# A Sharpe Ratio Based Reward Scheme in Deep Reinforcement Learning for Financial Trading

G. Rodinos, P. Nousi, N. Passalis, and A. Tefas

Computational Intelligence and Deep Learning Research Group
Artificial Intelligence and Information Analysis lab, Department of Informatics
Aristotle University of Thessaloniki, Thessaloniki, Greece
Emails: {grodinos, paranous, passalis, tefas}@csd.auth.gr

**Abstract.** Deep Reinforcement Learning (DRL) is increasingly becoming popular for developing financial trading agents. Nevertheless, the nature of financial markets to be extremely volatile, in addition to the difficulty of optimizing DRL agents, lead the agents to make more risky trades. As a result, while agents can earn higher profits, they are also vulnerable to significant losses. To evaluate the performance of the financial trading agent, the Profit and Loss (PnL) is usually calculated, which is also used as the agent's reward. However, in addition to PnL, traders often take into account other aspects of the agent's behavior, such as the risk associated with the positions opened by the agent. A widely used metric that captures the risk-related component of an agent's performance is the Sharpe ratio, which is used to evaluate a portfolio's risk-adjusted performance. In this paper, we propose a Sharpe ratio-based reward shaping approach that enables optimizing DRL agents by taking into account both PnL and the Sharpe ratio, with the objective to improve the overall performance of the portfolio, by mitigating the risk that occurs in the agent's decisions. The effectiveness of the proposed method to increase different performance metrics is illustrated using a dataset provided by Speedlab AG, which contains 14 instruments.

**Keywords:** Financial Trading · Reward Shaping · Deep Learning · Deep Reinforcement Learning.

## 1 Introduction

Using traditional machine learning methods for automated financial trading can be very challenging. Most of the time, the creation of supervised labels is needed. In works, such as [14–17], Deep Learning (DL) models were used to predict the price movement and depending on the direction, a trader is able to make a decision to either go long or short. However, this task might be challenging because of the uncertainty of the financial markets. The use of Deep Reinforcement Learning (DRL) is an efficient way to follow, yet tough, to avoid the limitations of supervised learning. In works such as [1, 3, 13, 18, 19], a DRL framework was used to overcome possible restrictions occurring on supervised problems.

DRL agents for automated financial trading are difficult to be developed since a carefully designed reward scheme is required [8]. As tasks get more complicated, reward shaping becomes more challenging, while recent applications have demonstrated that adapting it to the specific domain of its usage may considerably increase the agents' performance [10, 11].

There are works that use the Profit and Loss (PnL) as a reward but the agent doesn't take into account the risk that often arises in the trades. In addition, some works also use the Sharpe ratio as a reward, however, sometimes seems not to work effectively, such as in [21]. In this work, instead of using rewards that are based only on the agent's PnL or the Sharpe ratio, we force the agent to take into account both PnL and the Sharpe ratio in the reward function.

Sharpe ratio was originally mentioned in the 1960s by William F. Sharpe [5]. It is a measure of the risk-adjusted return of an investment or portfolio and constitutes one of the most widely used metrics in finance. The Sharpe ratio is calculated as the average return of an investment minus the risk-free rate of return, divided by the standard deviation of the investment's returns. In our case, the risk-free rate is assumed to be zero as a practical simplification. Keep in mind that in practice, the risk-free rate is never truly zero. The standard deviation measures the volatility of the investment's returns and captures the idea that higher returns should be associated with higher risk. A higher Sharpe ratio indicates that an investment has provided a better return for the amount of risk taken. The Sharpe ratio is used to evaluate the performance of individual investments as well as portfolios and is a useful metric for comparing different investment options and helping make investment decisions.

Even though the Sharpe ratio is widely used, currently there is no such work that takes advantage of it combined with a PnL-based reward, when training DRL agents. Sharpe ratio is usually calculated as an annualized metric, which means that in order to be calculated, takes into account the returns over a long period of time. In practice, the volatility of monthly returns is typically considered when using the Sharpe ratio, which is generally lower than that of daily returns, which are in turn less volatile than hourly returns. However, when training a trading agent, the returns that are available, are hourly sampled, and normally equal to the number of steps an agent makes in an RL episode. As a consequence, the existing volatility in an RL episode may be significant.

Our contribution can be summarized as follows. We propose a method to incorporate the Sharpe ratio into the training regime of a DRL agent, to mitigate the risk of the taken action by the agent. Specifically, to overcome the aforementioned limitations regarding the calculation of the Sharpe ratio, we introduce a window that dynamically changes its size, by taking into account the returns we have available inside an RL episode. Thus, we are able to have an approximation of the Sharpe ratio that can be included in the reward function.

