DYSANOS

Generative Dynamic Smooth Arbitrage-free Non-parametric Option Surfaces

Hans Buehler* Blanka Horvath† Anastasis Kratsios‡

August 3, 2026 (working draft)

Abstract

This article presents with DYSANOS a model framework for realistic generation of paths of daily smooth option surfaces for all strikes and expiries which are free of static arbitrage, and for which we therefore can find a change to a risk-neutral measure. DYSANOS is therefore the first production-grade "market model" for surfaces of European options. Our model is designed to generate entire path of daily spot and option prices for years in the future. We capture two days to expiry up to any expiry further out. In our experiments we use up to one year.

We discuss model setup, data pipeline, and training. We illustrate model performance using a simplistic autoregressive hidden state process on Option Metrics' IvyDB S&P Index data from 2020 to 2025.

Contents

*Mathematical Institute and Oxford-Man Institute, University of Oxford. hans.buehler@maths.ox.ac.uk
†Mathematical Institute and Oxford-Man Institute, University of Oxford. blanka.horvath@maths.ox.ac.uk
‡McMaster University and Vector Institute.

1 Introduction

The implied volatility surface of an option market changes dynamically over time, giving rise to Vega risk when risk managing option portfolios. Modelling the dynamics of a full continuous surface of options in a way that is simultaneously *statistically realistic*, *statically arbitrage-free* at every simulated date, and *dynamically arbitrage-free* in the sense that simulated daily paths for long periods over are consistent with an equivalent risk-neutral measure has remained an open problem since the foundational work of [CFD02] on dynamics of the implied volatility surface.

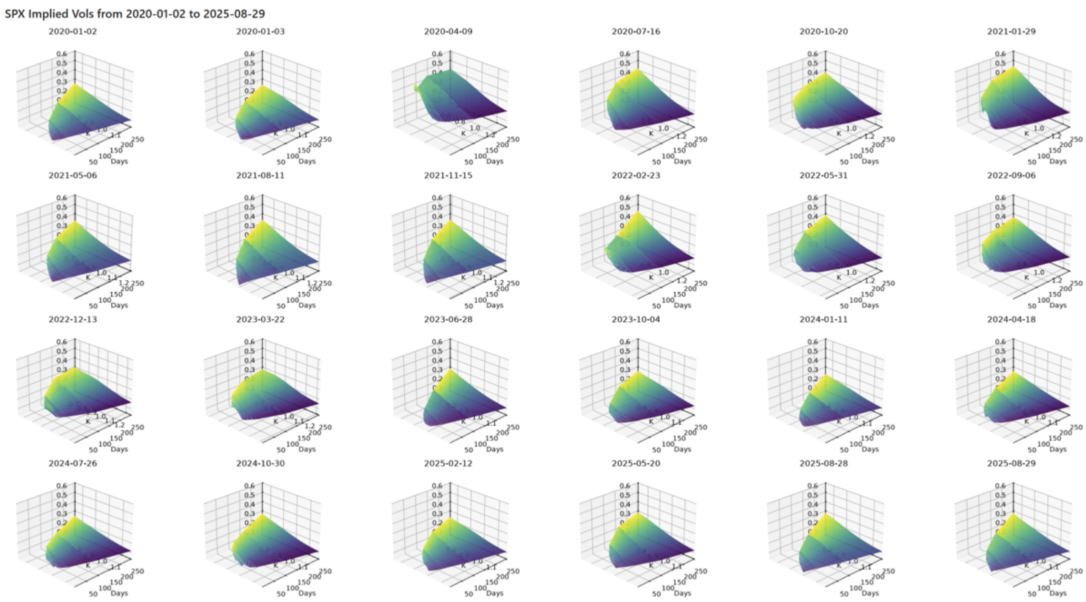

Figure 1: Historical implied volatilities for mid-prices of S&P 500 Index options from 2020-01-02 to 2025-08-29 using Option Metrics Ivy DB data via WRDS `https://wrds-www.wharton.upenn.edu/` (the latest data available at the time of writing).

The most advanced approach until recently were the simulators [WBWB19, WWP+21, CRW23] for realistic dynamics of a *discrete floating grid* of options which are strictly arbitrage-free. These simulators are using, in essence, the discrete local volatility representation pioneered in [BR15] to ensure absence of static arbitrage in every simulated option surface.

In addition, [BMPW22] provides a general framework how to change from the statistical measure to a risk-neutral measure, including in the presence of transaction cost. This is a generalization of the seminal work on Minimal Entropy Martingale Measures by [Fri00].

However, the restriction of the above models to a discrete set of floating option strikes and time-to-expiries means that the simulators do not have access to the daily returns $dC_t(\tau, k)$ of an actual tradable instruments with cash strikes k and calendar expiries τ (because the contract terms change from one simulated day to another). That means

- Hedging instruments chosen at simulation time determine the hedging instruments available to any hedging algorithm.
- For hedging algorithms such as Deep Hedging [BGTW19] for which we solve optimal trading along full paths, the calculation of the return of a hedging instrument requires the simulation of the spot until the expiry of the farthest option. This limits the expiries of tradable instruments made available during simulation.
- The approach is unsuitable for "daily" hedging algorithms such as Statistical Hedging [Bue17] (a generalization of local regression hedging) or Deep Bellman Hedging [BPW22] (the "actor

critic" version of Deep Hedging) which we wish to execute on a simulated path to hedge exotics to expiry – in both cases the P&L of option returns is central to the approach; see also the lecture notes [Bue26].

Of course, since the discrete option prices are free of arbitrage, one option would be to interpolate option prices linearly, which means all options are free of static arbitrage. However, as shown in [Bue06] and illustrated in [BHK+26] linear interpolation yields the most expensive possible arbitrage-free price for any option off the grid, rendering the approach unsuitable in practise.

This article presents a model framework for

- **(i) realistic generative dynamics** of
- **(ii) smooth option surfaces for all strikes and expiries** which are
- **(iii) free of static arbitrage** and for which we therefore can find a change to a
- **(iv) risk-neutral** measure.

1.1 Historic Context and Literature Review

The classical approach and its limitations. Quantitative finance has traditionally approached this problem by specifying analytically tractable risk-neutral dynamics for the underlying asset — Heston, rough volatility, SABR — and then *calibrating* model parameters to fit each day's option surface independently. In this paradigm, realistic surface dynamics are an afterthought: calibration is repeated from scratch on each date, and inter-day consistency is achieved, if at all, via a local-volatility overlay such as the seminal [GHL12] that lacks interpretable dynamics. This approach is both impractical for large-scale simulation and structurally inconsistent with the goal of generating realistic implied volatility or option price paths.

Why simulating directly in implied-volatility space fails. A natural data-driven alternative is to treat the implied volatility surface itself as the simulation object, much as implied in [CFD02]. The fundamental difficulty is that there is no known parametrization of the implied volatility surface which are free of static arbitrage. If a simulated surface admits a butterfly spread or calendar spread arbitrage, then no equivalent risk-neutral measure can exist, and numerical procedures for finding one — including the Deep Hedging drift-removal technique of [BMPW22] — fail numerically.

In [CV23] the authors proposed to simply zero the weights of paths where this exist. However, detection of arbitrage.

In addition, there are number of recent works which use machine learning methods to learn dynamics of implied volatilities who attempt to impose no-arbitrage via soft constraints; none of them strictly excludes arbitrage. Here, we aim higher and present a framework which does not require minutiae daily guesses on penalty weights and other soft parameters to try avoid static arbitrage along simulate pays of years of implied volatilities.

Also, most published works simplify the problem by only considering artificial expiries with large spacing. We consider real market expiries, in our examples starting from two days to expiry. SANOS was shown in [BHK+26] to be able to fit entire option surfaces from one day to expiry out to several years. Our model here is a generative version thereof.

Relation to Dupire's local volatility. An intuitive alternative is to use Dupire's local volatility as state space instead *(not as dynamical model)*. Given a non-negative local volatility function $\sigma_t(T,K)$ for $(T,K) \in \mathbb{R}^2_{\geq 0}$ defined on some day t we can define a smooth, strictly arbitrage-free option surface $C_t(T,K) := C^{\mathrm{LV}}(\sigma_t;T,K)$ by solving the Dupire forward-PDE $\partial_T C^{\mathrm{LV}}(\sigma_t;T,K) = \frac{1}{2}K^2 dT\, \partial^2_{KK} C^{\mathrm{LV}}(\sigma_t;T,K)$.

However, in order to price any simulated option off a simulated local volatility surface we would need top solve a forward-PDE on a suitably fine grid which is slow when used within batch-based gradient descent training.

The idea to address this conundrum is to use a grid of options and define an discrete version of local volatility on that grid, which is what we called *discrete local volatility (DLV)* in [BR15]. However, its native option prices for strikes not on its base grid are the result of linear interpolation and therefore, as mentioned above, "to expensive". We therefore use our smooth SANOS extension [BHK+26].

Remark 1.1. *We should also note that in order to find a Dupire local volatility surface for giver market data in the first place, we need to fit the market with smooth arbitrage-free option surface parametrization. Of course, that is what SANOS [BHK+26] was made for but once you have SANOS we might as well use it as the ground truth (since the respective Dupire local volatility will just be another representation of the same surface).*

Key contributions

SANOS Decoder:

We show how to implement and learn a coordinate transformation of our SANOS [BHK+26] model such that the state parameters h_t live in a $\mathbb{R}^{n_h}$ for a low-dimensional $n_h \sim 20$. The call prices remain smooth and statically arbitrage free.

This "ML-SANOS" is a natural decoder of a driving state process into an option surface of the form

$$h_t \longmapsto C(h_t; T, K) \ .$$

We show how ML-SANOS can be trained directly to suitably pre-processed option market data, yielding a trained h_t for each day. The pre-processing step means that DYSANOS, via SANOS, is fitted to real market bid/ask prices.

Generative Dynamics

We show that most of the variance of daily changes to the joint space of spot[1] and hidden state can be explained by essentially five parameters, in line with general PCA analysis of the implied vol surface, c.f. [CFD02]. We therefore present a simple baseline version of a generative state space model based on a simple auto-regressive low-dimensional factor model which has in spirit the form

$$d\tilde{h}_{t+1} = (u_0 - \kappa \tilde{h}_t)dt + \left(w\alpha'_t(\omega) + \sqrt{1-w^2} A Y_t(\omega) \right) du_t \sqrt{dt} \tag{1}$$

where $u_0 \in \mathbb{R}^{n_h}$ is the long-term mean, $\kappa \in \mathbb{R}^{n_h \times n_h}$ is the mean-reversion speed and $du_t \in \mathbb{R}^{n_\alpha, n_h}$ are the main PCA factors driving dh_t. The matrix A is the Cholesky decomposition of the historic covariance of the factor loadings $\alpha \in \mathbb{R}^{n_t, n_\alpha}$. Moreover, $Y_t(\omega) \in \mathbb{R}^{n_\alpha}$ is normal and $\alpha_t(\omega) \in \mathbb{R}^{n_\alpha}$ is sampled from historic factor loadings $(\alpha_t)_{t \in \mathcal{T}}$. The bandwidth parameter w can be used to mix between purely historical sampling with $w = 1$ and fully continuous state sampling $w = 0$. (The actual model we present with (18) on page 16 has an additional state $\tilde{h}^0$ without mean-reversion which represents log-spot; it is also simulated using a more robust representation for large time steps.)

We do not claim that this approach provides the most realistic dynamics; rather, we see this implementation as baseline to beat with more modern time series models.

[1] In the current paper we assume, essentially, that interest rates and drifts are deterministic and can therefore be assumed to be zero. This is not a real restriction as extending the state space with forward rates and drifts does not pose any practical hindrances.

