Learning To Trade
From Market Simulation to Hedging

Visiting Prof. Dr. Hans Buehler
University of Oxford

Cambridge University Algorithmic Trading Society Quant Conference
Cambridge, March 6th, 2026

Agenda

Quant Finance 2.0

- Instead of focusing on solvable low-parametric “interpolation“ models solve the hedging problem in real markets.

- As we do not usually have sufficient market data, we build market simulators which capture broad features of the market

- Then we learn to trade and risk manage in the simulated environments with reinforcement learning techniques.

- We present the broad outline of the approach, applications, and challenges in implementation … and active areas of research.

Framework

Framework

Environment

- We operate in discrete time $t = 0, \dots, \infty$ under the statistical measure P.
- We observe the market **state** s_t: time, prices, news, historical trading events etc and assume it generates our filtration.
- Any publicly observable market quantity X_t can be written for some measurable function X as $X_t = X(s_t)$.
- We also assume here that our trading causes no impact.
  - Our framework is easily extended to include impact [1].

[1] Lecture Notes Learning to Trade III, Buehler et al 2022 https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4151043

Framework

Trading Instruments

- We assume that we are trading in a market with liquid **tradable instruments**.
  - These include primary assets such as spot, FX as well as derivatives such as options on equity, indices etc. We assume interest rates are deterministic(1) and cashflows are carried to expiry or to the point of sale.
- At each time step we can trade a different set of n_t instruments $H^{(t)}$.
  - We denote by $H_t^{(t)} \in R^{n_t}$ their book value at time t, and by $\gamma_t \in R_{\geq 0}^{n_t}$ their bid/ask spreads. The book value is the accounting value.(2)
  - We do not require that an instrument that was tradable in some t is also tradable at a later time.

(1) If this is not the case, then optimal investment decisions of cash into the available interest rate instruments also need to be considered.
(2) For liquid asset that can be a mid-price; for OTC derivatives this would be the classic derivatives price.

Framework

Trading Instruments

- Our **trading cost** to trade $a \in R^{n_t}$ units of $H^{(t)}$ in excess of their book values is given by a non-negative and convex(1) function with $c_t(0) = 0$.
- That implies c_t which is increasing in all directions, i.e. $\partial_\epsilon c_t(\epsilon a) \geq 0$.
- Recall that $c_t(a) \equiv c(a; s_t)$.

(1) Convexity excludes fixed fee cost which are, in fact, common.

Framework

Risk Limits and other Trading Restrictions

- Convex transaction cost allow defining convex limits to trading capacity by setting $c_t(\neg A) = \infty$ outside a convex set A.

- That means we can use trading cost to impose a wide variety of convex **trading restrictions** of the following type:
  - Maximum liquidity: $A = \{a: \text{askcapacity}^i \leq a^i \leq \text{bidcapacity}^i\}$
  - Total Vega Traded: $A = \{a: |\sum_i a_i \text{Vega}_i| \leq \text{Limit}\}$

- Trading restrictions which refer to the current portfolio are not always convex, as the available capacity in the market might not be sufficient to hedge all our risk:
  - Total Vega Held: $A = \{a: |\text{PortfolioVega} + \sum_i a_i \text{Vega}_i| \leq \text{Limit}\}$

Framework

Monetary Utilities

- We wish to maximize a risk-adjusted return U objective over random variables X (the value of our trading strategies).
- Axiomatic properties:
  - **More is better:** U should be increasing, i.e. if $X \geq Y$ then $U(X) \geq U(Y)$.
  - **Risk aversion:** $U(X)$ should be concave, i.e. $U(E[X]) \geq U(X)$.

Framework

Motivating Cash Invariance

- Assume that $\widetilde{U}$ is monotone and concave. Assume our position is X.
- We assume we are allowed to "write off" (drop) any Y for its worst-case cost as long as it is essentially bounded from below (we cannot drop an infinite loss). Following [1]:

$$U(X) \coloneqq \sup_{-Y<\infty} \left\{\widetilde{U}(X-Y) - \operatorname{ess\,sup}(-Y)\right\} = (*)$$

Since $\widetilde{U}$ is monotone, we have

$$(*) = \sup_{y\in R, Y\geq 0} \left\{\widetilde{U}(X+y-Y) - y\right\} = \sup_{y\in R}\left\{\widetilde{U}(X+y) - y\right\}$$

- That means U is **cash-invariant**, i.e. $U(X+y) = U(X) + y$ for $y \in R$.
- Any optimization over X is independent of our current cash position.

[1] Statistical Hedging, Buehler, 2017 https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2913250

Framework

OCE Monetary Utilities

- We use *Optimized Certainty Equivalents, OCEs* from [1]: let u be a concave, increasing *utility function, then*

$$U[X] := \sup_{c \in R} E[u(X + c) - c]$$

  is a *monetary utility*: increasing, concave and cash-invariant.
  - $-U$ is a convex risk measure..
  - From an ML perspective, such measures lend themselves into batch-based optimization.

