Surfaces: Static and Dynamic Arbitrage

Throughout this thesis, we represent the SPX stock index as a non-negative semi-martingale $S$ on a filtered probability space $(\Omega, \mathcal{F}, \mathbb{P})$ with finite time-horizon $T_{\max}$. We assume arbitrage-freeness in the form of NFLVR [DS04] to guarantee the existence of an equivalent local martingale measure $\mathbb{Q}$, and in order to avoid technicalities arising from strict local martingales, we will in fact require $\mathbb{Q}$ to be an equivalent martingale measure. In order to force uniqueness of option prices, we also assume that $\mathbb{Q}$ is the *unique* equivalent martingale measure, rendering the market complete. These assumptions are not unrealistic in a market as liquid as the SPX option market, since arbitrageurs rapidly correct blatantly mispriced options, and bid-ask spreads are so tight that there is little freedom in the market-implied risk-neutral measure. Moreover, there exist complete martingale models like path-dependent volatility [GL22] which are capable of capturing the approximate shape and the dynamics of the SPX volatility surface. Interest rates are represented by the deterministic discount factor $d(t,T)$ but we make no assumptions on this since it will be implied by the option prices. Unlike American equity options, SPX options are European in nature, and their price at time $t$ is given by

$$C(T,K) = d(t,T)\mathbb{E}^{\mathbb{Q}}[(S_T - K)^+|\mathcal{F}_t], \tag{2.1}$$
$$P(T,K) = d(t,T)\mathbb{E}^{\mathbb{Q}}[(K - S_T)^+|\mathcal{F}_t] \tag{2.2}$$

for calls and puts, respectively. Assuming observability of prices for all maturities and strikes, we obtain the call and put surfaces

$$\begin{aligned} C : [0,T_{\max}] \times [0,\infty) &\to \mathbb{R}^+ \\ (T,K) &\mapsto d(t,T)\mathbb{E}^{\mathbb{Q}}[(S_T - K)^+|\mathcal{F}_t] \end{aligned}$$

$$\begin{aligned} P : [0,T_{\max}] \times [0,\infty) &\to \mathbb{R}^+ \\ (T,K) &\mapsto d(t,T)\mathbb{E}^{\mathbb{Q}}[(K - S_T)^+|\mathcal{F}_t]. \end{aligned}$$

Basic no-arbitrage considerations imply that both these surfaces have to be convex and monotone in strike and non-decreasing in time. Furthermore, options are lower-bounded by their intrinsic value $C(T,K) \geq (S_t - Kd(t,T))^+$ and $P(T,K) \geq (Kd(t,T) - S_t)^+$. Since we have

an equivalent martingale measure, put-call parity holds, allowing us to write

$$C(T,K) - P(T,K) = d(t,T)\,(F_t - K) \tag{2.3}$$

where $F_t = d(t,T)^{-1}S_t$ is the forward price with maturity $T$.

Although this shows that specifying either a call or put surface already characterises the other, calls are generally liquidly traded at $K \geq S_t$ while puts are usually traded at $K \leq S_t$ due to increased effective leverage of OTM options. To simplify the modelling of the surface, we will merge the two by obtaining call prices for $K \geq S_t$ and converting put prices for $K < S_t$ to call prices via put-call parity. This is not only necessary (since attempting to use call prices in regions that are not liquidly traded results in stale prices) but also results in highly accurate prices across all strikes because put-call parity is continuously enforced by low-latency traders. Henceforth we shall only consider the call surface as the object of interest. There are several classical results linking the shape of this surface to the evolution of the underlying; the simplest is the Breeden-Litzenberger formula [BL78] which asserts that one can reverse-engineer the marginal density $q_T$ of $S_T$ (under $\mathbb{Q}$, not $\mathbb{P}$!) via

$$q_T(K) = d(t,T)C_{\mathrm{KK}}(T,K)dK, \tag{2.4}$$

and one may even weaken the assumed twice differentiability. Another classical result is Kellerer's Theorem: given any surface satisfying the no-arbitrage conditions above, there exists a Markovian martingale whose call surface coincides with the specified surface. One of the cornerstones of volatility modelling - Dupire's local volatility formula - yields a closed-form representation of such a process. However, all of these insights depend on the quantity $\partial_{\mathrm{KK}}C(T,K)$, which is notoriously difficult to estimate with a small set of strikes. In fact, calibrating any model to the market call surface (or equivalently, specifying $\mathbb{Q}$) is inherently an ill-posed problem since the call operator mapping $\mathbb{Q}$ to its call surface is an integral operator and therefore difficult to invert [Mon13]. A more convenient parametrisation of the call surface which eliminates some of these difficulties is in terms of *implied volatility*.

We recall that the Black-Scholes model - the origin of all volatility modelling - posits that the underlying evolves according to geometric Brownian motion described by the SDE

$$dS_t = \mu S_t dt + \sigma S_t dW_t.$$

The call surface corresponding to this model is the celebrated Black-Scholes-Merton formula

$$C(T,K) = d(t,T)\,(\mathcal{N}(d_+)\,F_t - \mathcal{N}(d_-)\,K) \tag{2.5}$$

$$d_+ = \frac{1}{\sigma\sqrt{\tau}}\left(log\left(\frac{F}{K}\right) + \frac{1}{2}\sigma^2\tau\right) \tag{2.6}$$

$$d_- = d_+ - \sigma\sqrt{\tau}, \tag{2.7}$$

where $\tau = T - t$ is the time to maturity. Since this formula is monotone in $\sigma^2$, it may be

inverted, yielding the volatility *implied* by a given option price. Although complicated at first sight, implied volatility is a convenient way to recast prices since it usually normalises a range of price from $(0, S_t)$ to approximately $(0, 1)$ (commonly observed volatilities are 5%-100% per year). We stress that it is nothing more than a reparametrisation of price.

In some cases, an even more convenient reparametrisation is total implied variance in normalised strike (log-moneyness), given by

$$w(T, K) = \tau \sigma^2_{\text{imp}}(T, log\,(K/F_t)).$$

Total implied variances observed in the market are smooth in strike by default - likely due to market makers imposing smooth parametric models on their quotes - and monotone increasing in time as a no-arbitrage constraint. Modelling in terms of implied total variance has the added benefit of facilitating reasonable tail fitting. As quotes become sparse and wide for deep out-of-the-money options, traders require an extrapolation of traded volatilities. In terms of price, this is hopeless since prices decay rapidly in the tails. However, in terms of implied total variance, Roger Lee's moment formula shows that the tails must become asymptotically *linear* as normalised strike approaches $\pm\infty$ [Lee02]. However, by converting from price to total implied variance, we trade smoothness for complicated no-arbitrage relations. Recalling that the convexity of the price surface necessitates $\partial_{KK} C(T, K) \geq 0$, substituting the Black-Scholes-Merton formula into this ODE shows that the following non-linear ODE has to be satisfied in terms of implied variance [Gat12]:

$$0 \leq \left(1 + \frac{K w_K}{2w}\right) - \frac{w_K^2}{2}\left(\frac{1}{w} + \frac{1}{4}\right) + \frac{w_{KK}}{2} \qquad \forall (T, K) \in [0, T_{\max}] \times \mathbb{R}. \tag{2.8}$$

The calendar arbitrage constraint $C_T(T, K) \geq 0$ translates to $w_T(T, K) \geq 0$. Developing arbitrage-free models in total variance is therefore difficult, but it facilitates fitting of smooth curves to market data. The most competitive commercial volatility fitters like Vola Dynamics also seem to operate with implied total variances. In the public domain, the most popular model is the SVI-JW model by Gatheral which has five highly intuitive parameters controlling level, ATM skew, put slope, call slope, and the global minimum. Usually, implied total variances observed in markets are V-shaped with a smooth minimum and two linear upward-sloping tails. In the 0-5DTE regime, however, the implied total variances can become W-shaped, reflecting a bimodal risk-neutral density as the market forms views on an impending move of an asset. This splitting of the density becomes more pronounced as the time-to-maturity decreases and is now visible in SPX options roughly every fifth trading day. It is here where classical models like SVI-JW fail and implied volatility dynamics become complicated.

