Efficient Trading with Price Impact*

Xavier Brokmann[1], Lukas Gonon[2,3], Guangyi He[2], David Itkin[4], and Johannes Muhle-Karbe[2]

[1]Qube Research and Technologies†
[2]Imperial College London
[3]University of St. Gallen
[4]London School of Economics

November 14, 2024

Abstract

This paper studies how to compute trading strategies that trade off expected returns, risk, and the trading costs generated by price impact in an efficient manner. For a general model where impact is nonlinear and decays only gradually, we show that the performance of linear feedback policies can be computed in closed form. These are optimal for linear impact and linear decay dynamics only. However, when the linear impact parameters are optimized for the nonlinear impact model at hand, then the performance of the resulting policy is competitive with state-of-the-art neural network methods unless the market is so illiquid that even the performance of optimized strategies is no longer attractive net of trading costs.

1 Introduction

A crucial challenge in the implementation of systematic trading strategies is to mitigate the adverse price impact that the corresponding trades generate. If this is not done in an appropriate manner, then impact costs quickly turn apparently profitable strategies into losing bets.

A rich empirical literature documents the subtle features of price impact. On the one hand, its magnitude does not depend linearly on order sizes, but is instead more accurately described by a "square-root law" (Loeb, 1983; Hasbrouck, 1991). On the other hand, impact is also not static but gradually decays as liquidity recovers following a large order (Hasbrouck, 1991; Biais et al., 1995). A sizeable fraction of this decay happens quickly, whereas another component is surprisingly persistent, a multiscale behavior consistent with a power law (Bouchaud et al., 2009).[1]

Incorporating these empirical features into optimal trading is a challenging task for which analytical results are only available in special cases, e.g., when impact is linear (Gârleanu and Pedersen, 2013, 2016; Collin-Dufresne et al., 2020; Abi Jaber and Neuman, 2022), impact is small (Guasoni and Weber, 2020; Cayé et al., 2020), or if there are no risk constraints (Hey et al., 2023).

As a consequence, one naturally has to turn to numerical methods to incorporate price impact into more general optimal trading problems. Modern machine learning methods open the door to tackling complex problems of this kind, but the results are often difficult to interpret and lack simple, robust benchmarks. In the present paper, we therefore propose to use the class of linear feedback controls as such a benchmark. These strategies are optimal when price impact is linear and decays on a single exponential time scale (Gârleanu and Pedersen, 2016). If one simply plugs in point estimates for the price impact parameters, it is not surprising that this leads to substantial performance losses relative

*The authors gratefully acknowledge financial support from Qube Research and Technologies, and pertinent remarks by Hans Buehler, Natascha Hey and Peter Schmidt.

†The first author contributed to this work while actively working at Qube RT.

[1]For a detailed account of the more recent empirical literature, we refer to the textbooks Bouchaud et al. (2018); Webster (2023) and the references therein.

to the optimal strategy that takes into account the nonlinear nature of price impact (Hey et al., 2024). Whence, we instead view impact level and decay rate as tuning parameters that need to be chosen to optimize performance in the presence of a more realistic price impact model.

Moallemi and Sağlam (2017) propose an approach in this spirit in a very general context, where it allows to bring to bear efficient methods from convex optimization. Brokmann et al. (2024) focus on models with instantaneous impact decay as in Almgren et al. (2005), where each trade only affects its own execution price but not subsequent ones. They show that the performance of linear strategies can then be computed in closed form, so that no numerical optmization is required at all.

In the present study, we develop results of this kind for general price impact models where impact can be of an arbitrary concave form and decays gradually according to a general decay kernel. This impact model nests virtually all specifications from the empirical literature as special cases. Nevertheless the performance of linear feedback strategies can still be computed explicitly, up to some integrals of the decay kernel which can also be evaluated in closed form in concrete specifications. With this closed-form expression for the performance of an arbitrary linear policy at hand, the tuning parameters can in turn be optimized in a straightforward manner to obtain optimal "effective" values of the linear price impact and decay parameters.

The resulting trading rules are available in closed form and in turn easy to analyze and interpret. However, this analysis leaves open the key question how much performance is sacrificed by restricting to the subclass of linear policies. To assess this, their performance needs to be compared to numerical methods that solve for the globally optimal trading policy. In Brokmann et al. (2024), the Viterbi algorithm of Kolm and Ritter (2014) is used for this purpose in models with instantaneous impact decay. This approach can also be adapted to the present context with gradual impact decay, but achieving the precision necessary to improve on linear policies typically leads to infeasibly long runtimes. We therefore instead focus on feedback controls parametrized by a suitable family of neural networks, which we in turn optimize using stochastic gradient descent-type algorithms and backpropagation, akin to classical policy gradient methods in reinforcement learning (Sutton and Barto, 2018).[2] The resulting algorithm matches the results of the Viterbi algorithm run at very high accuracy, but does so in a small fraction of the runtime.

In a numerical case study, we then compare the performance of the linear and neural network strategies. As a baseline case, we first focus on the model of Alfonsi et al. (2010), where impact has a square root form and decays at an exponential rate. The parameters for this experiment are calibrated to to resemble realistic market regimes faced by mid-to-high capacity trading firms. We find that neural network strategies allow to capture about 48.7% of a gross Sharpe ratio of 3 after accounting for realistic trading costs, whereas the corresponding figure for the optimal linear strategy is 47.4%. Put differently, neutral networks improve the Sharpe ratio net of costs by about 2.7%. This is a larger improvement compared to the 1-2% observed for models with instantaneous impact decay by Brokmann et al. (2024), but still rather modest compared to estimation errors for trading signals and price impact parameters.[3]

To illustrate this point, we also compare the performance of the linear and neural network strategies optimized for a single decay timescale in a model that combines three different exponential decay kernels fitted to the same dataset (Hey et al., 2023). We observe that out-of-model, the already modest performance gap is reduced further and drops from 2.7% to way below 1%, illustrating the robustness of the simple linear policies. Optimizing either linear or neural network strategies for multiple decay timescales allows to – in the model – further improve performance by a few percent.

Implementing neural network strategies for the fully path-dependent dynamics implied by power-law decay kernels requires significantly more effort. In contrast, linear strategies can still be tuned in the same manner as before. We find that roughly similar power law and multi-exponential kernels lead to quite similar trading strategies, but their performance is once more influenced to a much bigger extent by the precise form of the decay kernel.

[2]Similar methods have been used to optimize hedging strategies in the presence of instantaneous trading costs by Buehler et al. (2019); Kolm and Ritter (2019), for example. Other machine learning methods for optimal trading with price impact have been proposed by Moallemi and Wang (2022); Leal et al. (2022); Cartea et al. (2023); Chen et al. (2024), for example. The contemporaneous learning of impact parameters and optimal trading strategies is studied for linear price impact models by Park and Van Roy (2015); Neuman and Zhang (2023), for example.

[3]Bigger improvements can be obtained using neural networks when the strategy is near capacity because i) trading costs are high, ii) impact decay is slow, or iii) price forecasts ("alpha signals") are not accurate or persistent enough. However, in most of these scenarios, trading costs degrade the performance so severely that even optimized strategies are not attractive anymore. Conversely, in most scenarios where the strategy does remain attractive also net of costs, the extra performance that can be obtained by adding nonlinear features to the optimal trading rules is small. The only notable exception is the case of very slow impact decay.

In summary, our analytical results for linear feedback policies are straightforward to apply and allow to obtain simple trading rules, whose performances nevertheless come close to state-of-the-art numerics based on neural networks. Incorporating nonlinear features is computationally very feasible with modern architectures and computing power but, in our experiments, only leads to significant performance gains when liquidity is too low to render any strategy attractive net of trading costs.

The remainder of this article is organized as follows. Section 2 introduces our general price impact model and shows how it nests the typical specifications from the literature. Section 3 in turn formulates the tradeoff between expected returns, trading costs, and risk. For linear price impact and exponential impact decay, the model simplifies to the setup of Gârleanu and Pedersen (2016); the linear feedback policies that are optimal in this special case are summarized in Section 4. The performance of strategies of this form in our general model with nonlinear impact and impact decay is analyzed in Section 5, and compared to the performance of neural network strategies in Section 6. Section 7 conclude and Appendices A and B contain the derivations of the goal functional and the performance of linear strategies, respectively. Finally, Appendix C describes the neural-network approach that we use for the numerical optimization of our goal functional.

2 Price Impact Model

2.1 Unaffected Prices

We consider two assets: the first one is safe (and its price is normalized to one); the second one is risky in that its *unaffected price* has dynamics

$$dS_t = \alpha f_t \, dt + \sigma dB_t. \tag{2.1}$$

Here, B_t is a standard Brownian motion driving the price shocks with volatility $\sigma > 0$. Future returns are partially predictable through the *trading signal* f_t; the parameter $\alpha > 0$ regulates the accuracy of these predictions. For tractability, we assume as in many related studies that the predictor has linear mean-reverting dynamics:

$$df_t = -\phi f_t \, dt + \sqrt{2\phi} dW_t, \tag{2.2}$$

for another standard Brownian motion W_t independent of B_t.[4] The mean reversion and volatility parameters of the signal are chosen so that its stationary distribution is normalized to a Gaussian with mean zero and unit variance; the parameter ϕ in turn controls the half life of the signal.

2.2 Nonlinear Price Impact and Impact Decay

When large trades are executed quickly, they are not settled at the unaffected price (2.1) but instead incur price impact. Empirically, this price impact is a nonlinear function of the trade size and decays over several different timescales – a sizeable part vanishes rapidly, while another component lingers for a surprisingly long time. We capture both of these effects in a general price impact model, which nests the typical specifications from the empirical literature as special cases, see Section 2.3 for a range of examples.

To wit, we denote by Q_t the quantities of the risky asset held by a large trader, and write $P_t = S_t + I_t$ for the *market price* of the risky asset. It is composed of the *unaffected price* S_t (which models price changes caused by other traders or news, for example) and the *price impact* generated by the trader's current and past transactions dQ_s, $s \leq t$:

$$I_t = \sum_{n=1}^{N} h(J_t^n), \quad \text{where} \quad J_t^n = \int_0^t K_n(t-s) dQ_s. \tag{2.3}$$

Here, each process J_t^n, $n = 1, \ldots, N$ is a different weighted average of current trades and past trades, for which the impact of past transactions decays according to a *propagator kernel* $K_n(t-s)$. As in Alfonsi et al. (2010), price impact in turn is a nonlinear function $h(J_t^n)$ of these weighted averages, allowing to capture concave dependencies.[5] Considering linear combinations of several such models in (2.3) allows to

[4] This means that the predictor f_t is optimal in that it is independent from the noise B_t. If the signal and noise were auto-correlated in time then one could construct a better predictor f_t.

[5] An apparently similar but different effect is the concavity of the price impact of individual trades. Bouchaud et al. (2009) provide an overview of the empirical literature on such "local concavities"; optimal trading in this context is studied by Curato et al. (2017); Muhle-Karbe et al. (2023). The present study in stead focuses on "global concavities" at the metaorder level.

incorporate different impact concavities at different decay timescales. For example, a nearly permanent impact component can be modeled with linear impact, whereas averages with faster decay can enter in a nonlinear fashion.

2.3 Examples

We now illustrate with a range of examples how the general framework from the previous section naturally includes the typical specifications from the empirical and optimal trading literatures.[6]

The linear model of Obizhaeva and Wang (2013) The analytically most tractable specification is to assume that the impact function

$$h(x) = \lambda x$$

is linear and the impact decay kernel is exponential:

$$K(t) = e^{-\beta t}.$$

In this case, $dJ_t = -\beta J_t dt + dQ_t$ is an exponentially weighted moving average and the corresponding impact state dynamics are linear. This allows to solve a wide range of optimal trading problems in closed form using the machinery of linear-quadratic stochastic control (see Section 5 for a summary and, e.g., Gârleanu and Pedersen (2013, 2016) or the textbook Cartea et al. (2015) for more details).

As the impact decay β tends to zero, price impact in the model of Obizhaeva and Wang (2013) becomes permanent ($I_t = \lambda Q_t$). Conversely, when the impact decay rate tends to infinity (and the impact level λ is rescaled accordingly), then the price impact becomes instantaneous (i.e., $I_t = \lambda \dot{Q}_t$ only depends on the current trading rate for smooth strategies $dQ_t = \dot{Q}_t dt$ and blows up otherwise, see Gârleanu and Pedersen (2016) for more details). Whence, these two limiting regimes correspond to the model of Almgren and Chriss (2000), another workhorse of the price impact literature.

The nonlinear model of Alfonsi, Fruth, and Schied (2010) Retaining the exponential decay kernel of Obizhaeva and Wang (2013) but considering a general impact function h leads to the model of Alfonsi et al. (2010). This allows to incorporate the nonlinear dependence of impact on trade sizes, with a square-root function typically providing the best fit among all power laws (cf., e.g., Hey et al. (2024, 2023) and the references therein):

$$h(x) = \lambda \mathrm{sign}(x)|x|^p.$$

When impact decay β becomes faster and faster (and the impact level λ is rescaled accordingly), price impact in the model of Alfonsi et al. (2010) converges to a nonlinear function $h(\dot{Q}_t)$ of the current trading speed. This limiting regime in turn recovers the model of Almgren (2003).

The model of Alfonsi et al. (2010) retains analytical tractability for risk-neutral optimization problems (Alfonsi et al., 2010; Hey et al., 2023, 2024), but there are no known analytical results for traders that are not only capacity but also risk constrained.

More general decay kernels Just like impact depends on trade sizes in a nonlinear fashion, its decay also does not happen on a single exponential time scale (Bouchaud et al., 2009; Bacry et al., 2015; Brokmann et al., 2015; Bucci et al., 2018). One way to incorporate this is to consider several exponential kernels

$$K_n(t) = e^{-\beta_n t}, \quad n = 1, \dots, N,$$

with different decay rates $\beta_n > 0$ as in Hey et al. (2023). For example, with one very fast and one very slow timescale, this specification nests, as a limiting case, the model of Almgren et al. (2005) with nonlinear instantaneous and linear permanent impact. Exponential kernels have the advantage of retaining Markovian price impact dynamics (albeit with each of the weighted averages becoming an additional state variable), but require fitting additional parameters for each extra decay timescale considered. A more parsimonious parametric model class are the *power law kernels* of Bouchaud et al. (2003),

$$K(t) = \frac{1}{(\epsilon + t)^\nu},$$

[6] To ease notation, we simply write h, J, and K without the superscript n, when dealing with a single impact function, state variable, or kernel, respectively.

for a *decay elasticity* parameter $\nu \in (0,1)$ and a *shift* parameter $\epsilon \geq 0$. Compared to exponential kernels, a single power law can capture several fast and slow decay timescales. However, power law kernels induce non-Markovian impact dynamics where the evolution of the current impact state truly depends on the entire history of trades and not just a few sufficient statistics. For linear price impact functions, some optimal trading results can still be obtained (Gatheral et al., 2012; Abi Jaber and Neuman, 2022), but nothing is known about optimal trading with nonlinear impact and such general decay kernels.