Risk-Neutral Density

We show that the absence of static arbitrage for each generated surface means that there exist in theory a risk-neutral measure equivalent to the statistical measure implied by (1).

We show how to apply the techniques pioneered in [BMPW22] to the joint vectors

$$dS_t;\ dC_t(K^1,T^2),\dots,dC_t(K^m,T^m)$$

over $m = 140$ options per time step, and discuss numerical challenges.

2 DYSANOS

2.1 The Market

We consider European call options on a single underlying for historic days $t \in \{t_1,\dots,t_{n_t}\}$. For notational simplicity we assume forwards, discount factors, and default intensities are zero; the extension to non-zero rates and dividends is standard. We comment in remark 2.10 on how to convert market data into driftless data.

We will use boldface for market observables.

We use $\boldsymbol{S}_t$ to denote the (positive) stock price at time t which we assume trades without cost (and therefore has no bid/ask spread). We assume we observe $L_t \in \mathbb{N}$ European call option prices with time-to-expiries $T_t^\ell \geq 0$ and relatives strikes $K_t^\ell \geq 0$ for $\ell = 1,\dots,L_t$ in the market. Their payoff is therefore

$$\left(\frac{\boldsymbol{S}_{t+T_t^\ell}}{\boldsymbol{S}_t} - K_t^\ell\right)^+ .$$

We denote by $\boldsymbol{B}_t^\ell$ and $\boldsymbol{A}_t^\ell$ their bid and ask prices; mid-price and half-spread are defined (in vector notation) as $\boldsymbol{C}_t := \frac{1}{2}(\boldsymbol{A}_t + \boldsymbol{B}_t)$ and $\boldsymbol{\gamma}_t := \frac{1}{2}(\boldsymbol{A}_t - \boldsymbol{B}_t)$. We generally assume that $\boldsymbol{B}_t \succ 0$ and $\boldsymbol{\gamma}_t \succ 0$.[2] Contrary to most derivative literature do not assume that bid/ask prices are in some way free of arbitrage; instead we will show how to handle arbitrage valuations gracefully.

The market Black–Scholes implied mid-volatilities $\boldsymbol{\sigma}_t^\ell$ are given by the unique relation $\mathrm{BSCall}(\boldsymbol{S}_t, K_t^\ell, \boldsymbol{\sigma}_t^\ell\sqrt{T_t^\ell}) = \boldsymbol{C}_t^\ell$ in terms of the Black & Scholes call price formula

$$\mathrm{BSCall}(S,K,W) := S\,\mathcal{N}(d_+) - K\,\mathcal{N}(d_-) \quad \text{for} \quad d_\pm := \frac{\log S/K}{W} - \frac{1}{2}W^2 . \tag{2}$$

where $\mathcal{N}$ denotes the normal distribution function.

2.2 Arbitrage-Free Call Price Functions

A call price function is simply a function $C : \mathbb{R}^2_{\geq} \longrightarrow \mathbb{R}_{\geq 0}$ which computes call prices $C(T,K)$ for time-to-expiries $T \geq 0$ and relative strikes $K \geq 0$ (i.e. the call prices are normalized to a spot price of 1). The objective of this section and the next is to generate such functions C_t for each simulated date t such that each C_t is "free of static arbitrage" in the following sense:

Definition 2.1. *A call price surface $C : \mathbb{R}^2_{\geq 0} \to \mathbb{R}_{\geq 0}$ is* statically arbitrage-free *or* free of static arbitrage *if there is no trading strategy which has non-negative returns with a non-zero probability of positive returns.*

Theorem 2.2. *A call price surface C is free of static arbitrage if and only if it does not admit negative butterfly or calendar spreads and satisfies the boundary conditions $C(T,0) = 1$, $C(0,K) = (1-K)^+$, $\lim_{K\uparrow\infty} C(T,K) = 0$ and $\partial_K C(T,0) \in [-1,0)$.*

The stricter condition $\partial_K C(T,0) = -1$ applies if the stock price cannot become zero.

[2] We define $x \succ y \Leftrightarrow \forall i : x_i > y_i$".

This has been proven in [BR15]. Here we give a practical guide on how to construct *actionable* arbitrage strategies if any of the above are violated. The simplicity of each of them illustrates the risk of admitting arbitrage in generative environments which are used to train agents to hedge: if such opportunities exist, an optimal agent will try learning them.

Remark 2.3 (Actionable arbitrage if the conditions of theorem 2.2 are not met)**.** *All of the below are meant as practical heuristics.*

- ***Butterfly arbitrage:*** *this means that the price of a positive butterfly payoff is negative. The arbitrage is to get paid to enter into the butterfly which can only ever generate a positive return.*

- ***Calendar arbitrage:*** *if for the same strike K the call price at T_2 is below the call price at T_1, then:*

  1. *Sell the call at T_1 and buy the call at T_2. This provides a positive cash flow $\pi_0 > 0$.*
  2. *In T_1 if the call expiries out of the money, then we have made money: both π_0 and a potential positive payoff at T_2 in our favour.*

     If the call expires in the money, then we deliver one share and receive K. We are now short that share. In T_2, if the call expiries in the money, then we receive one share to cover our short in return for K. Our total cashflow is π_0. If the call expiries out of the money, then $S_T < K$. We can therefore buy back the share and cover our short for less than K. Our profits are $\pi_0 + (K - S_T)^+ > 0$.

- *If $C(0, K) \neq (1 - K)^+$ we have an immediate pathological arbitrage.*

- *If $C(T, 0) > 1$ then sell the call and cover with one share; if $C(T, 0) < 1$ then buy the call and enter into a short equity position.*

- $\partial_K C(T, 0) \geq -1$*: assume on the contrary there is a $K \downarrow 0$ such that $(1 - C(T, K))/(-K) < -1$ i.e. $C(T, K) = 1 - K - \epsilon$ for some $\epsilon > 0$. Buy the call and short sell the equity, resulting a cash position of $\pi_0 = -K + \epsilon$; the payoff in T is $(S_T - K)^+$ which allows us to cover our position and cash with a total reward of $\epsilon > 0$.*

- $\partial_K C(T, 0) < 0$*: assume on the contrary there is a $K \downarrow 0$ such that $(1 - C(T, K))/(-K) \geq 0$. We show existence of arbitrage in the case of equality i.e. existence of some $K > 0$ such that $C(T, K) = 1$. In that case we sell the call and buy the equity for 1. If the call expires in the money, we deliver the share and receive $K > 0$. If the call expiries out of the money, then sell the share for $S_T \geq 0$.*

- $\lim_{K\uparrow\infty} C(T, K) = 0$ *which means for all $c > 0$ there exists some K such that $C(T, K') < c$ for all $K' \geq K$ (because of convexity). This is the only arbitrage condition which is technical because a violation means, essentially, that the equity can grow without bounds which violates integrability. Assume therefore on the contrary that there is some $c > 0$ such that $C(T, K) \geq c$ for all K. Notice that, formally,*

  $$S_T = -\int_0^\infty \frac{(S_T - K + dK)^+ - (S_T - K)^+}{dK} dK = S_T + \lim_{K\uparrow\infty} (S_T - K)^+ \ .$$

  Therefore sell (a discretization of) the strip

  $$\int_0^\infty \frac{C(T, K) - C(T, K + dK)}{dK} dK = 1 + e$$

  which delivers S_T, and buy the equity. This will generate profits of $e > 0$.

2.3 SANOS - Smooth Arbitrage-free Non-parametric Option Surfaces

In [BHK+26] we introduced the following non-parametric family of call price functions which are all statically arbitrage-free: each call price function is given in terms of model expiries $0 = \hat{T}_0 < \hat{T}_1 < \cdots < \hat{T}_M$ and, for each expiry j, model strikes $0 < \hat{K}_j^1 < \cdots < 1 < \cdots < \hat{K}_j^{N_j}$ with widening range across expiries. It is recommended that the two outer strikes are far away from the inner strikes.[3] Model strikes and expiries do not need to match market strikes or expiries. We also set $K_0 = (1)$; see [BHK+26] for general comments on grid construction.

We say that $q = (q_1, \ldots, q_M)$ with $q_j \in \mathbb{R}^{N_j}$ is a *martingale density* if it satisfies the linear constraints

$$\begin{cases} \text{Each } q_j \text{ is a density:} & q_j \geq 0,\ q_j' 1 = 1 \\ \text{Each } q_j \text{ has unit expectation:} & q_j' \hat{K}_j = 1 \\ \text{The } q \text{ are in convex order:} & c_j \geq c_{j|j-1} \end{cases} \tag{3}$$

where $c_j^i = q_j'(\hat{K}_j - \hat{K}_j^i)^+$ is the j-call price of for $\hat{K}_j^i$, and $c_{j|j-1}^i := q_{j-1}'(\hat{K}_{j-1} - \hat{K}_j^i)^+$ is the $(j-1)$-call price of for $\hat{K}_j^i$.

Definition 2.4 (SANOS call price)**.** *Let q be a martingale density and $0 < W_1 < \cdots < W_M$ increasing total volatilities.*[4] *For a smoothing parameter $\mu \in [0, 1)$ the SANOS call price for arbitrary $T \geq 0, K \geq 0$ is given by*

$$C(T, K) := \begin{cases} \alpha_j(T) & \mathrm{BSCall}(\hat{K}_j^i,\ K,\ \mu W(T))\, q_j^i \\ +(1 - \alpha_j(T)) & \mathrm{BSCall}(\hat{K}_j^i,\ K,\ \mu W(T))\, q_j^i \cdot\ , \end{cases} \qquad T \in (\hat{T}_{j-1}, \hat{T}_j], \tag{4}$$

with $\alpha_j(T) = (T - \hat{T}_{j-1})/(\hat{T}_j - \hat{T}_{j-1})$ and total volatilities $W(T) := \alpha_j(T) W_j + (1 - \alpha_j(T)) W_{j-1}$.

For $T > \hat{T}_M$ we set $\alpha_{j+1}(M) = 0$ and $W(T) := W_M \sqrt{T/T_M}$.

Theorem 2.5. *The call price function C is statically arbitrage-free. For $\mu > 0$ it is C^∞ in K. The function is as smooth as α in T.*

Remark 2.6. *Contrary to [BHK+26] we interpolated total volatility linearly, not variance. Accordingly, our μ here corresponds to $\sqrt{\eta}$ in [BHK+26]. Our typical value $\eta = 0.25$ corresponds to $\mu = 0.5$.*

Using total volatilities here is a mechanical choice to avoid taking square roots of variables during training which can lead to exploding gradients. If forward volatilities are bounded well away from zero, linear interpolation in variance is suitable, too.

Remark 2.7. *The Black & Scholes formula in (4) can be replaced by the call price function for any martingale in T. For practical implementations, the function should be fast on a GPU.*

Calibration as a linear program: For clarity, let us briefly fix t and drop it from the notation of market data. We assume we obtain $W_1 < \cdots < W_M$ from the market e.g. as square roots of ATM variance per expiry. For each quoted call option with Given W the model call price $C^\ell := C(T^\ell, K^\ell)$ for a market expiry T^ℓ and a market strike K^ℓ can be seen in (4) to be linear in q. The market-fitting problem is therefore the linear programme

$$\min_q \sum_\ell \frac{1}{\gamma_\ell} \left[(C_\ell - \boldsymbol{A}_\ell)^+ + (\boldsymbol{B}_\ell - C_\ell)^+ + \epsilon |C_\ell - \boldsymbol{M}_\ell| \right] \tag{5}$$

subject to the linear constraints (3) encoding non-negativity, normalisation, integration to 1, and the inter-expiry convex-order constraints that enforce absence of calendar-spread arbitrage. We

[3] In our experiments, the strikes $K_j^2 < \cdots < 1 < \cdots < K_j^{N_j - 1}$ where chosen and then extended by $K_j^1 := K_j^2/2$ and $K_j^{N_j} = 1.5\, N_j^{N_j - 1}$.

