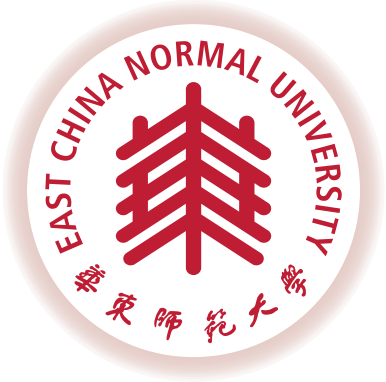
2023 届本科生学士学位论文　　学校代码：10269



華東師範大學

East China Normal University

本科生毕业论文

基于深度强化学习的加密货币投资组合管理

Cryptocurrency Portfolio Management Based on Deep Reinforcement Learning

姓　　名：周家纬

学　　号：10195501437

学　　院：数据科学与工程学院

专　　业：数据科学与大数据技术

指导教师：李翔

职　　称：研究员

2023 年 4 月

华东师范大学学位论文诚信承诺

本毕业论文是本人在导师指导下独立完成的，内容真实、可靠。本人在撰写毕业论文过程中不存在请人代写、抄袭或者剽窃他人作品、伪造或者篡改数据以及其他学位论文作假行为。

本人清楚知道学位论文作假行为将会导致行为人受到不授予/撤销学位、开除学籍等处理（处分）决定。本人如果被查证在撰写本毕业论文过程中存在学位论文作假行为，愿意接受学校依法作出的处理（处分）决定。

承诺人签名：  日期：2023 年 5 月 12 日

华东师范大学学位论文使用授权说明

本论文的研究成果归华东师范大学所有，本论文的研究内容不得以其它单位的名义发表。本学位论文作者和指导教师完全了解华东师范大学有关保留、使用学位论文的规定，即：学校有权保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅；本人授权华东师范大学可以将论文的全部或部分内容编入有关数据库进行检索、交流，可以采用影印、缩印或其他复制手段保存论文和汇编本学位论文。

保密的毕业论文（设计）在解密后应遵守此规定。

作者签名：  导师签名：  日期：2023 年 5 月 12 日

目录

基于深度强化学习的加密货币投资组合管理

摘要:

本文针对传统强化学习技术在波动市场中构建交易模型时易于出现控制力较弱和收益率不理想的问题，提出了一种基于 PPO（Proximal Policy Optimization）算法的加密货币投资组合管理模型，旨在提升超额收益。

首先，通过比对不同策略下的投资模型收益情况。结果显示，当采用技术指标观测和最近一期的 OHLCV（开盘价、最高价、最低价、收盘价、成交量）数据观测模型结合日线数据进行投资时，收益较为显著。相较于采用等权重买入与持有策略，投资模型在测试集和代表性代币构建的投资组合上，能够平均获得约 11% 的超额年化收益率。当智能体处理最近一期 OHLCV 数据时，其表现最佳，可以获得约 61% 的平均超额年化收益率。相反，如果采用最近五期的 OHLCV 观测或者小时线数据，模型的收益则较差。这主要因为五期的观测空间过大，导致模型输入端的特征解析能力不够；另外，小时线数据量较大，需要更长时间的模型训练才能得到收敛解。

其次，比较了不同策略下的模型风险控制情况。在添加了风险机制奖励的智能体中，发现其收益率在单边下行市场偏高。然而，在单边上行市场，其获益显著低于传统 pnl 奖励。在此，注意到无论是使用哪种策略，模型的最大回撤比例都与基线水平相似。这可能与参数设定有关。如果熔断系数进行调整，并对熔断后的行为加以控制，其最大回撤比例可能会有显著改善。

最后，我们比较了在不同系统上运行该算法的训练时间，发现使用 m2pro 芯片的训练速度是 2060s 显卡的 10 倍左右。说明在强化学习的设定下，显卡算力不是制约性能的关键，而是 cpu 性能和内存带宽。

总的来说，该模型行动表现提示这是一种指数增强型交易策略。在适合的条件下，相较于普通买入与持有交易而言，该模型能够提供一定的超额年化收益优势。

关键词：深度强化学习, 近端策略优化算法, 投资组合管理, 加密货币

Cryptocurrency Portfolio Management Based on Deep Reinforcement Learning

Abstract:

In this paper, we propose a cryptocurrency portfolio management model based on the PPO (Proximal Policy Optimization) algorithm, aiming to enhance the excess returns, in response to the problem that traditional reinforcement learning techniques are prone to weak control and unsatisfactory returns when constructing trading models in volatile markets.