A final but important distinction is static versus dynamic arbitrage-freeness. While static arbitrage is always disqualified in theory by conditions on the option price surface, and in practice by low-latency arbitrageurs, dynamic arbitrage concerns the existence of arbitrage opportunities by considering option contracts as separate tradeable securities over time. The resulting high-dimensional market of the spot asset together with all of its options may admit dynamic arbitrage even in the absence of static arbitrage at any time. Conditions for option prices to satisfy absence of dynamic arbitrage are complex and implied total variance does not seem to be the right framework for it [SW08], although modern machine-learning-based approaches exist [CRW23]. Since we are interested in single-day volatility increments, we lack

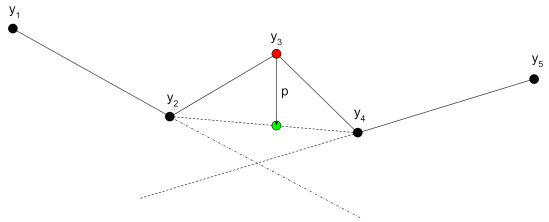
any notion of dynamics and focus purely on absence of static arbitrage. All path-dependence and encoding of the market state is left to the user to input into VolaDiff as a conditioning vector.

## 3 Arbitrage-free Volatility Fitting and Discretisation

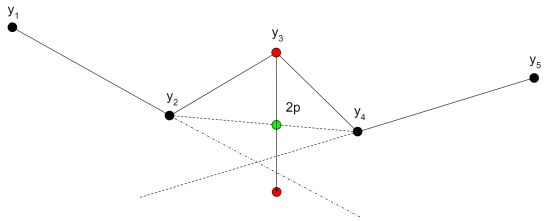
We now detail how to obtain highly accurate, arbitrage-free volatility surfaces from bid and ask prices on the SPX. The data we use is generously sponsored by ORATS Option Research and Technology Services. Each day, a snapshot of all bids and asks of all SPX option expiries is taken at 15:46 (14 minutes before close) for a total of approximately 5000 points in the (T,K)-plane. We reverse-engineer the forward price from put-call parity (equation 2.3) for each option maturity separately: since put-call parity implies that calls equal puts at the forward, we perform a minimally smoothed cubic spline fit to their difference and solve for zero numerically. ATM option liquidity is high enough for this to yield extremely accurate forward prices. Furthermore, obtaining forward prices from the options themselves (and not from interest rates) combats arbitrage in the surface as temporary supply and demand pressures can distort these rates. Implied volatilities are obtained from prices by a Newton method on the Black-Scholes-Merton formula with a tolerance of $10^{-12}$, equivalent to price errors several orders of magnitude smaller than the bid-ask spread and therefore completely negligible.

The dominant SPX options market makers like Citadel, SIG, Optiver, and IMC achieve a wire-to-wire time of <5ns (1.5m light-distance!) nowadays, rendering tradable static arbitrage impossible to observe in real prices. Due to the asynchronicity of option and spot order books, it is possible to end up with data containing static arbitrage opportunities, but these are not tradable for anyone but the fastest low-latency traders, and they should therefore be removed from data. We filter such arbitrageable prices using *arbitragerepair* [CRW20; Wan20], which finds the minimal $L^1$-perturbation to the option surface to remove all arbitrage. We avoid running *arbitragerepair* on the entire surface since it tends to return zero implied density at corrected prices, and has no respect for smoothness in implied total variances. This is because the minimal perturbation always projects arbitrageable prices onto the linear function interpolating its two neighbours. These artefacts bias the surface towards linear segments and are even visible in the examples documented on the GitHub repository [Wan20]. Instead, we apply *arbitragerepair* to each maturity individually, passing both the maturity itself and its two adjacent maturities as inputs. Furthermore, we present an ad-hoc improvement which alleviates the smoothness problem by performing two iterations of *arbitragerepair*: the first iteration computes the $L^1$-minimal perturbation removing all arbitrage, then we add *twice* this perturbation to the original surface, and run *arbitragerepair* once again. The final output is the mean of the two arbitrage-free surfaces produced, which is also guaranteed to be arbitrage-free. In figure 1, we illustrate why this solves the problem of zero-implied densities: the two surfaces produced are likely to output the maximum and minimum possible corrected prices, and their mean is a more realistic estimate of the price.

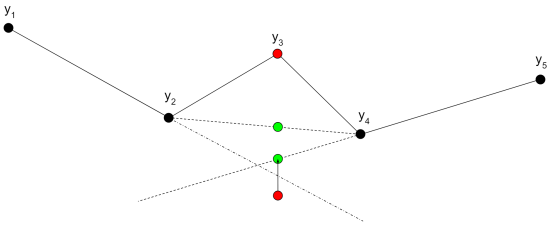
The final surfaces are guaranteed to be free of arbitrage while being smooth and entirely model-free. Note that other high-capacity arbitrage-removal methods exist, including *deep smoothing* [ATV20] and more recently, a neural operator approach [GJW24]. Finally, once



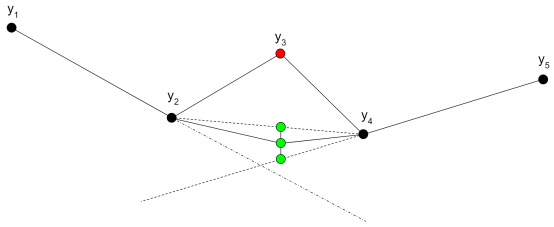

(a) The incorrect price is projected by the smallest amount that renders the resulting curve convex.




(b) Add twice the perturbation to the incorrect price.



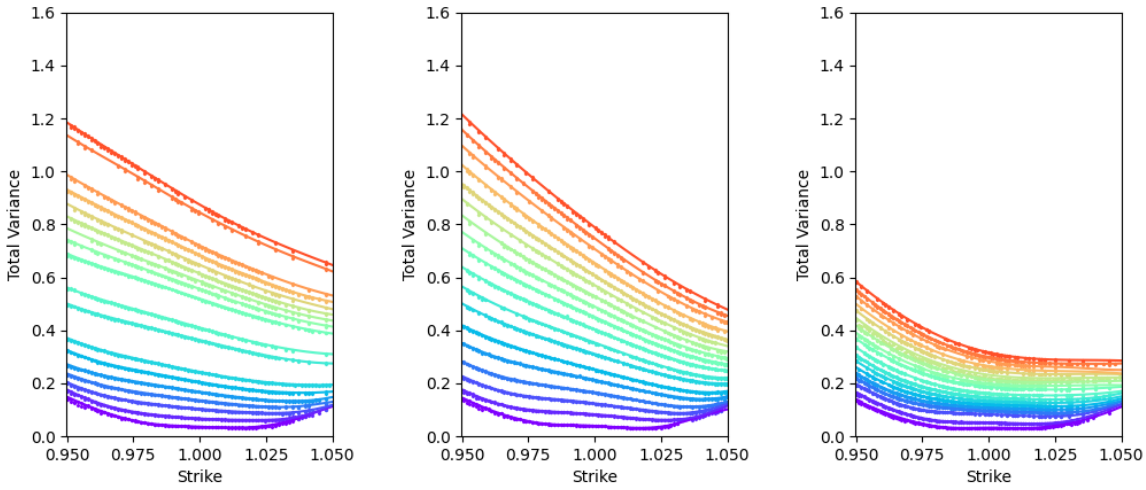

(c) *Arbitragerepair* projects the perturbed price back to the nearest convex surface.




(d) The mean of the two projections is guaranteed to be arbitrage-free and smoother than either.

Figure 1: Two-step modification to *arbitragerepair*.

cleaned prices have been converted to total implied variances, we perform local cubic spline fitting on each maturity and then evaluate the volatility surface on a grid of 32 points in normalised strike $[\log(0.95), \log(1.05)]$. This smoothing may introduce minuscule amounts of arbitrage which may again be filtered using *arbitragerepair*, but we do not find this to be relevant. Note that we avoid investigating the tails beyond $\pm 5\%$ OTM since these provide little information at short time horizons and are sometimes thinly traded. The final data consists of the first 16 expiries (3+ weeks) stacked vertically as vectors of length 32, yielding a volatility surface describing 16 business days to expiry at a resolution of 32 strikes per day.