[4] In our experiments we used square-roots of ATM variance.

note that the model strikes and time-to-expires do not need to correspond to market strikes or expiries.

On S&P 500 Index data with 1000 options across 48 expiries (2-day to 657 business-day TTE), the LP solves in sub-seconds on a desktop PC, achieving 91.4% of options within bid–ask on a representative date (2025-05-06), with a median error of 21% of the half-spread on the remaining options. Figure 2 illustrates the quality of the fit.

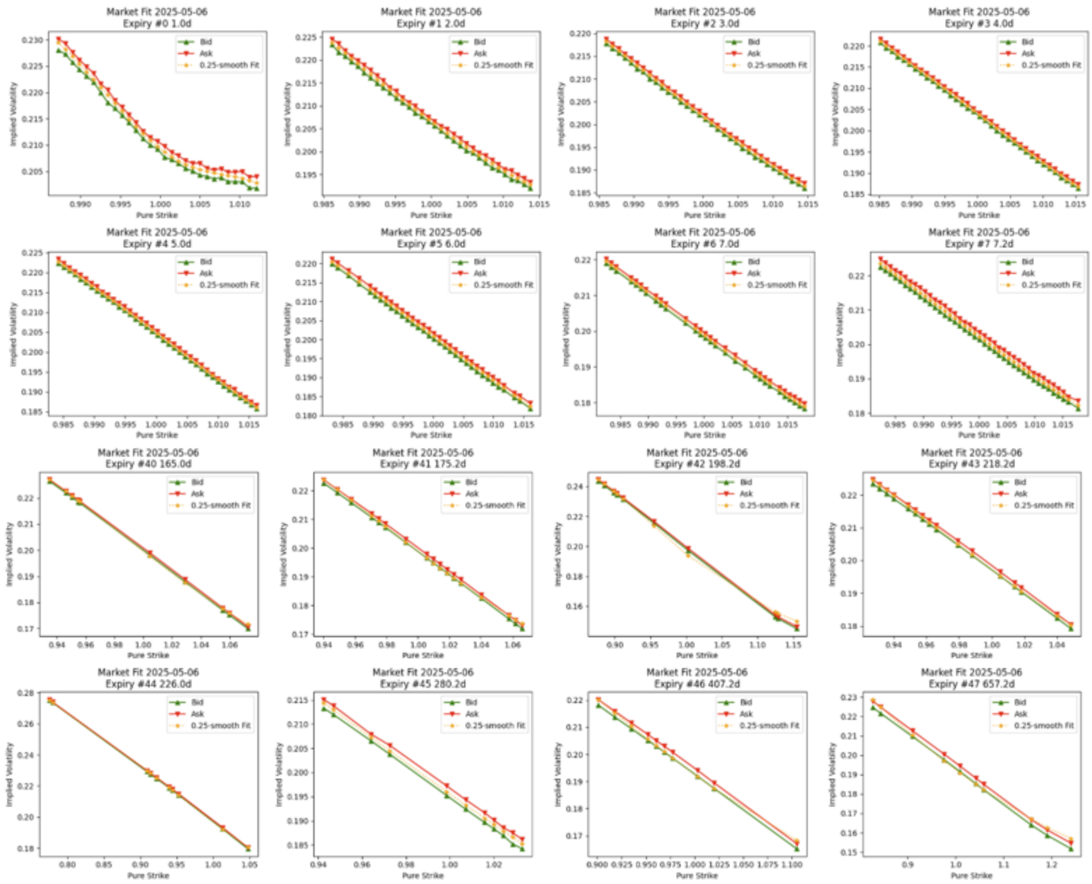

Figure 2: Example fit of SANOS.

2.4 The ML-SANOS decoder optimized for machine learning

We may view our call price operator as a map from the model parameters to a function space

$$(q, V;\ \hat{K}, \hat{T}; \mu) \overset{\text{SANOS}}{\longmapsto} C \tag{6}$$

for which we naturally want to learn q, V and, as we will see, $\hat{K}$ (as the right strike grid depends on the implied variance of the market). However, the density weights q_j in particular are subject to the linear constraints (3) for which no exact methods exist to integrate them into standard batch gradient descent methods. In this section we therefore reparameterize the map (6) in terms of learnable parameters.

To this end, recall the previously defined linear call prices $c_j^i := q_j'(\hat{K}_j - \hat{K}_j^i)^+$ and $c_{j|j-1}^i := q_{j-1}'(\hat{K}_{j-1} - \hat{K}_j^i)^+$. Given a martingale density q the squared discrete local volatilities (DLVs) $\Sigma_j^i \geq 0$ were then defined for $i = 2, \ldots, N_{j-1}$ in [BR15] via the discrete "backward" Dupire formula[5]

$$\frac{c_j^i - c_{j|j-1}^i}{T_j - T_{j-1}} \equiv \Sigma_j^i \frac{\hat{K}_j^{i2}}{(\hat{K}_{j+1}^i - \hat{K}_j^i)(\hat{K}_j^i - \hat{K}_{j-1}^i)} q_j^i \tag{7}$$

with $\Sigma_j^i := 0$ for $i \in \{1, N_j\}$ or whenever $q_j^i = 0$.

The key is that we can reverse this representation. We briefly recall the construction from [BHK+26]:

1. Start in $q_0 = (1)$ for $K_0 = (1)$.
2. For $j > 0$ let $c_{j|j-1}^i := \sum_{\ell=0}^{N_j+1} q_{j-1}^\ell (K_{j-1}^\ell - K_j^i)^+$ as before, and define for $i = 0, \ldots, N_j$

   $$dC_{j|j-1}^i := \frac{c_{j|j-1}^i - c_{j|j-1}^{i-1}}{K_j^i - K_j^{i-1}}$$

   with $K_j^0 = 0$, $c_j^0 = 1$. As a next step define for $i = 1, \ldots, N_j$ with $dC_{j|j-1}^{N_j+1} = 0$

   $$q_{j|j-1}^i := dC_{j|j-1}^{i+1} - dC_{j|j-1}^i \ .$$

   Then $q_{j|j-1}$ is a density over K_j, given by linear interpolation of call prices at the previous expiry. In [BHK+26] this operation was written as the linear expression

   $$q_{j|j-1} = L_j q_{j-1}$$

   in terms of the respective matrix $L_j \in \mathbb{R}^{N_j, N_{j-1}}$.
3. Let now

   $$\xi_j^{i\pm} := \pm \frac{K_j^{i2}}{(K_j^{i+1} - K_j^{i-1})\,(K_j^{i\pm1} - K_j^i)} \Sigma_j^i (T_j - T_{j-1}) \geq 0$$

   for $i = 2, \ldots, N_j - 1$:. The discrete local volatility transition operator is defined by the (inverseof) the tri-band matrix $\Xi_j \in \mathbb{R}^{N_j, N_j}$ with

   - upper diagonal $(\xi_j^{2-}, \ldots, \xi_j^{(N_j-1)+}, 0)$,
   - lower diagonal $(0, \xi_j^{2+}, \ldots, \xi_j^{(N_j-1)+})$, and
   - main diagonal $(1, 1 - \xi_j^{2+} - \xi_j^{2-}, \ldots, 1 - \xi_j^{(N_j-1)+} - \xi_j^{(N_j-1)-}, 1)$.

   Then, the inverse Ξ_j^{-1} exists, can be efficiently computed on GPU using a compiled Thomas' algorithm, and is a martingale transition operator in the sense of [BR15], and is used to define the linear operation

   $$q_j := \Xi_j^{-1} q_{j|j-1} \ .$$

   The operator Ξ_j is the in-homogeneous strike implicit transition operator of the PDE implied by (7), and is based on ideas of [AH11].

Definition 2.8 (ML-SANOS)**.** *Fix the normalized grid geometry for ML-SANOS with:*

- *Time-to-expiries (TTEs)* $0 = T_0 < \cdots < T_M$ *with* $dT_j = T_j - T_{j-1}$*;*[6]
- *Normalized strikes* $z^1 < \cdots < 0 < \cdots < z^N$ *which we assume contain zero;*[7]

[5] This actually *is* the Dupire formula applied to the implict foward-PDE of the density of a local volatility process, correctly computed on an irregular strike grid; c.f. [BR15].

[6] In our examples we used $2./255., 5./255., 10./255., 20./255., 40./255., 0.5, 1.$.

[7] In our examples we used a grid from $(-2, 1)$, computed to include zero.

- *Minimum and maximum forward volatilities* $\sigma_{\max} > \sigma_{\min} > 0$*; and*
- *Minimum and maximum squared discrete local volatilities* $\Sigma_{\max} > \Sigma_{\min} > 0$*.*

We translate the new real-valued model parameters $x = (x^\sigma, x^\Sigma) \in \mathbb{R}^N \times \mathbb{R}^{M,N}$ *into SANOS parameters as follows:*

- *Total volatilities: define the forward volatilities*

$$\sigma_j := \sigma_{\min}\sqrt{T_j} + \mathrm{sigmoid}(x^\sigma_j)(\sigma_{\max} - \sigma_{\max})$$

  and with them the total volatilities

$$W_j := \sum_{\ell=1}^{j} \sigma_\ell \left(\sqrt{T_\ell} - \sqrt{T_{\ell-1}}\right) \ . \tag{8}$$

- *Squared discrete local volatilities:*

$$\Sigma^i_j := \Sigma_{\min} + \mathrm{sigmoid}(x^\Sigma_{i,j})(\Sigma_{\max} - \Sigma_{\max}) \ . \tag{9}$$

- *Relative strikes:*

$$K^i_j := \exp\left(z^i_j W_j\right) \ . \tag{10}$$

  with boundaries $\hat{K}^0_j := \frac{1}{2}\hat{K}^1_j$ *and* $\hat{K}^{N+1}_j := \frac{3}{2}\hat{K}^N_j$*.*

Put together, this process describes a map

$$\left(x; z, \hat{T}; \mu\right) \overset{\substack{\text{ML}\\ \text{SANOS}}}{\longmapsto} \left(\Sigma, W;\ \hat{K}, \hat{T}; \mu\right) \overset{\text{DLV}}{\longmapsto} \left(q, W;\ \hat{K}, \hat{T}; \mu\right) \overset{\text{SANOS}}{\longmapsto} C \ . \tag{11}$$

In our new parametrization x lives in an unrestricted space and is therefore particularly suited for ML/AI based learning.[8]

2.5 Learning ML-SANOS from real data

Our reparametrization allows us learning the model parameters $x \in \mathbb{R}^{N+NM}$ from the market using standard batch-gradient descent methods.

However, to ensure a good fit, the state remains large: a typical SANOS setup uses around 10 expiries (7 in our examples) and around 200 strikes per expiry. That gives rise to around 2000 discrete local volatilities, a gross overparametrization of the option market when compared to the findings of few driving PCA factors in [CFD02]. We also note that DLVs are not particularly smooth as illustrated in figure 3.

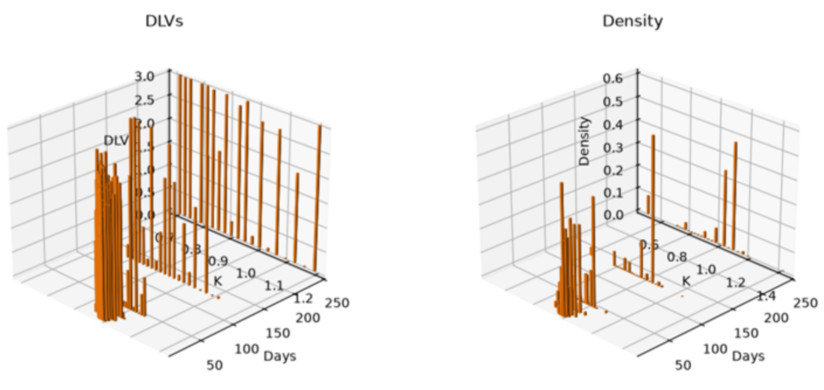

Figure 3: Example of the fitted density and the respective DLVs.