First, by comparing the returns of the investment model under different strategies. The results show that the returns are more significant when the model is invested using technical indicator observations and the most recent period of OHLCV (Open, High, Low, Close, Volume) data observations combined with daily data. Compared to using an equal-weighted buy-and-hold strategy, the investment model is able to achieve an average excess annualized return of approximately 11% on a portfolio constructed from the test set and representative tokens. The agent performs best when it processes the most recent period of OHLCV data, and can achieve an average excess annualized return of approximately 61%. On the contrary, the model returns are poorer when the five most recent periods of OHLCV observations or hourly data are used. This is mainly because the observation space of five periods is too large, resulting in insufficient feature resolution at the input of the model; in addition, the larger volume of hourly line data requires a longer time for model training to obtain a convergent solution.

Second, the model risk control under different strategies was compared. With added risk mechanism reward, the return is found to be high in one-sided downside markets. However, in one-sided upward markets, its benefit is significantly lower than the traditional pnl reward. Here, it is noted that the maximum percentage retracement of the model is similar to the baseline level regardless of the strategy used. This may be related to the parameter settings. If the meltdown coefficients are adjusted and post-meltdown behavior is controlled, their maximum retracement ratios may be significantly improved.

Finally, we compared the training time of running the algorithm on different systems and found that the training speed of the mac using the m2pro chip was about 10 times the performance improvement of the 2060s graphics card, indicating that the graphics card arithmetic is not the key to constrain the performance in the reinforcement learning setting, but rather the cpu performance and memory bandwidth.

Overall, the model action performance suggests that this is an enhanced index trading strategy. Under suitable conditions, the model can provide some excess annualized return advantage compared to a normal buy-and-hold trade.

Keywords: Deep reinforcement learning, Proximal policy optimization (PPO), Portfolio management, Cryptocurrency

1、 绪论

1.1 研究背景和研究目的

量化交易是一种依赖数学和统计模型自动识别市场投资机会的交易模式。[1]随着人工智能时代各类高度优化的交易策略和快速堆积的盈利方式到来，量化交易越来越受推崇。在发达市场（如美国）和发展中市场（如中国），交易量分别占据了 70% 和 40% 以上。总体而言，量化交易研究主要存在两个主要方向：在金融领域，主要关注模型设计和金融理论解释，如著名的资本资产定价模型马科维茨投资组合理论[2]、CAPM[3]和 Fama-French 三因子模型[4]等都是一些具有代表性的例子；而另一方面，计算机科学方向则更擅长利用数据驱动的机器学习技术分析交易数据。目前，深度学习因其出色的数据解析性能而成为金融市场备受推崇的研究工具。[5][6]

强化学习是机器学习的一个新兴子领域，可以从不断试错中获得有价值经验。通过使用强化学习，我们可以持续优化特定任务的奖励函数，来训练形成近乎最优行为策略的智能体。随着强化学习和深度学习的深度融合，在过去十年里，我们已经见证了许多深度强化学习技术模型在诸如围棋、视频游戏、机器人等领域取得的重要成果。深度强化学习方法也在许多量化交易任务上取得了较为明显的突破，如算法交易（AT）、投资组合管理（PM）、订单执行（OE）和市场做市（MM），如图1-1所示。

