An Application of Deep Reinforcement Learning for Order Execution

with a supporting analysis of Optimal Liquidation under a Transient Price Impact Model

Matthew B. Reiter
Supervised by Dr. Y. Lawryshyn

BASc. Thesis
University of Toronto
Spring 2020

Introduction

The main objective of the thesis is to develop an Asynchronous Advantage Actor-Critic (A3C) model for Optimal Order Execution. The subject of Optimal Order Execution is considered within the context of Volume-Weighted Average Price (VWAP) execution. Deep Reinforcement Learning methods offer an attractive models-free approach to Optimal Order Execution, where the complicated factors that drive asset prices can be handled under reduced modelling assumptions. A notable academic contribution in the space of Deep Reinforcement Learning comes from the work of [4], for first adapting a *Double-Deep Q-Network* that trains under an Optimal Order Execution directive.

The primary objective is fortified with two supplementary topics: the first relating to a Teacher/Student learning framework, and the second covering a models-dependent approach to Order Execution. The additional topics are covered to gain familiarity in the constructs that facilitate the application of A3C for VWAP execution. For instance, the Teacher/Student framework reveals that a Student model can be trained in less time, compared to a baseline model, when given parameter preinitializations from the Teacher, in addition to incorporating an *Advice & State-Driven* decisioning network. Furthermore, the models-dependent approach to Order Execution offers insight into how economic factors such as market depth, resilience, tightness and bias influence optimal trading decisions.

The purpose of this document is to summarize the thesis. The content in this document follows the chapter-layout of the thesis, and the conclusion provides recommendations for each of the three objectives studied.

Methodology

Teacher/Student Learning

The Teacher/Student learning framework is approached from the perspective of efficiently training a Student model by processing advice from a Teacher model. Baseline models are used for comparison purposes, and are trained on three Atari 2600 games: Breakout, Beamrider and Space Invaders. For the Student models, two learning networks are constructed that facilitate *Advice-Driven* (AD) decisioning and *Advice & State-Driven* (ASD) decisioning. The two learning networks differ in the sense that the ASD model incorporates both an advice-stream and an observation-stream, whereas the AD model takes only Teacher-fed advice as an input. Figure 1 is provided for an overview of the main network structures; note that the AD model is captured in that of the ASD structure.

To make the learning framework non-trivial, the Teacher and Student models each receive different objectives. The Teacher model inherits the same network structure as the baseline model, but the Teacher is incentivized to *stay alive for as long as possible*. To effectively reinforce the Teacher for a survival objective, Equation 1 presents the crafted reward signal, where t is the episode length, $t_{\max}$ is the maximum episode length, and parameters η, κ are interpreted respectively as the survival bonus and the hit penalty.

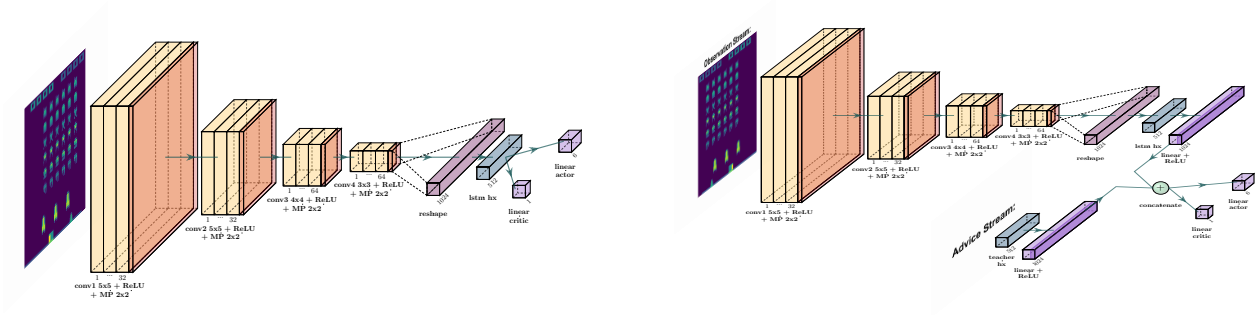
$$r(t) = \frac{t\eta}{t_{\max}}\left(1 + \mathbb{1}_{\{\texttt{done}\wedge\texttt{cleared}\}}\right) - \kappa\mathbb{1}_{\{\texttt{done}\wedge\neg\texttt{cleared}\}} \quad (1)$$

The primary performance metric to evaluate the success of the Teacher model is the average episode length over a training session. The expectation is that a proficient Teacher should realize an average episode length that approaches the maximum episode length of $10,000$.