(a) The SPX surface on 20/01/2023 exhibiting three major events priced in at 8, 10, and 17 business days to expiration. VIX=19.9

(b) The SPX surface on 20/10/2023 during the largest crash of the year. Notice the long maturity implied volatility skew. VIX=21.7

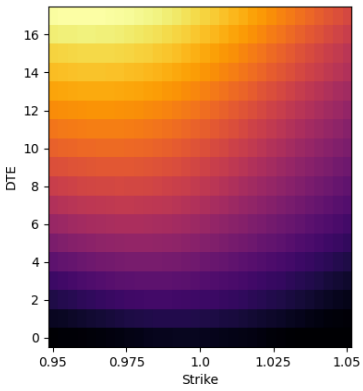
(c) The SPX surface on 30/01/2024 in a low volatility environment, but with three consecutive W-shaped IV curves. VIX=14.3

Figure 2: Implied total variances in three market regimes of the SPX, colour-coded by expiration. Bid and ask prices are marked as dots but are barely distinguishable due to tight spreads.

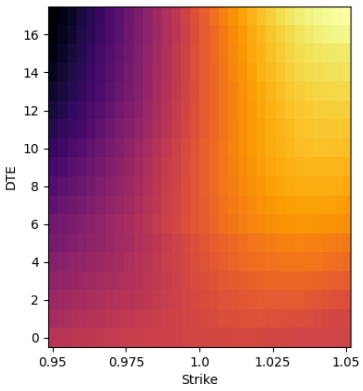
# 4 Endogenous Volatility Surface Dynamics in the 0DTE Regime

The literature has noted that the daily increments of implied volatilities are well-described by a small number of principal components [VC23; CD02]. However, these observations were made when 0DTE options were unavailable, bid-ask spreads were wide, and data was constrained to weekly or monthly expires. This low-rank structure does not extend at all to the 0DTE regime since 2022: daily increments of implied volatilities exhibit far more complicated structure in low-maturity regions as evident in figure 4. While long-maturity options are well-described by principal components corresponding to level, skew, and term structure, the low-maturity regime requires a completely different treatment. We note that the surfaces themselves - but not the increments - are well-described by a small number of principal components: figure 3 displays the first four principal components of the total variance surfaces in 2023. Only the first three are interpretable as level, skew, and a characteristic term structure shock near 1 week to expiration.

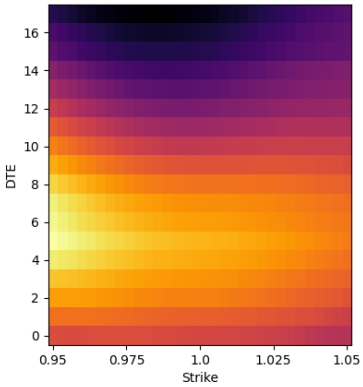
We propose a straightfoward model to simplify the complex evolution of the volatility surface from one day to the next, based on the following empirical observation: when a day passes and no extreme returns in the SPX transpire, the volatility surface approximately moves by one day, i.e. all implied total variances move down one day but otherwise remain unchanged. This is to be expected, since the surfaces we generate are capable of timing market events down to the day, and hence their maturation should respect this structure. In the case of a large movement or unexpected event, the entire volatility surface experiences a shock reflecting the updated market conditions: a large negative return causes all volatilities to spike, while a



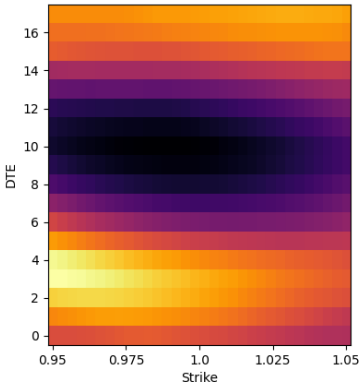

(a) The first principal component reflects the average level of the volatility surface.




(b) The second component is a long-maturity skew factor causing 5-30 day IV skew.




(c) The third principal component is a 1-week volatility shock.




(d) The fourth principal component is no longer clearly interpretable.

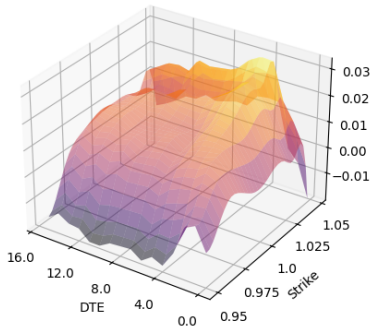
Figure 3: The first four principal components of the SPX total variance surfaces in 2023.

positive return attenuates the entire surface. Formally, we consider the implied total variance surface on day $n \in \mathbb{N}$ as $w_n(t_i, k_j)$ for $1 \leq i \leq 16, 1 \leq j \leq 32$ and decompose tomorrow's surface $w_{n+1}$ as

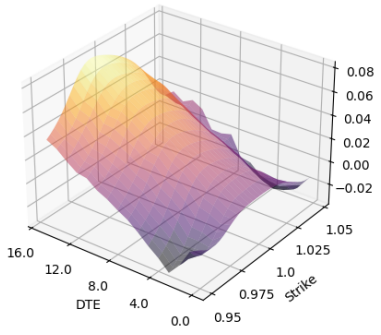
$$w_{n+1}(t_i, k_j) = w_{n+1}(t_1, k_j) + (w_n(t_{i+1}, k_j) - w_n(t_2, k_j)) + v_n(t_i, k_j) \tag{4.1}$$

where $v_n$ is the volatility surface shock not explained by the maturation of the surface. In other words, the 1DTE expiry on day $n$ is consumed and replaced by the 1DTE expiry on day $n + 1$, while all other inter-maturity relations mature by one day, and the residual is modelled by $v_n$. This residual is far smoother and more expressive than a direct prediction of tomorrow's implied total variance surface since the endogenous dynamics we posit subsume the complex inter-maturity relations. We avoid modelling the 1DTE expiry on the following day since understanding the inter-maturity relations is sufficient for pure vega strategies. Figure 4 illustrates how rich the volatility shocks are in the low-DTE regime.

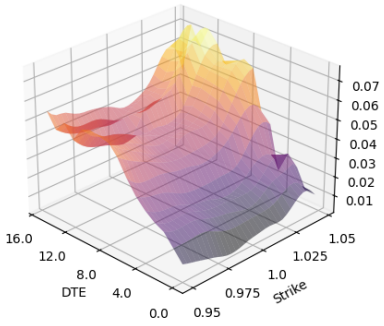
As smooth surfaces represented by $16 \times 32$ images, the volatility shocks may be directly



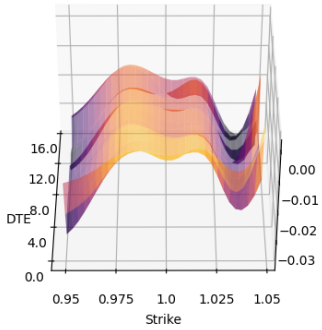

(a) SPX target vol shock on 4/1/2023. Despite a -1.1% return the following day, put volatilities decreased and a significant short-term call vol increase correctly marks the beginning of the bull market in Jan 2023.




(b) SPX target vol shock on 17/1/2023, characteristic of global volatility increases following a -1.6% return. Note, however, that there is no skew towards puts; the market rebounds 2 days later.




(c) SPX target vol shock on 1/2/2023, displaying global vol increases despite a +1.5% rally. Severe long-term kurtosis increases herald the beginning of the choppy market in Feb 2023.




(d) SPX target vol shock on 14/2/2023, two days before the Feb 2023 crash. The vol shock represents the formation of a bimodal 1-day risk-neutral distribution on 15/2/2023, the day of the crash.

Figure 4: Four sample vol shocks in early 2023 illustrating the complexity of the low-DTE surface evolution.

modelled by any machine learning model capable of learning conditional distributions. In the next section, we explain how diffusion models are a promising class of models for such tasks.

# 5 Diffusion Models for Volatility Surfaces

## 5.1 Origins and Shortfalls of Diffusion Models

Diffusion models are a class of generative models that learn to reverse a data corruption process, allowing them to transform pure noise into data. They have become the industry standard in generative image modelling since [HJA20] showed how score-matching methods can finally outperform GANs at sample quality while being remarkably simple to train. It is mainly this training stability that has allowed researchers to focus on maximising model capacity, rendering diffusion models sufficiently flexible to scale to gargantuan datasets and leading to recent waves of faster and more realistic multi-billion parameter models like stable diffusion XL and DALL·E. However, this simplicity also leads to a plethora of potential modelling pitfalls: even stable diffusion used to leak 7% of the image signal, leading to significant artefacts which went unnoticed for over half a year [Lin+24]. This was not a bug; rather, it originated from poor modelling choices. In order to avoid such mistakes, we will develop a custom diffusion model from first principles, leveraging recent insights into beneficial modelling choices which have led to state-of-the-art results.