[8] In our experiments we have not tried to also learn μ or the position of the expiries.

We therefore take one final step in our approach, and introduce a simple feed forward embedding NQ_θ to map a much smaller state space $\mathbb{R}^{n_h}$ to $\mathbb{R}^{N+NM}$. In our experiments, the state h_t per sample t has dimension $n_h = 20$.

In other words we learn the weights θ of the map

$$NQ_\theta : h \in \mathbb{R}^{n_h} \longmapsto x \in \mathbb{R}^{M+NM} \ , \tag{12}$$

extending the overall process to

$$(\, h;\, z, \hat{T}; \mu \,) \overset{NQ}{\longmapsto} (\, x; z, \hat{T}; \mu \,) \overset{\substack{\text{ML}\\ \text{SANOS}}}{\longmapsto} (\, \Sigma, W;\, \hat{K}, \hat{T}; \mu \,) \overset{\text{DLV}}{\longmapsto} (\, q, W;\, \hat{K}, \hat{T}; \mu \,) \overset{\text{SANOS}}{\longmapsto} C \ . \tag{13}$$

In our experiments NQ was a three-layer feed forward network with 100 hidden nodes and SELU activation function between the layers.

The result of our training yields over around five years of S&P 500 Index option surfaces an average volatility error of 0.3% using h vs. the same options fitted with x with up to one year expiry. Figure 4 shows the fit accross time.

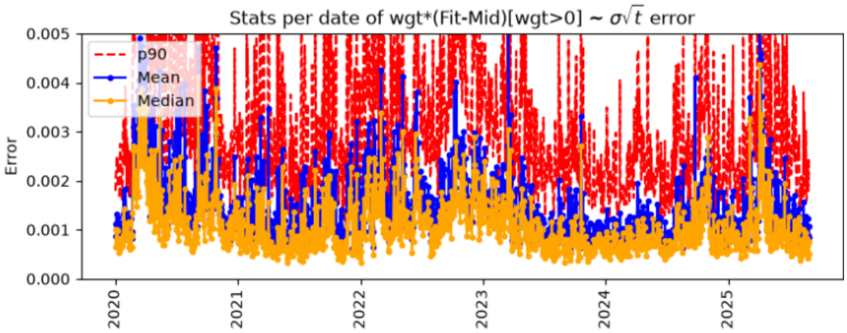

Figure 4: Statistics of fit, expressed in implied volatility (the actual fit-mid is in price, and weight is inverse of vega, divided by square root of expiry). For each date we compute the mean (blue), median (orange) and 90% percentile (red) fitting error, conditional on the fitting weight being positive. A value of 0.004 means that the error is 0.4% total volatility.

This training yields a time series of hidden states h_t for all historic samples $t \in \{t_1, \ldots, t_{n_t}\}$. Figure 5 shows some example fits, while figure 6 visualizes the 1+20 states used in our experiments.

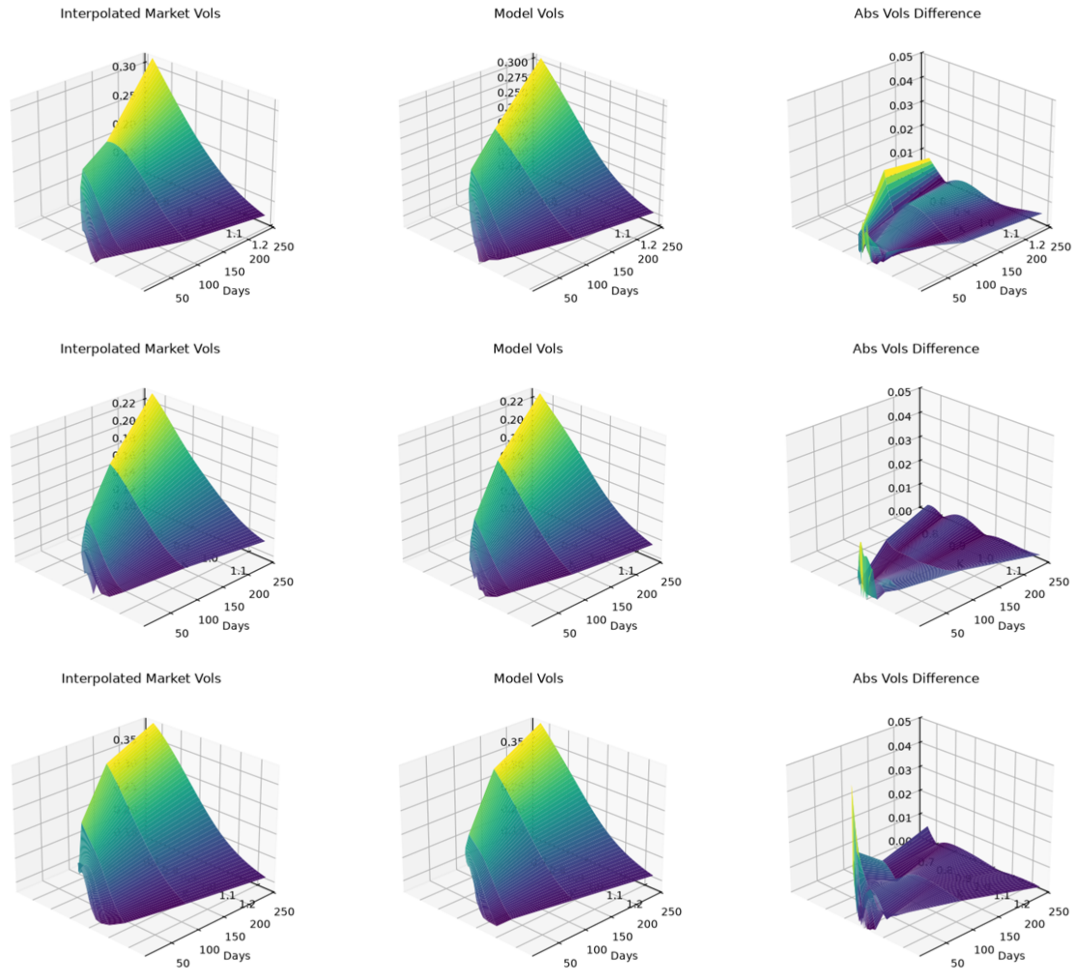

Figure 5: Examples of quality of fit using the encoding h for x. The fit is good in general with some deficiencies for 2 days to expiry.

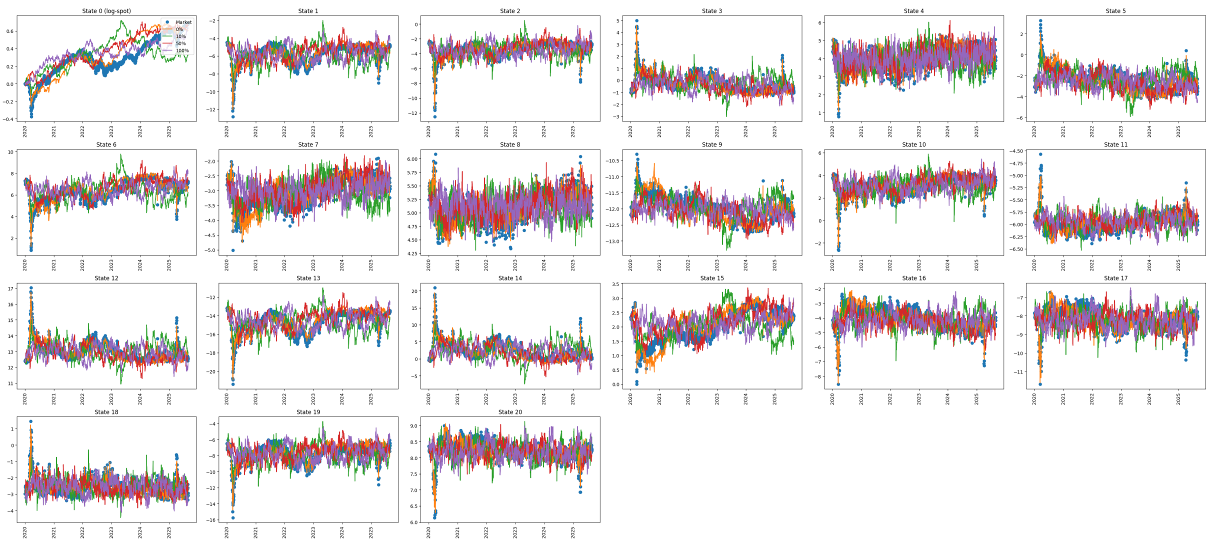

Figure 6: The states h of ML-SANOS when trained to 2020-01-02 to 2025-08-29. The first state represents log-spot.

2.6 Generative Option Surface State Space Models

We now assume are given historic samples of ML-SANOS states h_t for each historic date $t \in \{t_1, \ldots, t_{n_t}\}$ leading to a time series of differences $dh_t = h_{t+1} - h_t$ (with some abuse of notation). We also observe log-spot prices $h_t^0 := \log \boldsymbol{S}_t$ with returns $dh_t^0 := d \log \boldsymbol{S}_t$. With slight abuse of notation we now consider the extended state space $dh_t := (dh_t^0, \ldots, dh_t^{n_h}) \in \mathbb{R}^{1+n_h}$.

Theorem 2.9. *Any generative model for sequences $\tilde{h}_t$ of model parameters with real values gives rise to a* dynamic generative ML-SANOS *model which has for each time step t a spot price $\tilde{S}_t$ and a call price function $\tilde{C}_t$, which is free of static arbitrage and smooth (if $\mu > 0$).*

Remark 2.10. *If the real market has discount factors $\boldsymbol{DF}_t(T) = \exp(-R_t(T))$ and forwards $\boldsymbol{F}_t(T) = \boldsymbol{S}_t \exp(R_t(T) - D_t(T))$ for each sample t for each time-to-expiry T, then the spot returns are computed as*

$$dh_t^0 = \log \boldsymbol{S}_{t+1} - \log\left(\boldsymbol{DF}_t(dt)\boldsymbol{F}_t(dt)\right) = \log \boldsymbol{S}_{t+1} - \log \boldsymbol{S}_t + D_t(dt) \ .$$

We note that we do not need to adjust option prices, or their returns, as those were already normalized.

Baseline Model

The purpose of this article is to establish the overall framework of our approach. We therefore do not discuss advanced time series modelling for $\tilde{h}$ here. Instead, we will provide a robust baseline mirroring the ideas of [CFD02], namely to use PCA to drive implied volatility dynamics.

To this end we estimate the following PCA-AR(1) model for our time series $(h_t)_{t \in \mathcal{T}}$:

$$dh_t = (u_0 - \kappa h_t)dt + \alpha_t' du \sqrt{dt} \ . \tag{14}$$

We note that this formulation allows for both genuine mean-reversion for each factor as well as a plain drift, e.g. for the log-spot price h_0^0. We learn

- A mean reversion level $u_0 \in \mathbb{R}^{1+n_h}$,[9]
- a matrix of mean reversion speeds $\kappa \in \mathbb{R}^{1+n_h, 1+n_h}$ whose eigenvalue's real values should be non-negative,[10]
- historic PCA loadings $\alpha_t \in \mathbb{R}^{n_\alpha}$, and
- PCA factors $du \in \mathbb{R}^{n_\alpha, 1+n_h}$.