The Student models are trained to the specifications of the original Atari 2600 games. Both the AD and the ASD models source advice from the Teacher by "detaching" the output state of the Teacher's LSTM cell. In this sense, the advice given to the Student is based on the collective experience of the Teacher, from which the Student preserves an ability to learn a policy by generalizing the advice. Intuitively, an AD model is expected to perform well when the Teacher's task is similar to the Student's objective. Since the AD model does not directly process a state observation, decisions rely on a second-hand connection to the environment and may suffer from poor generalizations if the Teacher is suboptimally trained.

To introduce a first-hand connection to the environment is to add an observation-stream, and the ASD model does just that. The structure of the ASD network allows for the parameters in the observation-stream to be preinitialized with weights from the Teacher's network. Under such a scheme, the ASD learner would begin by making informed actions under the Teacher's directive, and gradually shift decisioning towards an optimal policy according to the different task.

The A3C algorithm is adapted to train the Teacher/Student learning framework, where the advantage function is estimated using the Generalized Advantage Estimator. Consult the thesis document for details pertaining to the preprocessing of the state observations and the configuration of the hyperparameters.



(a) The baseline Neural Network structure

(b) The Advice & State-Driven Neural Network structure

Figure 1: Diagrams of the Neural Network structures used for the Teacher/Student learning framework.

A Transient Price Impact Model for Order Execution

For a models-dependent approach to Order Execution under a transient price impact model, consider a trader who is liquidating an inventory of γ_{0^-} shares over a fixed time horizon. The trader's inventory process is captured by $\gamma_t = \gamma_{0^-} + \gamma_t^+ - \gamma_t^-$, where $\gamma_t^{\pm}$ are non-decreasing cádlág functions stipulating the quantity of shares purchased $(+)$ or sold $(-)$. The price model is based on the block shaped limit order book from [1] and is presented in Equation 2.

Note that $(P)_{t\geq 0}$ is the unperturbed price of the asset, which is realized in the absence of market activity, and is taken as an arithmetic Brownian Motion with constant exogenously-determined coefficients μ and σ. Furthermore, the positive constants λ and α account for the depth of the market and the rate of recovery for the market respectively. Under the dynamics of Equation 2, purchases appreciate the best available ask-price and sales erode the best available bid-price in proportion to the quantity traded. Price processes recover to P_t at a rate α.

$$\begin{aligned} dA_t &= dP_t + \lambda d\gamma_t^+ - \alpha(A_t - P_t)dt \\ dB_t &= dP_t - \lambda d\gamma_t^- - \alpha(B_t - P_t)dt \end{aligned} \tag{2}$$

Under the transient price impact model, the dynamics for the trader's liquidation wealth is captured by Equation 3. The liquidation wealth provides the profits from trading, from the start and right up to the end of the horizon τ. As an item of notation, denote $\Delta\gamma_t^{\pm} = \gamma_t^{\pm} - \gamma_{t^-}^{\pm}$ by the magnitude of a block/impulse trade made at time t.

If the trader carries constant absolute risk aversion, the utility function is exponential with risk aversion $\beta \geq 0$. Then, solving for the optimal trading policy amounts to maximizing the mean-variance objective function given in Equation 4, with the cost terms aggregated in $C_\tau(\gamma)$.

$$dw_t = \big(B_{t^-} - \lambda\Delta\gamma_t^-\big)d\gamma_t^- - \big(A_{t^-} + \lambda\Delta\gamma_t^+\big)d\gamma_t^+ \tag{3}$$

The objective function is both convex and quadratic in γ_t. This means that under an appropriate discretization, Equation 4 can be solved on a point-wise basis as a Quadratic Program. It will be convenient to evenly-space the time intervals between n distinct trades, in which case the discretization is taken over a grid $\Xi = \{0, h, 2h, \ldots, \tau\}$ for $h = \tau/n$. Under the discrete-time setting, the optimal trading policy is solved computationally under generalized cases. Additionally, when P_t is a martingale and there is no initial price tightness nor bias – meaning that the starting bid/ask spread $S_{0^-} = A_{0^-} - B_{0^-} = 0$ and the initial mid-quote $Q_{0^-} = (A_{0^-} + B_{0^-})/2 = P_{0^-}$ – a closed-form solution can be determined for a risk-neutral trader using Lagrangian multipliers. A limit of this closed-form solution shifts attention towards a continuous-time modelling approach, where a variational argument mirroring that from [3] confirms the previous discretization technique.