Like any generative model, diffusion models aim to learn, and then sample from some distribution $p_0 \in \mathcal{P}(\mathbb{R}^n)$. Here we focus mainly on compactly supported distributions and square-integrable tails, rendering the Wasserstein space $\mathcal{W}_2(\mathbb{R}^n)$ a suitable base space. GANs model the data distribution directly as a push-forward of a simple (Gaussian) distribution through a neural network. This is extremely taxing on the neural network since it has to learn how to transform Gaussians into complex, high-dimensional distributions. Diffusion models decompose this task by first continuously deforming the data distribution to a Gaussian, and then learning to reverse the process in small steps. While diffusion models were originally framed in discrete time, it was soon realised that a continuous-time formulation yields a more elegant theory [Son+21]. For this thesis, we will also stick to continuous-time, particularly because readers interested in volatility modelling will already be familiar with the required stochastic calculus.

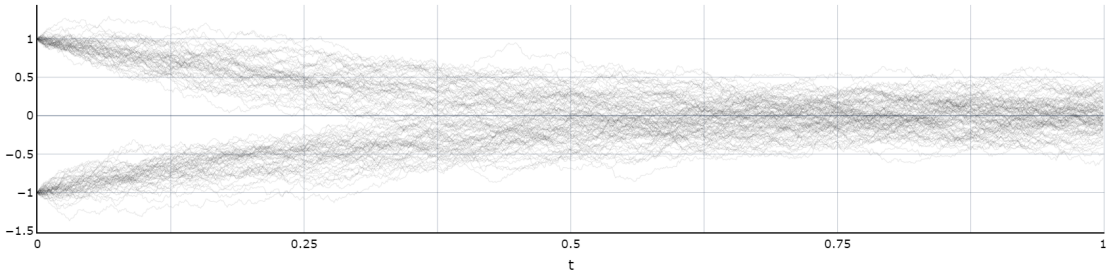
A diffusion model consists of three parts: a forward process, its corresponding backward process, and a neural network. The forward process is an SDE

$$dX_t = F_t X_t dt + G_t dW_t, \qquad X_0 \sim p_0 \tag{5.1}$$

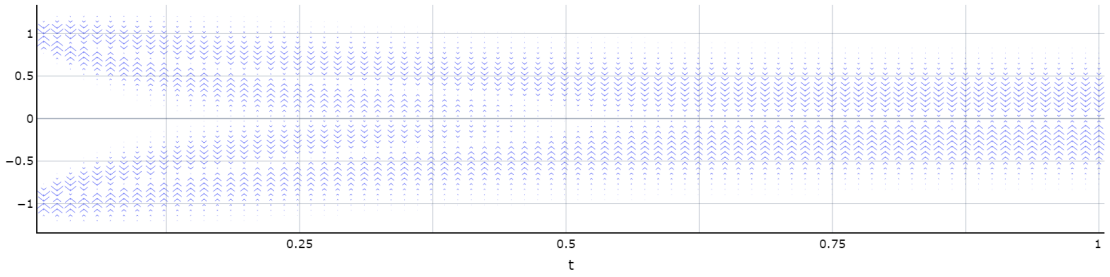
for carefully chosen linear maps $F_t$ and $G_t$, and the initial condition is the desired data distribution $p_0$. Omitting technical details, its *time-reversal* $\tilde{X}_t = X_{T-t}$ satisfies the SDE

$$d\tilde{X}_t = -\tilde{F}_t\tilde{X}_t dt + \tilde{G}_t\tilde{G}_t^T\nabla_x \log p_t(\tilde{X}_t)dt + \tilde{G}_t d\tilde{W}_t, \qquad \tilde{X}_0 \sim p_T \tag{5.2}$$

where $p_t$ is the density of $X_t$, and $\tilde{W}_t$ is another Brownian motion [HP86]. In other words, when volatility is reversed in time, it becomes drift. If we can learn this drift $\nabla_x \log p_t$, known as the *score function*, then the time-reversed SDE will map samples from $p_T$ to $p_0$, yielding our desired sampling process. Usually, the drift $F_t$ and volatility $G_t$ are chosen in such a way that the forward process has excellent mixing properties and converges rapidly to a simple invariant distribution $p_\infty$ (usually Gaussian or uniform). The most natural process of this type is a variant of the Ornstein-Uhlenbeck (OU) process with time-varying decay and noise rates, which amount to setting $F_t = -\mu_t I_{n\times n}$ and $G_t = \sigma_t I_{n\times n}$ for $\mu_t > 0$. This family of processes has closed-form Gaussian transition kernels, and converges to a Gaussian invariant distribution exponentially fast in the Wasserstein 2-metric as long as $\mu_t$ is bounded away from zero. Figure 1 illustrates the OU process ($\mu_t$ and $\sigma_t$ constant), and the corresponding score function on a one-dimensional bimodal initial distribution $p_0 = \frac{1}{2}(\delta_{-1} + \delta_1)$.




(a) Sample trajectories of an OU process.




(b) Associated un-normalised score function $\nabla_x p_t(x)$, depicted as a vector field.

Figure 5: Samples of an OU process together with its score function.

Virtually all diffusion models are of this type due to the rapid convergence and availability of closed-form transition kernels. Without loss of generality, this is commonly reparametrised (abusively reusing $\sigma_t$) as

$$P\left(X_t|X_0\right) = \mathcal{N}(X_t|\alpha_t X_0, \sigma_t), \tag{5.3}$$

where $\alpha : [0, T] \to [0, 1]$ is a monotone decreasing function with $\alpha_0 = 1$ and $\sigma : [0, T] \to [0, \infty)$

is monotone increasing. In the diffusion literature, this transition kernel is often written suggestively (and illegitimately) as

$$X_t = \alpha_t X_0 + \sigma_t \mathcal{N}(0,1) \tag{5.4}$$

as this is how the forward process is sampled algorithmically. Together, $\alpha_t$ and $\sigma_t$ are known as the *noise schedule*. The performance of diffusion models is extremely sensitive to the noise schedule and it has been the centre of much research due to its impact on the geometry of the score function [Kar+22]. In particular, noise schedules where $\alpha_T \neq 0$ are detrimental since the terminal distribution $p_T$ will not be a pure Gaussian even though the backward process samples from a Gaussian. Even a small nonzero $\alpha_T$ will cause the model to sample severely out of distribution. For instance, the first version of stable diffusion used $\alpha_T \approx 0.07$ and consequently suffered from an inability to generate a black background even when prompted for one [Lin+24]. Since this discovery, the diffusion community has embraced zero signal-to-noise (SNR) noise schedulers which enforce $\alpha_T = 0$.

In order to learn the score function, one may be tempted to train a neural network $s_\theta : \mathbb{R}^{n+1} \to \mathbb{R}^{n+1}$ to minimise

$$L_{SM}(\theta) = \mathbb{E}_{t\sim U(0,T), X_t \sim p_t}\left[\lambda(t) \left\| s_\theta(X_t, t) - \nabla_x \log p_t(X_t) \right\|^2\right] \tag{5.5}$$

for a neural network $s_\theta$ and a weighting function $\lambda : [0,T] \to \mathbb{R}^+$. This *score-matching loss* is a valid training objective, as it has been shown to upper-bound the KL distance and the Wasserstein 2-distance between the data distribution and the distribution obtained by the reversed SDE [KFL22]. However, evaluating the score function a single time requires passing over the entire dataset, so this is not a practical objective. Pascal Vincent [Vin11] was the first to show in 2011 that there is an equivalent loss - the denoising score matching loss - which eliminates this explicit dependence:

$$L_{DSM} = \mathbb{E}_{t\sim U(0,T), X_0 \sim p_0, X_t \sim p_t(\cdot|X_0)}\left[\lambda(t) \left\| s_\theta(X_t, t) - \nabla_x \log p_t(X_t|X_0) \right\|^2\right]. \tag{5.6}$$