3 Goal Functional for Optimal Trading

We now outline how to formulate a suitable goal functional that trades off expected returns, risk, and trading costs due to price impact. To this end, we first discuss the trader's PnL net of trading costs for smooth trading strategies[7] and how to complement this with a suitable risk penalty. Then, we show how to reformulate the resulting goal functional in a form that easily extends to general strategies, and will form the basis of our subsequent analytical and numerical results.

Formulating the goal functional in a consistent manner for general strategies is not just a technical subtlety but crucially important to obtain stable numerical results. To wit, if bulk trades are not penalized consistently with their smooth approximations, then naive discretizations of the model typically admit "price manipulation". Sufficiently flexible numerical solvers (e.g., based on neural networks) in turn learn to exploit these inconsistencies in the model rather than the trading signal they are supposed to implement.

3.1 The Trader's PnL

We now consider the trader's gains and losses from trading when price impact is given by (2.3). We start by deriving the PnL of smooth strategies $Q_t = \int_0^t \dot{Q}_s ds$. As these only execute small orders of order $O(dt)$ at each time t, the corresponding execution costs are $P_t \dot{Q}_t dt + O(dt^2)$, because the additional impact of each trade on the current market price $P_t = S_t + I_t$ is of order $O(dt)$.[8] On a finite time horizon T, the balance on the trader's cash account is then $-\int_0^T (S_t + I_t) dQ_t$.

To value the corresponding terminal position Q_T, we use the fundamental price S_T.[9] The trader's PnL from trading until time T is $\mathrm{PnL}_T = Q_T S_T - \int_0^T (S_t + I_t) dQ_t$. With the integration by parts $Q_T S_T = \int_0^T S_t dQ_t + \int_0^T Q_t dS_t$ this can be rewritten as

$$\mathrm{PnL}_T = \int_0^T Q_t dS_t - \int_0^T I_t \dot{Q}_t dt.$$

Here, the first term is the usual stochastic integral that describes the gains and losses from changes of the unaffected price. The second term, in turn, records the additional effects of price impact on the trader's PnL. To extend this representation to strategies that are not smooth, we rewrite it in a form that does not depend on the trading rate $\dot{Q}_t = dQ_t/dt$ anymore:

Proposition 3.1 (PnL Equation)**.** *Under Assumption A.1, the investor's PnL when trading a smooth strategy $dQ_t = \dot{Q}_t dt$ over the time interval $[0,T]$ is given by*

$$\begin{aligned} \mathrm{PnL}_T(Q) = \int_0^T Q_t dS_t - \sum_{n=1}^N \frac{H^n(J_T^n) - H^n(J_0^n)}{K_n(0)} \\ + \sum_{n=1}^N \int_0^T h^n(J_t^n) \left(\frac{K_n'(0)}{K_n(0)} Q_t + \int_0^t \frac{K_n''(t-s)}{K_n(0)} Q_s ds \right) dt, \end{aligned} \tag{3.1}$$

where $H^n(x) = \int_0^x h^n(y) dy$ is the antiderivative of $h^n(x)$.

[7] That is, holdings that do not change rapidly due to block trades or diffusive fluctuations.

[8] In contrast, for holdings with a nontrivial Brownian motion component, both trade sizes and additional price impact are of order $O(dt^{1/2})$ and therefore lead to an additional correction term of order $O(dt)$ as in Gârleanu and Pedersen (2016). We sidestep dealing with these terms and how to specify where block trades are executed between the pre- and post-trade prices by focusing on smooth strategies here. We then reformulate the resulting goal functional in a form in which it can readily be extended to general strategies.

[9] In contrast to using the market price $P_T = S_T + I_T$, this avoids "illusionary profits" based on inflating the market price with one's own impact.

The proof of this result is delegated to Appendix A. Here, we just note that no integrals against the trades dQ_t appear in (3.1) anymore, so this represeantation readily extends to any strategy that can be approximated by smooth strategies.

To simplify the exposition as much as possible, we focus on the long-run average PnL per unit time $\lim_{T\to\infty}\frac{1}{T}\mathbb{E}[\mathrm{PnL}_T(Q)]$. After plugging in the dynamics (2.1) of the unaffected price, we see that subject to a mild transversality condition this is given by[10]

$$\begin{aligned}\mathrm{PnL}(Q) &= \lim_{T\to\infty}\frac{1}{T}\mathbb{E}[\mathrm{PnL}_T(Q)] \\ &= \lim_{T\to\infty}\frac{1}{T}\int_0^T \mathbb{E}\left[\alpha f_t Q_t + \sum_{n=1}^N h^n(J_t^n)\left(\frac{K_n'(0)}{K_n(0)}Q_t + \int_0^t \frac{K_n''(t-s)}{K_n(0)}Q_s ds\right)\right] dt.\end{aligned} \tag{3.2}$$

3.2 Inventory Costs and Risk Constraints

The most tractable way to penalize large inventories when optimizing the expected PnL (3.2) is to include a running cost $-\frac{\gamma\sigma^2}{2}Q_t^2 dt$ on the trader's squared positions as in Gârleanu and Pedersen (2013, 2016), where $\gamma > 0$ is a risk aversion parameter. The corresponding mean-variance goal functional in turn is

$$\mathcal{J}(Q) = \lim_{T\to\infty}\frac{1}{T}\int_0^T \mathbb{E}\left[\alpha f_t Q_t + \sum_{n=1}^N h^n(J_t^n)\left(\frac{K_n'(0)}{K_n(0)}Q_t + \int_0^t \frac{K_n''(t-s)}{K_n(0)}Q_s ds\right) - \frac{\gamma\sigma^2}{2}Q_t^2\right] dt. \tag{3.3}$$

As in classical mean-variance optimization, this objective can be viewed as the Lagrangian formulation of maximizing the PnL per unit time net of costs subject to the constraint that the average *dollar risk* per unit time,

$$\mathcal{R}^2(Q) = \lim_{T\to\infty}\frac{1}{T}\int_0^T \mathbb{E}[\sigma^2 Q_t^2]dt, \tag{3.4}$$

matches a given risk limit $\mathrm{DR}^2 > 0$. Another equivalent formulation is to maximize the Sharpe ratio $\mathrm{SR}(Q) = \mathrm{PnL}(Q)/\mathcal{R}(Q)$ (net of trading costs) for a given risk budget $\mathcal{R}(Q) = \mathrm{DR}$.

4 Solution with Linear Impact and Exponential Decay

With nonlinear price impact and general decay kernels, the optimization problem (3.4) is intractable and can only be tackled with numerical methods such as the ones outlined in Appendix C. However, for linear price impact decaying at a single exponential time scale, the corresponding optimization problem is linear-quadratic and has a simple explicit solution. Indeed, in this case the goal functional (3.3) simplifies as follows:

Lemma 4.1. *For linear price impact $h(x) = \lambda x$ and the exponential decay kernel $K(t) = e^{-\beta t}$, the goal functional* (3.3) *can be rewritten as*

$$\mathcal{J}(Q) = \lim_{T\to\infty}\mathbb{E}\left[\frac{1}{T}\int_0^T\left(\alpha Q_t f_t - \beta\lambda J_t Q_t - \frac{\gamma\sigma^2}{2}Q_t^2\right)dt\right], \tag{4.1}$$

for strategies satisfying a suitable transversality condition, see Appendix B. This goal functional is equivalent to the one studied in (Gârleanu and Pedersen, 2016, Equation (2.1)) in the limit where the time-discount rate tends to zero.

The solution for this problem has been derived by Gârleanu and Pedersen (2016) in terms of an algebraic Riccati equation. For easy reference, we provide a short self-contained derivation in Appendix B for the time-average control problem that formally corresponds to sending the time discount rate to zero. In this case, the solution in fact becomes fully explicit:

Theorem 4.2 (Optimal Strategy with Linear Impact and Exponential Decay)**.** *The optimal strategy Q_t^* for* (4.1) *is a linear feedback function of the current signal and an exponential moving average of current and past trades:*

$$Q_t^* = C_f f_t - C_J J_t, \tag{4.2}$$

[10] More specifically, we tacitly assume here that the local martingale part of $\int_0^T Q_t dS_t$ is a true martingale and that $\lim_{T\to\infty}\frac{1}{T}\mathbb{E}[H^n(J_T^n)] = 0$. In Appendix B we verify that these conditions are satisfied for the linear feedback controls considered in Sections 4 and 5.

where

$$C_f = \frac{\alpha(1+\beta/\phi)}{\gamma\sigma^2\left(\sqrt{1+\frac{2\lambda\beta}{\gamma\sigma^2}}+\beta/\phi\right)}, \quad C_J = \sqrt{1+\frac{2\lambda\beta}{\gamma\sigma^2}} - 1. \tag{4.3}$$

The corresponding risk is given by

$$\mathcal{R}^2(Q^*) = \frac{\alpha^2\sigma^2}{(\gamma\sigma^2)^2} \frac{(\beta/\phi+1)^2\left(\beta/\phi\sqrt{1+\frac{2\lambda\beta}{\gamma\sigma^2}}+1\right)}{\sqrt{1+\frac{2\lambda\beta}{\gamma\sigma^2}}\left(\sqrt{1+\frac{2\lambda\beta}{\gamma\sigma^2}}+\beta/\phi\right)^3}. \tag{4.4}$$

From the closed-form expressions of the optimal strategy in Theorem 4.2, it is straightforward to explore its comparative statics. Indeed, the coefficients C_f and C_J are both positive, so that the trader's holdings in the risky asset are increasing in the strength of the predictive signal and decreasing in the accumulated impact. Moreover, we see that C_f is decreasing in λ, while C_J is increasing in this parameter. This is intuitive since larger price impact affects the investor adversely. As such, the trader exploits a given signal less aggressively, in particular, if impact has already been accumulated in the same direction.

The corresponding risk $\mathcal{R}(Q^*)$ can be shown to be decreasing in both the impact and risk aversion parameters. This is again intuitive as risk averse traders avoid large positions due to the associated risk, whereas the capacity constraints induced by large impact costs also lead to smaller positions. The risk (4.4) ranges from infinity to zero as the risk aversion parameter varies from 0 to ∞. Together with the monotonicity in γ it follows that for a fixed price impact model (λ, β), there is a unique value $\gamma(\lambda, \beta)$ for which the optimal policy matches the given given risk constraint.

5 Performance of Linear Feedback Policies

We now turn our attention back to the general impact model (2.3) and the corresponding objective function (3.3). With nonlinear price impact and general decay kernels, no analytical solutions are available. One way forward is to tackle the problem using numerical methods, such as the algorithm of Kolm and Ritter (2014) inspired by the classical Viterbi algorithm, or by approximating the optimal feedback controls using neural networks and optimizing these using stochastic gradient descent as in Buehler et al. (2019), for example.

In this section, we present a simple alternative in the spirit of Moallemi and Sağlam (2017); Brokmann et al. (2024), namely to use the *linear* feedback controls from Theorem 4.2. These are available in closed form as a function of only two tuning parameters, impact level and impact decay (when the corresponding risk aversion is fixed by a risk limit). Of course, these policies cannot be expected to perform well in models with nonlinear impact and impact decay if the level and decay rate are chosen arbitrarily or as the statistical point estimates for the linear model (Hey et al., 2024). Instead, the key is that as in Brokmann et al. (2024), one needs to choose the "effective" parameters in the linear model to take into account the nonlinearities in a suitable manner, tailored to the target risk levels and trading signals in the problem at hand.

In order to implement this program, we consider linear models with arbitrary risk aversion $\boldsymbol{\gamma}$, impact level $\boldsymbol{\lambda}$, impact decay $\boldsymbol{\beta}$ and the corresponding exponential moving averages $d\boldsymbol{J}_t = -\boldsymbol{\beta}\boldsymbol{J}_t dt + dQ_t$ of the trader's current and past transactions. For each of these models, we then consider the corresponding optimal linear policy $Q_t^* = C_f f_t - C_J \boldsymbol{J}_t$ with coefficients from (4.2) (with λ, β, and γ replaced by $\boldsymbol{\lambda}$, $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$, respectively). More specifically, we compute the performance of these linear policies for the nonlinear goal functional (3.3). Using the properties of Gaussian processes, the performance of the linear strategies can generally be computed in closed form up to some integrals of the price impact function against the standard Normal distribution and integrals of the decay kernel and its derivatives:

Theorem 5.1 (Performance of Linear Strategies)**.** *Suppose Assumptions A.1 and B.1 hold and consider the linear strategy $Q_t^* = C_f f_t - C_J \boldsymbol{J}_t$ with coefficients* (4.3) *that are optimal in a linear model with impact level $\boldsymbol{\lambda}$, impact decay $\boldsymbol{\beta}$, and risk aversion $\boldsymbol{\gamma}$.*

Then, the corresponding risk is given by (4.4) *(with $\boldsymbol{\lambda}$, $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$ in place of λ, β, γ) and the expected PnL per unit time net of nonlinear impact costs is*

$$\begin{aligned}\mathrm{PnL}(\boldsymbol{\lambda},\boldsymbol{\beta};\boldsymbol{\gamma}) = {} & \frac{\alpha^2\boldsymbol{\gamma}\sigma^2(\boldsymbol{\beta}+\phi)^2}{(\boldsymbol{\beta}\boldsymbol{\gamma}\sigma^2+\xi\phi)^2} + \sum_{n=1}^{N} \frac{K_n'(0)}{K_n(0)} \frac{\tau_n}{\sigma_n} \mathbb{E}[h^n(\sigma_n Z) Z] \\ & + \sum_{n=1}^{N} \frac{\mathbb{E}[h^n(\sigma_n Z) Z]}{\sigma_n} \left(\int_0^\infty K_n''(t)\rho(t)dt + \int_0^\infty \int_0^\infty \frac{K_n''(t)K_n'(s)}{K_n(0)} \rho(|t-s|)dsdt \right),\end{aligned} \tag{5.1}$$

where Z is a standard normally distributed random variable and

$$\xi = \sqrt{\boldsymbol{\gamma}\sigma^2(\boldsymbol{\gamma}\sigma^2 + 2\boldsymbol{\lambda}\boldsymbol{\beta})}, \tag{5.2}$$

$$\rho(t) = C_1 e^{-\phi t} - C_2 e^{-\frac{\boldsymbol{\gamma}\boldsymbol{\beta}\sigma^2}{\xi}t}, \tag{5.3}$$

$$C_1 = \frac{\alpha^2\boldsymbol{\gamma}^2\sigma^4(\boldsymbol{\beta}-\phi)(\boldsymbol{\beta}+\phi)^3}{(\boldsymbol{\beta}\boldsymbol{\gamma}\sigma^2 - \xi\phi)(\boldsymbol{\beta}\boldsymbol{\gamma}\sigma^2 + \xi\phi)^3}, \tag{5.4}$$

$$C_2 = \frac{2\alpha^2\boldsymbol{\gamma}^2\boldsymbol{\lambda}\boldsymbol{\beta}^2\sigma^4\phi(\boldsymbol{\beta}+\phi)^2}{\xi(\boldsymbol{\beta}\boldsymbol{\gamma}\sigma^2 - \xi\phi)(\boldsymbol{\beta}\boldsymbol{\gamma}\sigma^2 + \xi\phi)^3}, \tag{5.5}$$

$$\tau_n = K_n(0)\rho(0) + \int_0^\infty K_n'(t)\rho(t)dt,$$

$$\sigma_n^2 = K_n^2(0)\rho(0) + 2K_n(0)\int_0^\infty K_n'(t)\rho(t)dt + \int_0^\infty\int_0^\infty K_n'(t)K_n'(s)\rho(|t-s|)dsdt.$$

Remark 5.2. (i) *It is straightforward to check that $C_1 > C_2$ so that $\rho(0) > 0$;*

(ii) *When $\phi = \boldsymbol{\gamma}\boldsymbol{\beta}\sigma^2/\xi$, ρ in (5.3) should be understood as the limit $\phi \to \boldsymbol{\gamma}\boldsymbol{\beta}\sigma^2/\xi$.*

When the impact function is of power form and the decay kernel is a linear combination of exponentials, then the integrals appearing in Theorem 5.1 can all readily be computed in closed form, leading to a fully explicit formula for the performance of each linear strategy.