$$\begin{aligned} J_\tau(\gamma) = P_{0^-}\gamma_{0^-} + \mu\int_0^\tau \gamma_{t^-}\,dt - \frac{1}{2}C_\tau(\gamma) \\ - \frac{\beta\sigma^2}{2}\int_0^\tau \gamma_{t^-}^2\,dt \to \max_{\gamma}! \end{aligned} \tag{4}$$

An A3C Model for Order Execution

In devising the A3C model for VWAP execution, three training environments are developed that each prove conducive for Reinforcement Learning. The first environment is based on the block shaped limit order book (Equation 2), the second environment is built on historical limit order book data from the Nasdaq stock exchange, and the third environment is adapted from a simulated order-driven market maintained by the Rotman School of Management at the University of Toronto. Refer to the thesis for a detailed description of each environment, with accompanying references for the open-source code.

Figure 2 presents the Neural Network structure, the features, and the action space used for the A3C model. Here, $p_b^{(l)}$, $p_a^{(l)}$ are the bid- and ask-prices for the l^{th} level in the limit order book; $v_b^{(l)}$, $v_a^{(l)}$ are the bid- and ask-volumes for the l^{th} level in the limit order book; I is the inventory that the agent must liquidate (resp. acquire); ρ_t is the

position at time t which has been executed; $\tilde{\rho}_t$ is the magnitude of any pending orders at time t; and N_p, N_v, N_ρ are normalization factors for the prices, volumes and positions respectively.

The network structure is indeed similar to the baseline model in the Teacher/Student learning framework, and the two convolutional layers have a financial interpretation. As demonstrated in [5], convolutional networks can apply a feature-mapping on limit order book data that measures a micro-price $\tilde{p}^{(l)}$ on volume imbalance. The feature-mapping is captured in Equation 5 and provides a relevant technical indicator for the network's features.

$$\tilde{p}^{(l)} = \frac{v_b^{(l)}}{v_b^{(l)} + v_a^{(l)}} p_a^{(l)} + \frac{v_a^{(l)}}{v_b^{(l)} + v_a^{(l)}} p_b^{(l)} \tag{5}$$

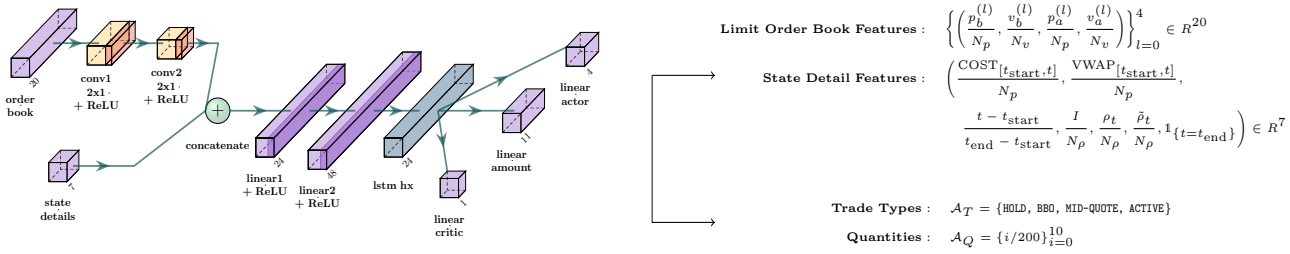


Figure 2: The Neural Network structure, the features, and the action space used for VWAP execution with A3C.

The design of the reward signal proves critical to the performance of the A3C model for VWAP execution. Drawing inspiration from [2] – where a closed-form VWAP strategy is shown to intrinsically track the Time-Weighted Average Price (TWAP) with appropriate offsets – the reward signal used in the thesis incorporates a TWAP tracking component. Equation 6 defines the reward signal separately for both a liquidation $(-)$ and an acquisition $(+)$ objective, noting that $\epsilon \in [0, 1]$ gives a tolerance parameter for deviating from TWAP, and is set to $\epsilon = 0.05$.