The structure of this paper is as follows. In Section 2 the background is mentioned along with the proposed method which is introduced and analytically described. Then the dataset as well as the experimental evaluation are presented in Section 3. Finally, Section 4 concludes this paper.

## 2 Proposed Method

This Section introduces the background related to DRL. The baseline PnL reward is presented, followed by the PnL and Sharpe ratio reward scheme and the proposed one. All of them are determined and thoroughly explained.

### 2.1 Background

The DRL setup is briefly described in the next paragraphs. We follow a similar approach for financial trading that was used in [3, 13].

In financial trading via DRL, the environment provides the agent with an observation, which consists of features generated from market data as presented in Section 3.2. Along with the observation, the current market position is provided, which is denoted as $e_t$, where $e_t \in \{1, 0, -1\} = \{long, neutral, short\}$. The combination of these two, forms the state of the environment, $s_t$, at time $t$, where time $t$, specifies the simulation moment in time. The dimensions of the state are equal to $d \times T$, where $d$ is the number of features and T specifies the time steps that occurred prior to time $t$.

Every time $t$, the agent has the choice to either buy, sell or stay out of the market, depending on the state, $s_t$, that receives from the environment. For every action, at time $t$ receives a reward, $r_t$. The proposed reward is received by the position currently held and is compared to two other methods. When the agent changes the current position held, a commission is paid to make the change. To make the simulation process easier, we chose a reasonable commission for all the transactions.

### 2.2 PnL reward

Rewarding an agent based on the profit of the positions taken is a common methodology for financial trading with Reinforcement Learning, e.g., [1–3]. This approach is our base and is also separately tested in this study. The profit-based reward is defined as:

$$r_t^{(PnL)} = \begin{cases} z_t, & \text{if agent going long} \\ -z_t, & \text{if agent going short} \\ 0, & \text{if agent has a neutral position} \end{cases} \tag{1}$$

where $z_t$ is the return change and is defined as:

$$z_t = \frac{p_c(t) - p_c(t-1)}{p_c(t-1)} \tag{2}$$

which is also referred to as the change of the close price $p_c$. With the return definition, the reward of Equation 1 can be written as:

$$r_t^{(PnL)} = e_t \cdot z_t \tag{3}$$

When the agent changes position is obligated to pay an extra *fee*. That is called the *commission*, in which case an additional reward is formulated as:

$$r_t^{(fee)} = -c \cdot |e_t - e_{t-1}| \tag{4}$$

where $c$ denotes the commission. The total PnL reward can be defined as:

$$r_t^{(total)} = r_t^{(PnL)} + r_t^{(fee)} \tag{5}$$

### 2.3 PnL and Sharpe ratio reward

As discussed previously, the Sharpe ratio is a metric that is usually calculated annually. However, in our study, we propose to include it in the reward as an approximation, in every RL episode. That means that we calculate it over a short period of time.

Let $m$ be the number of time steps that an episode consists of. We introduce a window, let $w$ be the window, over the period of $m$ steps, which increases its size dynamically. The agent, in order to calculate the approximation of the Sharpe ratio, will take into consideration the trades that took place in the last $m/2$ steps, and in each step, its size grows, up to $m$. Reward, based on the approximated Sharpe ratio is defined as:

$$r_t^{(sr)} = \frac{E[\mathbf{z}]}{\sqrt{Var[\mathbf{z}]}} \cdot \alpha \quad t \in \{w, ....., m\}, \quad \mathbf{z} = (z_0, ...., z_t) \tag{6}$$

where $w = m/2$, $\mathbf{z}$ is a vector with the returns as defined in Equation 2, and $\alpha$ is a constant value, typically less than 1, that can be adjusted and influence the agent's behavior. PnL rewards are normally in a very small range. Multiplying the approximated Sharpe ratio reward in Equation 6, with a scale factor less than 1, we avoid overpowering the PnL reward. The total PnL and Sharpe ratio reward is defined as:

$$r_t^{(total)} = \begin{cases} r_t^{(PnL)} + r_t^{(fee)}, & t < w \\ r_t^{(PnL)} + r_t^{(fee)} + r_t^{(sr)}, & for \quad t \geq w \end{cases} \tag{7}$$