This is in fact a manifestation of Tweedie's formula, and we refer the interested reader to [Thi23] for an insightful discussion. Note that the log gradient of the conditional density is simply that of a Gaussian,

$$\nabla_x \log p_t(X_t|X_0) = \nabla_x \log \mathcal{N}(X_t|\alpha_t X_0, \sigma_t) = -\frac{X_t - \alpha_t X_0}{\sigma_t^2}, \tag{5.7}$$

and this quantity is exactly the rescaled noise $\frac{1}{\sigma_t}\epsilon$ such that $X_t = \alpha_t X_0 + \sigma_t \epsilon$! The first generation of diffusion models trained a neural network $\hat{\epsilon}(X_t, t)$ on this so-called "noise prediction" objective. However, re-writing the equation as $\hat{X_0} = \frac{1}{\alpha_t}(X_t - \sigma_t \hat{\epsilon})$ shows that the predicted clean image becomes arbitrarily sensitive to the network's output as $t \to T$ since $\frac{1}{\alpha_t} \to \infty$. This should come as no surprise, as denoising an image consisting of almost entirely noise is obviously ill-posed. Once the backward SDE is discretised, these errors cause

an immediate distributional error during the first step of the backward process due to the division by $\alpha_T$. Public implementations often mask this effect by simply clipping the perturbation; this is clearly illegitimate and disfigures the initial Gaussian distribution into an unpredictable distribution which the network must now denoise in the remaining steps. Once researchers became aware of these discretisation problems, two new objectives rapidly formed: *x-prediction*, and *v-prediction*. The former tasks a denoising neural network $D(X_t, t)$ with ingesting a noisy image $X_t$ and returning its best prediction of the clean image $X_0$. This solves the previous issue, but introduces another at $t = 0$: now the score function divides by $\sigma_0$, which causes the same issue. The $v$-prediction target solves both these problems by combining them into a joint objective $v = \alpha_t \epsilon - \sigma_t X_0$ and is reported by many practitioners to yield superior results. Nevertheless, $x$-prediction remains popular since the final denoising step can be made irrelevant by ensuring that $\dot{\sigma}_t$ remains bounded as $t \to 0$. In fact, the most performant noise schedules choose $\sigma_t \approx t$ as $t \to 0$ due to improved stability of the network's predictions [Kar+22]. For this reason, we also choose $x$-prediction.

We set $\alpha_t = 1 - t^2$ and $\sigma_t^2 = 1 - \alpha_t = t^2$, which is known as the *variance-preserving* noise schedule with linear standard deviation in time. This is a competitive noise schedule and ensures a terminal SNR of zero unlike many other noise schedules. Critically, $\alpha_t$ is locally parabolic at time zero, which eliminates the denoising problem discussed previously [Kar+22]. One of the core insights in this landmark paper is that implementations which use a linear *variance* in time - like the OU process - are wasteful since the errors made by the neural network concentrate at time zero.

Finally, it remains to discretise the reverse SDE and sample from it. Let us introduce an extra hyperparameter $\lambda \in [0, 1]$ and modify the backward SDE to

$$d\tilde{X}_t = -\tilde{F}_t \tilde{X}_t dt + \frac{1+\lambda^2}{2} \tilde{G}_t \tilde{G}_t^T \nabla_x \log p_t(\tilde{X}_t) dt + \lambda \tilde{G}_t d\tilde{W}_t. \tag{5.8}$$

It is easily verified that the Fokker-Planck equation of this SDE is independent of $\lambda$, meaning that the marginal distributions of $X_t$ are also independent of $\lambda$. In particular, $\lambda = 1$ corresponds to the usual backward SDE while $\lambda = 0$ corresponds to a deterministic ODE, the so-called *probability flow ODE*. Both equations are valid, but the presence of stochasticity ($\lambda > 0$) has been shown to produce superior sample quality due to its tendency to correct discretisation errors [Kar+22]. We also find that $\lambda = 1$ yields the best results. A valid discretisation of the SDE above is the following DDIM scheme from [Kar+22]

$$X_{t-\Delta t} = \frac{\sigma_{t-\Delta t}^2}{\sigma_t^2} X_t + (1 - \frac{\sigma_{t-\Delta t}^2}{\sigma_t^2}) D(X_t, t) + \frac{\sigma_{t-\Delta t}}{\sigma_t} \sqrt{\sigma_t^2 - \sigma_{t-\Delta t}^2} \epsilon, \tag{5.9}$$

where $\epsilon$ is a $\mathcal{N}(0, 1)$-distributed random variable. In the case $\lambda = 0$, this DDIM sampler is far more accurate than anticipated; it was discovered in [ZC23] that this DDIM scheme is in fact the exponential integrator of the probability flow ODE under the variance-preserving schedule, and therefore yields better samples with fewer time-steps compared to the Euler-Maruyama scheme proposed in the first paper on diffusion models (DDPM) [HJA20]. Much of the research on diffusion models has focused on obtaining SDEs and samplers that can yield accurate samples with as few neural network evaluations as possible since calling the

denoiser is by far the most expensive part of the process. However, this will not be a problem since we deal with small models. In our case, 1000 time steps are enough to yield convergence of generated samples.

### 5.2 U-Net Architectures

Designing highly performant denoising networks capable of approximating $D(X_t, t) \approx X_0$ is a challenging task. The network must be able to handle noise at wildly different levels reflecting signal-to-noise ratios between 0 and $\infty$. Near $t = 0$, the network operates close to the identity function as the signal-to-noise ratio is high. Meanwhile, near $t = 1$, the network mainly draws its predictive power from inductive biases since there is almost no signal to recover. The dominant architecture in the diffusion community has been the *U-Net*, a multiscale residual convolutional neural network originally designed for image segmentation. Residual layers allow the network to operate near the identity at $t \approx 0$, while the multiscale convolutional blocks impose a strong convolutional inductive bias near $t \approx 1$. While it would make sense to utilise separate networks for the high and low SNR regimes, the diffusion community has settled on *time-embedding* to handle these two vastly different regimes. Commonly, the network ingests the time parameter by embedding from one dimension to the number of convolutional and residual channels, and then multiplying or adding to all convolutional outputs and residual outputs channel-wise. This allows the network to learn functions near the identity at $t \approx 0$ by blocking convolutional layers, and to generate sensible outputs at $t \approx 1$ by fully blocking residual connections and returning the noise filtered through a cascade of convolutional blocks.

The final architecture we use for unconditional generation of $16 \times 32$ images is a U-Net with two downsampling layers, depicted in figure 6. The input of shape $(d, 1, 16, 32)$ - by which we mean a batch of $d$ images with 1 channel, height 16, and width 32 - is processed by two convolutional layers with 64 channels and $5\times5$ kernels. Large kernel sizes promote smoothness which is desirable for volatility surfaces. Each of the downsampling blocks performs a $5\times5$ convolution followed by a SiLU non-linearity and $2\times2$ max pooling to halve image dimensions and provide further non-linearity. The upsampling blocks reverse this halving of the image dimensions by performing bilinear interpolation of pixels to double dimensions, followed by another convolution and SiLU. At all scales, time-embeddings map $t$ to the corresponding channel dimension using feedforward networks, and multiply each channel output. Residual connections are also multiplied channel-wise by time-embeddings. The final layer consists of more convolutions to collapse the 64 convolutional channels back to a shape of $(d, 1, 16, 32)$. See github.com/JPNotleks/VolaDiff for a compact implementation of this architecture.

Most of the modelling choices in the architecture above stem from the observation that many public implementations of U-Nets for diffusion yield rather poor image predictions. Of course it is ill-defined what constitutes objectively superior modelling choices, but by considering the inductive bias that U-Nets should impose - namely the characteristic smoothing of multiscale convolutions - three overarching trends emerge that lead to these issues:

- First, networks that inject time embeddings by *adding* to convolutional outputs (effectively adding a single-colour image to the layers) perform worse than models that use multiplication. Given that the network must interpolate between the identity at

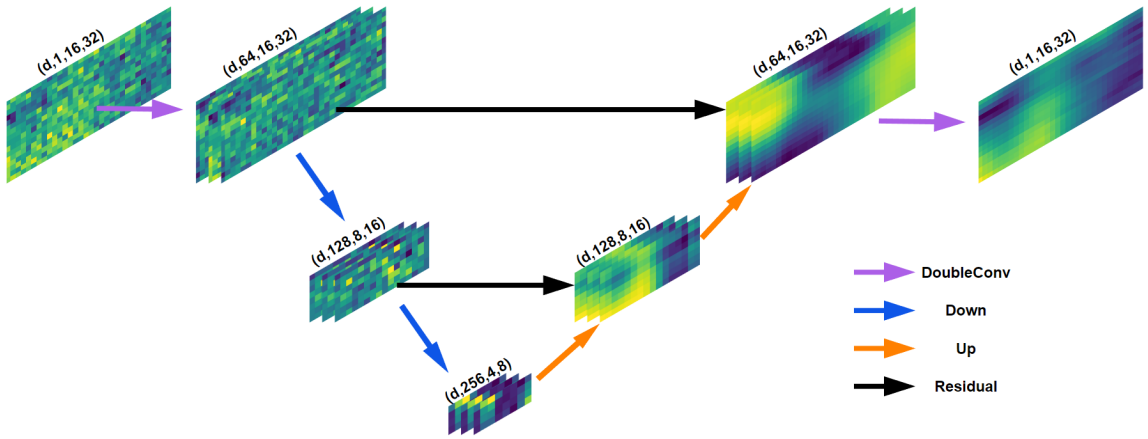



Figure 6: The final U-Net architecture depicting the processing of an actual implied vol surface. The structure of the output image is visible in the noisy input. Multiplication by time-embeddings is suppressed.