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

Framework

OCE Monetary Utilities

- The classic risk management tool in finance, CVaR, was shown in [1] to be given by

$$u(x) \coloneqq (1+\lambda)\min\{x, 0\}$$

in which case $U(X) = E[X|X \leq p(\alpha)]$ where

  - $p(x) \coloneqq P[X \leq \cdot]^{-1}(x)$ is the percentile function and
  - $\alpha = 1/(1+\lambda)$ is the confidence level, i.e. if $\lambda = 0$ then $\alpha = 1$ and $p(x) = \operatorname{ess\,sup} X$.

- Robust measure: maximize expectation by ignoring the top $1-\alpha$ fraction of returns.

[1] "Optimization of conditional value-at-risk", Rockafellar, Uryasev, Journal of Risk, 2, 21-41, 2000

Framework

OCE Monetary Utilities

- The prototype monetary utility for risk aversion $\lambda > 0$ is the **entropy** given by

$$u(x) := \frac{1 - \exp(-\lambda x)}{\lambda}$$

in which case $U(X) = -\frac{1}{\lambda} \log E\left[e^{-\lambda X}\right]$.

- In case X is normal with mean μ and variance v we obtain the mean-variance objective $U(X) = m - \frac{\lambda}{2} v$.
- The entropy penalizes losses heavily. The monetary utility of a short position in a Black & Scholes process is negative infinity.
- Truncated entropy $u(x) := \frac{1-\exp(-\lambda x)}{\lambda} 1_{x>0} + \left(x - \frac{1}{2}\lambda x^2\right) 1_{x \leq 0}$

Vanilla Deep Hedging

- In **Vanilla Deep Hedging** [1] we are given a fixed horizon T to hedge a fixed current portfolio Z with terminal value $Z_T \in R$.
  - Practically, we will compute $Z_T(\omega)$ for $\omega = 1, \dots, N$ samples.
  - This can be done ahead of our simulation using standard Monte-Carlo implementation.
- We are able to trade different options $H^{(t)} \in R^{n_t}$ in every step with a book price $H_t^{(t)}$.
  - We use their total return to expiry, denoted by

$$DH_{t:T} \coloneqq H_T^{(t)} - H_t^{(t)}.$$

  - Can also be pre-computed ahead of training.

[1] Deep Hedging, Buehler et all 2018, https://arxiv.org/pdf/1802.03042.pdf

Vanilla Deep Hedging

- Let a with $a_t \equiv a(s_t) \in R^{n_t}$ be a **trading policy**.
- The **gains** of following a trading policy a until a fixed horizon T in the presence of a portfolio Z are given as:

$$G(Z;a) := Z_T + \sum_{t=0}^{T-1} a_t' D H_{t:T} - c_t(a_t)$$

Existing portfolio cashflows at expiry

Hedge cashflows to expiry

Trading cost and limits

Vanilla Deep Hedging

- Let U be a monetary utility. Then the **Vanilla Deep Hedging** problem of hedging a portfolio Z until T is given as

$$U^*(Z) \coloneqq \sup_a : U\left(Z_T + \sum_{t=0}^{T-1} a_t' DH_{t:T} - c_t(a_t) \right)$$

- The sup is taken over all admissible strategies.
- Essentially *hedging under convex risk measures*, c.f. [1]

[1] Convex Risk Measures, Foellmer, Schied, Wiley 2010

Vanilla Deep Hedging

Marginal Pricing – *the price of a derivative is the cost of its hedge*

- We wish to sell a derivative D to our client while we already have a position Z.
- The marginal price of selling D in the presence of Z is given as

$$p(D) \coloneqq U^*(Z) - U^*(Z - D)$$

- It represents the **minimal price** we should charge to not be worse off vs our existing position Z.
- It satisfies the invariance condition

$$U^*\big(Z - D + p(D)\big) = U^*(Z)$$

Implementation

Vanilla Deep Hedging

- To solve

$$U^*(Z) \coloneqq \sup_a : U\left(Z_T + \sum_{t=0}^{T-1} a_t' D H_{t:T} - c_t(a_t)\right)$$

  we are going to use neural networks:
  - Compress current market state into a recurrent hidden state h_t.
  - Decode the relevant action a_t^i as a function of "the past" h_t and current relevant state.
- Networks or learned matrices are written in **bold**.

Vanilla Deep Hedging

Set Invariant State Embedding

- To trade different options $H^{(t)} \in R^{n_t}$ in every step:
- For simplicity assume these are equity options, each identified by $w_t^i = (k_t^i, \tau_t^i, c_t^i)\ i = 1, \ldots, n_t$ given in terms of
  - relative strikes k_t^i,
  - time to expiries τ_t^i, and
  - call/put indicators c_t^i.
- At each time t we observe a general market state m_t and option features $f_t^1, \ \ldots, f_t^{n_t}$ such as mid-price, spread, trade volume etc.
- Naturally $w_t^i \subset f_t^i$.