### 2.4 Proposed reward

The proposed Sharpe ratio-based reward shaping scheme allows for training agents that handle the risk taken in every transaction, significantly improving their risk-adjusted performance and the total profits, as it is experimentally illustrated in Section 3. The total reward of the proposed scheme is defined as:

$$r_t^{(total)} = \begin{cases} r_t^{(PnL)} + r_t^{(fee)}, & t < w \\ r_t^{(PnL)} + r_t^{(fee)} + r_t^{(sr)}, & for \quad t = w \\ r_t^{(PnL)} + r_t^{(fee)} + r_t^{(sr)}, & if \quad r_t^{(sr)} > r_{t-1}^{(sr)} \quad for \quad t > w \\ r_t^{(PnL)} + r_t^{(fee)} - r_t^{(sr)}, & if \quad r_t^{(sr)} < r_{t-1}^{(sr)} \quad for \quad t > w \end{cases} \tag{8}$$

The objective in Equation 8 is to achieve a higher Sharpe ratio in each step. For this reason, we compare the approximated Sharpe ratio from two consecutive steps. If we achieve a higher Sharpe ratio in the current step compared to the prior one, we enhance the agent by adding this value to the PnL reward, otherwise, we penalize the agent by subtracting the approximated Sharpe ratio.

## 3 Experimental Evaluation

The DRL setup is briefly described in this Section. In addition, the dataset used to run the simulation that interacts with the RL agents is presented. The impact of the proposed reward shaping is then evaluated and compared to the two reward schemes from Sections 2.2 and 2.3. The number of steps that an RL episode consists of is equal to 100. Since we have hourly candles, as is analytically described in Section 3.2, the agent is trained for approximately 4 days in each episode. The constant value $\alpha$ in Equation 6 is set to 0.01. Each experiment is executed 10 times, with each instance using a different random seed. The PnLs presented, were averaged throughout the 10 experiments as well as the annualized Sharpe ratios.

### 3.1 DRL setup

The RL agent is trained using the Policy Gradient (PG) approach. More specifically, Proximal Policy Optimization (PPO) [4]. In addition, the neural network architecture is Long-Short Term Memory (LSTM)-based [6]. Finally, the loss was proposed in [7] for estimating the advantage from the temporal difference residual, and the optimizer used is Rectified Adam (RAdam) and was introduced in [9]. It is worth noting that the proposed method is not restricted to the aforementioned architecture.

### 3.2 Dataset

The proposed method was tested on a financial dataset that included Crypto trading data of 14 currency pairs such as the BTC/BUSD, BTC/USDT, and ETH/USDT among others. The Open-High-Low-Close (OHLC) price level technique was used to subsample the market data [20], which reduces the raw data into 4 values. The dataset consists of minute price candles gathered by SpeedLab AG from 2017-08-17 up to 2022-02-12.

To utilize the dataset, the minute-price candles are resampled to hour candles. More specifically, these values are the open price or the first traded price of the set interval, the highest and lowest traded prices within the interval, and finally, the last price that a trade did occur during the interval, also referred to as the close price. The following features are inspired by [12] and were created using the OHLC values:

1. $x_{t,1} = \dfrac{p_c(t) - p_c(t-1)}{p_c(t-1)}$
2. $x_{t,2} = \dfrac{p_h(t) - p_h(t-1)}{p_h(t-1)}$
3. $x_{t,3} = \dfrac{p_l(t) - p_l(t-1)}{p_l(t-1)}$
4. $x_{t,4} = \dfrac{p_h(t) - p_c(t)}{p_c(t)}$
5. $x_{t,5} = \dfrac{p_c(t) - p_l(t)}{p_c(t)}$

where $p_c(t)$ is the close price that occurred during an interval at time $t$ and $p_h(t)$, $p_l(t)$ are the high and low prices within the same interval, respectively. Additionally, time-related features are created, including day, month, week, and year features. Note that $x_{t,1}$ denotes the return as specified in Section 2.2 in Equation 2. The described features are concatenated into a feature vector $\mathbf{x}_t$ for each time $t$.

The dataset was divided into two parts, a training set, and a test set, with the training set spanning from the start of each instrument's period to 2021-03-15, and the test set ranging from there to 2022-02-12. In total, the dataset contains 439.737 candles, where the train/test candles are 327.596 and 112.141 candles, respectively.