Corollary 5.3 (Power Law Impact and Exponentially Decay Kernels)**.** *For $n = 1, \dots, N$ suppose that $h^n(x) = \lambda_n \text{sign}(x)|x|^{p_n}$ for some $p_n \in (0,1)$ and $\lambda_n > 0$, and let $K_n(t) = e^{-\beta_n t}$ for some $\beta_n > 0$. Then for any linear feedback strategy $Q_t = C_f f_t - C_J \boldsymbol{J}_t$ with coefficients C_f, C_J from (4.3) with impact level $\boldsymbol{\lambda}$, impact decay $\boldsymbol{\beta}$ and risk aversion $\boldsymbol{\gamma}$, we have*

$$\begin{aligned}\text{PnL}(\boldsymbol{\lambda},\boldsymbol{\beta};\boldsymbol{\gamma}) &= \frac{\alpha^2\boldsymbol{\gamma}\sigma^2(\boldsymbol{\beta}+\phi)^2}{(\boldsymbol{\beta}\boldsymbol{\gamma}\sigma^2 + \xi\phi)^2} \\ &- \sum_{n=1}^N \Gamma\left(\frac{p_n}{2}\right)\frac{p_n\beta_n\lambda_n}{2\sqrt{\pi}}\left(\frac{2\alpha^2\boldsymbol{\gamma}^2\sigma^4\phi(\boldsymbol{\beta}+\phi)^2(\boldsymbol{\beta}^2\xi + \beta_n\phi\xi + \boldsymbol{\gamma}\boldsymbol{\beta}\sigma^2(\beta_n+\phi))}{\xi(\beta_n+\phi)(\boldsymbol{\gamma}\boldsymbol{\beta}\sigma^2 + \beta_n\xi)(\phi\xi + \boldsymbol{\gamma}\boldsymbol{\beta}\sigma^2)^3}\right)^{\frac{p_n+1}{2}},\end{aligned} \tag{5.6}$$

where $\Gamma(x)$ is the gamma function and we recall that $\xi = \xi(\boldsymbol{\lambda},\boldsymbol{\beta};\boldsymbol{\gamma})$ is given by (5.2).

Similarly, for shifted power law kernels the corresponding integrals can still be expressed in terms of special functions:

Corollary 5.4 (Power Law Impact and Shifted Power Law Decay Kernel)**.** *Let $N = 1$ and suppose that (i) $h(x) = \lambda\text{sign}(x)|x|^p$ for some $p \in (0,1)$ and $\lambda > 0$, and (ii) $K(t) = (t+\epsilon)^{-\nu}$ for some $\nu \in (0,1)$ and $\epsilon > 0$. Then for any linear feedback strategy $Q_t = C_f f_t - C_J \boldsymbol{J}_t$ with coefficients C_f, C_J from (4.3) with impact level $\boldsymbol{\lambda}$, impact decay $\boldsymbol{\beta}$ and risk aversion $\boldsymbol{\gamma}$ we have*

$$\begin{aligned}&\text{PnL}(\boldsymbol{\lambda},\boldsymbol{\beta};\boldsymbol{\gamma}) \\ &= \frac{\alpha^2\boldsymbol{\gamma}\sigma^2(\boldsymbol{\beta}+\phi)^2}{(\boldsymbol{\beta}\boldsymbol{\gamma}\sigma^2 + \xi\phi)^2} - \lambda\nu\frac{2^{\frac{p+1}{2}}\Gamma(1+\frac{p}{2})}{\sqrt{\pi}}\frac{\tau}{\sigma_{\text{power}}^{1-p}} \\ &+ \nu(\nu+1)\frac{2^{\frac{p+1}{2}}\Gamma(1+\frac{p}{2})}{\sqrt{\pi}\sigma_{\text{power}}^{1-p}}\left(C_1\phi^{\nu+1}e^{\phi\epsilon}\Gamma(-\nu-1,\phi\epsilon) - C_2\left(\frac{\boldsymbol{\gamma}\boldsymbol{\beta}\sigma^2}{\xi}\right)^{\nu+1}e^{\frac{\epsilon\boldsymbol{\gamma}\boldsymbol{\beta}\sigma^2}{\xi}}\Gamma\left(-\nu-1,\frac{\epsilon\boldsymbol{\gamma}\boldsymbol{\beta}\sigma^2}{\xi}\right)\right) \\ &- \frac{\nu^2(\nu+1)}{\epsilon^\nu}\frac{2^{\frac{p+1}{2}}\Gamma(1+\frac{p}{2})}{\sqrt{\pi}\sigma_{\text{power}}^{1-p}}\Bigg(C_1\left(\mathcal{I}(\phi,\nu+1,\nu+2) + \mathcal{I}(\phi,\nu+2,\nu+1)\right) \\ &\qquad - C_2\left[\mathcal{I}\left(\frac{\boldsymbol{\gamma}\boldsymbol{\beta}\sigma^2}{\xi},\nu+1,\nu+2\right) + \mathcal{I}\left(\frac{\boldsymbol{\gamma}\boldsymbol{\beta}\sigma^2}{\xi},\nu+2,\nu+1\right)\right]\Bigg),\end{aligned} \tag{5.7}$$

where

$$\tau = \frac{C_1 - C_2}{\epsilon^\nu} - \nu C_1 \phi^\nu e^{\phi\epsilon}\Gamma(-\nu, \phi\epsilon) + \nu C_2 \left(\frac{\boldsymbol{\gamma\beta\sigma}^2}{\xi}\right)^\nu e^{\frac{\epsilon\boldsymbol{\gamma\beta\sigma}^2}{\xi}}\Gamma\left(-\nu, \frac{\epsilon\boldsymbol{\gamma\beta\sigma}^2}{\xi}\right),$$

$$\sigma^2_{\text{power}} = \frac{C_1 - C_2}{\epsilon^{2\nu}} - \frac{2\nu}{\epsilon^\nu} C_1 \phi^\nu e^{\phi\epsilon}\Gamma(-\nu, \phi\epsilon) + \frac{2\nu}{\epsilon^\nu} C_2 \left(\frac{\boldsymbol{\gamma\beta\sigma}^2}{\xi}\right)^\nu e^{\frac{\epsilon\boldsymbol{\gamma\beta\sigma}^2}{\xi}}\Gamma\left(-\nu, \frac{\epsilon\boldsymbol{\gamma\beta\sigma}^2}{\xi}\right)$$
$$+ 2\nu^2 C_1 \mathcal{I}(\phi, \nu+1, \nu+1) - 2\nu^2 C_2 \mathcal{I}\left(\frac{\boldsymbol{\gamma\beta\sigma}^2}{\xi}, \nu+1, \nu+1\right),$$

$$\mathcal{I}(x, q, r) = x^{q+r-2}\left(\frac{{}_2F_2(1, 2-q-r; 2-r, 3-q-r; \epsilon x)}{(\epsilon x)^{q+r-2}(r-1)(q+r-2)},\right.$$
$$\left. + \frac{\pi(-1)^q \csc(\pi r)(\Gamma(1-q) - \Gamma(1-q, -\epsilon x)) - \pi\Gamma(1-q)\csc(\pi(q+r))}{\Gamma(r)}\right).$$

Here, ξ, C_1, C_2 are given by (5.2)*,* (5.4) *and* (5.5) *respectively, $\Gamma(\cdot)$ is the Gamma function, $\Gamma(\cdot,\cdot)$ is the upper incomplete Gamma function and ${}_2F_2$ is the generalized hypergeometric function.*

Using the explicit expressions (5.1) and especially (5.6),(5.7), the linear policy with the maximal Sharpe ratio net of trading costs can be determined in a straightforward manner. Indeed, to this end, we consider a grid of values ($\boldsymbol{\lambda}$, $\boldsymbol{\beta}$) of impact levels and decay rates. For each of these, we then use the formula (4.4) for the risk of the corresponding strategy to find the value of the risk aversion parameter $\boldsymbol{\gamma}(\boldsymbol{\lambda}, \boldsymbol{\beta})$ for which a given risk constraint binds. Using the formulas for the PnL (Equation (5.1), resp. (5.6),(5.7)), we then compute the net Sharpe ratio of each linear strategy on the grid and finally select the value for which the optimal value is attained. As a byproduct, this analysis not only yields the optimal linear policy, but also provides a sensitivity analysis with respect to changes of the tuning parameter $\boldsymbol{\lambda}$ and $\boldsymbol{\beta}$. The performance of this approach under realistic market conditions is investigated in the next section.

6 Numerical Results

The linear feedback policies from the previous sections offer a simple and interpretable approach to trade scheduling for general price impact dynamics. Using our closed-form expressions for their performance, comparative statics can be computed directly and the tuning parameters can readily be optimized to take into account nonlinearities of the impact function and decay dynamics.

As in Moallemi and Sağlam (2017), this leads to an optimal policy in the chosen parametric class, but leaves open the key question how much performance is sacrificed relative to the global optimum. For a model with instantaneous impact decay, Brokmann et al. (2024) compared the performance of optimal linear policies to policies computed using the Viterbi algorithm of Kolm and Ritter (2014), run at very high accuracy. In the present study, we instead benchmark the performance of the linear policies against feedback controls parametrized by neural networks, which are in turn optimized via stochastic gradient descent and backpropogation methods. Appendix C.2 describes the network architecture, implementation and related methods in the financial literature. In Appendix D we verify that this approach recovers the results of Brokmann et al. (2024) but, at the same time significantly reduces computation time and readily generalizes to models with impact decay as well.[11] This allows us to assess the suboptimality of linear strategies across a range of parameter regimes and price impact models.

6.1 Parameter Calibration

A key step for a meaningful comparison between the different computational methods is to choose reasonable model parameters that match impact dynamics and trading costs observed in real markets.

For the trading signals and risk constraints, we follow Brokmann et al. (2024) and choose the model parameters to resemble realistic market regimes faced by mid-to-high capacity trading firms. To calibrate the market parameters, we assume that there are $d = 500$ uncorrelated homogenous assets with daily volatility $\sigma = 0.02$. Then, the multivariate optimization problem decouples into d individual optimization problems of the form studied in Section 3. We suppose each asset has an independent predictor with a half-life of 5 days (corresponding to $\phi = 0.139$) and set the maximal (yearly) frictionless portfolio Sharpe

[11] We have also adapted and implemented the Viterbi algorithm for models with impact decay, but do not report the results here since it does not outperform the neural network strategies and is computationally far more expensive.

ratio to SR = 3 by choosing $\alpha = \sigma \times \mathrm{SR}/\sqrt{256d} = 1.67 \times 10^{-4}$ (in daily units). More or less persistent trading signals and lower or higher frictionless Sharpe ratios are considered in a robustness check below.

For the price impact model, we focus on an impact function of square-root form ($p = 0.5$), as corroborated by many empirical studies in different contexts. There is less agreement about the speed of impact decay and estimates vary between more and less liquid securities. We first focus on a single decay timescale ($N = 1$) for simplicity. As a baseline case, we use an exponential kernel with impact half life of a third of a trading day ($\beta = 2$), in line with estimates from CFM meta order data reported in Hey et al. (2023). The impact of slower or faster decay is in turn explored in another one of our robustness checks below. We also consider the three-timescale model from Hey et al. (2023), where $K(t) = w_1 e^{-\beta_1 t} + w_2 e^{-\beta_2 t} + w_3 e^{-\beta_3 t}$ for timescale weights w_i and decay rates β_i estimated as

$$\beta_1 = 2, \quad \beta_2 = 1/65, \quad \beta_3 = 1/7, \qquad\qquad w_1 = 1.2, \quad w_2 = 0.6, \quad w_3 = 0.2.$$

In addition to these exponential decay profiles, we also illustrate that our methods readily extend to (shifted) power law kernels $K(t) = (\epsilon+t)^{-\nu}$, which capture multiple timescales of decay in a parsimonious manner. We take $\nu = 0.8$ as estimated in Brokmann et al. (2015) on proprietary CFM meta order data. Finally, we set the shift parameter equal to $\epsilon = 0.5$, as this choice leads to a similar one-day accumulated impact trajectory as the three-exponential model for a TWAP strategy trading 1% of average daily volume (ADV), roughly in line with the meta orders studied in Hey et al. (2023).

In all specifications, the push factor λ is chosen to obtain a realistic level of trading costs. More specifically, we assume the ADV for each of the traded securities is \$15.8M in line with US equity markets. We then choose the parameter λ (for all of the kernel choices above) so that the execution of a TWAP day order accounting for 1% of ADV induces an average price impact of 1.8bps in line with Brokmann et al. (2024). In the baseline setting of a single exponential decay kernel this led us to setting $\lambda = 0.0035$. The impact of higher trading costs is explored in another robustness check below.