Vanilla Deep Hedging

Set Invariant Market State Embedding

- We embed our option features using a set-invariant architecture, e.g. Deep Sets [1]:

$$e_t := \boldsymbol{E}\left(m_t; \frac{1}{n_t}\sum_{k=1}^{n_t} \boldsymbol{F}\left(m_t; f_t^k\right)\right) \in R^{n_e}$$

  with networks $\boldsymbol{E}, \boldsymbol{F}$.
- Transformers are also set-invariant.
- Note that such an embedding can be computed in parallel in t.

[1] Deep Sets, Zaheer et al, 2017, https://arxiv.org/abs/1703.06114

Vanilla Deep Hedging

History State

- Once we have obtained a sequence e_t we can employ a time series model to generate our recursive "hidden" history state h, e.g. with selective SSM-like [1] models:

$$h_t := \boldsymbol{C}(e_t)h_{t-dt} + \boldsymbol{B}(e_t, z_t)$$

Neural CDE [2]:

$$h_t = (1 - \boldsymbol{C}(e_t)de_t)h_{t-dt}$$

- Implemented via compiled forward scan.
- Usually forced into unit space by applying tanh.

[1] Mamba: Linear-Time Sequence Modeling with Selective State Spaces, Gu et al 2024, https://arxiv.org/abs/2312.00752
[2] Neural Controlled Differential Equations for Irregular Time Series, Kidger et al 2020, https://arxiv.org/abs/2005.08926

Vanilla Deep Hedging

Portfolio State Encoding

- Many OTC payoffs have complicated state conditions on the past
  - Barriers depending on past KO/KI states
  - Cliquets depend on past reset dates

- *Should* be able to learn these.
  - However, we only look at terminal values of Z_T.
  - Identifying e.g. a monthly barrier breach requires noisy state detection, with possibly very long look back windows.
  - Simply add a "portfolio state" z_t to which captures portfolio-specific events.

- Decode into $a_t^i = \boldsymbol{A}(w_t^i; f_t^i, m_t, h_t, z_t)$.

Vanilla Deep Hedging

- We then solve the problem

$$U^*(Z) \coloneqq \sup_a : U\left(Z_T + \sum_{t=0}^{T-1} a_t' DH_{t:T} - c_t(a_t)\right)$$

  using Monte Carlo **Periodic Policy Search**.

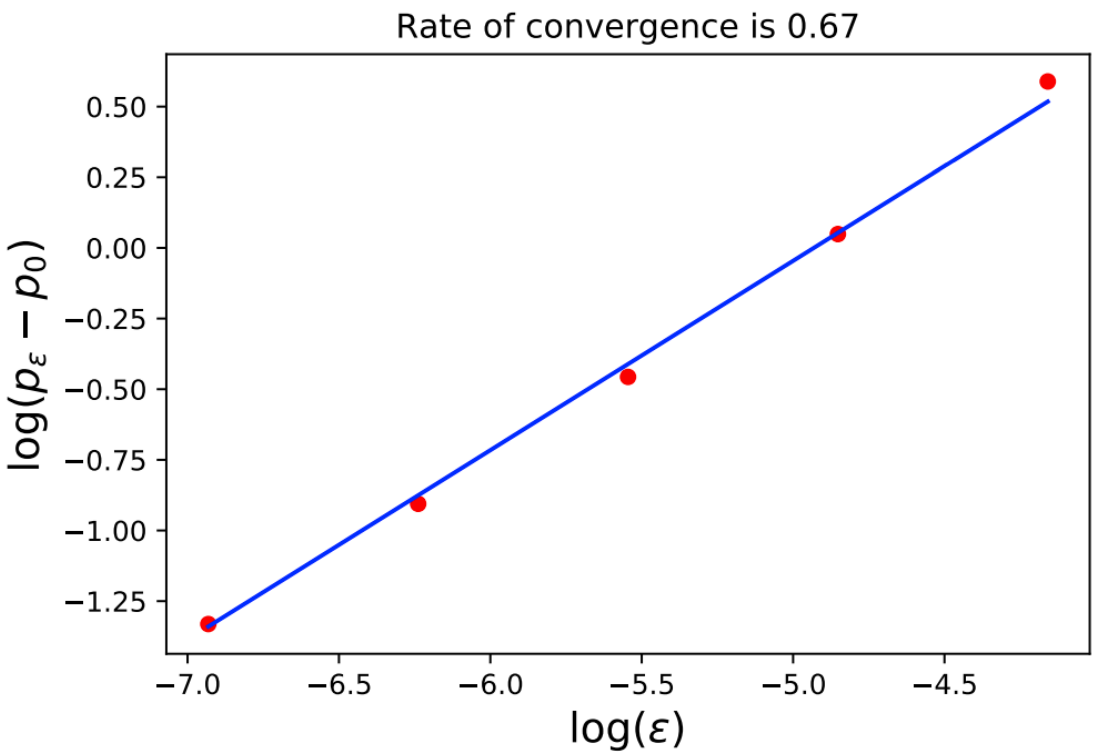
- For N paths find the network weights θ and $y \in R$ for