These are estimated using standard statistical AR(1) methods and a PCA on the residuals.[11] For the period 2020-01-01 to 2025-08-29 (the latest day available on WRDB Option Metrics' Ivy DB at the time or writing) just $n_\alpha = 5$ eigenvectors explain 99.4% of the daily variance, c.f. figure 7. This finding is in line with other results [CFD02]. Figure 8 shows the factors and their loadings over the training period.

[9] We found $u_0^0 = -0.407$ for the log-spot state, h^0.

[10] In our training the min real eigenvalue was 0.001283, the maximum Euler transition spectral radius was 1.151073, and the max matrix-exp transition spectral radius was 0.998726.

We note that the inclusion of log-spot h^0 in the mean-version term is motivated by the observation that its drift contains a variance term. However, it is straight forward to impose zero mean reversion by zeroing out the respective row of κ.

[11] A least squared fit between de-meaned dh_t/dt and h_t yields u_0 and κ. We then perform an SVD on the residual, scaled by $\sqrt{dt}$.

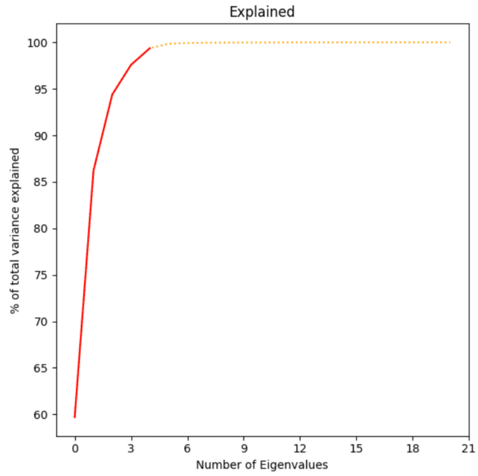

Figure 7: Variance explained by number of factors.

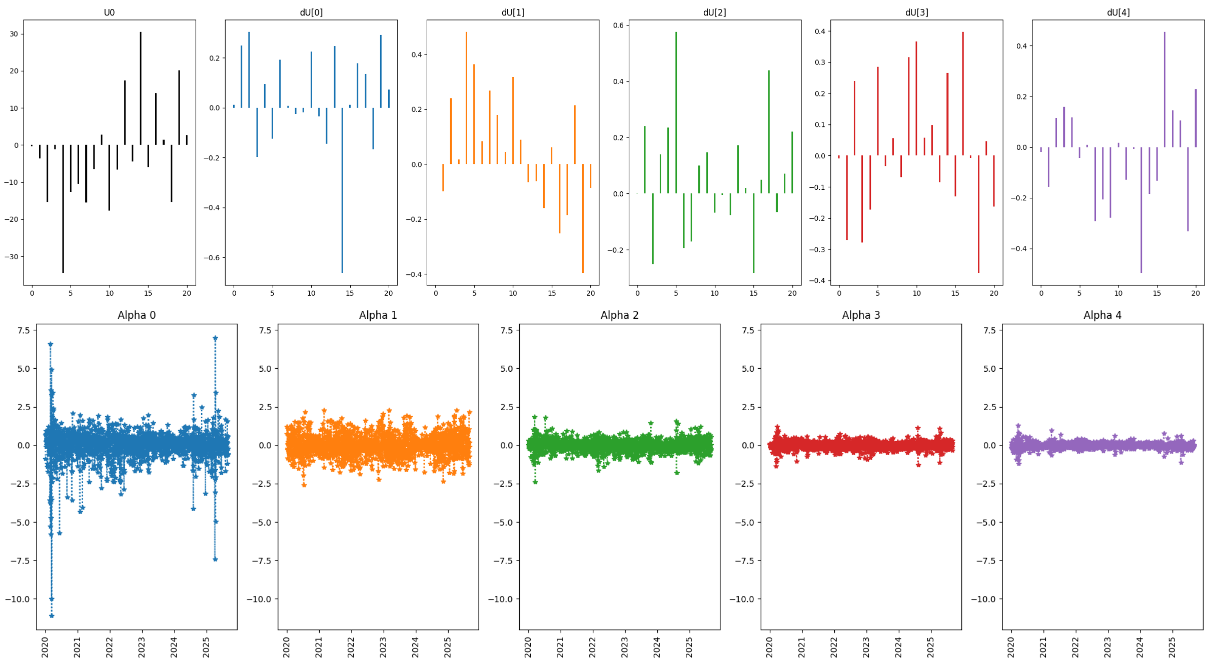

Figure 8: Factors u_0 and du (top) and their loadings α (bottom).

To define our baseline generative model:

- Let $A^2 := \alpha'\alpha \in \mathbb{R}^{n_\alpha, n_\alpha}$ be the covariance of the factor loadings, which we Cholesky-decompose into $A^2 = AA'$ for $A \in \mathbb{R}^{n_\alpha, n_\alpha}$.
- Define the standardized factor loadings

$$\hat{\alpha}_t := A^{-1}\alpha_t \in \mathbb{R}^{n_\alpha} \tag{15}$$

  such that $\hat{\alpha}'\hat{\alpha} = I_{n_\alpha}$.
- Finally, we define the standardized volatility matrix

$$\Sigma := du' A \in \mathbb{R}^{n_h, n_\alpha} \tag{16}$$

such that

$$\alpha_t' du = \Sigma \hat{\alpha}_t$$

Definition 2.11 (Baseline DYSANOS)**.** *For each sample,*

1. *Draw randomly* $\tilde{h}_0$ *from* $\{h_t\}_{t\in\{1,\ldots,n_t\}}$.
2. *Then generate a path of states via*

$$d\tilde{h}_{t+1} = \left(u_0 - \kappa\tilde{h}_t\right) dt + \Sigma Z_t \sqrt{dt} \tag{17}$$

where each of the random variables $Z_t \in \mathbb{R}^{n_\alpha}$

$$Z_t = w\,\hat{\alpha}_{t(\omega)} + \bar{w} Y_t(\omega)$$

is generated for a blend parameter $w \in [0,1]$, $\bar{w} := \sqrt{1-w^2}$ *by drawing a historical sample* $t(\omega) \in \{t_1, \ldots, t_{n_t}\}$ *and a normal* $Y_t(\omega) \in \mathbb{R}^{n_\alpha}$*. (We note that* Z_t *has unit covariance.)*

That means for $w = 1$ *we obtain a process generated by resampling historic factor loadings, while* $w = 0$ *yields a process driven by a normal with a consistent covariance process.*

Remark 2.12. *The Euler-step (17) is not numerically stable for large time steps. For large time steps of* n *days simulate instead*

$$d\tilde{h}_{t+n} = e^{-\kappa\, n\, dt}\tilde{h}_t + u_0 \int_0^{n\,dt} e^{-\kappa s} ds + +w \sum_{q=0}^{n-1} e^{-K\, q dt/n} \Sigma \hat{\alpha}_{t+q} \sqrt{dt} + \bar{w} \sqrt{\int_0^{n\,dt} e^{-\kappa\, s} \Sigma' \Sigma e^{-\kappa\, s'} ds}\; \tilde{Z}_t \tag{18}$$

with $\tilde{Z}_t \in \mathbb{R}^{n_h}$ *standard normal.*

We note that all the matrix exponentials in this expression can be pre-computed given n *and* dt *which makes this formulation suitable for in-training generation of market data for downsteam models (instead of path generation of paths ahead of downstream training as was used in [BGTW19].*

The visual performance of even such a simple process is impressive. Figure 9 shows paths of the generated process started in some historic h_t compared with the original data.

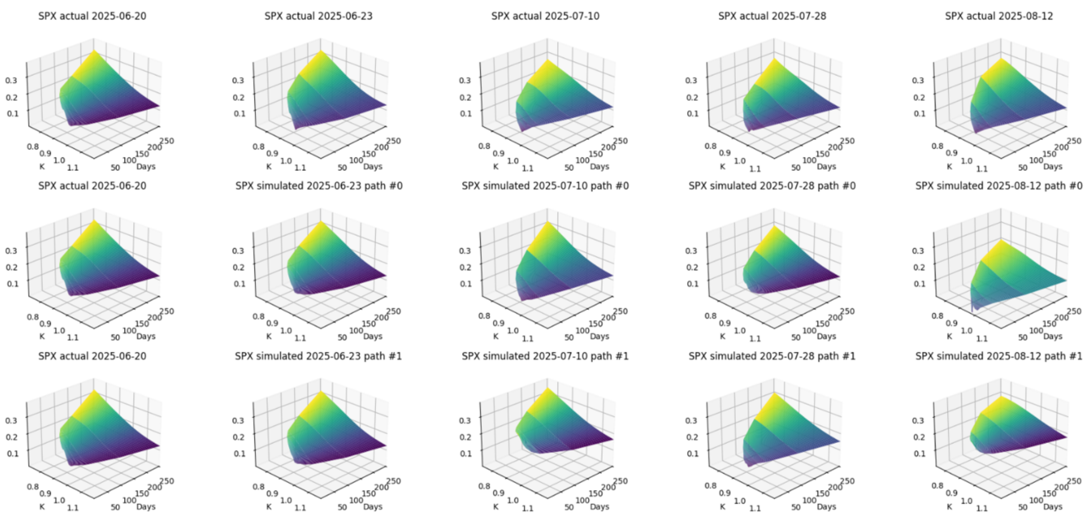

Figure 9: Actual historic path (top) vs two simulated paths, all starting in the historic data of 2025-060-25. Data are simulated daily but only every eights sample is shown.

Clearly, (18) is but the simplest of dynamics one might want to use to generate option surfaces.

2.7 Summary

We have provided a baseline model to generate long paths of daily spot and option prices. The option prices are free of static arbitrage. Moreover, we can price *any* option on each path given the ML-SANONS parameters for the respective sample and time step. That means contrary to previous results, here we can train agents to chose which options to trade in each time step.

3 Risk-Neutral Dynamics

In the previous section we described with our "Baseline DYSANOS" (18) a simple PCA-AR(1) model for the dynamics of the reduced underlying state of ML-SANOS under the observed statistical measure $\mathbb{P}$. As discussed in [BMPW22] using the statistical measure, either directly via historic paths, or indirectly like here using a trained model, risks picking up spurious market behaviour. Indeed, for most of the period 2020-01-02 to 2025-08-29 which we used to train our model the S&P 500 went up as is shown in figure 10.

Figure 10: Spot path for the S&P 500 Index during the modelling period (blue) and since then, up to the latest data available from Yahoo Finance.

However, the mean positive daily annualized return of the S&P of 14.4% are barely statistically significant as table 1 shows. Indeed, at the time of writing (2026) the market has shown that this positive drift is by no means a statistical certainty.

Mean	14.4%	Median	22%
Std	21%	q10	-317%
		q90	350%

Table 1: Statistics of daily annualized returns of the S&P 500 2020-01-02 to 2025-08-29, divided by dt or $\sqrt{dt}$ in the case of "Std".

We therefore have two motivations to remove spurious drifts in the market

1. We are not aiming to extract alpha with such a simple model; therefore we make a baseline assumption of zero drift for tradable instruments, consistently applied to spot and options.
2. From a classic quantitative finance perspective, a risk-neutral model gives us the ability to compute a frictionless price for any exotic by computing risk-neutral expectations.

We therefore now follow [BMPW22] and switch to a risk-neutral measure. We will focus here on the following scenario assume we are first generating $\Omega \in \mathbb{N}$ paths for $\mathcal{T} \in \mathbb{N}$ time steps. Note that we do not need to know the strikes or time-to-expiries of the options we will want to trade during agent training at path generation time, which sets our method apart from to grid-based multi-step arbitrage-free simulators such as [WBWB19, WWP+21, CRW23].

Once simulated at each t we are able to trade the spot $S_t = \exp h_t^0$ and a grid of options with time-to-expires T_t^ℓ and spot-relative strikes K_t^ℓ for $\ell = 1, \ldots, L$. In our experiments below we chose the same $L = 140$ options we used to fit the model to, but that is not necessary.