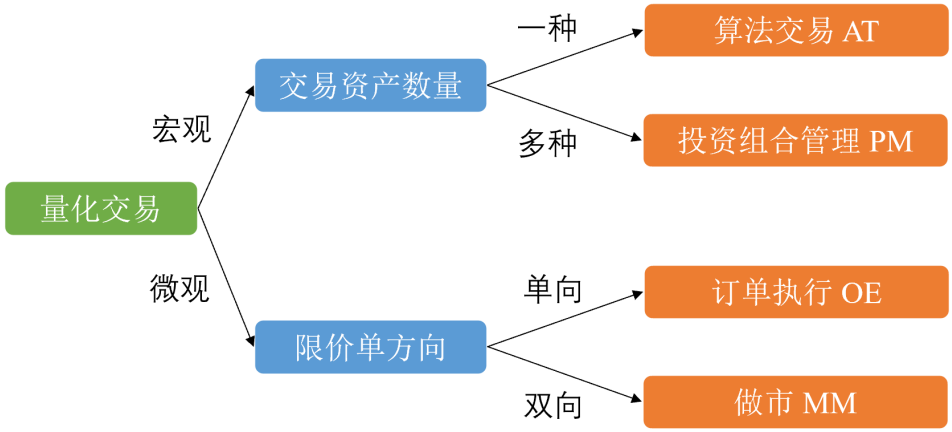


图 1-1 量化交易分类

利用深度强化学习技术承担量化交易任务是一个充满挑战和富有前景的研究方向，近几年中已涌现出大量基于深度强化学习的高频交易研究[7][8][9]。本文主要讨论投资组合管理领域的一些探索工作。

Deng 等[10]表明 DRL 可以获得超过传统方法更多的利润，该文章使用了模糊神经网络进行特征建模。Jiang 等[11]使用 EIIE 拓扑结构，包含资产组合记忆单元（PVM），在线随机批量学习（OSBL），在这个框架下可以对接使用 CNN 和 LSTM 进行特征数据提取。Lucarelli 等[12]通过深度 Q 网络建模来进行加密货币交易。Liu 等[13]构建了

一个较为成熟的“即插即用”深度强化学习金融交易库 FinRL，极大的促进了该领域的发展，该团队还在强化模型进行集成增强鲁棒性[14]，以及可解释性[15]上有较大突破。最近，TradeMaster 在 GitHub 上线。其项目设计与 FinRL 的目标类似，但在可视化[16]方面有了新的发展。还有更多工作在多模态数据（融合新闻情感提取[17-18]）上进行实验也取得良好效果。

相对于国外在高频交易领域较高发展水平，国内在该领域尚处于起步阶段，目前相关论文较少[19-21]，仅有少量使用深度强化学习进行资产组合管理的探索，但也大部分集中在中国股票和期货市场。目前还没有在加密货币交易中发现使用深度强化学习进行研究的报道。

针对国内深度强化学习在加密货币交易领域探索的不足，研究一种基于深度强化学习对加密货币进行投资组合管理的方式是本文的主要目的。

1.2 研究现状

长期以来，金融界一直对交易时间序列分析感兴趣，如果在交易市场中即时发现较强交易信号，就有可能基于此信号的交易获得短期收益。而机器学习与深度学习在筛选交易信号方面比传统交易员人工方式更有效率。此外，在任何一个金融市场进行交易，本质都是一个多方博弈过程，任何突发消息都有可能使资产发生较强价格波动，而普通交易通过各种集聚效应也会放大成一个较强的资产收益。因此，如何在低信噪比金融市场中设计出一种稳定的优势交易策略，是现代量化交易的核心问题。

近期研究中有大量使用深度学习预测资产价格[5]的研究报告，这类预测算法构成主要取决于用于训练的数据质量，以及模型对于数据的整合能力，但也局限于应用在比较短期的价格预测，而且，由于金融市场信噪比很低，容易受到外界突发信息影响[22]。因此，对于中低频的交易预测，其预测价格一般是不被置信，即使知道短期结果，也很难捕捉合适的交易时机并形成一定规模的交易量，最终还是需要依赖大量人工规则构建合适的交易策略。作为机器学习的一部分，强化学习更加关注在复杂场景中，通过算法自我最大化奖励方式构建智能体，并自行生成较优行动策略[23]。智能体能够直接在历史资料中自我学习并生成对应策略，同时用获得的经验去规避后期探索中所面临的风险。由此可见，融合深度学习与强化学习的深度强化学习模型[24]在面对复杂序列的决策上可以取得较为显著优势，因此，目前深度强化学习应用于量化投资已经成为一种主要趋势[7]。

目前运用深度强化学习技术实现高频交易还需要解决以下几个问题：第一，如何筛选初始投资组合，以确定其具有较高的探索和投资价值。虽然确定交易标的不会直接影响模型训练，但投资目的就是为获得最大化收益，因此如何在交易前确定更具升值潜力的投资组合是极为重要的。第二，如何选择数据供智能体预学习，或选择何种