$$U^*(Z) \equiv \max_{a,y \in R} : \frac{1}{N} \sum_{\omega=1}^{N} u\left(y + Z_T + \sum_{t=0}^{T-1} a_t' DH_{t:T} - c_t(a_t)\right) - y$$

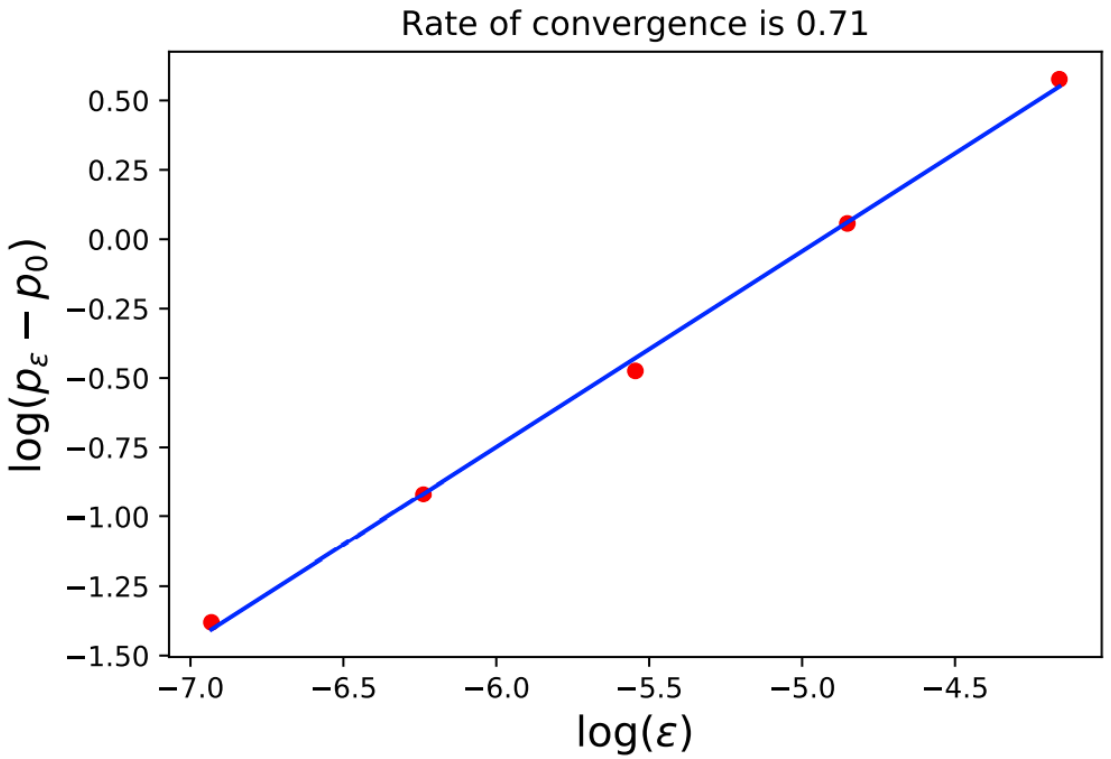
  using batch gradient descent e.g. Adam.

Vanilla Deep Hedging

- Test vs cases where we know or guess the theoretical answer [1]



Black-Scholes model price asymptotics.