$t = 0$ and a complex convolutional network at $t = 1$, this should be no surprise, since multiplication of residual and convolutional outputs yields far better control over this distinction. Addition does not seem to afford sufficient model capacity to cover the entire range $t \in [0, 1]$.

- Upsampling blocks that use transposed convolutions yield striped checkerboard artifacts. This issue was already known in 2016, and we refer to [ODO16] for a clear explanation of this phenomenon. It is also trivial to fix by replacing transposed convolution layers with a bilinear upsampling layer followed by a standard convolutional layer. Furthermore, many public diffusion U-Nets do not use proper padding at image edges - the PyTorch default is automatic zero-padding which yields severe image border artifacts. Reflection padding solves this issue and is a more natural choice since reflecting an i.i.d. Gaussian sample image yields a Gaussian sample too (though independence is lost).

- Finally, some implementations utilise self-attention mechanisms without positional encoding. QKV attention is inherently permutation-equivariant, meaning that permuting the pixel inputs to the attention layer permutes the pixel outputs in the same way. This leads to a complete loss of spatial coherence and we find that self-attention without positional encoding damages performance, while incorporating positional encoding corrects this and yields improved performance.

### 5.3 Classifier-free Guidance on Alpha Signals

So far, we have discussed how diffusion models learn to sample from a distribution $p_0 \in \mathcal{P}(\mathbb{R}^n)$, known as *unconditional* generation. The first breakthrough in *conditional* generation (sampling from some class $c$) was the invention of classifier guidance [DN21], where a classifier gradient is added to the learnt score function $\epsilon_\theta$:

$$\tilde{\epsilon}_\theta(X_t, t, c) = \epsilon_\theta(X_t, t) + w\nabla_{X_t} \log p_{\tilde{\theta}}(c|X_t) = \nabla_{X_t} \left( \log p_\theta (X_t, t) + w \log p_{\tilde{\theta}} (c|X_t) \right). \quad (5.10)$$

Consequently, sampling from this model yields approximate samples from

$$\tilde{p_\theta}(X_0|c) \propto p_\theta(X_0) p_{\tilde{\theta}}(c|X_0)^w.$$

Higher values of the weighting $w > 0$ lead to a higher concentration of samples that the classifier considers to belong to class $c$. Even though this yields a method to successfully generate conditional samples, training a classifier on noisy samples is unstable. Nowadays, the preferred method to condition a diffusion model is classifier-free guidance, which trains a denoising network while feeding it either the class label of the target image, or a null token [HS22]. As such, the model simultaneously learns to denoise towards generic samples when fed the null token, or specific classes when fed the class token. At sampling time, the denoiser used to predict the clean image is then taken to be

$$\hat{X_0} = (1 + w)D(X_t, t, c) - wD(X_t, t, \emptyset). \quad (5.11)$$

This means that during sampling, the image is guided towards samples from class $c$ while being pushed away from generic samples. The weighting $w$ is a hyperparameter which trades adherence to the label $c$ for sample diversity.

Conditioning was originally introduced for discrete classes with $n$ labels. The standard approach to condition the U-Net on a class is to perform a one-hot encoding of the classes (yielding a basis vector in $\mathbb{R}^n$ for each class) followed by a linear layer to the desired embedding dimension. This embedding is then treated identically to the time embedding discussed previously. We may extrapolate this idea to *continuous* conditioning vectors $c \in \mathbb{R}^m$ in the same way: ensure reasonable scaling and orthogonality of the conditioning vectors - for example by pre-whitening them - and then embed into a high dimension with feedforward networks. Conditioning on the null token $\emptyset$ is intricate, since it cannot be treated as a class unlike in discrete conditioning. While training a batch of $d$ images, we feed the model a random mask $c_{\text{mask}} \in \mathbb{R}^d$ which is i.i.d. Bern(0.9)-distributed, and multiply all condition embeddings by this mask, effectively zeroing all conditioning embeddings 10% of the time. Consequently, the model cannot access any conditioning information and learns to denoise without guidance.

### 5.4 Training Improvements and Variance Reduction

Diffusion models exhibit notoriously strange training behaviour where the sample quality often improves even though the loss function does not [Kar19]. Presumably, this is due to the ability of the backward process to correct model errors due to its Langevin-type mixing. Furthermore, diffusion model loss curves have a huge variance which obfuscates whether the model is making any progress. A popular solution to this problem - which is in fact a trick recommended in section 7.2 of the original Adam paper [KB17] - is to save an exponential moving average of training weights during training. The smoothed model generally vastly outperforms any checkpoint of the raw model, but the optimal smoothing filter is a matter of current research. We choose a power law filter recently proposed by [Kar+24] which approximately averages the model weights of the last 10% of the training time, irrespective of the scale of the training time. Finally, the best model with the lowest smoothed loss is selected for the sampling process. Figure 7 depicts the loss function and its smoothed version during an average training run, as well as the checkpoint where we harvest the best smoothed model. See appendix A for the diffusion flow of one sample.

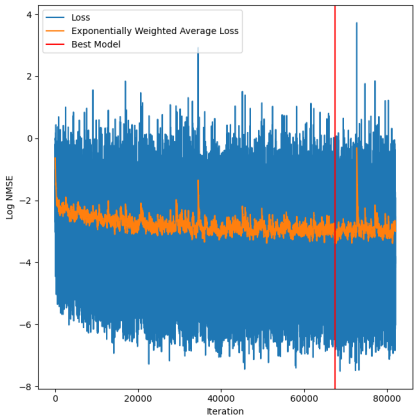



Figure 7: The log of the normalised mean square error for an average training run. The exponential moving average tracks a smoothed error and selects the best model a posteriori.

## 6 Testing VolaDiff

We test VolaDiff in two ways: unconditional and conditional generation. Investigating unconditional generation shows whether the model is capable of learning the underlying distribution of volatility shocks in the absence of conditioning information. Moreover, any generative model should be capable of performing meaningful generalisation; the model should generate samples which are reasonably out-of-distribution. This is subjective and tunable by modifying the model's capacity, but we show that VolaDiff is capable of matching the first few principal components of the data distribution while generating novel samples by up-weighting the explained variances of the higher-order principal components. In the presence of conditioning information in the form of price forecasts, VolaDiff correctly learns the correlation between the SPX and VIX but is also capable of generating the 'outliers' found in the dataset of early 2024 where price increases occur jointly with volatility increases and vice versa.

### 6.1 Unconditional Generation

First, we test the ability of our chosen model to generate meaningful samples of the volatility shock. We train an unconditional model on all volatility surfaces in 2023 with the Adam

optimiser for 3000 epochs with a batch size of 3 - diffusion models require surprisingly long training times [Kar19]. The model reaches a minimum exponentially weighted normalised mean square error of 3.6% during training, meaning that a random sample from $p(X_t, t)$ for uniformly random $t \in [0, 1]$ is reconstructed on average with 96.4% of the energy. We try several public U-Net implementations on GitHub and find our chosen architecture to be the best; others manage to achieve an NMSE of 4%-12%. It is unclear whether this is due to the rigorous design choices we made, or simply due to the difficulty of comparing different models. We then generate 1000 samples and compare the PCA of these generated samples to the empirical samples. Principal components are found to be stable across different runs.

Figure 8 contrasts the first five generated and empirical principal components. The principal components suggest sensible generalisation: while they are macroscopically similar, the diffusion model's ability to generate novel samples distorts them slightly. The discrepancy between the second and third principal components is caused by the similarity of their empirical explained variances of 11% and 10%, causing the model to confuse them. Interestingly, the fourth and fifth components are still reproduced almost perfectly with smoothing, despite low explained variances of 5% and 3%, respectively.