粒度的数据以及如何使用时序网络进行信息提取，对智能体的构建具有关键作用。如果在状态空间中加入过去时间步，或直接使用循环神经网络等时间序列，那么马尔可夫决策过程 MDP 问题就会变成部分可观测马尔可夫决策过程的 POMDP 问题[25]，两者在建模和模型训练上表现会有所差异。第三，智能体的动作和奖励如何设计，不同的设计方式对于深度强化学习模型最后呈现的智能体水平也会有一定的差距。第四，如何考虑交易费与滑点[26]带来的交易损失。我们需要加入这些交易摩擦来使得交易更加贴近真实场景。第五，如何加入风险控制边际，使模型能够形成趋于稳健的交易策略。当出现交易量突然放大，或交易价格出现异常波动时，模型需要加入一些风险控制因子来规避交易损失。综上所述，运用深度强化学习进行线上交易存在大量值得讨论的问题。

1.3 研究内容和技术路线

本文以加密货币作为交易标的，讨论如何将深度强化学习应用于投资组合管理。选择加密货币的主要原因：首先，目前整个加密货币交易市值在一万亿美金[1]左右，体量较大，而且其流动性好，交易手续费低且任何时刻均可交易，因此适合高频交易；其次，加密货币相比传统的股票等有价证券可以进行更细粒度分割，不以“手”作为交易单位[2]，因此可供选择的交易动作方式比较多，对于小资金的用户更加友好，也可以避免因为滑点导致资产损失。最后，加密货币市场价格波动大，历史上也出现过多次闪崩情况，能够更好模拟出针对异常事件的处理策略。

本文主要技术路线如下：设计并实现从加密交易数据获取、预处理、模型选择、训练和回测的全流程活动。针对数据获取与预处理部分，使用 yfinance 以及 binance 进行数据获取（日线数据和小时级别数据），并使用 stockstats 处理技术指标；算法部分采用了 PPO 算法技术，并且设计了多种不同的智能体观测，对每一种观测都进行详尽测试，同时采用 gym 框架针对模拟器设计进行处理。本文还探讨不同的奖励是否能有效提升模型性能，最终在模型评估部分实现模拟量化交易的性能对比。

1.4 章节安排

本文共分为六个章节，内容如下所示：

- 第一章，**绪论**：介绍选题背景与意义以及与本文相关的文献。
- 第二章，**相关工作**：本文所需要的相关深度强化学习算法简要介绍，以及在自动化交易领域的一些技术进展。
- 第三章，**实验方法**：本文采用的主要实验方法以及相关工程实现细节。

1 目前加密货币总市值为人民币 8.5 万亿元左右 (截止时间：2023.4.6) 数据来源：https://www.coingecko.com/zh

2 具体的交易粒度可以参考币安交易所：https://www.binance.com/en/trade-rule

- 第四章，**实验结果与分析**：基于深度强化学习技术构建的算法模型进行模拟交易，对初步结果进行分析。
- 第五章，**总结**：归纳本文的主要成果并给出结论。
- 第六章，**不足与展望**：讨论本项课题主要不足之处和将来可以改进的方法，以及未来可能的研究方向。

2、 相关工作

回顾基本的深度强化学习算法，系统整理强化学习在自动化交易领域的相关研究，更好了解目前研究现状，为实现本文做好技术准备。

2.1 深度强化学习

深度学习内容可以参考一些相关教材[27][28]。这里默认读者对与深度学习有基本认识，包括但不限于常见模型：如全连接神经网络、卷积神经网络、循环神经网络、基于自注意力的 Transformer、常见损失函数与激活函数、常见优化器、常见正则化方式以及常见超参数优化方式等基本内容，这些都是深度学习领域的基础。

本节我们不讨论基于模型的深度强化学习，所用均是**无模型的深度强化学习算法**，价值拟合和策略梯度是本节关注的重点。基于价值的优化主要介绍基于梯度的方法，如使用深度神经网络的深度 Q 网络。基于策略的优化主要分为确定性策略梯度和随机性策略梯度，我们主要介绍随机性策略梯度算法。基于价值和基于策略两种优化方法结合，产生了著名的 Actor-Critic 结构，并进一步衍生出大量高级深度强化学习算法，我们在这里介绍其中最有名的一个算法之一：PPO。