6.2 One Decay Timescale

Performance comparison We now compare the performance of the linear strategies from Section 5 and the neural network strategies. For our baseline parameters with one exponential decay timescale, Figure 1 displays the optimal effective parameters $(\boldsymbol{\lambda}^*, \boldsymbol{\beta}^*)$ and the performance of the corresponding trading strategies for three different risk constraints on the portfolio of 500 uncorrelated assets.[12]

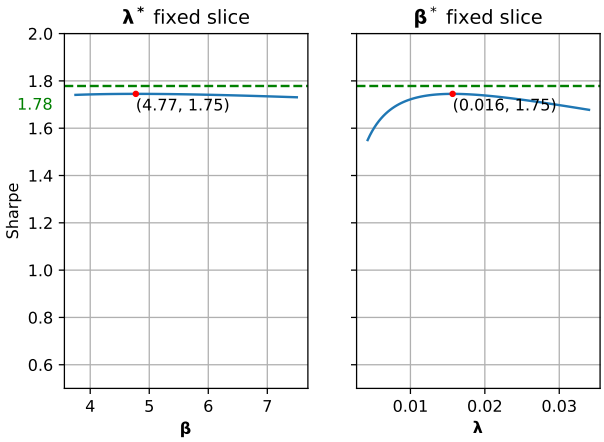
In all three cases, we see that the performance of the linear policies is relatively insensitive to the choice of the impact decay β, whereas varying the effective impact level has a considerably bigger effect. At the optimum, we see that the outperformance of the neural networks increases as the impact costs of the strategy grow for higher risk levels. With gradual impact decay, the gap between the optimal linear policy and the global optimum generally increases compared to the models with instantaneous impact decay studied in Brokmann et al. (2024). For example, in our baseline case with a daily risk of 1.3M, the optimal neural network policy allows to capture a fraction 1.46/3=48.7% of the optimal frictionless Sharpe ratio, whereas the corresponding figure for the optimal linear strategy is 1.42/3=47.4%. Put differently, adding nonlinear features improves the performance net of costs by about 2.7% in this case, whereas the corresponding figure with instantaneous impact decay is 1.3% in Brokmann et al. (2024).

However, unless the strategy is severely capacity constrained as in the lower panel of Figure 1 (where trading costs drastically degrade the Sharpe ratio by more than two thirds to less than one), the relative loss is still very modest, in particular, when compared to the challenges in fitting predictive models and price impact parameters.

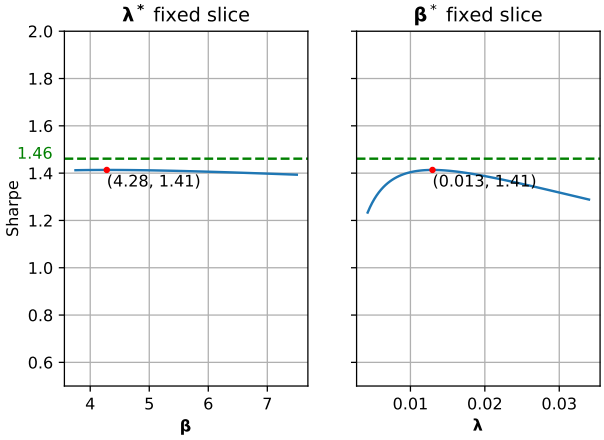
Robustness checks Table 1 assesses the robustness of the baseline results from the previous section with respect to variations of the model parameters. Each row corresponds to our baseline model from Section 6.1, with only the stated parameter perturbed. The outperformance of the neural network generally increases when the trading costs have a bigger effect because i) the level of price impact is higher, ii) it decays more slowly, or iii) the trading signal is weaker or less persistent.

Crucially, we see that as in the previous section, the neural networks typically only yield significant improvements in the regimes where the net Sharpe ratio is reduced too much to render any strategy attractive. The only notable exception is the case of very slow impact decay, which does not degrade performance too much but nevertheless allows nonlinearities in the optimal policy to improve performance by about 4%.

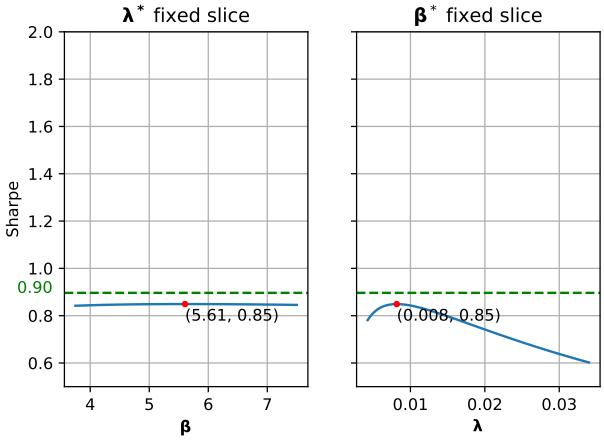
[12] For N uncorrelated homogenous assets, the daily risk associated with holding Q shares split evenly between assets is $\mathrm{DR} = \sigma Q/\sqrt{N}$, so the smallest risk level we use corresponds to $Q = 560$M shares overall. The largest risk level corresponds to a position that is about ten times as large.



(a) DR=0.5M



(b) DR=1.3M



(c) DR=5.3M

Figure 1: Performance of neural networks (dashed green) and linear policies (solid blue) when one of the effective parameters $\boldsymbol{\beta}$,$\boldsymbol{\lambda}$ is fixed to its optimal value and the other varies. The daily risk (DR) is increasing from the upper to the lower panels.

Perturbed parameter value	NN Sharpe	Linear Sharpe	% Outperformance
$\phi \sim$ Half-life = 3 days	0.9	0.85	5.9%
$\phi \sim$ Half-life = 10 days	2.04	2.01	1.5%
Frictionless SR = 2	0.76	0.72	5.6%
Frictionless SR = 4	2.21	2.17	1.8%
$\lambda \sim$ 5bp per 1% ADV	0.65	0.61	6.6%
$\lambda \sim$ 10bp per 1% ADV	0.235	0.215	9.3%
$\beta \sim$ Half-life = 0.5 days	1.37	1.33	3%
$\beta \sim$ Half-life = 1 days	1.26	1.21	4.1%

Table 1: Performances with perturbed parameters at DR $\approx$ 1.3M. In the table "$\sim$" means "corresponds to".

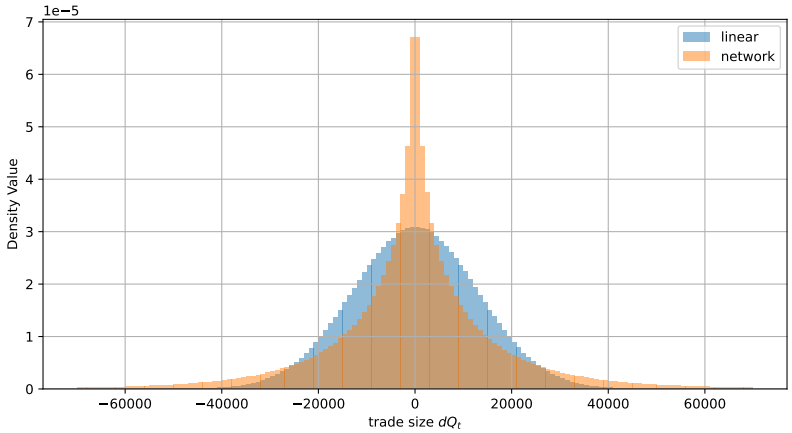


Figure 2: Trade histogram at the DR = 1.3M level

Strategy comparison To gain a better understanding of the differences between linear and neural network strategies, we now compare the corresponding trades and holdings. By construction, the linear strategy leads to trades dQ_t which are normally distributed. The network strategy, by contrast, learns to trade taking into account the true nonlinear underlying price impact leading to a different, non-Gaussian, distribution of the trades. Figure 2 displays histograms of the trades dQ_t of the two strategies, obtained by simulation on a discrete time grid with 100 trades per day. We see that the network trades have both a higher peak and fatter tails than the linear strategy, so that both very small and very large trades occur more frequently, and mid-size trades less frequently. This allows the strategy to benefit from the concavity of price impact, which marginally (i.e., per share) penalizes large trades less than it penalizes mid-size trades.

To understand this effect better, in Figure 3 we display the behaviour of the strategies for some fixed signal values ($f_t \in \{-1, 0, 1\}$) corresponding to negative, zero and positive future price returns of the risky asset. For the linear strategy, at any fixed slice, holdings depend on the accumulated distortion $\boldsymbol{J}_t$ in a linear fashion. The network, however, shows a nonlinear dependence on $\boldsymbol{J}_t$ leading to larger positions than the linear strategies when the accumulated distortion is large in absolute magnitude. A similar observation is made in Brokmann et al. (2024); however, in their model with instantaneous impact decay, the same relation appears as a feedback function of the current holdings, rather then a moving average of the current and past trades. With gradual impact decay, Figure 4 shows a scatter plot of the trades dQ_t against the current holdings Q_t. Unlike in Figure 3, there is no discernible relationship between the two quantities either for the linear or for the network strategies. This highlights the significance of closely monitoring the accumulated "volume impact" $\boldsymbol{J}_t$ and demonstrates a key difference between settings with transient and instantaneous impact decay.

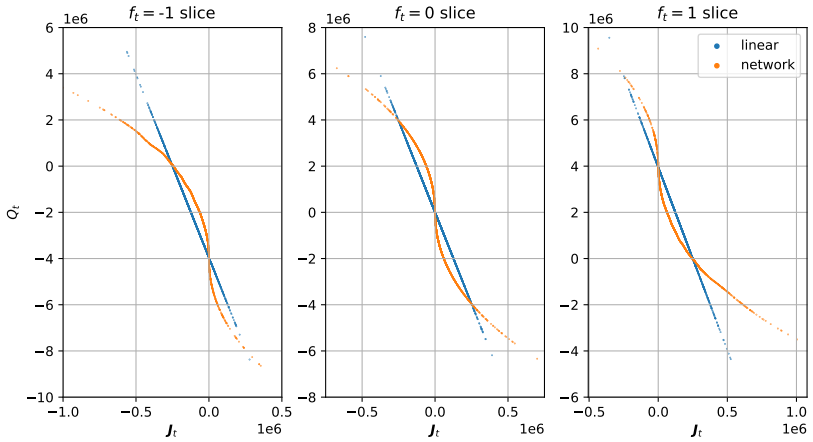


Figure 3: Feedback strategy slices at representative levels of the trading signal f_t.

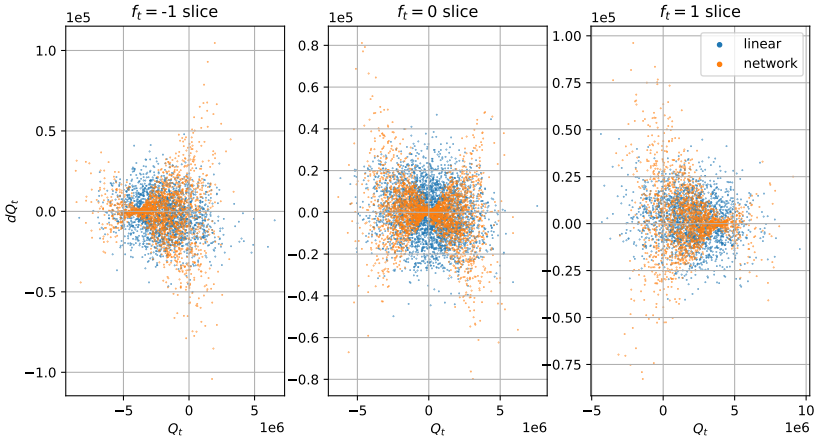


Figure 4: Scatter plot of trades dQ_t against current position Q_t at different signal values f_t.

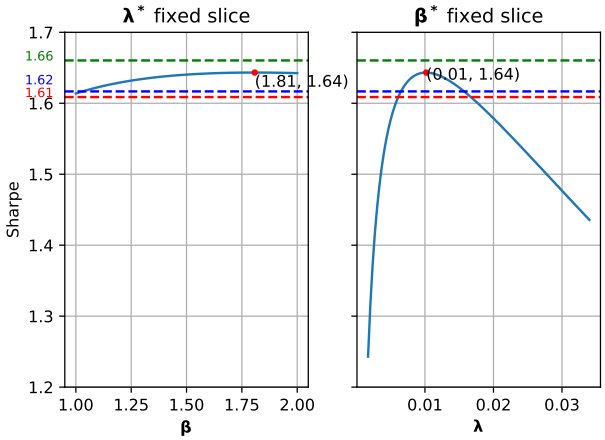


Figure 5: Performance of linear policies for three exponential decay timescales at DR = 1.3M when one of the effective parameters $\boldsymbol{\beta}$,$\boldsymbol{\lambda}$ is fixed to its optimal value and the other varies. The dashed lines indicate the performances of the optimized NN strategy (green), the NN strategy trained for one exponential timescale (blue), and the best linear strategy for one exponential timescale (red).

6.3 Multiple Decay Timescales

So far, we have compared the performance of linear and neural network strategies in a model where square-root impact decays at a single exponential timescale. However, empirical data suggests a more complex multiscale behavior, where some fraction of the price impact dissipates very quickly while another fraction remains visible for a surprisingly long time.

To take this into account while retaining the Markovian structure of the model, one can consider a combination of several exponential decay kernels as in Hey et al. (2023). The price impact of each order then depends on several exponential moving averages with different smoothing parameters, that in turn appear as additional state variables for the optimal trading problem. Figure 5 compares the performance of the optimal neural network strategy that learns the dependency on all these variables to a number of competitors: (i) the linear strategy optimized for the one-exponential model of Section 6.2, (ii) the neural network strategy optimized for the one-exponential model of Section 6.2, and (iii) a linear strategy which still only uses a single decay parameter but now tunes it to the multiscale impact decay as in Corollary 5.4.

A first observation is that when the strategies optimized for the simpler one-timescale model are applied out-of-model, the performance gap between them narrows substantially. This suggests that the simplicity of the linear strategies offers additional robustness. Second, we see that the performance of both strategies is markedly different from the single-exponential model, but rather close to the performances of the strategies that are optimized for the multiscale decay. This suggests that the decay kernel crucially determines the performance of trading strategies, which can be determined by simulation without the need for numerical optimization. In contrast, the additional gains that can be achieved by optimizing for the multiple timescales are relatively modest, in particular, when recalling that these figures apply in the simulated model but may well be reduced out-of-model just like when passing from one to three exponential decay kernels.