### 3.3 Annualized Sharpe ratio

The Sharpe ratio is used to compare the return of an investment with its risk and provides an insight that returns over a period of time may indicate volatility and risk. Let $\mathbf{z}$ be a vector with the hourly returns over the test period, since our dataset consists of hour candles as described in Section 3.2. When calculating the annualized Sharpe ratio using monthly returns, $\mathbf{z}$ is resampled to the frequency of 1 month and it is defined as:

$$sr^m_{ann} = \frac{E[\mathbf{z}_m]}{\sqrt{Var[\mathbf{z}_m]}} \times \sqrt{12} \tag{9}$$

where $E[\mathbf{z}_m]$, $\sqrt{Var[\mathbf{z}_m]}$ are the mean and the standard deviation of the resampled monthly returns, respectively. We multiply by the square root of 12 to annualize the Sharpe ratio. In the same manner, we calculate the annualized Sharpe ratio from the hourly returns. This time, there is no need for resampling since the returns are in the frequency of hours. It can be formulated as:

$$sr^h_{ann} = \frac{E[\mathbf{z}_h]}{\sqrt{Var[\mathbf{z}_h]}} \times \sqrt{8640} \tag{10}$$

where $E[\mathbf{z}_h]$ is the mean of the hourly returns and $\sqrt{Var[\mathbf{z}_h]}$ the standard deviation. In order to annualize the Sharpe ratio, we multiply by $\sqrt{8640}$ since there are approximately 8640 trading hours in a year for Crypto currencies.

### 3.4 Proposed reward evaluation

In Table 1, it is clearly shown from the annualized Sharpe ratio, that agents trained with the proposed reward scheme, outperform the baseline PnL-based reward and the PnL with the added Sharpe ratio since the greater a portfolio's Sharpe ratio, the better its risk-adjusted performance.

In addition, in Table 1, the annualized Sharpe ratio from hourly returns is presented. In practice, it is not a usual phenomenon since the volatility of the hourly returns is typically greater than the monthly returns. However, we consider that it is worth to be also calculated since in the proposed reward, we calculate the approximated Sharpe ratio from the hourly returns.

**Table 1.** Backtesting Annualized Sharpe Ratio.

| Reward type | $sr^m_{ann}$ | $sr^h_{ann}$ |
|---|---|---|
| PnL | 1.462 ± 0.055 | 2.374 ± 0.079 |
| PnL + Sharpe ratio | 1.499 ± 0.060 | 2.484 ± 0.090 |
| Proposed | **1.617 ± 0.056** | **2.641 ± 0.083** |

In Figure 1, the cumulative PnL is depicted, comparing the profits achieved from the three different reward schemes that were described in Sections 2.2, 2.3, and 2.4 respectively. The standard deviation of the PnL is also demonstrated for the three agents, to illustrate the statistical significance of the obtained results.

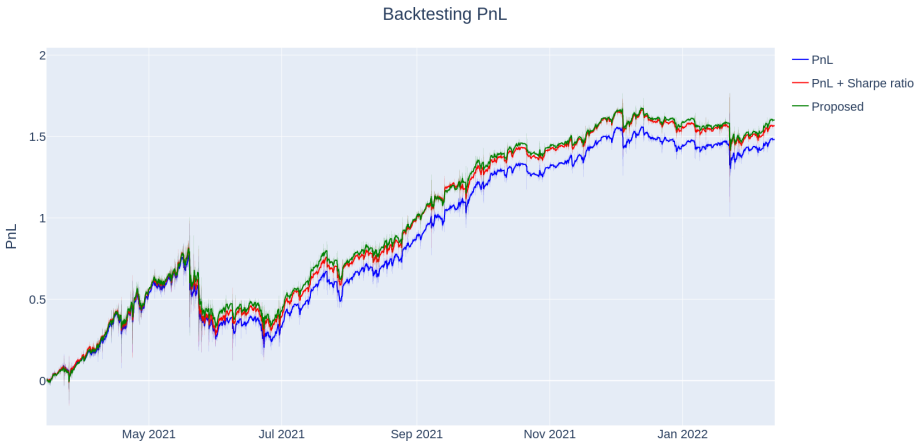


**Fig. 1.** Mean performance across 14 Cryptocurrency pairs of an agent trained with proposed reward vs. PnL vs. PnL + Sharpe ratio. The y-axis represents the cumulative Profit and Loss (PnL), while the x-axis represents the date.