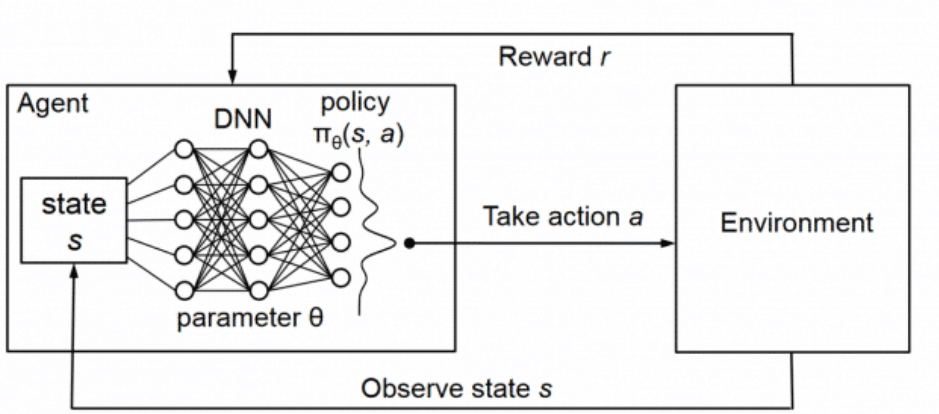


图 2-1　深度强化学习

此外，策略是否在线对于数据训练也会产生较大影响。在线策略是一种在直接交互环境中评估并提升数据的方法，而离线策略的数据生成不需依靠直接交互环境。这表明在线策略要求智能体与环境交互和提升的条件必须相同，而离线策略不需要遵循这个约束，后者可以利用其他智能体与环境交互得到的数据来提升自己的策略，但前者生成的数据可靠性更高，更能反应现实趋势。由于在线策略使用的样本复杂度很高，在实际使用中有较大限制，因此在策略选择上需要作出一定的权衡。

2.1.1 基于价值优化

为了将基于价值的强化学习应用到大规模的任务上，我们可以使用神经网络作为函数拟合器去拟合价值函数（参数化价值函数），其优势不仅包括可以进行大规模任务扩展，也便于在连续状态空间中进行从所见状态到未见过状态的泛化，而且可以减少人为设计特征来表示状态的需要。通常而言，我们的优化目标设置成价值函数估计，采用动作价值函数估计和对应真实函数之间的均方误差进行优化。我们以动作价值函数为例，优化目标如下：

$$J(w)=\mathbb{E}_{\pi}\left[\left(Q^{\pi}(s,a;w)-q_{\pi}(s,a)\right)^{2}\right] \tag{2.1}$$

其梯度如下

$$\Delta w=\alpha\left(Q^{\pi}(s,a;w)-q_{\pi}(s,a)\right)\nabla_{w}Q^{\pi}(s,a;w) \tag{2.2}$$

由于我们无法直接获得动作价值函数，因此需要进行估计

- **MC** 的方式

$$\Delta w_{t}=\alpha\left(Q^{\pi}\left(S_{t},A_{t};w_{t}\right)-G_{t+1}\right)\nabla_{w_{t}}Q^{\pi}\left(S_{t},A_{t};w_{t}\right) \tag{2.3}$$

- **TD**(0) 的方式

$$\Delta w_{t}=\alpha\left(Q^{\pi}\left(S_{t},A_{t};w_{t}\right)-\left(R_{t+1}+\gamma Q_{\pi}\left(S_{t+1},A_{t+1};w_{t}\right)\right)\nabla_{w_{t}}Q^{\pi}\left(S_{t},A_{t};w_{t}\right)\right) \tag{2.4}$$

- **TD**(λ) 的方式

$$\Delta w_{t}=\alpha\left(Q^{\pi}\left(S_{t},A_{t};w_{t}\right)-G_{t}^{\lambda}\right)\nabla_{w_{t}}Q^{\pi}\left(S_{t},A_{t};w_{t}\right) \tag{2.5}$$

深度 Q 学习就是使用 TD(0) 的方式进行更新。当然，还需要一些技巧使得训练过程变得稳定。

- **回放缓存**。目的：降低样本相关性与样本复杂度。实现上，通过构造 FIFO 队列。每个时间步需要将智能体经验 $\left(S_{t},A_{t},R_{t},S_{t+1}\right)$ 存入回访缓存中，然后从该缓存中均匀采样小批量样本用以 Q-learning 的更新。
- **目标网络**。目的：使得训练稳定。实现上，每常数步需要对目标网络进行同步，同步可以直接复制也可以使用指数移动平均的方式。
- **目标值修正**。目的：防止目标过高估计。实现上，同样是使用目标网络，外层就用目标网络替换。

$$R_{t}+\gamma\hat{Q}\left(S_{t+1},\arg\max_{a}Q\left(S_{t+1},a;\theta\right);\hat{\theta}\right) \tag{2.6}$$