A more parsimonious approach to capture multiscale impact decay is to replace the exponential decay kernel by a (shifted) power law. This substantially reduces the number of model parameters, but makes the model fully path dependent. Optimizing neural network strategies becomes much more challenging, as there no longer is a natural finite set of state variables in this context, which is beyond our scope here. In contrast, using the explicit formulas from Corollary 5.4, the optimal linear strategies can readily be determined in the same straightforward manner as before, see Figure 6. Encouragingly, we see that the optimal linear strategy does not change too much when replacing the three-exponential kernel by a roughly similar power law. In contrast, the performance of the strategies is much more sensitive to the precise specification of the decay kernel.[13]

[13]This parallels results for option hedging with various frictions, where the Black-Scholes delta hedge also turns out to be

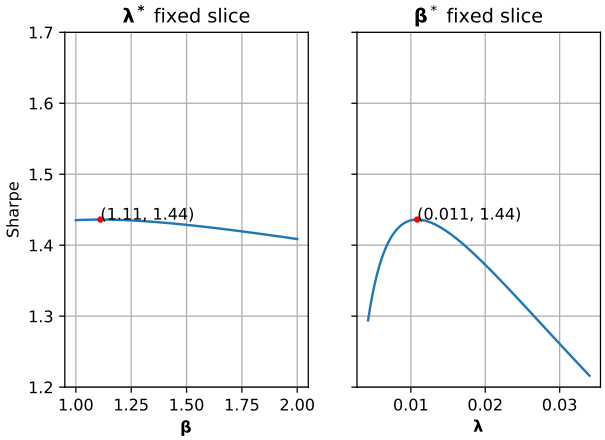


Figure 6: Performance of linear policies for power-law decay at DR = 1.3M when one of the effective parameters $\boldsymbol{\beta}$,$\boldsymbol{\lambda}$ is fixed to its optimal value and the other varies.

7 Conclusion

In this paper, we show that the performance of linear feedback controls remains tractable even in price impact models that account for the nonlinearities observed empirically. When the parameters of the corresponding "effective" linear model are tuned appropriately, the resulting strategies turn out to be competitive with modern machine learning methods, where the optimal feedback controls are approximated using neural networks. In particular, the already modest performance gap is reduced even further if the strategies' performances are tested outside the model on which they were trained. This suggests that while it is crucial to accurately measure the nonlinear features of price impact, taking these into account for optimal trading can be done in a straightforward and efficient manner by focusing on a simple parametric class of linear strategies.

A Deriving The Goal Functional

We first state a very mild technical assumption on the price impact functions h^n and decay kernels K_n, which is needed for the proof of Proposition 3.1:

Assumption A.1. *For each $n = 1, \dots, N$ we assume that:*

(i) *The price impact function $h^n : \mathbb{R} \to \mathbb{R}$ is continuous;*

(ii) *The decay kernel $K_n : [0, \infty) \to (0, \infty)$ is twice continuously differentiable and nonincreasing.*

Proof of Proposition 3.1. For a smooth strategy $dQ_t = \dot{Q}_t dt$ the investor's cash balance C_T from trading on $[0, T]$ is

$$C_T = -\int_0^T P_t dQ_t = -\int_0^T S_t dQ_t - \sum_{n=1}^N \int_0^T h^n(J_t^n) dQ_t. \tag{A.1}$$

We now rewrite both integrals using integration by parts. For the first one, we have

$$\int_0^T S_t dQ_t = S_T Q_T - \int_0^T Q_t dS_t. \tag{A.2}$$

For the second one, we recall the kernel in the definition of J^n in (2.3). We integrate by parts to see that for any $0 \le u \le t < \infty$,

$$K_n(t-u)Q_u = \int_0^u K_n(t-s) dQ_s - \int_0^u K_n'(t-s) Q_s ds.$$

surprisingly robust and often close to optimal, but the corresponding hedging error is typically far from zero and crucially depends on the actual market dynamics (Denkl et al., 2013).

Evaluating at $u = t$ yields

$$J^n(t) = K_n(0)Q_t + \int_0^t K_n'(t-s)Q_s ds. \tag{A.3}$$

Isolating for Q_t and writing $A_t^n = \int_0^t K_n'(t-s)Q_s ds$ we obtain that

$$dQ_t = \frac{1}{K_n(0)}\left(dJ_t^n - dA_t^n\right).$$

As a consequence, Itô's formula gives

$$\begin{aligned}\int_0^T h^n(J_t^n)dQ_t &= \frac{1}{K_n(0)}\left(\int_0^T h^n(J_t^n)dJ_t^n - \int_0^T h_n(J_t^n)dA_t^n\right)\\ &= \frac{H^n(J_T^n) - H^n(J_0^n)}{K_n(0)} - \frac{1}{K_n(0)}\int_0^T h^n(J_t^n)dA_t^n.\end{aligned} \tag{A.4}$$

Now plugging (A.2) and (A.4) into (A.1) yields

$$\begin{aligned}C_T &= -S_T Q_T - \sum_{n=1}^N \frac{H^n(J_T^n) - H^n(J_0^n)}{K_n(0)} + \int_0^T Q_t dS_t\\ &\quad + \sum_{n=1}^N \frac{1}{K_n(0)}\int_0^T h^n(J_t^n)dA_t^n\\ &= -S_T Q_T - \sum_{n=1}^N \frac{H^n(J_T^n) - H^n(J_0^n)}{K_n(0)} + \int_0^T Q_t dS_t\\ &\quad + \sum_{n=1}^N \frac{1}{K_n(0)}\int_0^T \left(h^n(J_t^n)K_n'(0)Q_t + h^n(J_t^n)\int_0^t K_n''(t-s)Q_s ds\right)dt,\end{aligned}$$

where in the final equality we used the identity $\frac{dA_t^n}{dt} = K_n'(0)Q_t + \int_0^t K_n''(t-s)Q_s ds$. As such, the investor's PnL is given by

$$\begin{aligned}\mathrm{PnL}_T &= C_T + S_T Q_T\\ &= \int_0^T Q_t dS_t - \sum_{n=1}^N \frac{H^n(J_T^n) - H^n(J_0^n)}{K_n(0)}\\ &\quad + \sum_{n=1}^N \frac{1}{K_n(0)}\int_0^T \left(h^n(J_t^n)K_n'(0)Q_t + h^n(J_t^n)\int_0^t K_n''(t-s)Q_s ds\right)dt,\end{aligned}$$

which completes the proof. □

B Proofs for Linear Strategies

Proof Lemma 4.1. Plugging the linear impact function and exponential kernel into (4.1) and computing the integral leads to the expression

$$\mathcal{J}(Q) = \lim_{T\to\infty}\mathbb{E}\left[\frac{1}{T}\int_0^T\left(\alpha Q_t f_t - \lambda\beta J_t\left(Q_t - \beta\int_0^t e^{-\beta(t-s)}Q_s ds\right) - \frac{\gamma\sigma^2}{2}Q_t^2\right)dt\right]. \tag{B.1}$$

We note $J_t = Q_t - \beta\int_0^t e^{-\beta(t-s)}Q_s ds$ (see Equation (A.3)) so that the middle term in the right hand side of (B.1) is simply $\lambda\beta J_t^2$. To put this into the form (4.1) we again use integration by parts to write

$$(Q_T - J_T)^2 = 2\beta\int_0^T (J_t^2 - J_t Q_t)dt.$$

Hence taking expectation, dividing by T and sending $T\to\infty$ we see that under the mild trasversality condition $\lim_{T\to\infty}\frac{1}{T}\mathbb{E}[(Q_T - J_T)^2] = 0$ (which we verify for linear strategies at the end of this appendix) we have the identity $\lim_{T\to\infty}\frac{1}{T}\int_0^T \mathbb{E}[J_t^2]dt = \lim_{T\to\infty}\frac{1}{T}\int_0^T \mathbb{E}[J_t Q_t]dt$. Making these substitutions in (B.1) and recalling that $I_t = \lambda J_t$ gives precisely (4.1). □

Proof of Theorem 4.2; Optimal Strategy. To derive the optimality of (4.2), we first make a change of variables to put the problem at hand into a standard stochastic control framework. To this end we define $J_t^0 = J_t - Q_t$ and note that

$$dJ_t^0 = -\beta J_t dt = (-\beta J_t^0 + \beta Q_t)dt. \tag{B.2}$$

Plugging this in for J in the objective function (4.1), we obtain

$$\mathcal{J}(x) = \lim_{T\to\infty} \mathbb{E}\left[\frac{1}{T}\int_0^T \left(\alpha Q_t f_t - \lambda\beta J_t^0 Q_t - \lambda\beta Q_t^2 - \frac{\gamma\sigma^2}{2}Q_t^2\right)dt\right].$$

This is now a standard ergodic stochastic control problem with one uncontrolled state variable f and one controlled state variable J^0. As such, the HJB equation for the corresponding value function $V(J^0, f)$ is

$$0 = \sup_Q \left\{(\alpha f - \lambda\beta J^0)Q - \left(\lambda\beta + \frac{\gamma\sigma^2}{2}\right)Q^2 - \beta(J^0 + Q)\partial_{J^0}V(J^0, f) \right.$$
$$\left. -\phi f \partial_f V(J^0, f) + \frac{\eta^2}{2}\partial_{ff}V(J^0, f)\right\}.$$

We now make a quadratic ansatz for V,

$$V(J^0, f) = -\frac{a}{2}(J^0)^2 + bJ^0 f + \mathcal{F}(f)$$

for some constants a and b to be determined. The functional $\mathcal{F}(\cdot)$ will be determined by an exogenous ODE but does not impact the optimal strategy so we do not pursue this further here. Plugging in and maximizing over Q yields

$$Q^* = \frac{\alpha - b\beta}{2\beta\lambda + \gamma\sigma^2}f - \frac{\lambda\beta - a\beta}{2\beta\lambda + \gamma\sigma^2}J^0. \tag{B.3}$$

Now we plug this back into the HJB equation and collect terms. Comparison of coefficients for the terms proportional to $(J^0)^2$ and $J^0 f$ leads to the following two equations for the two unknown parameters a and b:

$$a^2\beta + a\left(2\gamma\sigma^2 + 2\lambda\beta\right) + \lambda^2\beta = 0,$$
$$b\left(a\beta^2 + \lambda\beta^2 + \phi\left(\gamma\sigma^2 + 2\lambda\beta\right) + \gamma\beta\sigma^2\right) - a\alpha\beta + \alpha\lambda\beta = 0.$$

The first equation involves only a and is quadratic in it, while the second equation is linear in b. Hence solving the equation for a (and taking the positive solution) and then solving for b yields

$$a = \frac{-\beta\lambda - \gamma\sigma^2 + \sqrt{\gamma\sigma^2(2\beta\lambda + \gamma\sigma^2)}}{\beta},$$
$$b = \frac{\alpha\left(\sqrt{\gamma\sigma^2\left(\gamma\sigma^2 + 2\beta\lambda\right)} - \left(\gamma\sigma^2 + 2\beta\lambda\right)\right)}{\gamma\sigma^2\phi + 2\beta\lambda\phi + \sqrt{\gamma\beta^2\sigma^2\left(\gamma\sigma^2 + 2\beta\lambda\right)}}.$$

Inserting these expressions into (B.3) in turn pins down the optimal feedback trading strategy:

$$Q^* = \frac{\alpha(\beta + \phi)}{\gamma\sigma^2\phi + 2\beta\lambda\phi + \sqrt{\gamma\beta^2\sigma^2\left(\gamma\sigma^2 + 2\beta\lambda\right)}}f - \frac{\gamma\sigma^2 + 2\beta\lambda - \sqrt{\gamma\sigma^2\left(\gamma\sigma^2 + 2\beta\lambda\right)}}{\gamma\sigma^2 + 2\beta\lambda}J^0.$$

Recalling that $J^0 = J - Q^*$, isolating for Q^* and simplifying now finally yields the expression (4.2) for the optimal feedback strategy with coefficients (4.3). □

To complete the proof of Theorem 4.2 it now remains to establish formula (4.4) for the risk of the strategy. To accomplish this we first establish Lemma B.2 below, which is also used in the proof of Theorem 5.1, which follows. The elements of the lemma concerning the nonlinear impact h^n and decay kernel K_n require the following mild assumption.

Assumption B.1. *For all $\kappa > 0$ and every $n = 1, \ldots, N$ we have $\int_{-\infty}^{\infty} xh^n(x)e^{-\kappa x^2}dx < \infty$ and $\int_0^\infty K_n''(t)e^{-\kappa t}dt < \infty$.*

We now state the lemma and its proof, using the notation from Theorem 5.1 throughout.

Lemma B.2. *Consider the linear policy $Q_t = C_f f_t - C_J \boldsymbol{J}_t$ with coefficients from* (4.3)*. Then there exists $\kappa > 0$ such that:*

(i) $\{(f_t, \boldsymbol{J}_t)\}_{t\geq 0}$ *is a bivariate Gaussian process satisfying, as* $t \to \infty$,

$$\mathbb{E}[f_t^2] = 1 + o(e^{-\kappa t}), \tag{B.4}$$

$$\mathbb{E}[f_t \boldsymbol{J}_t] = \frac{\phi C_f}{\boldsymbol{\beta} + \phi(1+C_J)} + o(e^{-\kappa t}), \tag{B.5}$$

$$\mathbb{E}[\boldsymbol{J}_t^2] = \frac{\phi C_f^2}{(1+C_J)(\boldsymbol{\beta} + \phi(1+C_J))} + o(e^{-\kappa t}); \tag{B.6}$$

(ii) $\{Q_t\}_{t\geq 0}$ *is a Gaussian process satisfying*

$$\mathbb{E}[Q_t Q_s] = \rho(|t-s|) + o(e^{-\kappa \max\{t,s\}}); \tag{B.7}$$

(iii) $\{(Q_t, J_t^n)\}_{t\geq 0}$ *is a bivariate Gaussian process satisfying for* $0 \leq t \leq T < \infty$,

$$\mathbb{E}[J_T^n Q_t] = K(0)\rho(T-t) + \int_0^T K'(T-s)\rho(|t-s|)ds + o(e^{-\kappa T}), \tag{B.8}$$

$$\mathbb{E}[(J_T^n)^2] = K^2(0)\rho(0) + 2K(0)\int_0^T K'(t)\rho(t)dt + \int_0^T \int_0^T K'(t)K'(s)\rho(|t-s|)dsdt + o(e^{-\kappa T}); \tag{B.9}$$

(iv) J_T^n *converges in distribution as* $T \to \infty$ *to a centered Gaussian random variable with variance* σ_n^2.