The prices of the options at time t are given as

$$C_t^\ell := C_t(T_t^\ell, K_t^\ell) \equiv C(\tilde{h}_t, z, \hat{T}, \mu;\ T_t^\ell, K_t^\ell)\ ,. \tag{19}$$

The value at time $t + dt$ of the actually tradable instrument with cash strike $S_t K_t^\ell$ and expiry $t + T_t^\ell$ is given as

$$C_{t+1}^{\ell|t} := \frac{S_{t+dt}}{S_t}\ C_{t+dt}\left(\ T_t^\ell - dt,\ \frac{S_t}{S_{t+dt}} K_t^\ell\ \right) \tag{20}$$

We note that this expression requires interpolation of option prices in $t + dt$, exactly the use case DYSANOS was designed to handle. It allows us to compute per path the return

$$dC_t^\ell = C_{t+1}^{\ell|t} - C_t^\ell\ . \tag{21}$$

For convenience define $dC_t^0 := S_{t+dt} - S_t$. We assume[12] that $|\mathbb{E}^\mathbb{P}[dC_t^0]| < \infty$ for all t which then also implies that $|\mathbb{E}^\mathbb{P}[dC_t^\ell]| < \infty$ due to our static arbitrage constraints.

3.1 Trading Policies

We call $a = (a_0, \ldots, a_{\mathcal{T}-1})$ a *trading policy* if each $a_t \in \mathbb{R}^L$ is a t-measurable random variable (i.e. we cannot look into the future when making our decision) and if

$$\mathbb{E}^\mathbb{Q}\left[G_t^\ell\right] < \infty \quad \text{for the gains} \quad G_t^\ell := a_t^\ell\, dC_t^\ell\ . \tag{22}$$

We call each a_t^ℓ a *trading decision*. The total gain of the policy is given by

$$G := \sum_{t=0}^{\mathcal{T}-1} \sum_{\ell=0}^{L} G_t^\ell = \sum_{t=0}^{\mathcal{T}-1} \sum_{\ell=0}^{L} a_t^\ell dC_t^\ell\ .$$

3.1.1 Arbitrage and Risk-Neutrality

Definition 3.1. *We say* $\mathbb{P}$ admits arbitrage *if there exists a trading policy* $a = (a_t)_{t=0,\ldots,\mathcal{T}-1}$ *with* $a_t \in \mathbb{R}^{1+L}$ *which has positive expected return but cannot lose money, i.e.*

$$\mathbb{E}^\mathbb{P}[G] > 0 \quad \textit{and} \quad \mathbb{P}[G < 0] = 0\ . \tag{23}$$

Definition 3.2. *A measure* $\mathbb{Q} \approx \mathbb{P}$ *is risk-neutral if no trading policy* $a = (a_t)_t$ *can make money,*

$$\mathbb{E}^\mathbb{Q}[G] = 0 \tag{24}$$

(symmetry means that any strategy with negative returns can be made into a strategy with positive returns, hence the condition here is that all returns equal zero; this is not the case in the presence of market frictions).

Definition 3.3 (Statistical Arbitrage)**.** *We say a trading policy is a* statistical arbitrage *[opportunity] if* $\mathbb{E}^\mathbb{P}[G] > 0$.

[12] For our current model and for finite M this is clearly the case as S_t is the exponential of a finite sum of normals.

Absence of such opportunities is risk-neutrality:

Proposition 3.4 (Risk-Neutrality is a Local Property)**.** *The measure $\mathbb{Q} \approx \mathbb{P}$ is risk-neutral if and only if for all ℓ, t and any trading decision a_t^ℓ*

$$\mathbb{E}^{\mathbb{Q}}\left[G_t^\ell\right] = 0 \ . \tag{25}$$

This condition is equivalent to the mathematical expression that the conditional expectation of the returns is zero,

$$\mathbb{E}_t^{\mathbb{Q}}\left[dC_t^\ell\right] = 0 \ . \tag{26}$$

This property is a key insight: risk-neutrality for finite instruments and time steps is a local property. It is well-know that absence of arbitrage is equivalent to the existence of a risk-neutral pricing measure. The reference for discrete time continuous state spaces is [DMW90]:

Theorem 3.5 (First Fundamental Theorem of Asset Pricing)**.** *A market is arbitrage-free if and only if there exists at least one risk-neutral probability measure consistent with current asset prices.*

As shown out in [BMPW22] contrary to continuous time in discrete time a change of measure can also affect the volatility of an asset – in other words, it can correct for mispricing of implied volatility vs realized volatility which is not possible in continuous time. It is also noteworthy that such measure is in general not unique; in particular not in discrete time.

Remark 3.6 (Continuous Time)**.** *We like to point in continuous time the problem of finding a risk-neutral measure is very constrained. There, the volatility of an asset cannot be affected, and all is left is to find a consistent drift, often referred to as the* Heath-Jarrow-Morton *condition. We sketch briefly the impact:*

The hidden state process h satisfies with (17) the diffusion form

$$d\tilde{h}_t = \underbrace{(u_0 - \kappa\tilde{h}_t)}_{=:A(\tilde{h}_t)}\,dt + \Sigma dB_t$$

where we assume for clarity $w = 0$ in (17) and write the normal innovation in terms of a Brownian motion $B_t \in \mathbb{R}^{n_\alpha}$.

This means that (21) has the principle form

$$\begin{aligned} dC_t^\ell &= C_{t+dt}^{\ell|t}(\tilde{h}_t) - C_t^\ell(\tilde{h}_t) + \sum_{i=0}^{n_h} \partial_{\tilde{h}^i} C_{t+1}^{\ell|t}(\tilde{h}_t)\, d\tilde{h}_t^i + \frac{1}{2}\sum_{i,k=0}^{n_h} \partial^2_{\tilde{h}^i,\tilde{h}^k} C_{t+1}^{\ell|t}(\tilde{h}_t)\, d\tilde{h}_t^i d\tilde{h}_t^k \\ &= C_{t+dt}^{\ell|t}(\tilde{h}_t) - C_t^\ell(\tilde{h}_t) + \sum_{i=0}^{n_h} \partial_{\tilde{h}^i} C_{t+1}^{\ell|t}(\tilde{h}_t) \left(A^i(\tilde{h}_t)\,dt + \Sigma^i dB_t\right) + \frac{1}{2}\sum_{i,k=0}^{n_h} \partial^2_{\tilde{h}^i,\tilde{h}^k} C_{t+1}^{\ell|t}(\tilde{h}_t)\,\Sigma^i\Sigma^{k\prime} dt \end{aligned} \tag{27}$$

Imposing the martingale property means that the drift needs to be zero, i.e. there exists a μ such that

$$0 = \frac{C_{t+dt}^{\ell|t}(\tilde{h}_t) - C_t^\ell(\tilde{h}_t)}{dt} + \left(A(\tilde{h}_t) + \mu(\tilde{h}_t)\right)\partial_{\tilde{h}} C_{t+1}^{\ell|t}(\tilde{h}_t) + \frac{1}{2}\Sigma^2 \partial^2_{\tilde{h},\tilde{h}} C_{t+1}^{\ell|t}(\tilde{h}_t)$$

simultaneously for all $\ell = 1, \ldots, L$.

Since $L \gg 1 + n_h$ and A, μ and Σ do not dependent on ℓ this equation does not in general have a solution.

This is the reason so far no analytic market model for option markets has been published.

We circumvent this challenge by focusing on discrete time steps and using machine learning methods to find a suitable measure change.

3.2 Risk-Neutrality for Sample Paths of Generative Market Simulators

The above conditions are theoretical conditions on a market with infinitely many paths. We now define *discrete risk-neutrality* as a property of $\Omega \in \mathbb{N}$ simulated paths over $\mathcal{T} \in \mathbb{N}$ time steps for $L \times \mathcal{T} \ll \Omega$ tradable instruments.

We recall our *Deep Hedging* framework [BGTW19] where we train an agent to risk-manage a derivative under risk-adjusted monetary utilities. We discussed there that if there is no derivative to manage, then a Deep Hedging agent learns to trade in the market to make money. This is not possible if the market is risk-neutral: mechanically, Deep Hedging is trying to find the trading policy in (3.2) to make money; if this fails, the market is risk-neutral.

We note, however, that in a practical simulation not finding a policy which can make money is subject to sampling noise: even if a market is simulated under a risk-neutral measure we do not expect the returns of its assets to have numerically zero drift. (This is only an issue because we simulate derivatives: if we only had equity assets we could just divide by the statistical expected value per time step; however with options this is not a consistent operation.)

We therefore impose a practical criterion, which requires that under $\mathbb{Q}$ the time-zero expected returns of all options dC_t^ℓ are within $\delta = 2$ of the MC sampling error noise:

$$\epsilon_t^{\mathbb{Q},\ell} := \sqrt{\frac{1}{\Omega}\mathbb{E}^{\mathbb{P}}\left[\left(\frac{d\mathbb{Q}}{d\mathbb{P}}dC_t^\ell - \mathbb{E}^{\mathbb{P}}\left[\frac{d\mathbb{Q}}{d\mathbb{P}}dC_t^\ell\right]\right)^2\right]} \tag{28}$$

(we notice that because we sampled under $\mathbb{P}$ the corret sampling error "under $\mathbb{Q}$" is computed for the variable $\frac{d\mathbb{Q}}{d\mathbb{P}}dC_t^\ell$).

The proposed solution is therefore to consider ϵ as *cost* which we can take into account when looking for an optimal trading strategy. This is therefore what we will present next: use Deep Hedging with cost set to the sampling error to find an optimal strategy. If this is successful, change measure such that under the new measure the expected returns of all options are within the sampling error. We then iterate this procedure one more time.

We start with introducing *monetary utilities*:

Definition 3.7. *Let $u : \mathbb{R} \to \mathbb{R}$ be a C^1 utility function, i.e. it is concave, decreasing and normalized in $u(0) = 1$ with limits $\lim_{x\downarrow-\infty} u'(x) = \infty$ and $\lim_{x\downarrow-\infty} u'(x) = 0$. We define the associated* monetary utility *for an integrable random variable X as*

$$U(X) := \sup_{y\in\mathbb{R}} \mathbb{E}^{\mathbb{P}}\left[u(X+y) - y\right] \ . \tag{29}$$

Theorem 3.8 ([BTT07])**.** *$U(X)$ is concave (risk averse), monotone increasing (more is better) and cash-invariant in the sense that $U(X + c) = U(X) + c$ for all $c \in \mathbb{R}$; as pointed out first in [Bue17] the latter property is equivalent to being able to write off finite valued liabilities at their worst value.*[13]

A canonical example for the utility u is the entropy given with $u(x) := (1 - \exp(-\lambda x))/\lambda$ for risk-aversion $\lambda > 0$ in which case $U_\lambda(X) = -\frac{1}{\lambda}\log \mathbb{E}^{\mathbb{P}}[\exp(-\lambda X)]$. For normal X we obtain the classic mean-reversion objective $U_\lambda(X) \equiv \mathbb{E}^{\mathbb{P}}[X] - \frac{\lambda}{2}\mathrm{Var}^{\mathbb{P}}[X]$.

The *Deep Hedging problem* for an empty initial portfolio is:

$$U^* := \sup_a U(G) = \sup_a U\left(\sum_{t=0}^{\mathcal{T}-1}\sum_{\ell=0}^{L} a_t^\ell dC_t^\ell\right) \tag{30}$$

Theorem 3.9 ([BGTW19])**.** *The market is free of arbitrage (in a theoretical sense) if and only if $U_\lambda^* = 0$ where U_λ is the entropy with risk aversion λ. (There exist no statistical arbitrage opportunities.)*

[13] Assume we have a position X. Then $U(X+W) \geq U(X + \inf W) = U(X) + \inf W$ as long as $\inf W > -\infty$.