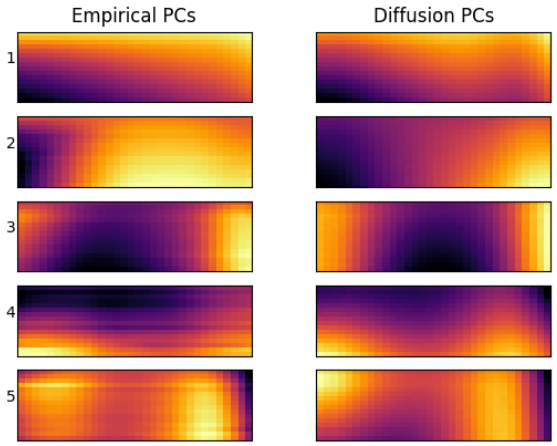



Figure 8: The first five principal components of the empirical and diffusion samples.

It is worth noting that the explained variances of the higher-order principal components are significantly higher for the diffusion model than for the empirical samples - consistently almost twice as high. This is likely due to the ability of the diffusion model to produce a meaningful manifold-like interpolation of the data points, which necessitates higher weightings of low-energy principal components.

### 6.2 Conditional Generation on Alpha Signals

Since our chosen model architecture is capable of ingesting arbitrary conditioning vectors, it may be fed with proprietary alpha signals to predict the conditional mean of tomorrow's volatility surface. We demonstrate the ability of VolaDiff to condition on alpha signals in the form of scenario forecasting: if we knew tomorrow's SPX return, how would the volatility surface change?

For this purpose, we train a conditional model ingesting only one conditioning scalar, tomorrow's return. Since the VIX reacts with a strong negative correlation to SPX returns, we expect the volatility shocks to be positive on days with negative returns and vice versa. For every day in 2023, we condition the model on tomorrow's return and generate 1000 samples with guidance strength $w = 2$ which is a common choice for diffusion models.

Unfortunately, no matter the hyperparameters tested, the model collapses during sampling and does not generate sensible volatility shocks. Moreover, partitioning the surfaces into two classes - positive and negative days - and then performing standard, discrete class conditioning yields the same problem, even using a model which we verified extensively to work on MNIST. This suggests that the dataset is too small, and the complexity too high, for the model to learn.

Instead, we suggest conditioning on arbitrary alpha signals in the following way: select the target condition vector, then k-NN search the training set for datapoints whose labels are close to the desired condition vector, and run the backward sampling process with these datapoints as the target (instead of the denoiser $D(X_t, t)$) for around half the sampling time. Finally, run the other half of the backward process using the denoiser. The exact split is a hyperparameter and we note that it is highly dependent on the noise schedule and distribution of the dataset. Geometrically, running part of the backward process with the 'true' target $X_0$ from $T$ to $t$ maps the terminal Gaussian distribution to the closed-form marginal $p(X_t|X_0)$ up to negligible discretisation errors. From that point onward, the denoiser can deform the closed-form marginal into its predicted distribution $\hat{X}_0$ while performing sensible generalisation. This form of conditioning inherits the generalisation of the unconditional model while having the ability to condition on chosen data points in a tunable way. We find this strategy to work extremely well. Figure 10 illustrates the effect of the percentage of time spent denoising with the true target on the conditional mean of the predicted distribution (1000 samples each). As the time spent denoising with the true targets changes, the conditional mean continuously interpolates between the unconditional mean and the fully conditional mean.

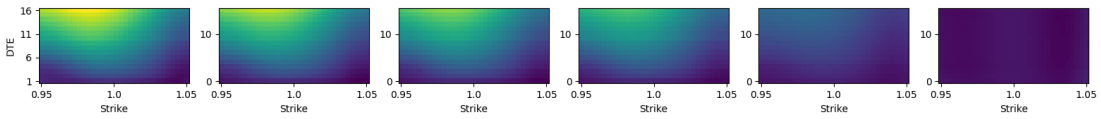



Figure 9: Conditional means when targeting the 20% most negative return days in 2023. From left to right: 100%, 80%, 60%, 40%, 10%, 0% of time denoised with the target. The rightmost figure is the unconditional mean and the leftmost is the conditional mean of the target points. Note the characteristic put volatility increase on strongly negative days near the top left.

Because the conditioning set can be chosen arbitrarily, as well as the conditioning strength, VolaDiff can condition on arbitrary signals tagged to the volatility surfaces. Further promising conditioning signals may include features derived from today's 1DTE volatility curve since this is likely to have a disproportionately large causal effect on tomorrow's volatility shock. Another signal could be the ratio of a proprietary price forecast and the VIX1D as a measure of the surprise of the return magnitude, or market maker gamma exposure since this affects 1-day realised volatility. Finally, one might consider conditioning on volatility surfaces which are close (in $L^2$ for instance) to today's surface to learn from similar market conditions in the past.

# 7 Conclusion and Further Research

Volatility dynamics have become significantly more complex since the introduction of 0DTE SPX options in 2022. Macroeconomic events priced into the option surface cause complex implied volatility shapes whose evolution requires disentangling volatility and time-to-maturity. We proposed a simple model splitting the evolution of the volatility surface into its endogenous maturation and an exogenous volatility shock. By training an unconditional diffusion model, and developing a novel conditioning method, we showed that VolaDiff accurately learns the distribution of volatility shocks and is able to generate realistic volatility shocks conditioned on arbitrary alpha signals. Given that the volatility shock does not model the 1DTE volatility curve but merely the relationships between longer maturities, VolaDiff may be used for pure vega strategies and scenario-forecasting.

A possible extension of VolaDiff could be modelling the 1DTE volatility curve simultaneously. In view of the complexity of the volatility shock surfaces, it is not clear how to jointly model these. One could simply add the 1DTE volatility curve to the volatility chock surface, but this will mix two different concepts - the evolution of volatilities relative to one another, and the pricing of an absolute volatility. Furthermore, one may extend VolaDiff to high-frequency volatility data and study the evolution of volatilities during market events which are characterised by widening bid-ask spreads, followed by a jump discontinuity in volatilities as the event passes. Alternatively, one might focus on improving the model capacity of VolaDiff, or its ability to fit small datasets. A particularly promising class of diffusion models well-suited towards fitting small datasets are blurring diffusion models [HS24; Dar+22; RHS23] as they have been shown to provide sensible generalisation given only 20 training samples on MNIST.

## References


[BL78] Douglas T. Breeden and Robert H. Litzenberger. "Prices of State-Contingent Claims Implicit in Option Prices". In: *Journal of Business* 51.4 (1978), pp. 621–651.

[HP86] U. G. Haussmann and E. Pardoux. "Time-Reversal of Diffusions". In: *The Annals of Probability* (1986).

[CD02] Rama Cont and José Da Fonseca. "Dynamics of Implied Volatility Surfaces". In: *Quantitative Finance* (2002). DOI: 10.1088/1469-7688/2/1/304.

[Lee02] Roger Lee. "The Moment Formula for Implied Volatility at Extreme Strikes". In: *Mathematical Finance* (2002).

[DS04] Freddy Delbaen and Walter Schachermayer. "What Is a Free Lunch?" In: *Notices of the AMS* 51.5 (2004), pp. 526–528. URL: https://www.ams.org/notices/200405/.

[SW08] Martin Schweizer and Johannes Wissel. "Arbitrage-free market models for option prices: the multi-strike case". In: *Finance and Stochastics* 12.4 (2008), pp. 469–505. DOI: 10.1007/s00780-008-0068-6.

[Fen09] Matthias R. Fengler. "Arbitrage-free smoothing of the implied volatility surface". In: *Quantitative Finance* 9.4 (2009), pp. 417–428. DOI: 10.1080/14697680802595585. URL: https://www.tandfonline.com/doi/abs/10.1080/14697680802595585.

[Vin11] Pascal Vincent. "A Connection Between Score Matching and Denoising Autoencoders". In: *Neural Computation* 23.7 (2011), pp. 1661–1674. DOI: 10.1162/NECO_a_00142.

[Gat12] Jim Gatheral. "Arbitrage-free SVI volatility surfaces". In: (2012). URL: https://mfe.baruch.cuny.edu/wp-content/uploads/2013/01/OsakaSVI2012.pdf.

[Mon13] Jean-Baptiste Monnier. *Risk-neutral density recovery via spectral analysis*. Tech. rep. Accessed: 2024-12-07. arXiv, 2013. URL: https://arxiv.org/abs/1302.2567.