- **引入优势函数**。目的：动作与状态解耦 $Q^{\pi}(s,a)=V^{\pi}(s)+A^{\pi}(s,a)$。实现上，需要确保 Q 值能够唯一对应状态和动作优势，因为多了一个自由度。理论上需要减去 $\max_{a'} A\left(s,a';\theta,\theta_a\right)$ 以保证贝尔曼最优方程。实际使用期望效果更好。

$$Q\left(s,a;\theta,\theta_v,\theta_a\right)=V\left(s;\theta,\theta_v\right)+\left(A\left(s,a;\theta,\theta_a\right)-\frac{1}{|\mathcal{A}|}\sum_{a'}A\left(s,a';\theta,\theta_a\right)\right) \tag{2.7}$$

- **优先经验回放**。目的：更好的经验回放策略。实现上，首先计算状态转移的采样概率。

$$P(i)=\frac{p_i^{\alpha}}{\sum_k p_k^{\alpha}} \tag{2.8}$$

其中：$p_i=\frac{1}{\text{rank}(i)}$，rank($i$) 是基于 $\left|\delta_i\right|$ 的等级评定，其中 $\left|\delta_i\right|$ 是状态转移 i 的 TD 误差。α 是一个指数超参数。最后的权重则是通过 β 超参退火计算得到的。

$$w_i=(NP(i))^{-\beta} \tag{2.9}$$

其中：N 指回放缓存的容量。对于 β 超参退火的方式可以自行设置。

2.1.2 基于策略优化

除了可以参数化价值函数，当然也可以直接参数化策略。分确定性和随机性策略两类：$A_t=\mu_\theta\left(S_t\right)$ 和 $A_t\sim\pi_\theta\left(\cdot\mid S_t\right)$。随机性策略有很多形式，比如常见的有类别分布和高斯分布等。一般而言，我们常用梯度对策略进行优化。基于梯度的优化方法是使用在期望回报上的梯度估计来进行梯度上升以改进策略，而这个期望回报是从采样轨迹中得到的。这里我们把关于策略参数的梯度叫作策略梯度。在这里，我们不考虑确定性策略梯度方法，只关注随机性策略梯度方法。

$$\Delta\theta=\alpha\nabla_\theta J\left(\pi_\theta\right) \tag{2.10}$$

给出**随机性策略梯度定理**，证明较为复杂，可以参考[29]的文章。下式为定理内容：

$$\begin{aligned}\nabla_\theta J\left(\pi_\theta\right)&=\mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{\mathrm{T}}\nabla_\theta\left(\log\pi_\theta\left(A_t\mid S_t\right)\right)Q^{\pi_\theta}\left(S_t,A_t\right)\right]\\&=\mathbb{E}_{S_t\sim\rho^{\pi},A_t\sim\pi_\theta}\left[\nabla_\theta\left(\log\pi_\theta\left(A_t\mid S_t\right)\right)Q^{\pi_\theta}\left(S_t,A_t\right)\right]\end{aligned} \tag{2.11}$$

其中 $S_t\sim\rho^{\pi}$ 代表折扣状态分布 $\rho^{\pi}\left(s'\right):=\int_{\mathcal{S}}\sum_{t=0}^{\mathrm{T}}\gamma^{t-1}\rho_0(s)p\left(s'\mid s,t,\pi\right)\mathrm{d}s$，$p\left(s'\mid s,t,\pi\right)$ 代表在 t 时刻 $s\to s'$ 的概率。举例：REINFORCE 算法就是最简单的随机性策略梯度。

$$\nabla J(\theta)=\mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t'=0}^{\infty}\nabla_\theta\log\pi_\theta\left(A_{t'}\mid S_{t'}\right)\left(\sum_{t=t'}^{\infty}\gamma^{t-t'}R_t-b\left(S_{t'}\right)\right)\right] \tag{2.12}$$