Proof. Throughout we assume for simplicity and without loss of generality that that $f_0 = \boldsymbol{J}_0 = J_0^n = 0$. We also note that if $g(t) = o(e^{-\kappa_1 t})$ then $e^{-\kappa_2 t}\int_0^t g(s)ds = o(e^{-\kappa_3 t})$ for any $\kappa_1, \kappa_2 > 0$ and some $\kappa_3 > 0$. We don't keep track of this constant and just generically write κ in the computations that follow.

Proof of (i) Using the given form of Q we see that

$$d\boldsymbol{J}_t = -\boldsymbol{\beta}\boldsymbol{J}_t dt + C_f df_t - C_J d\boldsymbol{J}_t \implies d\boldsymbol{J}_t = -\frac{\boldsymbol{\beta}}{1+C_J}\boldsymbol{J}_t dt + \frac{C_f}{1+C_J} df_t. \tag{B.10}$$

From here it is clear that $\{(f_t, \boldsymbol{J}_t)\}_{t\geq 0}$ is jointly Gaussian, which can be most easily seen using the fact that f is an Ornstein-Uhlenbeck process and that $\boldsymbol{J}_t = \frac{C_f}{1+C_J}\int_0^t e^{-\frac{\boldsymbol{\beta}}{1+C_J}(t-s)} df_s$.

Equation (B.4) follows immediately from well-known properties of the Ornstein-Uhlenebck process process. To obtain (B.5) we integrate by parts,

$$\begin{aligned} d(f_t\boldsymbol{J}_t) &= f_t d\boldsymbol{J}_t + \boldsymbol{J}_t df_t + d[\boldsymbol{J}, f]_t \\ &= \left(-\frac{\boldsymbol{\beta} + \phi(1+C_J)}{1+C_J} f_t \boldsymbol{J}_t - \frac{\phi C_f}{1+C_J} f_t^2 + \frac{2\phi C_f}{1+C_J}\right) dt + \left(\frac{\sqrt{2\phi}C_f}{1+C_J} f_t + \sqrt{2\phi}\boldsymbol{J}_t\right) dW_t. \end{aligned}$$

Taking expectation we obtain the ODE

$$\frac{\partial}{\partial t}\mathbb{E}[f_t\boldsymbol{J}_t] = -\frac{\boldsymbol{\beta} + \phi(1+C_J)}{1+C_J}\mathbb{E}[f_t\boldsymbol{J}_t] - \frac{\phi C_f}{1+C_J}\mathbb{E}[f_t^2] + \frac{2\phi C_f}{1+C_J},$$

which has solution

$$\mathbb{E}[f_t\boldsymbol{J}_t] = \int_0^t \exp\left(-\frac{\boldsymbol{\beta} + \phi(1+C_J)}{1+C_J}(t-s)\right)\left(\frac{2\phi C_f}{1+C_J} - \frac{\phi C_f}{1+C_J}\mathbb{E}[f_s^2]\right) ds.$$

Using (B.4) to replace $\mathbb{E}[f_s^2]$ and integrating yields (B.5). For (B.6) we proceed similarly by computing

$$d\boldsymbol{J}_t^2 = \left(-\frac{2\boldsymbol{\beta}}{1+C_J}\boldsymbol{J}_t^2 - \frac{2\phi C_f}{1+C_J} f_t \boldsymbol{J}_t + \frac{2\phi C_f^2}{(1+C_J)^2}\right) dt + \frac{2\sqrt{2\phi}C_f}{1+C_J} dW_t.$$

Taking expectation yields the ODE

$$\frac{\partial}{\partial t}\mathbb{E}[\boldsymbol{J}_t^2] = -\frac{2\boldsymbol{\beta}}{1+C_J}\mathbb{E}[\boldsymbol{J}_t^2] - \frac{2\phi C_f}{1+C_J}\mathbb{E}[f_t\boldsymbol{J}_t] + \frac{2\phi C_f^2}{(1+C_J)^2},$$

which has solution

$$\mathbb{E}[\boldsymbol{J}_t^2] = \int_0^t \exp\left(-\frac{2\boldsymbol{\beta}}{1+C_J}(t-s)\right)\left(\frac{2\phi C_f^2}{(1+C_J)^2} - \frac{2\phi C_f}{1+C_J}\mathbb{E}[f_s\boldsymbol{J}_s]\right) ds.$$

Now using (B.5) to replace $\mathbb{E}[f_s\boldsymbol{J}_s]$ and integrating yields (B.6).

Proof of (ii) The Gaussianity of Q follows immediately from that of f and $\boldsymbol{J}$. To prove (B.7) we assume without loss of generality that $s \leq t$ and note that

$$\mathbb{E}[Q_t Q_s] = C_f^2 \mathbb{E}[f_t f_s] - C_f C_J \mathbb{E}[f_t \boldsymbol{J}_s] - C_f C_J \mathbb{E}[f_s \boldsymbol{J}_t] + C_J^2 \mathbb{E}[\boldsymbol{J}_t \boldsymbol{J}_s]. \tag{B.11}$$

For the OU process, we have

$$\mathbb{E}[f_t | \mathcal{F}_s] = f_s e^{-\phi(t-s)}. \tag{B.12}$$

Similarly, using the same approach as in the proof of part (i) above, we can derive an ODE for $\mathbb{E}[\boldsymbol{J}_t | \mathcal{F}_s]$, which leads to the expression

$$\begin{aligned}
\mathbb{E}[\boldsymbol{J}_t | \mathcal{F}_s] &= \boldsymbol{J}_s e^{-\frac{\boldsymbol{\beta}}{1+C_J}(t-s)} - \frac{\phi C_f}{1+C_J} f_s \int_s^t e^{-\phi(u-s)} e^{-\frac{\boldsymbol{\beta}}{1+C_J}(t-u)} du \\
&= \boldsymbol{J}_s e^{-\frac{\boldsymbol{\beta}}{1+C_J}(t-s)} + \frac{\phi C_f}{1+C_J} f_s \frac{e^{-\phi(t-s)} - e^{-\frac{\boldsymbol{\beta}}{1+C_J}(t-s)}}{\phi - \frac{\boldsymbol{\beta}}{1+C_J}}.
\end{aligned} \tag{B.13}$$

Hence, we (a) use the tower property in (B.11) to condition on $\mathcal{F}_s$, (b) sub in the expressions (B.12), (B.13) for the conditional expectations, (c) use the identities (B.4)-(B.6) and collect like terms and finally (d) express C_f and C_J in terms of the model parameters (4.3) to obtain (B.7).

Proof of ((iii)) The joint Gaussianity of Q and J^n is immediate from the fact that Q is Gaussian and the integral expression (2.3) for J^n. By taking the differential of (2.3), as we did in the proof of Proposition 3.1, we recall that J^n has the representation (A.3). Hence

$$\begin{aligned}
\mathbb{E}[J_T^n Q_t] &= K_n(0)\mathbb{E}[Q_T Q_t] + \int_0^T K_n'(T-s)\mathbb{E}[Q_t Q_s] ds \\
&= K_n(0)\rho(T-t) + \int_0^T K_n'(T-s)\rho(|t-s|)ds + o(e^{-\kappa T}) + \int_0^T K_n'(T-s) o(e^{-\kappa s}) ds,
\end{aligned}$$

where the second equality followed from (B.7). By changing variables we see that $\int_0^T K_n'(T-s)o(e^{-\kappa s})ds = \int_0^T K_n'(s)o(e^{-\kappa(T-s)})ds = o(e^{-\kappa T})$, since K' is integrable as a consequence of K being nonincreasing. This establishes (B.8).

To prove (B.9) we again use (A.3) to infer that

$$\mathbb{E}[(J_T^n)^2] = K_n^2(0)\mathbb{E}[Q_T^2] + 2K_n(0)\int_0^T K_n'(T-s)\mathbb{E}[Q_T Q_s]ds + \int_0^T \int_0^T K_n'(T-t)K_n'(T-s)\mathbb{E}[Q_t Q_s] ds dt.$$

Using (B.7) and arguing a similar manner as in the proof of (B.8), we obtain that

$$\begin{aligned}
\mathbb{E}[(J_T^n)^2] &= K_n^2(0)\rho(0) + 2K_n(0)\int_0^T K_n'(T-s)\rho(T-s)ds \\
&\quad + \int_0^T \int_0^T K_n'(T-t)K_n'(T-s)\rho(|t-s|)ds dt + o(e^{-\kappa T}) \\
&= K_n^2(0)\rho(0) + 2K_n(0)\int_0^T K_n'(t)\rho(t)dt + \int_0^T \int_0^T K_n'(t)K_n'(s)\rho(|t-s|)ds dt + o(e^{-\kappa T}),
\end{aligned}$$

where the final equality follows by a change of variables. This proves (B.9).

Proof of (iv) Since J^n is a Gaussian process we just have to show that $\lim_{T\to\infty}\mathbb{E}[J_T^n] = 0$ and that $\lim_{T\to\infty}\mathbb{E}[(J_T^n)^2] = \sigma_n^2$. The latter is immediate from (B.9). The former is also straightforward. Indeed, $\mathbb{E}[f_T] = 0$ for every T and from the dynamics (B.10) it is straightforward to establish that the same is true for $\boldsymbol{J}$. Hence $\mathbb{E}[Q_T] = 0$ for every T, from which it follows that $\mathbb{E}[J_T^n] = 0$ for every T using (A.3). □

Proof of Theorem 4.2; Equation 4.4. Using L'Hôpital's rule we have that

$$\lim_{T\to\infty} \frac{1}{T}\int_0^T \mathbb{E}[Q_t^2]dt = \lim_{T\to\infty}\mathbb{E}[Q_T^2].$$

From Lemma B.2(ii) we have that $\lim_{T\to\infty}\mathbb{E}[Q_T^2] = \rho(0) = C_1 - C_2$. Simplifying this expression and multiplying by σ^2 gives the formula (4.4) for $\mathcal{R}^2(x)$. □

Proof of Theorem 5.1. We start by computing the return term $\lim_{T\to\infty}\frac{1}{T}\int_0^T \alpha\mathbb{E}[Q_t f_t]\,dt$. By L'Hôpital's rule this limit is given by $\alpha\lim_{T\to\infty}\mathbb{E}[Q_T f_T]$. Using Lemma B.2(i) we obtain that this can be computed as

$$\lim_{T\to\infty}\alpha\mathbb{E}[f_T Q_T] = \alpha C_f \lim_{T\to\infty}\mathbb{E}[f_T^2] - \alpha C_J \lim_{T\to\infty}\mathbb{E}[f_T \boldsymbol{J}_T] = \alpha C_f - \frac{\alpha\phi C_f C_J}{\boldsymbol{\beta}+\phi(1+C_J)}. \tag{B.14}$$

Next we study the impact term and note that again by L'Hôpital's rule,

$$\begin{aligned}&\lim_{T\to\infty}\mathbb{E}\left[\frac{1}{T}\int_0^T\sum_{n=1}^N h^n(J_t^n)\left(\frac{K_n'(0)}{K_n(0)}Q_t + \int_0^t \frac{K_n''(t-s)}{K_n(0)}Q_s ds\right)dt\right]\\ &= \sum_{n=1}^N\left(\frac{K_n'(0)}{K_n(0)}\lim_{T\to\infty}\mathbb{E}[h^n(J_T^n)Q_T] + \lim_{T\to\infty}\int_0^T\frac{K_n''(T-t)}{K_n(0)}\mathbb{E}[h^n(J_T^n)Q_t]dt\right)\end{aligned} \tag{B.15}$$

Since J_T^n and Q_t are jointly Gaussian random variables, they are independent if and only if uncorrelated. Hence:

$$\mathbb{E}[h^n(J_T^n)Q_t] = \mathbb{E}\left[h^n(J_T^n)\left(Q_t - \frac{\mathbb{E}[J_T^n Q_t]}{\mathbb{E}[(J_T^n)^2]}J_T^n\right)\right] + \frac{\mathbb{E}[J_T^n Q_t]}{\mathbb{E}[(J_T^n)^2]}\mathbb{E}[h^n(J_T^n)J_T^n] = \frac{\mathbb{E}[J_T^n Q_t]}{\mathbb{E}[(J_T^n)^2]}\mathbb{E}[h^n(J_T^n)J_T^n]. \tag{B.16}$$

Lemma B.2(iii) and (iv) yield

$$\lim_{T\to\infty}\mathbb{E}[J_T^n Q_T] = \tau_n, \quad \lim_{T\to\infty}\mathbb{E}[(J_T^n)^2] = \sigma_n^2, \quad \lim_{T\to\infty}\mathbb{E}[h^n(J_T^n)J_T^n] = \mathbb{E}[h^n(\sigma_n Z)\sigma_n Z], \tag{B.17}$$

where Z is a standard normal random variable.

Hence, we can sub (B.16) into (B.15) and use the identities in (B.17) to obtain that (B.15) is given by

$$\sum_{n=1}^N\left(\frac{K_n'(0)}{K_n(0)}\frac{\tau_n}{\sigma_n}\mathbb{E}[h^n(\sigma_n Z)Z] + \frac{\mathbb{E}[h^n(\sigma_n Z)Z]}{\sigma_n K_n(0)}\lim_{T\to\infty}\int_0^T K_n''(T-t)\mathbb{E}[J_T^n Q_t]dt\right). \tag{B.18}$$

It therefore just remains to compute

$$\lim_{T\to\infty}\int_0^T K_n''(T-t)\mathbb{E}[J_T^n Q_t]dt.$$

To compute the final limit we note by (B.8) that $\mathbb{E}[J_T^n Q_t] = K_n(0)\rho(T-t) + \int_0^T K_n'(T-s)\rho(|t-s|)ds + o(e^{-\kappa T})$. From the integrability condition on K'' in Assumption B.1 it follows that the $o(e^{-\kappa T})$ term does not contribute in the limit yielding

$$\begin{aligned}&\lim_{T\to\infty}\int_0^T K_n''(T-t)\mathbb{E}[J_T^n Q_t]dt\\ &\quad= K_n(0)\lim_{T\to\infty}\int_0^T K_n''(T-t)\rho(T-t)dt + \int_0^T\int_0^T K_n''(T-t)K_n'(T-s)\rho(|t-s|)dsdt\\ &\quad= K_n(0)\int_0^\infty K_n''(t)\rho(t)dt. + \int_0^\infty\int_0^\infty K_n''(t)K_n'(s)\rho(|t-s|)dsdt.\end{aligned} \tag{B.19}$$

Hence using (B.14), (B.18) and (B.19) and recalling the definitions of C_f, C_J given by (4.3) we obtain (5.1). Corollary 5.3 now follows by explicitly computing the integrals, which only involve exponential functions, and inserting the explicit formula for the absolute moment of the standard normal random variable into $\mathbb{E}[h^n(\sigma_n Z)Z] = |\sigma_n|^{p_n}\mathbb{E}[|Z|^{p_n+1}]$. Corollary 5.4 follows similarly by known integral representations that involve the incomplete Gamma and hypergeometric functions. □

We finish off this Appendix by showing that linear strategies satisfy the transversality conditions mentioned in Footnote 10 and the proof of Lemma 4.1. From Lemma B.2(iv) it follows that $\mathbb{E}[H^n(J_T^n)] \to \mathbb{E}[H^n(\sigma_n Z)] < \infty$. This establishes the transversality condition $\lim_{T\to\infty}\frac{1}{T}\mathbb{E}[H^n(J_T^n)] = 0$. Analogously we have $\mathbb{E}[\langle\int_0^\cdot Q_t dS_t\rangle_T] = \sigma^2\int_0^T\mathbb{E}[Q_t^2]dt$, which is finite by Lemma B.2(ii). Hence, the local martingale part of $\int_0^T Q_t dS_t$, which is $\sigma\int_0^T Q_t dW_t$, is a true martingale. Finally, we note that $\mathbb{E}[(Q_T - \boldsymbol{J}_T)^2] = \mathbb{E}[Q_T^2] - 2\mathbb{E}[Q_T\boldsymbol{J}_T] + \mathbb{E}[\boldsymbol{J}_T^2]$ which converges to a finite value as $T\to\infty$ courtesy of Lemma 4.1(i). As such, $\lim_{T\to\infty}\frac{1}{T}\mathbb{E}[(Q_T - \boldsymbol{J}_T)^2] = 0$ establishing the transversality condition used in the proof of Lemma 4.1.