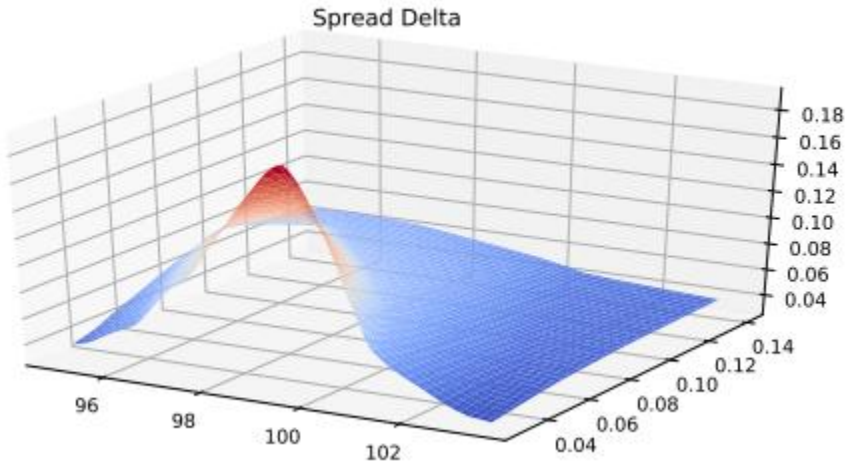


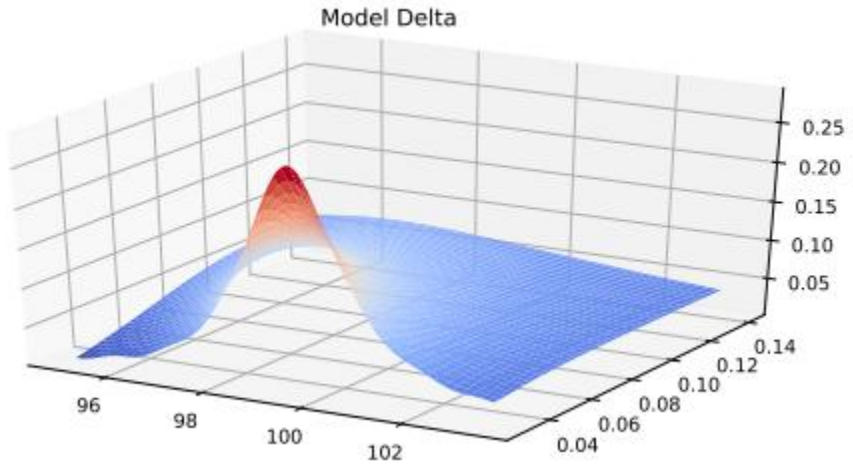
Heston model price asymptotics

[1] Deep Hedging, Buehler et all 2018, https://arxiv.org/pdf/1802.03042.pdf

Vanilla Deep Hedging

- Delta of a call spread in Black & Scholes [1]





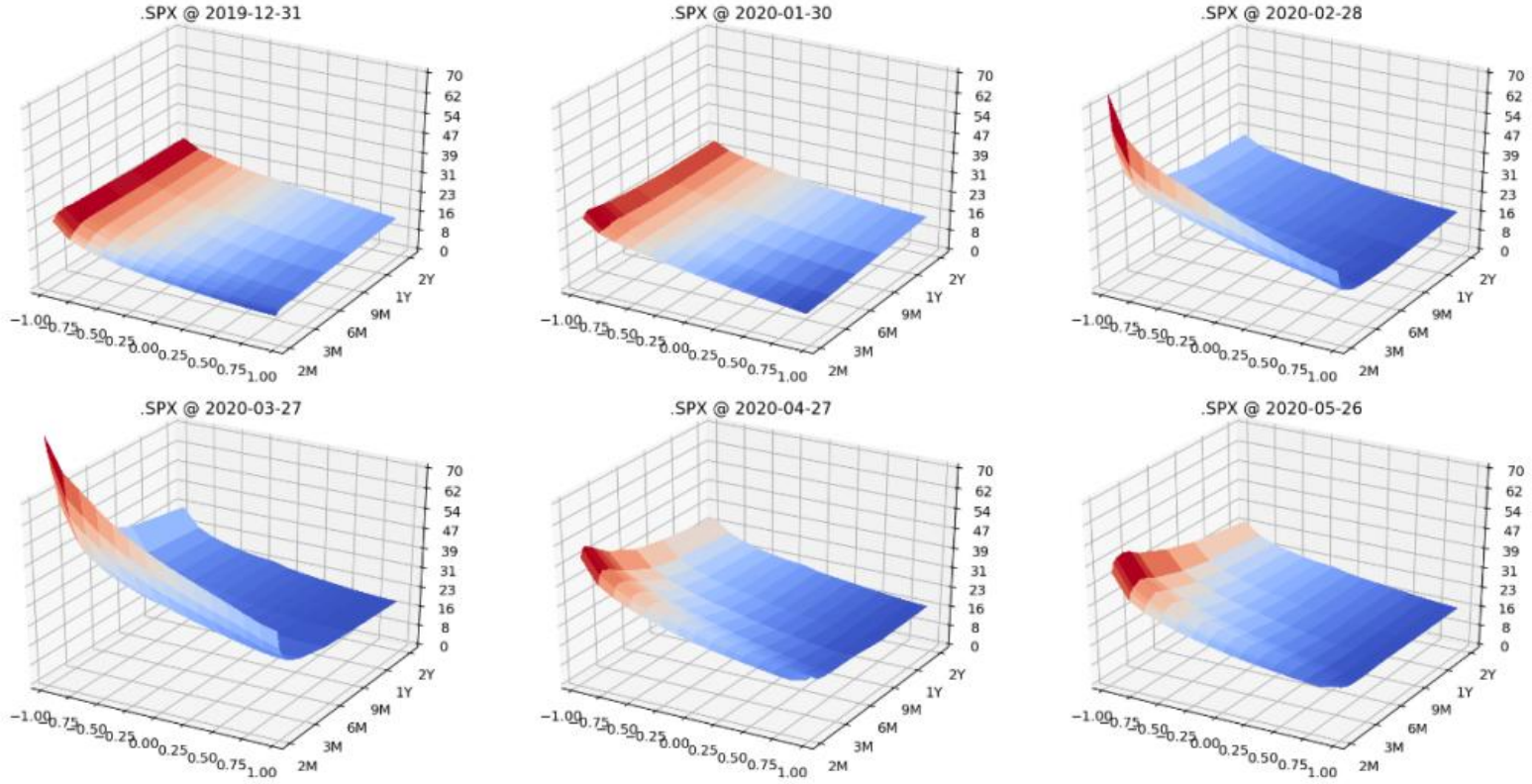
[1] Deep Hedging, Buehler et all 2018, https://arxiv.org/pdf/1802.03042.pdf