$$r^{\pm}(t) = \underbrace{\mathbb{1}_{\left\{\left|\frac{\rho_t}{I} - \frac{t - t_{\text{start}}}{t_{\text{end}} - t_{\text{start}}}\right| \leq \epsilon\right\}} - 0.5}_{\text{incentive to track TWAP}} + \underbrace{\mathbb{1}_{\{\rho_{t-1} \neq \rho_t\}} \mathbb{1}_{\left\{\mp\left(\text{COST}_{[t_{\text{start}}, t]} - \text{VWAP}_{[t_{\text{start}}, t]}\right) \geq 0\right\}}}_{\text{incentive to outperform VWAP}} \tag{6}$$

The agent trades by specifying an action tuple $(a_t^{(T)}, a_t^{(Q)})$, where $a_t^{(T)} \in \mathcal{A}_T$ gives the trade type and $a_t^{(Q)} \in \mathcal{A}_Q$ gives the quantity to be traded. Trading is restricted to be unidirectional and a `MARKET-ORDER` is imposed to clear unspent inventory at the end of the horizon. The A3C model is executed with 16 training processes. Each training agent is assigned, uniformly at random, either a liquidation or acquisition directive. The following parameters are also assigned randomly based on the discrete uniform distribution $\mathcal{U}(a, b)$: $I = 10,000\big(10 + \mathcal{U}(-4, 4)\big)$, $t_{\text{start}} = 5 + \mathcal{U}(0, 75)$ and $t_{\text{end}} = 290 - \mathcal{U}(0, 100)$. Appendix A outlines the adapted A3C algorithm that facilitates VWAP execution.

Main Results

Teacher/Student Learning

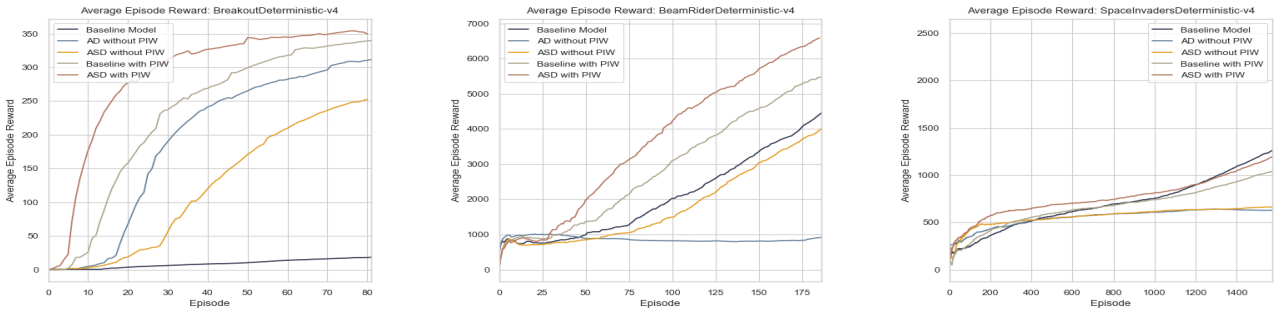
As evidenced by Table 1, the baseline models were trained to proficiency more effectively than the Teacher models. This is reflected in both the average episode reward and the average episode length. Both Teacher models trained on Breakout and Beamrider were eventually able to demonstrate a policy that achieves maximum survival in their respective environment. The same result was not secured for Space Invaders, where at no point during the training session did the Teacher come within fifty percent of the maximum survival target of $10,000$.

The shortcomings of the Teacher models trained under Equation 1 indicate that a proficient reward signal must provide a direct link between positive reinforcement and desired behaviour; the causality of the linkage directly impacts learning progression. To this tune, Figure 3 demonstrates how the Teacher models were able to transfer their survival-centric policies to assist with the training of the Student models.

Table 1: Performance of the baseline and Teacher models.

	Baseline Model			Teacher Model		
	Number of Episodes	Average Episode Reward	Average Episode Length	Number of Episodes	Average Episode Reward	Average Episode Length
▸ Breakout	304	202	3553	278	110	1693
▸ Beamrider	278	6793	5093	938	901	4968
▸ Space Invaders	2256	2525	2895	4101	394	1683

The best performing Student model was the ASD learner with preinitialized weights (PIW) from the Teacher. Two of the ASD models realized an average episode reward that was 3% (ref. Breakout) and 20% (ref. Beamrider) larger than their comparable preinitialized baseline models. While the results are encouraging, the ASD model for Space Invaders underperformed the baseline model by 6% on account of the Teacher being suboptimally trained. As a result, the engineering of the reward signal proves critical to the training scheme of Teacher model, and has a downstream effect on the learning ability of the Student.