C Numerical Methods

In this appendix, we provide a detailed description of the numerical method we use to benchmark the performance of the linear policies: a machine learning-based approach where the feedback controls are parametrized by neural networks. To simplify the exposition, we focus on optimizing the objective function (3.3) under the assumption of a single exponential decay timescale:[14]

$$\begin{aligned}\mathcal{J}(Q) &= \lim_{T\to\infty} \mathbb{E}\left[\frac{1}{T}\int_0^T \left(\alpha Q_t f_t - \lambda\beta \text{sign}(J_t)|J_t|^p \left(Q_t - \beta\int_0^t e^{-\beta(t-s)}Q_s ds\right) - \frac{\gamma\sigma^2}{2}Q_t^2\right) dt\right] \\ &= \lim_{T\to\infty} \mathbb{E}\left[\frac{1}{T}\int_0^T \left(\alpha Q_t f_t - \lambda\beta |J_t|^{p+1} - \frac{\gamma\sigma^2}{2}Q_t^2\right) dt\right], \end{aligned} \tag{C.1}$$

where the last equality follows from (A.3). To optmize this numerically, we first derive a discretized version of the problem. Subsequently, we discuss the machine learning algorithm used to optimize it.

C.1 Discrete Setting

Discrete objective function To deploy numerical methods, we consider discrete strategies with a fixed trading frequency Δt. Each holding sequence $\{Q_n\}$ leads to a right-continuous holding process Q_t, where for every $n = 0, 1, \ldots$

$$\begin{aligned} t_n &= n\Delta t, \\ Q_t &= Q_n \text{ for } t \in (t_{n-1}, t_n]. \end{aligned} \tag{C.2}$$

With the exponential decay kernel $K(t) = e^{-\beta t}$ we then have

$$J_t = e^{-\beta(t-t_{n-1})}(J_{t_{n-1}} + Q_n - Q_{n-1}) \text{ for } t \in (t_{n-1}, t_n]. \tag{C.3}$$

In particular, we consider the sequence $\{J_n\}$ where

$$J_n = J_{t_n} = e^{-\beta\Delta t}(J_{n-1} + Q_n - Q_{n-1}). \tag{C.4}$$

The next step is to express the objective function (C.1) in terms of the sequences $\{Q_n\}$ and $\{J_n\}$. The expected asset return can be calculated as follows:

$$\begin{aligned}\int_0^{N\Delta t} Q_t dS_t &= \sum_{n=1}^N \mathbb{E}\left[\int_{t_{n-1}}^{t_n} Q_t dS_t | f_{t_{n-1}}\right] \\ &= \sum_{n=1}^N \int_{t_{n-1}}^{t_n} Q_t \alpha \mathbb{E}\big[f_t | f_{t_{n-1}}\big] dt \\ &= \sum_{n=1}^N Q_n \alpha \int_{t_{n-1}}^{t_n} e^{-\beta(t-t_{n-1})} f_{t_{n-1}} dt \\ &= \sum_{n=1}^N \alpha \frac{(1-e^{-\beta\Delta t})}{\beta} Q_n f_{n-1}. \end{aligned} \tag{C.5}$$

In the last equation, we denote $f_n = f_{t_n}$, thus simplifying the expression to handle the signal sequence $\{f_n\}$, which, recalling (2.2), is governed by the dynamics

$$f_n = e^{-\phi\Delta t} f_{n-1} + \sqrt{1 - e^{-2\phi\Delta t}} Z_n \text{ where } Z_n \sim N(0,1). \tag{C.6}$$

Regarding the impact term, we have $J_t = e^{-\beta(t_n - t)} J_{t_n}$ for $t \in (t_{n-1}, t_n]$. Hence,

$$\begin{aligned}\int_0^T \beta\lambda |J_t|^{p+1} dt &= \sum_{n=1}^N \mathbb{E}\left[\int_{t_{n-1}}^{t_n} \beta\lambda |e^{-\beta(t_n - t)} J_{t_n}|^{p+1} dt\right] \\ &= \sum_{n=1}^N \beta\lambda \frac{(1 - e^{-\beta(p+1)p\Delta t})}{\beta(p+1)} |J_{t_n}|^{p+1}. \end{aligned} \tag{C.7}$$

[14] We have also implemented a similar algorithm for multiple exponential decay timescales to obtain the results reported in Section 6.3. An extension to a fully path-dependent power law kernel is more challenging and not pursued here, as the results from Section 6.3 suggest that the additional performance gains that can be achieved in this manner are rather modest.

Lastly, the risk term is given by:

$$\int_0^T \frac{\gamma\sigma^2}{2} Q_t^2 dt = \sum_{n=1}^N \frac{\gamma\sigma^2}{2} Q_n^2 \Delta t. \tag{C.8}$$

In summary, the goal functional becomes

$$\begin{aligned}\mathcal{J}(Q) &= \lim_{N\to\infty} \frac{1}{N\Delta t} \sum_{n=1}^N \Big\{ \alpha \frac{(1-e^{-\beta\Delta t})}{\beta} Q_n f_{n-1} - \beta\lambda \frac{(1-e^{-\beta(p+1)p\Delta t})}{\beta(p+1)} |J_{t_n}|^{p+1} - \frac{\gamma\sigma^2}{2} Q_n^2 \Delta t \Big\} \\ &= \lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^N \Big\{ \hat{\alpha} Q_n f_{n-1} - \beta\hat{\lambda} |J_{t_n}|^{p+1} - \frac{\gamma\sigma^2}{2} Q_n^2 \Big\}, \end{aligned} \tag{C.9}$$

where

$$\hat{\alpha} = \alpha \frac{(1-e^{-\beta\Delta t})}{\beta\Delta t}, \qquad \hat{\lambda} = \lambda \frac{(1-e^{-\beta(p+1)p\Delta t})}{\beta(p+1)\Delta t}. \tag{C.10}$$

With large N and small Δt, we obtain the discrete strategy (Q_n) by optimizing the following loss function:

$$\mathcal{J} = \sum_{n=1}^N \Big(\hat{\alpha} Q_n f_{n-1} - \frac{\gamma\sigma^2}{2} Q_n^2 - \hat{\lambda}\beta |J_n^0 + Q_n|^{p+1} \Big). \tag{C.11}$$

where

$$J_n^0 = J_n - Q_n = e^{-\beta\Delta t}(J_{n-1} + Q_n - Q_{n-1}) - Q_n = e^{-\beta\Delta t} J_{n-1}^0 - (1-e^{-\beta\Delta t}) Q_n. \tag{C.12}$$

Evaluating sharpe ratios After obtaining a discrete trading strategy by optimizing the discretized objective (C.11), we aim to evaluate its Sharpe ratio (SR). This allows to compare the trading performance of the discrete and continuous strategies. To obtain accurate results, this step can be run with a long sequence length $M > N$, as only a simple Monte-Carlo simulation but no more optimization is performed here.

In view of (C.5), (C.7), and (C.8), we can express the profit and loss PnL_M and the risk $\mathcal{R}_M$ in terms of the sequence $\{Q_n\}$:

$$\begin{aligned}\mathrm{PnL}_M(\{Q_n\}) &= \int_0^{M\Delta t} Q_t \, dS_t - \int_0^{M\Delta t} \beta\lambda |J_t|^{p+1} \, dt = \sum_{n=1}^M \Big(\hat{\alpha} Q_n f_{n-1} - \hat{\lambda}\beta |J_n|^{p+1} \Big), \\ \mathcal{R}_M(\{Q_n\}) &= \int_0^{M\Delta t} \frac{\gamma\sigma^2}{2} Q_t^2 \, dt = \sqrt{\sum_{n=1}^M \frac{\gamma\sigma^2}{2} Q_n^2 \Delta t}. \end{aligned}$$

The Sharpe ratio for the discrete strategy $\{Q_n\}$ is then evaluated as follows:

$$\mathrm{SR}(\{Q_n\}) = \frac{\mathrm{PnL}_M(\{Q_n\})}{\mathcal{R}_M(\{Q_n\})}. \tag{C.13}$$

C.2 Deep Learning Algorithm

In this section, we describe our machine-learning method for optimizing the objective function (C.11). We employ a method akin to policy gradient methods from reinforcement learning (Sutton and Barto, 2018): we parametrize strategies by deep neural networks and then optimize the neural network parameters using stochastic gradient descent-type algorithms, with gradients computed through algorithmic differentiation by the backpropagation algorithm.

Algorithm overview Let us now describe in more detail the deep learning approach employed here. The dynamic programming principle suggests that the optimal holdings Q_n are a feedback function of the same state variables as for the linear version of the model discussed in Section 4, namely the current signal f_{n-1} and the moving average of past trades J_{n-1}^0.

Motivated by this observation, we parametrize the current holdings Q_n by a neural network NN_θ with input (f_{n-1}, J_{n-1}^0). With the new holdings Q_n chosen, we then update the moving average of

trades J_n^0 according to (C.12). Once the next signal arrives, we in turn compute the new holdings $\mathrm{NN}_\theta(f_n, J_n^0)$. Note that in this iterative procedure the neural network NN_θ is applied repeatedly and the input J_n^0 depends on the previous neural network output. This can be summarized in the following general structure:

$$Q_n^\theta = \mathrm{NN}_\theta(f_{n-1}, J_{n-1}^{0,\theta}), \tag{C.14}$$

$$J_n^{0,\theta} = e^{-\beta\Delta t} J_{n-1}^{0,\theta} - (1 - e^{-\beta\Delta t}) Q_n^\theta \tag{C.15}$$

$$J_n^\theta = Q_n^\theta + J_n^{0,\theta}, \tag{C.16}$$

where NN_θ, parameterized by θ, is the function we aim to learn during the training phase and the θ-superscripts indicate that the holdings and moving averages depend on the parameter θ.

For any given choice of neural network parameters θ, the procedure described in (C.14)–(C.16) thus takes the sequence $\{f_n\}$ as input and outputs $\{Q_n^\theta\}$ as well as $\{J_n^\theta\}$. Inserting these sequences in the discrete objective function (C.11) in turn leads to the following optimization problem:

$$\mathrm{argmax}_\theta \quad \mathbb{E}\left[\sum_{n=1}^{N}\left(\hat{\alpha} Q_n^\theta f_{n-1} - \frac{\gamma\sigma^2}{2}(Q_n^\theta)^2 - \hat{\lambda}\beta|J_n^\theta|^{p+1}\right)\right]. \tag{C.17}$$

We use stochastic gradient-type algorithms to iteratively update the parameter θ and compute an approximately optimal parameter θ^*. These algorithms require estimates of the gradient of (C.17) with respect to the parameters θ, which can be computed efficiently using algorithmic differentiation via the backpropagation algorithm. We refer, for example, to Goodfellow et al. (2016) and Buehler et al. (2019) for standard references that explain these algorithms in detail.

Neural network architecture In our experiments, we tested different choices of the network NN_θ. One straightforward form of NN_θ is an ordinary feed-forward neural network. This is a standard choice that does not incorporate any domain knowledge about the problem. To wit,

$$\mathrm{NN}_\theta(x) = L_{8,1} \circ \sigma \circ L_{32,8} \circ \sigma \circ L_{128,32} \circ \sigma \circ L_{2,128}(x), \tag{C.18}$$

where $L_{d_1,d_2} : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$ are affine functions and $\sigma(r) = \max(r, 0)$ is the ReLU activation function applied componentwise.

Alternatively, motivated by results for models without risk constraints (Hey et al., 2023), one can try to exploit that the optimal impact state $I_t \propto |Q_t + J_t^0|^p$ typically has a far more linear relationship to the state variables than the corresponding optimal position Q_t. This suggests that a ReLU neural network of the form (C.18), for example, can approximate the impact state more efficiently. Equivalently, one can parametrize the corresponding positions by

$$Q_n^\theta = \mathrm{NL}(\mathrm{NN}_\theta(f_{n-1}, J_{n-1}^0)) - J_{n-1}^0, \tag{C.19}$$

where

$$\mathrm{NL}(x) = \begin{cases} \mathrm{sign}(x)|x|^{1/p}, & \text{if } |x| < B, \\ ax + \mathrm{sign}(x)b, & \text{otherwise,} \end{cases} \tag{C.20}$$

where the neural network NN_θ is defined as in (C.18) and the state variable $J^0 = J^{0,\theta}$ is updated according to (C.15). In this formulation, B represents a large threshold beyond which the nonlinearity $\mathrm{NL}(x)$ transitions to a linear function. The parameters a and b are calibrated so that $\mathrm{NL}(x)$ is continuously differentiable at the boundary. This design ensures that within the bound B, the nonlinear relation between impact states and positions is captured; outside this region, $\mathrm{NL}(x)$ behaves linearly to prevent blow-up in the output when the network is applied iteratively.

Finally, we also evaluated a linear discrete strategy parametrized by a single linear layer $\mathrm{NN}_\theta = L_{2,1}$, and compared its performance against the linear continuous strategy detailed in Section 5 to explore differences between discrete and continuous approaches.