加入了一个基准函数降低方差，一般可以选择状态价值函数。在此基础上，借鉴之前提到过的 TD(λ)，使用 GAE[30]作为优势函数可以提高模型性能

$$\hat{A}_t^{GAE(\gamma,\lambda)}=\sum_{l=0}^{\infty}(\gamma\lambda)^l\delta_{t+1}^V=\sum_{l=0}^{\infty}(\gamma\lambda)^l\left[R_{t+l}+\gamma V\left(s_{t+l+1}\right)-V\left(s_{t+l}\right)\right] \tag{2.13}$$

特殊解就是时间差分和蒙特卡洛

$$\begin{aligned}\text{GAE}(\gamma,0):\quad \hat{A}_t&:=\delta_t &&=R_t+\gamma V\left(s_{t+1}\right)-V\left(s_t\right)\\ \text{GAE}(\gamma,1):\quad \hat{A}_t&:=\textstyle\sum_{l=0}^{\infty}\gamma^l\delta_{t+l} &&=\textstyle\sum_{l=0}^{\infty}\gamma^lR_{t+l}-V\left(s_t\right)\end{aligned} \tag{2.14}$$

这在之后的模型比如 TRPO、PPO 等都有用到。

2.1.3 结合基于策略和基于价值的方法

使用 REINFORCE 常常面临着无法准确的估计基准函数，因此设计出了一套既能估计动作价值，也能使用梯度优化的方式进行更新的结构 Actor-Critic。我们从 Actor-Critic 开始，介绍一下常用的经典算法。

2.1.3.1 Actor-Critic

把 REINFORCE 中的误差项通过时间差分取代

$$R_i+\gamma V^\pi\left(S_{i+1}\right)-V^\pi\left(S_i\right) \tag{2.15}$$

一般而言，我们可以采用 L 步时间差分误差。通过最小化平方误差可以更新批判者 Critic，而行动者 Actor 的更新实际上其实与 REINFORCE 梯度的更新类似。下面给出具体的更新公式。

- Critic 函数 $V_\psi^{\pi_\theta}$ 的期望回报梯度计算

$$J_{V_\psi^{\pi_\theta}}(\psi)=\frac{1}{2}\left(\sum_{t=i}^{i+L-1}\gamma^{t-i}R_t+\gamma^LV_\psi^{\pi_\theta}\left(S'\right)-V_\psi^{\pi_\theta}\left(S_i\right)\right)^2 \tag{2.16}$$

$$\nabla J_{V_\psi^{\pi_\theta}}(\psi)=\left(V_\psi^{\pi_\theta}\left(S_i\right)-\sum_{t=i}^{i+L-1}\gamma^{t-i}R_t-\gamma^LV_\psi^{\pi_\theta}\left(S'\right)\right)\nabla V_\psi^{\pi_\theta}\left(S_i\right). \tag{2.17}$$

符号说明：ψ 是 Critic 参数，S' 是智能体 L 步后到达的状态。

- Critic 参数更新（梯度下降）

$$\psi \leftarrow \psi - \eta_{\psi} \nabla J_{V_{\psi}^{\pi_\theta}}(\psi) \tag{2.18}$$

符号说明：η_{ψ} 是学习步长。

- Actor 利用 Critic 优化的基线来进行梯度计算

$$\nabla J(\theta) = \mathbb{E}_{\tau,\theta}\left[\sum_{i=0}^{\infty} \nabla \log \pi_\theta\left(A_i \mid S_i\right)\left(\sum_{t=i}^{i+L-1} \gamma^{t-i} R_t + \gamma^L V_{\psi}^{\pi_\theta}\left(S'\right) - V_{\psi}^{\pi_\theta}\left(S_i\right)\right)\right] \tag{2.19}$$

- Actor 参数更新（梯度上升）

$$\theta = \theta + \eta_{\theta} \nabla J_{\pi_\theta}(\theta) \tag{2.20}$$

对于 A2C 和 A3C[31]对于 AC 的提升主要是在**并行计算**上。在 A2C，Master 节点协调多个 Worker 与环境进行交互，当 Master 收集到 Worker 的所有梯度信息之后对全局模型进行更新，并将更新好的模型给到 Worker 节点。A3C 则是异步更新，只要 Worker 完成之后就对全局模型进行更新，这样做能够提高计算效率。