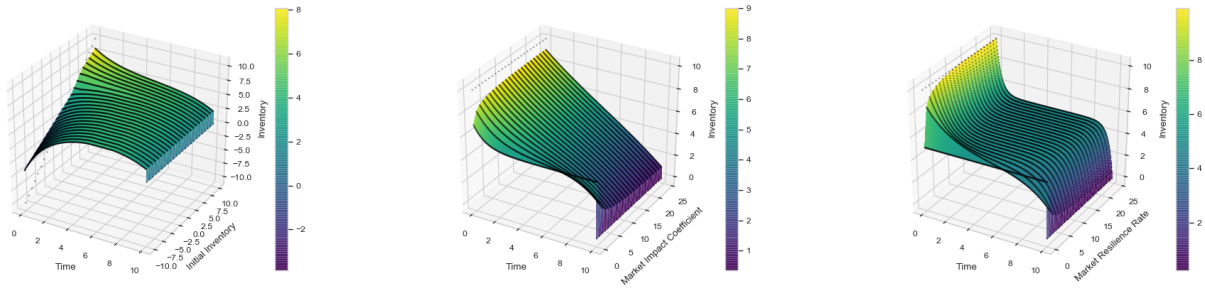
(a) Breakout *(b) Beamrider* *(c) Space Invaders*

Figure 3: The performance of the Student models.

In reference to Figure 3, when the baseline models were prescribed PIW, the Neural Network structure was slightly modified to accept a fully-connected layer after the LSTM cell. Accordingly, the baseline models with PIW are akin to the observation-stream from Figure 1. The relevance of this modification is to show that PIW prescription is not trivial for Transfer Learning models, and the process still relies on an adequate generalization method, for which the ASD learner appears to demonstrate. In all, such work is extendable outside the context of a Teacher/Student learning framework and offers relevance when designing a Deep Reinforcement Learning application from scratch.

A Transient Price Impact Model for Order Execution

Optimizing the discretization of Equation 4 is done as a Quadratic Program. In order to investigate the optimal trading policies, a hypothetical market is constructed under varying agent- and market-specific parameters. Shown below in Figure 4 are the classes of optimal trading policies under different initial inventory levels, market impact coefficients, and market resiliency rates. Addressing the latter two, it becomes clear that when the market is governed by a strict impact coefficient, then liquidity becomes scarce, and the optimal policies respond by trading in smaller allotments. The opposite nature is observed when the resiliency rate is reduced; when prices recover slowly to the fundamental level, then optimality calls for larger block trades and a reduced effective rate of trading.



(a) Initial Inventory Position *(b) Market Impact Coefficient* *(c) Market Resilience Rate*

Figure 4: Optimal policies when different agent and market-specific parameters are varied.

The analysis of the discrete-time system offers meaningful insights into the influence of market depth, resilience, tightness and bias on the optimal trading policies. Further generalizations though are difficult to make in the discrete-time setting; without a closed-form solution, it is challenging to infer a clear picture as to how each parameter changes the nature of the optimal trading policy. What remains however, is that the solution space is richly diverse and, at least at a surface level, offers insights that are intuitive from an economic perspective. The thesis document explores a continuous-time generalization and recovers results from [1] and [3], by analytically evaluating a limit as $n \to \infty$, and also exploring a similar setup under a variational argument with sub-gradients for the cost functional.

An A3C Model for Order Execution

The A3C model for VWAP execution posted a modest profit throughout the training period, but only after including a one-cent discount from VWAP. When it came to tracking VWAP throughout the training session, the agent underperformed by three basis-points on average.

Figure 5 provides insight into the learning progression of the agent. Within the first 200 training episodes, the agent primarily posted limit orders at the mid-quote, and traded almost exclusively in 2% allotment sizes. The result led to the most volatile profit-levels realized throughout training, as the agent traded through inventory levels early-on and was detrimentally unable to capture price appreciations towards the end of the trading horizon. For a brief period, from about the 250^{th} episode through to the 500^{th}, the agent began trading in smaller quantities and at more favourable prices. During this period, the agent outperformed VWAP consistently. However, as the agent continued to explore the action space, active orders became more frequent and larger swings in profits followed.