## 4 Conclusion

In this work, a sharpe ratio-based reward shaping scheme was presented that was utilized in a Deep Reinforcement Learning (DRL) approach for training agents that are capable of trading profitably by boosting the risk-adjusted returns. The most notable contribution of this work is the introduction of a reward shaping scheme for decreasing the risk that often occurs in agents' trading decisions. The suggested scheme utilizes an approximation of the Sharpe ratio as an additional term to the Profit and Loss (PnL)-based reward, which motivates the agent to avoid trades that could incur losses. It was demonstrated through extensive experiments that using the proposed scheme can increase the profit and the overall portfolio performance with increased both PnL and the Sharpe ratio. To the best of our knowledge, this is the first attempt to use PnL and the Sharpe ratio as a reward function in financial trading with DRL.

## 5 Acknowledgement

This work has been co-financed by the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH - CREATE - INNOVATE (project code: T2EDK-02094).

## References

1. Y. Deng, F. Bao, Y. Kong, Z. Ren, and Q. Dai, "Deep direct reinforcement learning for financial signal representation and trading," IEEE Transactions on Neural Networks and Learning Systems, vol. 28, no. 3, pp. 653–664, 2017.
2. C. Y. Huang, "Financial trading as a game: A deep reinforcement learning approach," arXiv preprint arXiv:1807.02787, 2018.
3. Avraam Tsantekidis, Nikolaos Passalis, Anastasia Sotiria Toufa, Konstantinos Saitas-Zarkias, Stergios Chairistanidis, and Anastasios Tefas, "Price trailing for financial trading using deep reinforcement learning, "IEEE Trans. on Neural Networks and Learning Systems, 2020.
4. J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," arXiv preprint arXiv:1707.06347, 2017.
5. Sharpe, W. F., 'Mutual fund performance', Journal of Business, (1966), 119±138.
6. S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (8) (1997) 1735–1780.
7. J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," arXiv preprint arXiv:1506.02438, 2015.
8. Dewey, Daniel. "Reinforcement learning and the reward engineering principle." 2014 AAAI Spring Symposium Series. 2014.
9. Liu, L., Jiang, H., He, P., Chen, W., Liu, X., Gao, J. and Han, J. (2019). On the variance of the adaptive learning rate and beyond. arXiv preprint arXiv:1908.03265.

10. A. Hussein, E. Elyan, M. M. Gaber, and C. Jayne, "Deep reward shaping from demonstrations," in 2017 International Joint Conference on Neural Networks (IJCNN), 2017, pp. 510–517. [Online]. Available: https://academic.microsoft.com/paper/2596874484
11. M. Grzes, "Reward shaping in episodic reinforcement learning," ´adaptive agents and multi agents systems, pp. 565–573, 2017. [Online]. Available: https://academic.microsoft.com/paper/2620974420
12. Murphy, Technical analysis of the financial markets: A comprehensive guide to trading methods and applications. Penguin, 1999
13. Avraam Tsantekidis, Nikolaos Passalis, and Anastasios Tefas, "Diversity-driven knowledge distillation for financial trading using deep reinforcement learning," Neural Networks, vol. 140, pp. 193–202, 2021.
14. A. Tsantekidis, N. Passalis, A. Tefas, J. Kanniainen, M. Gabbouj, and A. Iosifidis, "Forecasting stock prices from the limit order book using convolutional neural networks," in Proceedings of the IEEE Conference on Business Informatics (CBI), 2017, pp. 7–12
15. A. Tsantekidis, N. Passalis, A. Tefas, J. Kanniainen, M. Gabbouj, and A. Iosifidis, "Using deep learning to detect price change indications in financial markets," in Proceedings of the European Signal Processing Conference (EUSIPCO), 2017, pp. 2511–2515.
16. A. Ntakaris, J. Kanniainen, M. Gabbouj, and A. Iosifidis, "Mid-price prediction based on machine learning methods with technical and quantitative indicators," SSRN, 2018.
17. D. T. Tran, A. Iosifidis, J. Kanniainen, and M. Gabbouj, "Temporal attention-augmented bilinear network for financial time-series data analysis," IEEE Transactions on Neural Networks and Learning Systems, 2018.
18. J. Moody and M. Saffell, "Learning to trade via direct reinforcement," IEEE Transactions on Neural Networks, vol. 12, no. 4, pp. 875–889, 2001.
19. J. E. Moody and M. Saffell, "Reinforcement learning for trading," in Proceedings of the Advances in Neural Information Processing Systems, 1999, pp. 917–923.
20. S. Nison, Japanese candlestick charting techniques: a contemporary guide to the ancient investment techniques of the Far East. Penguin, 2001.
21. Liang, Z., Chen, H., Zhu, J., Jiang, K., and Li, Y., "Adversarial Deep Reinforcement Learning in Portfolio Management", 2018. doi:10.48550/arXiv.1808.09940.