The network strategies mentioned in Section 6 were trained using a sequence length of $N = 5120$ and stepsize of $\Delta t = 0.01$ (i.e., 100 trades per day). For evaluating the Sharpe ratio in (C.13) the strategies were then evaluated using a longer sequence length corresponding to 10 years of trading ($M = 256,000$ for $\Delta t = 0.01$, averaged over $5,000$ sample paths).

Comparison of network architectures In our experiments, the second network choice (C.20) consistently achieved the best performance. Whence, in the discussion in Section 6 we only focus on this network architecture. For completeness, we also report here a detailed comparison of the three architectures.

As summarized above, three types of networks are trained. **NetSimple** is a simple network where NN_θ is defined as in (C.18) and the corresponding holdings are given by (C.14)–(C.15). **NetPower** incorporates a nonlinear transformation as in (C.19). Finally, we consider the linear network **NetLinear**, where $\mathrm{NN}_\theta = L_{2,1}$ is an affine function and the corresponding holdings are given by (C.14)–(C.15).

Figure 7 reports the dynamics of the objective function (C.17) on out-of-sample validation data during training. At the end of training, **NetPower** reaches a somewhat higher value of the objective function than the other two networks. This result aligns with the performance of the three strategies shown in Table 2. **NetPower** achieves a higher Sharpe Ratio (SR) at very similar risk level compared to the other networks, while the performance of **NetSimple** lies between **NetPower** and **NetLinear**.

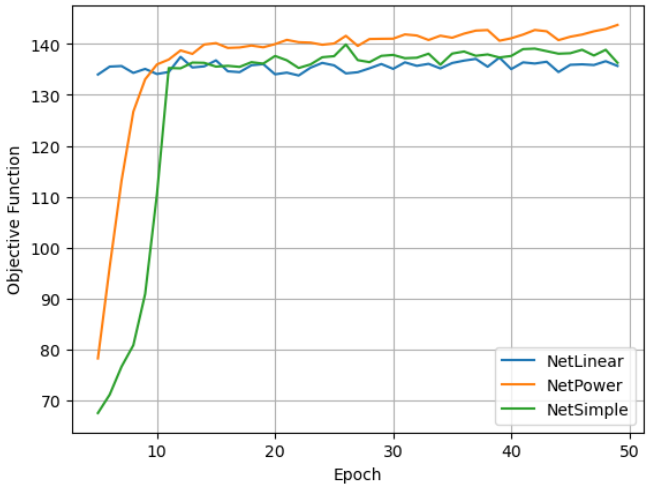


Figure 7: Evolution of the objective function during training, evaluated on out-of-sample validation data.

Table 2: Out-of-sample performance of different network architectures

Network	SR	Risk
NetSimple	1.443	1,223,859
NetLinear	1.438	1,209,254
NetPower	1.461	1,252,336

Implementation details We now detail the hyper-parameters used in training. For each gradient update step we simulate a batch of 1024 signals with sequence length $N = 5120$ according to the dynamics described by equation (C.6). These signals are used as inputs to the neural network to obtain, for each signal, the corresponding trading strategy according to (C.14)-(C.15). The objective function in (C.17) is then computed by a Monte Carlo approximation based on the 1024 samples and optimized using the ADAM optimizer (proposed in Kingma and Ba (2015)) across 50 batches per epoch for a total of 50 epochs. The learning rate is set to 10^{-4} (and 10^{-2} for the linear network) for the first 30 epochs, then reduced by a factor of $1/4$ every 5 epochs thereafter.

A crucial training technique to approximate the ergodic problem at hand involves non-zero initialization of strategy states. To wit, the initial signal f_0, holding Q_0, and the price impact J_0 are not set to zero, but instead sampled from a three-dimensional normal distribution. The covariance matrix for this distribution is dynamically updated throughout training based on the strategies generated in the preceding gradient update step. Specifically, for each gradient update step, we compute the sample covariance matrix from "batch size $\times$ N" triples (f_n, Q_n, J_n) and update the covariance matrix using a

moving average:

$$\text{Cov}_{\text{distribution}} := 0.999\text{Cov}_{\text{distribution}} + 0.001\text{Cov}_{\text{sample}} \tag{C.21}$$

This technique is crucial for several reasons. Firstly, a large price impact encourages the network to produce significant outputs, resulting in strategies with a high initial risk level during the early stages of training. Without this technique, transitioning to the desired risk level can be lengthy and sometimes unsuccessful, thus it effectively speeds up the training process. Secondly, we aim to evaluate the objective function in an ergodic state where the triplet (f_n, Q_n, J_n) follows a time-independent ergodic distribution. It is, therefore, reasonable to aim at initializing (f_0, Q_0, J_0) from this distribution, which we approximate as a normal distribution. This approach ensures that the initial conditions are representative of the expected long-term behavior of the strategy and allows to accurately train the networks for the ergodic problem with a rather short time series of simulated data.

An additional adjustment involves scaling the signal component of the neural network input by 10^6. This scaling balances the input magnitude between the signal and the price impact, which is essential for effective network training.

D Comparison of Neural Network and Viterbi Strategies

In this section, we compare the performance of the neural network strategies to the Viterbi algorithm of Kolm and Ritter (2014) used in Brokmann et al. (2024) for models with instantaneous impact decay. To obtain feasible runtimes for the Viterbi algorithm, we focus on the discrete-time setting with one trade per day from Brokmann et al. (2024), where the objective is to maximize

$$J(Q) = \frac{1}{N}\sum_{n=1}^{N} \mathbb{E}[\alpha Q_n f_n - \frac{\gamma\sigma^2}{2}Q_n^2 - \lambda|\Delta Q_n|^p].$$

Note that no adjustment to parameters α and λ is necessary in this case, as we directly work in the discrete-time case. For neural network training we use a sequence length of $N = 512$ days. For evaluating the Sharpe ratio, we then use $N = 25,600$ time steps, corresponding to around 100 years of trading. We employ model parameters as in Brokmann et al. (2024) and the same neural network hyperparameters as in Section C.2.

Similar to Section C.2, we use neural network strategies with the following general structure:[15]

$$Q_n = \text{NN}_\theta(f_n, Q_{n-1}) \quad \text{for} \quad n = 1, 2, ... \tag{D.1}$$

For comparison we train three types of networks. **NetSimple** is a standard feed-forward network where NN_θ is defined as in (C.18). **NetPretrain** has the same architecture as **NetSimple**, but NN_θ is pretrained using the Viterbi strategy, aiming to minimize the mean-squared error between the neural network output and the Viterbi updating rule from Brokmann et al. (2024) by supervised learning. **NetPower** does not use any such information, but instead incorporates a nonlinear component and trains the more linear residual component, with NN_θ defined as in (C.20).

These three neural network strategies and the Viterbi strategy from Kolm and Ritter (2014); Brokmann et al. (2024), are evaluated by computing the corresponding Sharpe Ratios. For each of those strategies, we also generate the best linear strategy with the same risk level by using the procedure proposed in Brokmann et al. (2024) and compute its Sharpe ratio. The results are reported in Table 3.

Table 3: Out-of-sample SR comparison of neural network, Viterbi and linear strategies.

Network	**SR**	**Risk**	**Linear SR**	**Improvement over Linear SR**
NetSimple	1.5814	1,236,161	1.5745	0.438%
NetPower	1.5900	1,245,040	1.5716	1.174%
NetPretrain	1.5919	1,247,775	1.5707	1.353%
Viterbi	1.5772	1,285,591	1.5584	1.209%

From the table, we observe that – unlike the simplest network **NetSimple** – **NetPretrain** and **NetPower** exhibit similar performances to the Viterbi strategy. Notably, **NetPower** achieves this

[15] For this problem with instantaneous impact decay, dynamic programming suggests that the state variables parametrizing the optimal strategy are the signal f_n and the incoming holdings Q_{n-1}, which motivates (D.1).

without leveraging any information from the Viterbi strategy. These consistent results suggest that the outputs of the algorithms indeed approximate the optimizer of the problem well. In doing so, employing the neural network strategies yields a significant improvement in computational efficiency. Indeed, using a NVIDIA GeForce RTX 4060 Laptop GPU, training **NetSimple** and **NetPreTrain** takes only 15 minutes, while **NetPower** requires 23 minutes. In contrast, it takes almost one day to generate the Viterbi strategy.

References

E. Abi Jaber and E. Neuman. Optimal liquidation with signals: the general propagator case. Preprint, 2022.

A. Alfonsi, A. Fruth, and A. Schied. Optimal execution strategies in limit order books with general shape functions. *Quantitative Finance*, 10(2):143–157, 2010.

R. Almgren and N. Chriss. Optimal execution of portfolio transactions. *Journal of Risk*, 3:5–39, 2000.

R. Almgren, C. Thum, E. Hauptmann, and H. Li. Direct estimation of equity market impact. *Risk*, 18 (7):57–62, 2005.

R. F. Almgren. Optimal execution with nonlinear impact functions and trading-enhanced risk. *Applied Mathematical Finance*, 10(1):1–18, 2003.

E. Bacry, A. Iuga, M. Lasnier, and C.-A. Lehalle. Market impacts and the life cycle of investors orders. *Market Microstructure and Liquidity*, 1(02):1550009, 2015.

B. Biais, P. Hillion, and C. Spatt. An empirical analysis of the limit order book and the order flow in the Paris bourse. *Journal of Finance*, 50(5):1655–1689, 1995.

J.-P. Bouchaud, Y. Gefen, M. Potters, and M. Wyart. Fluctuations and response in financial markets: the subtle nature of 'random' price changes. *Quantitative Finance*, 4(2):176, 2003.

J.-P. Bouchaud, D. Farmer, and F. Lillo. How markets slowly digest changes in supply and demand. In *Handbook of Financial Markets: Dynamics and Evolution*, pages 57–160. North-Holland, Amsterdam, 2009.

J.-P. Bouchaud, J. Bonart, J. Donier, and M. Gould. *Trades, Quotes and Prices*. Cambridge University Press, Cambridge, UK, 2018.

X. Brokmann, E. Sérié, J. Kockelkoren, and J.-P. Bouchaud. Slow decay of impact in equity markets. *Market Microstructure and Liquidity*, 1(2):1550007, 2015.

X. Brokmann, D. Itkin, J. Muhle-Karbe, and P. Schmidt. Tackling nonlinear price impact with linear strategies. Forthcoming in *Mathematical Finance*, 2024.

F. Bucci, M. Benzaquen, F. Lillo, and J.-P. Bouchaud. Slow decay of impact in equity markets: insights from the ANcerno database. *Market Microstructure and Liquidity*, 4(3/4):1950006, 2018.

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

Á. Cartea, S. Jaimungal, and J. Penalva. *Algorithmic and High-Frequency Trading*. Cambridge University Press, Cambridge, UK, 2015.

Á. Cartea, S. Jaimungal, and L. Sánchez-Betancourt. Deep reinforcement learning for algorithmic trading. In *Machine Learning and Data Sciences for Financial Markets*, pages 230–250. Cambridge University Press, Cambridge, UK, 2023.

T. Cayé, M. Herdegen, and J. Muhle-Karbe. Trading with small nonlinear price impact. *Annals of Applied Probability*, 30(2):706–746, 2020.

T. Chen, M. Ludkovski, and M. Voß. On parametric optimal execution and machine learning surrogates. *Quantitative Finance*, 24(1):15–34, 2024.

P. Collin-Dufresne, K. Daniel, and M. Sağlam. Liquidity regimes and optimal dynamic asset allocation. *Journal of Financial Economics*, 136(2):379–406, 2020.

G. Curato, J. Gatheral, and F. Lillo. Optimal execution with non-linear transient market impact. *Quantitative Finance*, 17(1):41–54, 2017.

S. Denkl, M. Goy, J. Kallsen, J. Muhle-Karbe, and A. Pauwels. On the performance of delta hedging strategies in exponential Lévy models. *Quantitative Finance*, 13(8):1173–1184, 2013.

N. Gârleanu and L. H. Pedersen. Dynamic trading with predictable returns and transaction costs. *Journal of Finance*, 68(6):2309–2340, 2013.

N. Gârleanu and L. H. Pedersen. Dynamic portfolio choice with frictions. *Journal of Economic Theory*, 165:487–516, 2016.

J. Gatheral, A. Schied, and A. Slynko. Transient linear price impact and Fredholm integral equations. *Mathematical Finance*, 22(3):445–474, 2012.

I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, Cambridge, MA, 2016.

P. Guasoni and M. H. Weber. Nonlinear price impact and portfolio choice. *Mathematical Finance*, 30 (2):341–376, 2020.

J. Hasbrouck. Measuring the information content of stock trades. *Journal of Finance*, 46(1):179–207, 1991.

N. Hey, I. Mastromatteo, J. Muhle-Karbe, and K. Webster. Trading with concave price impact – theory and evidence. Preprint, 2023.

N. Hey, J.-P. Bouchaud, I. Mastromatteo, J. Muhle-Karbe, and K. Webster. The cost of misspecifying price impact. *Risk*, January, 2024.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

P. Kolm and G. Ritter. Multiperiod portfolio selection and Bayesian dynamic models. *Risk*, 18(7):50–54, 2014.

P. Kolm and G. Ritter. Dynamic replication and hedging: A reinforcement learning approach. *Journal of Financial Data Science*, 1:1 59–171, 01 2019.

L. Leal, M. Laurière, and C.-A. Lehalle. Learning a functional control for high-frequency finance. *Quantitative Finance*, 22(11):1973–1987, 2022.

T. F. Loeb. Trading cost: The critical link between investment information and results. *Financial Analysts Journal*, 39(3):39–44, 1983.

C. Moallemi and M. Sağlam. Dynamic portfolio choice with linear rebalancing rules. *Journal of Financial and Quantitative Analysis*, 52(3):1247–1278, 2017.

C. Moallemi and M. Wang. A reinforcement learning approach to optimal execution. *Quantitative Finance*, 22(6):1051–1069, 2022.

J. Muhle-Karbe, Z. Wang, and K. Webster. Stochastic liquidity as a proxy for nonlinear price impact. *Operations Research*, 72(2):444–458, 2023.

E. Neuman and Y. Zhang. Statistical learning with sublinear regret of propagator models. Preprint, 2023.

A. Obizhaeva and J. Wang. Optimal trading strategy and supply/demand dynamics. *Journal of Financial Markets*, 16(1):1–32, 2013.

B. Park and B. Van Roy. Adaptive execution: Exploration and learning of price impact. *Operations Research*, 63(5):1058–1076, 2015.

R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, second edition, 2018.

K. Webster. *Handbook of Price Impact Modeling*. CRC Press, Boca Raton, FL, 2023.