The agent appears to have learned to track TWAP, as evidenced by the large concentration of small order quantities. This would suggest that the TWAP tracking factor dominates the VWAP outperformance incentive in Equation 6. A proposed reward signal is given below in Equation 7, wherein the agent would directly receive a reward based on the magnitude of outperformance with respect to VWAP.

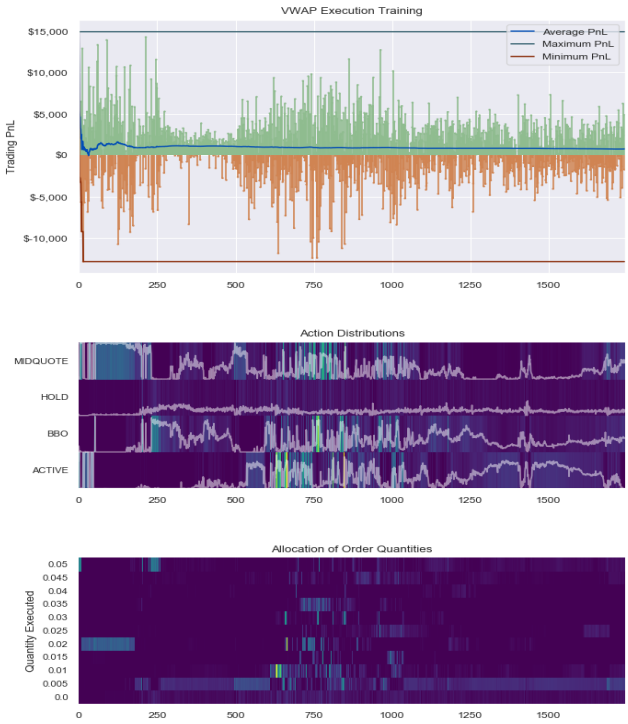


Figure 5: Training the A3C model for VWAP execution.

$$r^{\pm}(t) = \underbrace{\mathbb{1}_{\left\{\left|\frac{\rho_t}{I} - \frac{t - t_{\text{start}}}{t_{\text{end}} - t_{\text{start}}}\right| \leq \epsilon\right\}} - 0.5}_{\text{incentive to track TWAP}} \mp \underbrace{100 \times \mathbb{1}_{\{\rho_{t-1} \neq \rho_t\}} \left(\text{COST}_{[t_{\text{start}}, t]} - \text{VWAP}_{[t_{\text{start}}, t]}\right)}_{\textbf{stronger} \text{ incentive to outperform VWAP}} \tag{7}$$

With a modified reward signal comes an opportunity to adopt a Teacher/Student learning framework. Under an ASD learner, the agent trained under Equation 6 would act as a Teacher for a Student receiving reinforcement from Equation 7. The thesis document motivates this approach with a small-scale test, under which the ASD learner outperformed VWAP by approximately 0.5 basis points on average.

Conclusions and Recommendations

The success of the Teacher/Student learning framework depends on the complexity of the environment, the relevance of the Teacher's advice for the Student's objective, and the level of mastery achieved by the Teacher. The Teacher/Student learning framework culminates with the following recommendation:

▶ **Recommendation 1:** The Teacher/Student framework is evidenced to train a reference Deep Reinforcement Learning model more efficiently. To leverage this framework, the parameters for the Student model should be preinitialized from the Teacher's network, and the structural component for the Student's network should be under the *Advice & State-Driven* decisioning approach. The resulting ASD learner would be equipped to learn enriched generalizations from the Teacher's advice, and efficiently reconcile this input with the observation-stream.

Next, the transient price impact model provides an accessible entry point to the study of Optimal Order Execution. The accessibility comes from a solution method which avoids a complicated Hamilton-Jacobi-Bellman type equation, and instead uses Quadratic Programming and Variational Calculus. In light of the convenient interpretations, the recommendation coming from this work is given:

▶ **Recommendation 2:** The block shaped limit order book is a convenient setting to define a transient price impact model on and outline an Optimal Liquidation objective to. The modelling in discrete-time is easily achieved and the dynamics allow for insightful continuous-time representation. The take-away from completing this work is a characterization of the richly diverse trading policies under varying agent- and market-specific parameters.