Market Simulation

- *Ideally simulate against purely historic data.*
- *Lack of data: for example, S&P 500 index spot and option prices.*
  - *10Y of data ~ 2500 data points*
  - *Typically index option surface has 100's of options*

→ Market Simulator

Market Simulation



- Our approach relies on training under full market data as opposed to classic “interpolation” derivatives models with ~2 factors.

Market Simulation

- Observe in every historical sample different tradable options w_t^i in with features f_t^i for $i = 1, \dots, N_t$ and a general market state m_t.
  - Embed "the market" into e_t using a set-invariant embedding.
  - Implement a hidden state process h_t.
  - Decode into arbitrage-free option prices using DLV/SANOS.

- Main difference: for market generators we are after *generative* models which generate distributions, not point outputs.
- Basic idea: use noise dW_t and write

$$h_t = \mathbf{N}(h_{t+dt}, m_t; dW_t)$$

- *No standardized architecture in the literature at this point – lots of ideas.*

Market Simulation

- There are several machine learning methods, all discussed in various papers
  - PCA [1]
  - Autoencoders [2], Variational Autoencoders [2]
  - Multi-asset versions [3]
  - SDE methods [4]
  - Neural SDEs [5]

[1] Deep hedging: learning to remove the drift, Buehler et al 2022 https://www.risk.net/cutting-edge/banking/7932226/deep-hedging-learning-to-remove-the-drift and https://arxiv.org/abs/2111.07844
[2] Deep Hedging: Learning to Simulate Equity Option Markets, Wiese et al 2020 https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3470756
[3] Multi-Asset Spot and Option Market Simulation, Wiese et al 2021 https://arxiv.org/abs/2112.06823
[4] Arbitrage-free neural-SDE market models, Cohen et al, 2021, https://arxiv.org/abs/2105.11053
[5] Neural SDEs as Infinite-Dimensional GANs, Kidger et al, 2021, https://arxiv.org/abs/2102.03657

Static Arbitrage

- We map the simulated h_t to option prices via a map

$$w_t^i, h_t \longmapsto H_t^i$$

- This map must prevent *static arbitrage* opportunities i.e. linear combinations $a'(H_T - H_t) - c_t(a) \geq 0$with a non-zero probability of generating a positive return.
- Deep Hedging will otherwise learn trading the maximum allowed size of such opportunities.

Static Arbitrage

- **Discrete Local Volatility** "DLV" [1] is a numerically robust form of local volatility.
  - Bijection between strictly arbitrage-free call prices C and DLV's σ.
  - Positivity of σ is sufficient to define a strictly arbitrage-free option surface.
  - Our recent paper [2] "SANOS" adds smooth interpolation to essentially the same method.

→ Decode h_t into statically arbitrage-free prices H_t using DLV/SANOS.

[1] Discrete Local Volatility for Large Time Steps (Extended Version), Buehler et al 2015, https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2642630
[2] SANOS – Smooth Arbitrage-Free Non parametric Option Surfaces, Buehler et al 2026, https://arxiv.org/abs/2601.11209