Deep Hedging can not only be used to detect statistical arbitrage opportunities; in can also provide us with a change of measure under which such opportunities disappear, aka a risk-neutral measure:

Theorem 3.10 (Near-Martingale Measures [BMPW22]). *Let $c_t^\ell \geq 0$ be proportional transaction cost for each C_t^ℓ. Assume that $y^* \in \mathbb{R}$ and a^* are optimal solutions for the Deep Hedging problem with transaction cost:*

$$\sup_{a,y} : \mathbb{E}^{\mathbb{P}}\left[u\left(y + \sum_{t=0}^{\mathcal{T}-1}\sum_{\ell=0}^{L} a_t^\ell dC_t^\ell - |a_t^\ell| c_t^\ell \right) - y \right] \tag{31}$$

Then the measure $\mathbb{Q}$ defined by the density

$$\frac{d\mathbb{Q}}{d\mathbb{P}} := u'\left(y^* + \sum_{t=0}^{\mathcal{T}-1}\sum_{\ell=0}^{L} {a^*}_t^\ell dC_t^\ell - |{a^*}_t^\ell| c_t^\ell \right) \tag{32}$$

is a near-martingale measure *i.e. for all options*

$$\left| \mathbb{E}_t^{\mathbb{Q}}\left[dC_t^\ell \right] \right| \leq c_t^\ell \ . \tag{33}$$

If $c \equiv 0$ then $\mathbb{Q}$ is a risk-neutral measure.

Interpretation: Recall from (27) that dC_t^ℓ is locally approximately normal so can be written as $dC_t = A_t\, dt + \Sigma_t\, dB_t \in \mathbb{R}^{1+L}$ for some drift $A_t \equiv A_t(h_t) \in \mathbb{R}^{1+L}$, some square-root of covariance $\Sigma_t \equiv \Sigma_t(h_t) \in \mathbb{R}^{1+L,n_\alpha}$, and some Brownian motion $dB_t \in \mathbb{R}^{n_\alpha}$. In the specific case of the entropy with risk aversion $\lambda = 1$ and we obtain

$$\frac{d\mathbb{Q}}{d\mathbb{P}} := \exp\left(-y^* - \sum_{t=0}^{\mathcal{T}-1} \left(a_t^{*\prime}\left(A_t\, dt + \Sigma_t\, dB_t \right) - |a_t^{*\prime}| c_t \right) \right) \tag{34}$$

where y^* is a normalization factor. Because of the independence of the increments of dB_t and the need to integrate to one we notice that the density of our near-martingale measure in t must be

$$\left.\frac{d\mathbb{Q}}{d\mathbb{P}}\right|_t := \exp\left(-a_t^{*\prime}\Sigma_t\, dB_t - \frac{1}{2} a_t^{*\prime}\Sigma_t\Sigma_t' a_t^* dt \right) \equiv 1 - a_t^{*\prime}\Sigma_t\, dB_t \ . \tag{35}$$

We therefore obtain

$$\frac{1}{d}\mathbb{E}_t^{\mathbb{Q}}[dC_t] \equiv \left(1 - a_t^{*\prime}\Sigma_t\, dB_t \right)\left(A_t\, dt + \Sigma_t\, dB_t \right) = A_t - \Sigma_t\Sigma_t' a_t^* \ = (*) \ . \tag{36}$$

In case $c = 0$ we obtain the classic mean-variance problem where $(*)$ is (in theory, not numerically) zero. In other words,

$$a_t^{*\prime} = \left(\Sigma_t\Sigma_t' \right)^{-1} A_t \quad (c = 0) \tag{37}$$

which the classic risk-premium. when using the entropy, our "Remove the Drift" approach [BMPW22] approach is therefore a numerically elegant way to compute (37) directly without the need to compute the numerically challenging $(\Sigma_t\Sigma_t')^{-1}$. The ability to impose transaction cost is financially sound, while it acts numerically as a L^1 stabilization penalty. We note, however, that this normal view of our problem runs easily into the same issues we flagged in remark 3.6: since $\Sigma_t\Sigma_t'$ is heavily underdetermined there does not always exist a "mean-variance" minimizer for our portfolio.

While the entropy is in many ways the most natural utility function (to start with, it allows for many analytical results) it has the strong drawback that it produces negative infinite utility for a short position in a Black & Scholes asset. For this reason we introduce the following alternative:

Definition 3.11 (Power-Entropy). *Define*

$$u_{\lambda,p}(x) := \begin{cases} \frac{1}{\lambda}\left(1 - e^{-\lambda x} \right) & x \geq 0 \\ x - \frac{\lambda}{p}(-x)^p & x < 0 \end{cases} \tag{38}$$

with in our case $p = 2$.

3.2.1 Recipe

We now apply our results to our practical problem of finding a suitable *numerical risk-neutral measure* as follows:

1. **Absence of Deterministic Arbitrage**: we first confirm that our data does not exhibit trivial arbitrage with cost equal to one Monte Carlo sampling error:
$$\begin{cases} \sup a'\mathbb{E}^{\mathbb{P}}[dC] \\ \\ a'dC - |a|'\epsilon^{\mathbb{P}} \geq 0 \text{ (pathwise)} \\ |a| \leq 1 \end{cases} \tag{39}$$

2. **Normalizing** dS: As a pre-processing step we compute a simple initial measure $\mathbb{P}'$ wtih density $D \in \mathbb{R}^{\Omega}$ by enforcing zero drift on the equity itself: to this end, we solve the convex program
$$\begin{cases} \inf_D \mathbb{E}[D\log(D)] \\ \\ D \geq 0 \\ D'1 = 1 \\ D'dS_t = 0 \text{ for all } t \ . \end{cases}$$
This step speeds up convergence.

3. **Iterative Step**: Define the "cost" of trading any option as the time-zero estimation noise, i.e. $c_t^\ell := \epsilon_t^{\mathbb{P}',\ell}$ as defined in (28). Find an optimal Deep Hedging strategy and therefore a new measure $\mathbb{O}$. This will mean that under $\mathbb{O}$
$$\left| \mathbb{E}_t^{\mathbb{O}}\left[dC_t^\ell\right] \right| \leq \epsilon_t^{\mathbb{P}',\ell} \ .$$
up to $\mathbb{O}$-sampling noise.

4. **Final Measure**: We then iterate this procedure one more time with cost $\epsilon_t^{\mathbb{O},\ell}$ to obtain our final measure $\mathbb{Q}$. This is a heuristic based on the observation that iterative applications of theorem 3.10 provide ever tighter fits.

   Recall that we aim at limiting the expected value of dC_t^ℓ under $\mathbb{Q}$ to $\delta = 2$ of $\epsilon_t^{\mathbb{Q},\ell}$. By using one unit of MC error we are ensuring the the algorithm converges to that outcome.

3.3 Results

We parametrize a_t^ℓ as feed forward neural network NA_θ with depth 2 (e.g. three layers) with 40 hidden states and "leaky RELU" activation function and a learnable constant $\alpha_t^\ell \in \mathbb{R}$ as
$$a_t^\ell := \alpha_t^\ell + NA_\theta\left(h_t; \frac{t}{t_{\mathcal{T}-1}}, \log \hat{K}_t^\ell, \hat{T}_t^\ell\right) \tag{40}$$
with the convention $\log \hat{K}_t^0 = 0$ and $\hat{T}_t^0 = 0$ for spot.

We present the results for the case $\mathcal{T} = 10$, $L = 140$ and $\Omega = 40,000$ which fits well on a standard consumer GPU. We train the model on the whole data set without validation set because we do not require an out-of-sample: we require that the the measure we compute for those paths is numerically risk-neutral on the given paths.

Absence of Deterministic Arbitrage: We confirm on 40,000 path that there is no trivial time-zero arbitrage beyond the MC simulation error. However, we note there *is* arbitrage present in the sense of (39) without cost. We therefore confirm that this is indeed spurious by checking that the optimal a from (39) when applied to 400,000 samples does not create arbitrage. This is indeed the case.

Cost	Sample	Minimum	Mean	Negative paths	Negative fraction	Has Arb
No cost	40k training	-1.0000×10^{-10}	0.0754443	0	0.0000%	Yes
No cost	400k frozen	-0.0375553	0.0755699	874	0.2185%	No
$\epsilon^{\mathbb{P}}$	40k training	-2.3987×10^{-10}	9.4204×10^{-10}	0	0.0000%	No
$\epsilon^{\mathbb{P}}$	400k frozen	-3.8857×10^{-9}	9.4514×10^{-10}	0	0.0000%	No

Table 2: Training and frozen out-of-sample empirical-arbitrage results. Values below 10^{-8} are treated as numerical zero.

Normalizing dS: this is an initial step to reduce the drift of dS and the inherent drift of each dC_t^ℓ with respect to its delta.

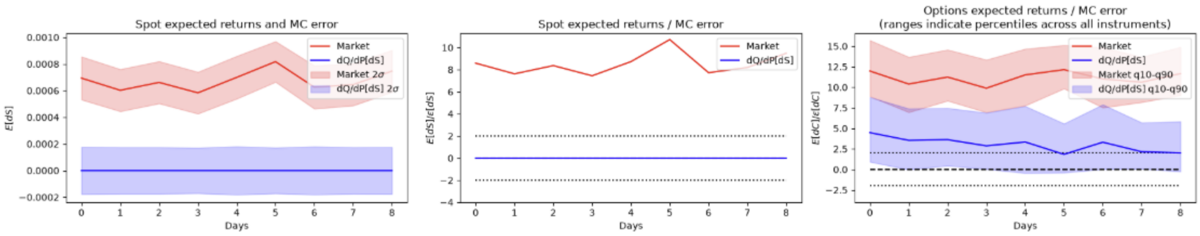

Figure 11: Adjusting an initial measure per path to center dS. The left hand side shows the initial ("market") and trained result as mean and two standard deviations. The second graph shows the return divided by the MC error for the spot, and the third shows the same for the options, with bands representing quantiles across all contracts. We see some reduction in drift due to the implicit delta-correction.

Training $\mathbb{O}$ and $\mathbb{Q}$: we report our results for both the entropy and the power-entropy

4 Conclusion

We presented with our generative DYSANOS model the first strictly arbitrage-free option *market model* with consistent robust spot and option dynamics. Our approach does not rely on approximate solutions; each smooth option surface is strictly arbitrage free and allows pricing of any option on GPU. We have shown how to leverage our representation to build a simple, but already fairly powerful baseline model which can generate spot and option paths for many days in the future up to years.

Finally, we have shown how to construct numerically a risk-neutral measure under which the expectation of a sample payoff matches the price computed with Deep Hedging in the absence of cost.

5 Thanks

We want to thank the team at Wharton for their WRDS https://wrds-www.wharton.upenn.edu portal and Option Metrics https://optionmetrics.com/united-states for providing their IvyDB data via this service. Without this data our work would not have been possible.

A Appendix: Data Pipeline

In the interest of full transparency we here discuss main elements of the data pipeline used in our experiments.

The market-data pipeline constructs a standardized daily time series of SPX option surfaces. Raw option quotes are cleaned, normalized, filtered, made arbitrage-free, interpolated onto a fixed expiry–moneyness grid, and finally collected into tensors suitable for model training.

The pipeline processes SPX option data between January 2nd, 2020 and August 29th, 2025. The latter is the latest day available at the time of writing. Only valid full-day NYSE trading days are considered.