Finally, the A3C model for VWAP execution is constructed from the ground-up, and involves the careful implementation of trading environments which are conducive for Reinforcement Learning applications. While the A3C model for VWAP execution remains simple, and the accompanying results prove modest, the extensive detailing of the setup offers ample room for improvements. Immediately, evidence shows that the ASD learning framework would be well suited for supplemental training efforts. Under this pretext, the final recommendation is summarized as:

▶ **Recommendation 3:** The A3C model is able to modestly track VWAP on average. The limitations of the modelling scheme introduce an opportunity to consider an enhanced reward signal, whereby the *Advice & State-Driven* learning model is a prime candidate for supplemental training.

References

[1] Anna A. Obizhaeva and Jiang Wang. "Optimal trading strategy and supply/demand dynamics". In: *Journal of Financial Markets* 16.1 (2013), pp. 1–32. DOI: https://doi.org/10.1016/j.finmar.2012.09.001.

[2] Álvaro Cartea and Sebastian Jaimungal. "A Closed-Form Execution Strategy to Target Volume Weighted Average Price." In: *SIAM J. Financial Math.* 7.1 (2016), pp. 760–785. URL: http://dblp.uni-trier.de/db/journals/siamfm/siamfm7.html#CarteaJ16.

[3] Moritz Voß. "Dynamic Hedging in Illiquid Financial Markets". PhD thesis. Berlin Institute of Technology, 2017.

[4] Brian Ning, Franco Ho Ting Ling, and Sebastian Jaimungal. "Double Deep Q-Learning for Optimal Execution." In: *CoRR* abs/1812.06600 (2018). URL: http://dblp.uni-trier.de/db/journals/corr/corr1812.html#abs-1812-06600.

[5] Zihao Zhang, Stefan Zohren, and Stephen J. Roberts. "DeepLOB: Deep Convolutional Neural Networks for Limit Order Books." In: *IEEE Trans. Signal Process.* 67.11 (2019), pp. 3001–3012. URL: http://dblp.uni-trier.de/db/journals/tsp/tsp67.html#ZhangZR19.

Appendix A

Algorithm 1 Asynchronous Advantage Actor Critic for VWAP execution

Inherit shared global parameters θ_T, θ_Q, θ_v and common global counter $\tau = 0$
Initialize process-specific parameters θ'_T, θ'_Q, θ'_v
Initialize process step counter $t \leftarrow 1$
repeat
  Reset gradients $d\theta_T \leftarrow 0$, $d\theta_Q \leftarrow 0$, $d\theta_v \leftarrow 0$
  Synchronize process-specific parameters $\theta'_T \leftarrow \theta_T$, $\theta'_Q \leftarrow \theta_Q$, $\theta'_v \leftarrow \theta_v$
  $t_{\text{start}} \leftarrow t$
  repeat
    Choose trade type $a_t^{(T)}$ and trade quantity $a_t^{(Q)}$ from s_t using π
    Compute entropies $e_t^{(T)}$, $e_t^{(Q)}$
    Execute trade $\left(a_t^{(T)}, a_t^{(Q)}\right)$, observe new state s_{t+1}
    construct reward r_t according to Equation 6
    $t \leftarrow t + 1$
    $\tau \leftarrow \tau + 1$
  until s_t is terminal or $t - t_{\text{start}}$ reaches the maximum step count
  $$R = \begin{cases} 0 & s_t \text{ terminal} \\ V(s_t, \theta'_v) & \text{otherwise} \end{cases}$$
  Initialize Generalized Advantage Estimator GAE $\leftarrow 0$
  $i \leftarrow t - 1$
  repeat
    $R \leftarrow r_i + \gamma R$
    $\text{GAE} \leftarrow \tau\gamma\text{GAE} + r_i + \gamma V(s_{i+1}, \theta'_v) - V(s_i, \theta'_v)$
    $d\theta_T \leftarrow d\theta_T + \nabla_{\theta'_T}\left[\log \pi(a_i|s_i; \theta'_T, \theta'_Q)\text{GAE} - 0.01 e_i^{(T)}\right]$
    $d\theta_Q \leftarrow d\theta_Q + \nabla_{\theta'_Q}\left[\log \pi(a_i|s_i; \theta'_T, \theta'_Q)\text{GAE} - 0.01 e_i^{(Q)}\right]$
    $d\theta_v \leftarrow d\theta_v + \frac{\partial}{\partial \theta'_v}\left(R - V(s_i, \theta'_v)\right)^2$
    $i \leftarrow i - 1$
  until $i < t_{\text{start}}$
  Asynchronous updates of shared global parameters
until $\tau >$ maximum time for the episode