Statistical Arbitrage

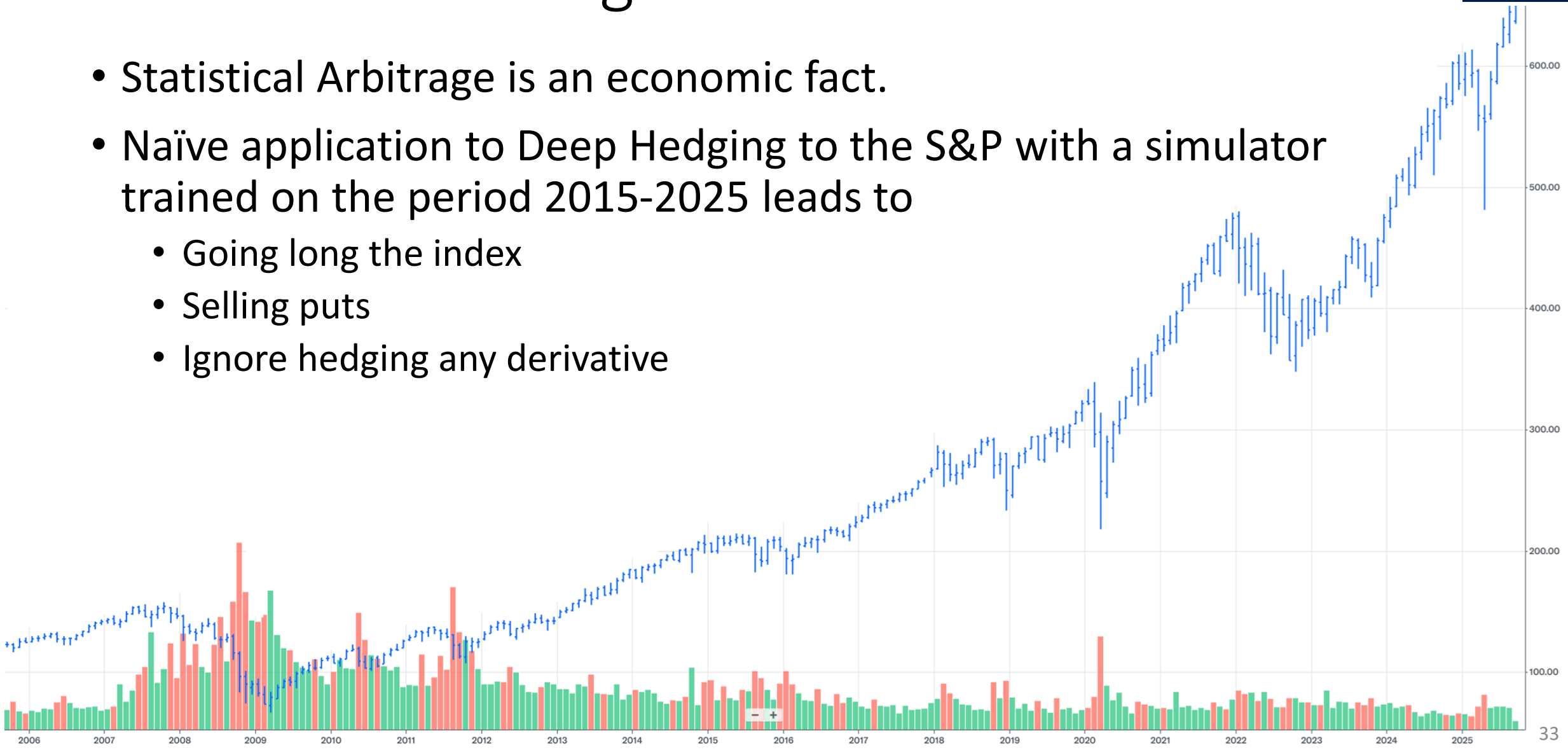
- We have earlier defined

$$U^*(Z) \coloneqq \sup_a : U\left(Z_T + \sum_{t=0}^{T-1} a_t' D H_{t:T} - c_t(a_t)\right)$$

- We have no static arbitrage.
- What about $U^*(0)$ which represents the value of an empty portfolio?
- We say that the market exhibits **statistical arbitrage** if $U^*(0) > 0$.
- Happens naturally.

Statistical Arbitrage

- Statistical Arbitrage is an economic fact.
- Naïve application to Deep Hedging to the S&P with a simulator trained on the period 2015-2025 leads to
  - Going long the index
  - Selling puts
  - Ignore hedging any derivative



Statistical Arbitrage

Removing the Drift [1]

- Solve the Deep Hedging problem for $Z = 0$ which yields a^0, y^0.
- Then change measure:

$$\frac{dQ}{dP} := u'\left(y_0 + \sum_{t=0}^{T-1} {a^0}'_t DH_{t:T} - c_t(a_t^0)\right)$$

- Then Q is a **near-martingale measure** in the sense that there is no trading strategy which makes money vs. the prevailing trading cost.
- It basically weights down all paths which are "too good".