A.1 Pipeline Overview

The complete transformation can be summarized as

$$\begin{aligned}\text{raw SPX option quotes} &\longrightarrow \text{daily cleaning and enrichment}\\ &\longrightarrow \text{arbitrage-free quote surface using linear SANOS}\\ &\longrightarrow \text{liquidity and moneyness filters}\\ &\longrightarrow \text{smooth SANOS surface}\\ &\longrightarrow \text{interpolate to fixed expiry–moneyness grid}\\ &\longrightarrow \text{daily ML-SANOS fit using Torch.}\end{aligned}$$

The key insight when working with Option Metrics IvyDB data is that the forward curves from the `Forward_Price` table do not line up with the option data themselves. We therefore compute our own forward from weighted put-call parities. We also convert the tightest options into call prices. Such data still has arbitrage, hence we use a linear SANOS pre-processing step to obtain fitted prices within bid/ask which adhere to basic linear no-arbitrage conditions (the absence of which would indicate actionable arbitrage). We drop any strikes where such a fit cannot be achieved, assuming that fit issues are due to data alignment challenges rather than genuine trading opportunities.

A.2 Raw Daily Market Data

For every eligible date d, the underlying data contain quantities such as

- the SPX spot price S_d and previous spot price S_{d-1};
- option strikes and expiration dates; call and put bid and ask quotes; volume and open interest; AM/PM settlement indicators;
- forward and discount-curve information (with forwards being re-calculated below).

The default raw-data cleaning removes

- same-day expirations;
- options with bids below 0.011 (the minimum tick size is 0.05$ for series trading below 3$);
- observations with missing or invalid quotes; observations with an invalid settlement flags; irregular expirations; dates or ticker-date combinations known to contain bad data (e.g. 2025-01-08 which was a holiday).

A.3 Enrichment and Expiry Filtering

The cleaned quotes are enriched with

- business days and year fractions to expiry where AM options are counted as 20% of the day;
- implied forward prices computed off put-call parity per expirty for all strikes for which both call and put were traded and have valid bid/ask prices and sizes, weighted by put-call implied forward bid-ask spread.[14]
- using the new forward, define "market" mid call prices between the actual call priced and synthetic call prices from the puts;
- compute implied volatilities; forward- and discount-adjusted prices, strikes, and Greeks.

An expiration is retained only if it satisfies the default liquidity and quality requirements. In particular,

- at least 10 strikes must have either a traded call or put;
- at least 4 strikes must have both a traded call and put;
- the discounted forward normalized spread must be at least 10^{-6};
- implied volatility must lie in
$$0.01 \le \sigma_{\text{impl}} \le 5.$$

A.4 Arbitrage-Free Preprocessing

Before the final option filter is applied, an arbitrage-free representation of the daily market is constructed using [BHK+26]. The default configuration uses surface mode and performs the following operations:

1. Extend the strike grid to improve boundary treatment.
2. Fit a linear (DLV) no-arbitrage option-price curve separately for each expiry.
3. Drop expiries for which a valid fit cannot be obtained.[15]
4. Remove locally redundant strikes associated with negligible fitted density.
5. Fit a surface across strike and expiry to control both butterfly and calendar arbitrage.

At least five strikes per expiry are required. The default minimum fitted density is 10^{-4}, and the minimum admissible ATM volatility is 0.05.

[14] We compute $\text{fwd}^{\text{ask}}(K) := +\text{Call}^{\text{ask}} - \text{Put}^{\text{bid}} - \text{df}K$ and $\text{fwd}^{\text{bid}}(K) := +\text{Call}^{\text{bid}} - \text{Put}^{\text{ask}} - \text{df}K$ using the discount factor sourced from IvyDB. Then we define $\text{fwd}^{\text{mid}}(K) := 0.5(\text{fwd}^{\text{ask}}(K) + \text{fwd}^{\text{bid}})(K)$ and $\gamma^{\text{mid}}(K) := 0.5(\text{fwd}^{\text{ask}}(K) - \text{fwd}^{\text{bid}})(K)$ and
$$\text{fwd} := \frac{\sum_K \gamma(K)^{-1}\text{fwd}^{\text{mid}}(K)}{\sum_K \gamma(K)^{-1}}$$

[15] Existence of such a fit is equivalent to the market not having actionable arbitrage using bid/asks as quoted.

A.5 Option Filtering

The default option filter retains observations satisfying

$$\mathrm{BDTE} \leq 3 \times 252 = 756,$$

where BDTE denotes business days to expiry.

The normalized-moneyness filter is

$$-5 \leq z_{\mathrm{ATM}} \leq 2,$$

where z_{ATM} is normalized moneyness computed using the ATM volatility.

Additional requirements are

$$\begin{aligned} \mathrm{VegaSqrtVar} &\geq 10^{-3}, \\ \log(\mathrm{volume}) &\geq 1, \\ \log(\mathrm{open\ interest}) &\geq 1. \end{aligned}$$

Here, "VegaSqrtVar" refers to Vega divided by sqrt of expiry. It is therefore the sensitivty to total volatility of the option.

Each retained expiry must contain at least 10 options. At most 1000 options are kept per day. If more than 1000 observations survive, the filter first reserves observations around the money for every expiry and then fills the remaining capacity using increasing absolute normalized moneyness.

It should be noted that this filter removes many options post 1Y for the early years of the training period due to the minimum volume traded filter. We suspect that this information in IvyDB is not reliable in the past. Future tests might want to remove the volume and/or open interest filters. (It is an option in our data pipeline parametrization.)

A.6 Spot Normalization

Price-like quantities are normalized by the current spot S_d to obtain variables of scale around 1. For example,

$$\widetilde{K}_{d,i} = \frac{K_{d,i}}{S_d}, \qquad \widetilde{C}_{d,i} = \frac{C_{d,i}}{S_d}, \qquad \widetilde{F}_{d,i} = \frac{F_{d,i}}{S_d}.$$

The original spot is retained as $\mathrm{spot\_ref}_d = S_d$. (The normalized log-spot level is passed on as h^0 later.)

A.7 Smooth Interpolation to the Reference Grid

The previous steps, in the end, simply cleaned market data. All remaining options are market tradable instruments with market bid/ask spreads.

The next step is now to interpolate from the cleaned market to a reference grid to which we train ML-SANOS. To this end, for each day, the filtered pure (forward normalized) strikes, bids, and asks are sorted by strike and separated by expiry. A smooth arbitrage-free SANOS surface is then fitted to these irregularly spaced market quotes using linear programming [BHK+26].

The grid-fitting configuration uses $\mu = 0.5$, $\Sigma_{\min} = 0.01$, $\Sigma_{\max} = 2$, $\sigma_{\min} = 0.01$ and $\sigma_{\max} = 2$. The step seeks an arbitrage-free representation consistent with the observed bid and ask intervals.

From that smooth interpolation we define the default target expiries, measured in years, as

$$\left\{ \frac{2}{255}, \frac{5}{255}, \frac{10}{255}, \frac{20}{255}, \frac{40}{255}, \frac{1}{2}, 1 \right\}.$$

Thus, the grid contains $M = 7$ target expiries, corresponding approximately to 2, 5, 10, 20, 40, 127.5, and 255 business days. If the longest target expiry exceeds the longest observed market expiry on a given date, the implementation emits a warning and extrapolates the fitted surface.

For every expiry, the surface is sampled at $\tilde{N} = 100$ strike locations; the default normalized-moneyness range is $\tilde{z} \in [-2, 1]$. The grid explicitly contains the ATM location at zero. Since the interval is asymmetric, more grid points are allocated below ATM than above ATM.

Let $\tilde{W}_j$ be the interpolated total volatility for expiry $\hat{T}_j$. The normalized-moneyness coordinates are converted into spot-normalized strikes using the same formulas as (8)

$$\tilde{K}_j^i = \exp\left(z^i \tilde{W}_j\right).$$

The continuous surface is then evaluated at every pair $\left(\hat{T}_j, \tilde{K}_j^i\right)$,. Two artificial boundary strikes are appended:

$$\tilde{K}_j^0 = \frac{1}{2}\tilde{K}_j^1, \qquad \tilde{K}_j^{N+1} = \frac{3}{2}\tilde{K}_j^N$$

Consequently, the extended strike dimension contains $N + 2$ points.

A.8 Fitting ML-SANOS

We now fit ML-SANOS to this data. Recall that ML-SANOS contains the network map $NQ_\theta : h \in \mathbb{R}^{n_h} \longmapsto x \in \mathbb{R}^{M+NM}$ defined in (12) and that its effective strikes $\hat{K}$ are dynamic as a function of learned total volatility. We train at the same grid expiries as introduced above, but this time with Torch.

The ML-SANONS model then fits all $\tilde{N}$ smoothed strikes per expiry using $N = 20$ model strikes. It uses the same normalized strike range $(-2, 1)$. Weights are inverse BS vega divided by sqrt to expiry.

References

[AH11] J. Andreasen and B. Huge. Volatility interpolation. *Risk*, pages 86–89, 2011.

[BGTW19] H. Buehler, L. Gonon, J. Teichmann, and B. Wood. Deep hedging. *Quantitative Finance*, 19(8):1271–1291, 2019.

[BHK+26] Hans Buehler, Blanka Horvath, Anastais Kratsios, Yannick Limmer, and Raeid Saqur. Sanos - smooth arbitrage-free non-parametric option surfaces. https://arxiv.org/abs/2601.11209, Februar 2026.

[BMPW22] Hans Buehler, Phillip Murray, Mikko S. Pakkanen, and Ben Wood. Deep hedging: learning to remove the drift. *Risk*, February 2022.

[BPW22] Hans Buehler, Murray Phillip, and Ben Wood. Deep bellman hedging. https://ssrn.com/abstract=4151026, June 2022.

[BR15] Hans Buehler and Evgeny Ryskin. Discrete local volatility for large time steps (extended version). https://papers.ssrn.com/abstract=2642630, 2015.

[BTT07] Aharon Ben-Tal and Marc Teboulle. An old-new concept of convex risk measures: The optimized certainty equivalent. *Mathematical Finance*, 17(3):449–476, 2007.

[Bue06] Hans Buehler. Expensive martingales. *Quantitative Finance*, 6(3):207–218, 2006.

[Bue17] Hans Buehler. Statistical hedging. https://ssrn.com/abstract=2913250, Feb 2017.

[Bue26] Hans Buehler. Learning to trade i - greek, parameter and statistical hedging. https://ssrn.com/abstract=7086398, March 2026.

[CFD02] Rama Cont, Jose da Fonseca, and Valdo Durrleman. Stochastic models of implied volatility surfaces. *Economic Notes*, 31(2):361–377, 2002.

[CRW23] Samuel N. Cohen, Christoph Reisinger, and Sheng Wang. Arbitrage-free neural-sde market models. *Applied Mathematical Finance*, 30(1):1–46, 2023.

[CV23] Rama Cont and Milena Vuletić. Simulation of arbitrage-free implied volatility surfaces. *Applied Mathematical Finance*, 30(2):94–121, 2023.

[DMW90] Robert C. Dalang, Andrew Morton, and Walter Willinger. Equivalent martingale measures and no-arbitrage in stochastic securities market models. *Stochastics and Stochastics Reports*, 29(2):185–201, 1990.

[Fri00] Marco Frittelli. The minimal entropy martingale measure and the valuation problem in incomplete markets. *Mathematical Finance*, 10(1):39–52, 2000.

[GHL12] Julien Guyon and Pierre Henry-Labordère. Being particular about calibration. *Risk*, January 2012.

[WBWB19] Magnus Wiese, Lianjun Bai, Ben Wood, and Hans Buehler. Deep hedging: Learning to simulate equity option markets. https://arxiv.org/abs/1911.01700, 2019.

[WWP+21] Magnus Wiese, Ben Wood, Alexandre Pachoud, Ralf Korn, Hans Buehler, Phillip Murray, and Lianjun Bai. Multi-asset spot and option market simulation. https://arxiv.org/abs/2112.06823, 2021.