[ODO16] Augustus Odena, Vincent Dumoulin, and Chris Olah. "Deconvolution and Checkerboard Artifacts". In: *Distill* (2016). DOI: 10.23915/distill.00003. URL: http://distill.pub/2016/deconv-checkerboard.

[KB17] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: 1412.6980.

[Kar19] Andrej Karpathy. *A Recipe for Training Neural Networks*. 2019. URL: https://karpathy.github.io/2019/04/25/recipe/#6-squeeze-out-the-juice.

[ATV20] Damien Ackerer, Natasa Tagasovska, and Thibault Vatter. "Deep Smoothing of the Implied Volatility Surface". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 11552–11563. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/858e47701162578e5e627cd93ab0938a-Paper.pdf.

[CRW20] Samuel N. Cohen, Christoph Reisinger, and Sheng Wang. "Detecting and Repairing Arbitrage in Traded Option Prices". In: *Applied Mathematical Finance* 27.5 (Sept. 2020), pp. 345–373. ISSN: 1466-4313. DOI: 10.1080/1350486x.2020.1846573. URL: http://dx.doi.org/10.1080/1350486X.2020.1846573.

[HJA20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. "Denoising Diffusion Probabilistic Models". In: *CoRR* abs/2006.11239 (2020). URL: https://arxiv.org/abs/2006.11239.

[Wan20] Victor Wang. *arbitragerepair*. Python modules and Jupyter notebook examples for the paper "Detecting and Repairing Arbitrage in Traded Option Prices". 2020. URL: https://github.com/vicaws/arbitragerepair.

[Ber+21] Maxime Bergeron et al. *Variational Autoencoders: A Hands-Off Approach to Volatility*. 2021. arXiv: 2102.03945.

[DN21] Prafulla Dhariwal and Alex Nichol. *Diffusion Models Beat GANs on Image Synthesis*. 2021. arXiv: 2105.05233.

[Son+21] Yang Song et al. *Score-Based Generative Modeling through Stochastic Differential Equations*. 2021. arXiv: 2011.13456.

[Dar+22] Giannis Daras et al. *Soft Diffusion: Score Matching for General Corruptions*. 2022. arXiv: 2209.05442.

[GL22] Julien Guyon and Jordan Lekeufack. "Volatility Is (Mostly) Path-Dependent". In: *SSRN* (2022).

[HS22] Jonathan Ho and Tim Salimans. *Classifier-Free Diffusion Guidance*. 2022. arXiv: 2207.12598.

[Kar+22] Tero Karras et al. *Elucidating the Design Space of Diffusion-Based Generative Models*. 2022. arXiv: 2206.00364.

[KFL22] Dohyun Kwon, Ying Fan, and Kangwook Lee. *Score-based Generative Modeling Secretly Minimizes the Wasserstein Distance*. 2022. arXiv: 2212.06359.

[Nin+22] Brian Ning et al. *Arbitrage-Free Implied Volatility Surface Generation with Variational Autoencoders*. 2022. arXiv: 2108.04941.

[Che+23] Jacky Chen et al. "A Variational Autoencoder Approach to Conditional Generation of Possible Future Volatility Surfaces". In: *SSRN* (2023).

[Che23] Ting Chen. *On the Importance of Noise Scheduling for Diffusion Models*. 2023. arXiv: 2301.10972.

[CGL23] Ying Chen, Maria Grith, and Hannah Lai. "Neural Tangent Kernel in Implied Volatility Forecasting: A Nonlinear Functional Autoregression Approach". In: *SSRN* (2023).

[CRW23] Samuel N. Cohen, Christoph Reisinger, and Sheng Wang. "Arbitrage-Free Neural-SDE Market Models". In: *Applied Mathematical Finance* 30.1 (2023), pp. 1–46. DOI: 10.1080/1350486X.2023.2257217. URL: https://econpapers.repec.org/RePEc:taf:apmtfi:v:30:y:2023:i:1:p:1-46.

[RHS23] Severi Rissanen, Markus Heinonen, and Arno Solin. *Generative Modelling With Inverse Heat Dissipation*. 2023. arXiv: 2206.13397.

[Thi23] Alexandre Thiérry. *Reversing a Diffusion*. 2023. URL: https://alexxthiery.github.io/posts/reverse_and_tweedie/reverse_and_tweedie.html.

[VC23] Milena Vuletić and Rama Cont. "VolGAN: a generative model for arbitrage-free implied volatility surfaces". In: *SSRN* (2023).

[Zai23] Jonathan Zaionz. "The Rise of SPX 0DTE options". In: (2023). URL: https://go.cboe.com/l/77532/2023-08-11/ffh41m/77532/1691776633zvMs4Hml/The_Rise_of_SPX___0DTE_Options_WP_v1.0.4___Final.pdf.

[ZC23] Qinsheng Zhang and Yongxin Chen. *Fast Sampling of Diffusion Models with Exponential Integrator*. 2023. arXiv: 2204.13902.

[GJW24] Lukas Gonon, Antoine Jacquier, and Ruben Wiedemann. *Operator Deep Smoothing for Implied Volatility*. 2024. arXiv: 2406.11520.

[HS24] Emiel Hoogeboom and Tim Salimans. *Blurring Diffusion Models*. 2024. arXiv: 2209.05557.

[Kar+24] Tero Karras et al. *Analyzing and Improving the Training Dynamics of Diffusion Models*. 2024. arXiv: 2312.02696.

[Lin+24] Shanchuan Lin et al. *Common Diffusion Noise Schedules and Sample Steps are Flawed*. 2024. arXiv: 2305.08891.

# A Appendix A

The following figure visualises the flow of one sample from i.i.d. Gaussian noise to denoised image. Note that the generation is oversampled 10x here, yielding 10 denoising steps between the final two images. This is where some diffusion implementations do not expend the effort to denoise in sufficiently small time steps to result in clean images. More samples are available on GitHub.

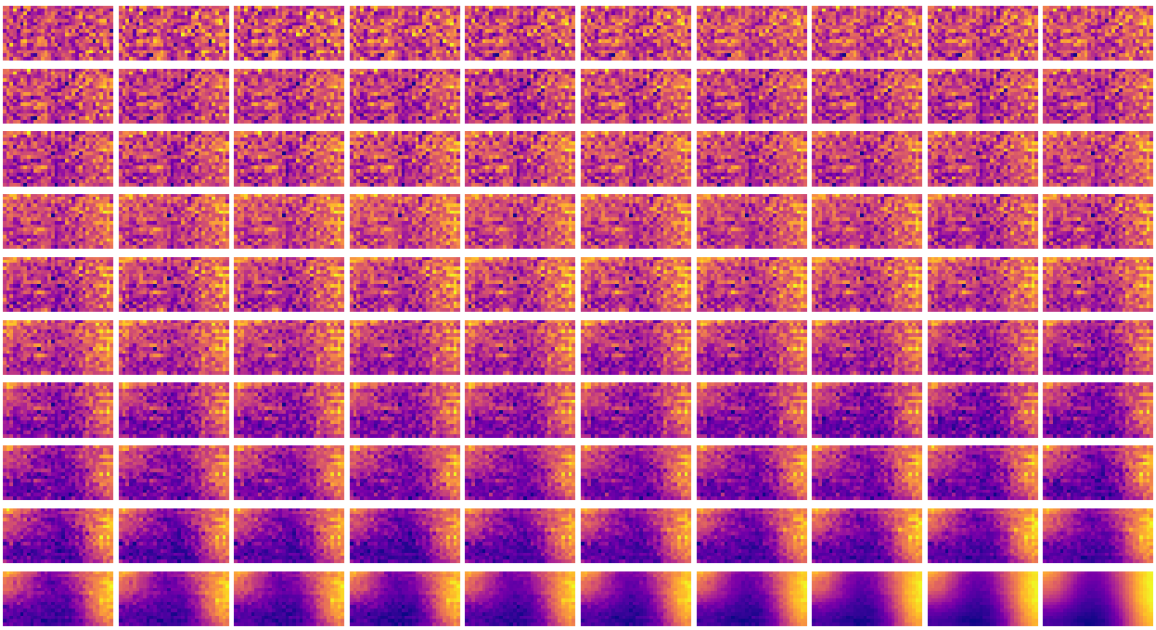


Figure 10: Snapshots of the latent state during the sampling process, to be read from left to right and top to bottom.