[1] Learning Risk-Neutral Implied Volatility Dynamics, Buehler et al 2021, https://arxiv.org/pdf/2103.11948.pdf

Deep Hedging & Market Simulation

- Used in JP Morgan for Cliquet Options
  https://www.risk.net/derivatives/equity-derivatives/7921526/jp-morgan-testing-deep-hedging-of-exotics

- Good
  - Promising, versatile, intuitive machinery
  - Model-free in the sense that the machine is easily transferrable
  - Takes into account market frictions
  - Results deviate from classic models
  - Recent work on robust “bootstapped” hedging [1] and [2]

- But
  - Very heavy numerically
  - Results deviate from classic models and need explanation
  - Vanilla Deep Hedging requires re-training every day and every time Z changes

[1] Uncertainty-Aware Strategies: A Model-Agnostic Framework for Robust Financial Optimization through Subsampling, Buehler et al, 2025, https://arxiv.org/abs/2506.07299
[2] Distributional Adversarial Attacks and Training in Deep Hedging, He 2025, https://arxiv.org/abs/2508.14757

Deep Bellman Hedging

(on-going research)

Deep Bellman Hedging

- Assume that we are given a portfolio $Z^{(t)}$ with book-value $Z_t^{(t)}$ today. Here book value excludes past cash flows.
  - Tomorrow's book value of today's portfolio is $Z_{t+1}^{(t)}$.
  - Hence, cashflows and changes in book value are reflected in $dZ_t^{(t)}$.

- The one-day deterministic discount factor is $\beta_t \in (0, \beta^*]$ with $\beta^* < 1$.

- We can trade today in hedging instruments $H^{(t)}$ with book values $H_t^{(t)}$ for cost $c_t(a) \geq 0$.
  - We assume they do not pay any cashflows today when bought.
  - Hence $dH_t^{(t)}$ represents only the change in book value.

Deep Bellman Hedging

- If we trade a units of $H^{(t)}$ then tomorrow's portfolio is formally defined as

$$Z^{(t+1)} \coloneqq Z^{(t)} \oplus a' H^{(t)}$$

- Our **rewards** are the mark-to-book values

$$r_t \coloneqq dZ_t^{(t)} + a' dH_t^{(t)} - c_t(a)$$

- This gives rise to the Bellman equation c.f. [1]:

$$V^*\left(Z^{(t)}, s_t\right) = \sup_{\mathrm{a}} U_t \left[\beta_t V^*\left(Z^{(t)} \oplus a' H^{(t)}, s_{t+1}\right) + dZ_t^{(t)} + a' dH_t^{(t)} - c_t(a)\right]$$

[1] Deep Bellman Hedging, Buehler et al, https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4151026

Deep Bellman Hedging

- Theorem [1]: define the Bellman Operator T for candidate value functions V such that

$$TV\left(Z^{(t)}, s_t\right) = \sup_{\mathrm{a}} U_t\left[\beta_t V\left(Z^{(t)} \oplus a' H^{(t)}, s_{t+1}\right) + dZ_t^{(t)} + a' dH_t^{(t)} - c_t(a)\right]$$

Assume that there is no statistical arbitrage such that $T0 < \infty$.

Then the sequence $V^{n+1} \coloneqq TV^n$ starting in $V^0 \coloneqq 0$ converges to a finite optimal solution V^* for any monetary utility U.
The solution V^* then satisfies the Belman equation.

[1] Deep Bellman Hedging, Buehler et al, https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4151026

Deep Bellman Hedging

- Bellman relationship for an optimal value V^*.

$$V^*\left(Z_t^{(t)}, s_t\right) \coloneqq \sup_{\mathrm{a}} U_t\left[\beta_t V^*\left(Z^{(t)} \oplus a' H^{(t)}, s_{t+1}\right) + dZ_t^{(t)} + a' dH_t^{(t)} - c_t(a)\right]$$

- How do we represent our portfolio → use what trading has been using for years: the greeks and risk scenarios available to them as well as any cashflows.
  - Typically, we have these values historically at individual hedging and OTC instrument level.
  - We can therefore sample over historic periods and solve for V^*.

[1] Deep Bellman Hedging https://arxiv.org/abs/2207.00932

Practical Implementation

- Numerical solution via Actor-Critic:

$$TV\left(\mathrm{Z}_t^{(t)}, s_t\right) \coloneqq \sup_{\mathrm{a,y}} E_t\left[u\left(y + \beta_t V\left(Z^{(t)} \oplus a' H^{(t)}, s_{t+1}\right) + dZ_t^{(t)} + a' dH_t^{(t)} - c_t(a)\right) - y\right]$$

- **Actor:** Start in $V^0 \coloneqq 0$; given V^{n-1} maximize a^n and y^n which also yields samples of TV^n.

Practical Implementation

- **Critic:** given samples of TV^{n-1} solve for a neural network V^n to satisfy

$$V^n(w, s_t) \equiv TV^{n-1}(w, s_t)$$

- This interpolation program may also be solved by simpler, classic methods such as kernel interpolators.

Deep Bellman Hedging

- Implementation with fixed T: worked [1]
- Actual actor/critic: pretty unstable with results so far for only simple cases (such as portfolios of vanillas)
- [2] was/is the first of its kind; much more work to be done

[1] Deep Hedging: Continuous Reinforcement Learning for Hedging of General Portfolios across Multiple Risk Aversions, Murray et al, https://arxiv.org/abs/2207.07467
[2] Deep Bellman Hedging, Buehler et al, https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4151026

Thank you for your attention.