2.1.3.2 一个基于 Actor-Critic 框架的模型，PPO

由于论文中主要用到的 Actor-Critic 模型是 PPO，所以我们这部分仅介绍这个模型。首先需要介绍一下信赖域策略优化 TRPO[32]。策略梯度的一个问题是步长这个超参数如何选择的问题，太大容易出现性能突降、太小则学习进度太慢。而策略梯度是在线学习的，也就是说梯度学习的不好还会影响到数据样本的质量。因此诞生出了信赖域策略优化，它能够更好地处理步长的问题。

首先需要给出引理

$$J\left(\theta'\right) = J(\theta) + \mathbb{E}_{\tau \sim \pi_\theta'}\left[\sum_{t=0}^{\infty} \gamma^t A^{\pi_\theta}\left(S_t, A_t\right)\right] \tag{2.21}$$

其中：π_θ' 是 π_θ 的一个提升，优势函数 $A^{\pi_\theta}(s,a) = Q^{\pi_\theta}(s,a) - V_{\psi}^{\pi_\theta}(s)$，原期望回报 $J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t R\left(S_t, A_t\right)\right]$。因此我们需要去最大化后一项期望使得新的期望回报最大化。由于我们无法直接优化这个期望（因为是通过 π_θ' 采样得到的），因此需要使用重要性采样。这里可以假设两者的状态度量相差很小，于是有

$$\mathscr{L}_{\pi_\theta}\left(\pi_\theta'\right) = \mathbb{E}_{\tau \sim \pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t \frac{\pi_\theta'\left(A_t \mid S_t\right)}{\pi_\theta\left(A_t \mid S_t\right)} A^{\pi_\theta}\left(S_t, A_t\right)\right] \tag{2.22}$$

实际上，这个误差可以被策略间的 KL 散度 $D_{\mathrm{KL}}^{\max}\left(\pi_\theta \| \pi_\theta'\right)$ 进行控制

$$\left|J\left(\theta'\right)-J(\theta)-\mathscr{L}_{\pi_\theta}\left(\pi'_\theta\right)\right| \leqslant C D_{\mathrm{KL}}^{\max}\left(\pi_\theta \| \pi'_\theta\right). \tag{2.23}$$

这样，如果策略相差很小的话，刚刚的假设成立。因此可以通过 $\mathscr{L}_{\pi_\theta}\left(\pi'_\theta\right)$ 这个代理误差来优化

$$\begin{aligned} \max_{\pi'_\theta} \quad & \mathscr{L}_{\pi_\theta}\left(\pi'_\theta\right) \\ \text{s.t.} \quad & \mathbb{E}_{s\sim\rho_{\pi_\theta}}\left[D_{\mathrm{KL}}\left(\pi_\theta \| \pi'_\theta\right)\right] \leqslant \delta. \end{aligned} \tag{2.24}$$

这也是 TRPO 的优化目标。一般使用二阶近似进行优化求解，但这样做计算量很大。我们不再讨论其数值计算的过程，直接转向其后继版本 PPO[33]，也是我们本课题的核心算法。其直接优化其拉格朗日目标

$$\max_{\pi'_\theta} \mathscr{L}_{\pi_\theta}\left(\pi'_\theta\right)-\lambda \mathbb{E}_{s\sim\rho_{\pi_\theta}}\left[D_{\mathrm{KL}}\left(\pi_\theta \| \pi'_\theta\right)\right] \tag{2.25}$$

但这里出现了一个超参 λ，这个超参依赖于 KL 散度。因此 PPO 提出一个比较好的替代方法，记 $\ell_t\left(\theta'\right)$ 表示前后两个策略的比值

$$\ell_t\left(\theta'\right)=\frac{\pi'_\theta\left(A_t \mid S_t\right)}{\pi_\theta\left(A_t \mid S_t\right)} \tag{2.26}$$

于是计算

$$\mathscr{L}^{\text{PPO-Clip}}\left(\pi'_\theta\right)=\mathbb{E}_{\pi_\theta}\left[\min\left(\ell_t\left(\theta'\right) A^{\pi_\theta}\left(S_t, A_t\right), \operatorname{clip} A^{\pi_\theta}\left(S_t, A_t\right)\right)\right] \tag{2.27}$$

其中剪切函数 clip 定义如下

$$\text{clip}=\operatorname{clip}\left(\ell_t\left(\theta'\right), 1-\epsilon, 1+\epsilon\right) \tag{2.28}$$

其将误差的截断在一个小区间内。通过这样做避免计算 KL 散度，且效果在实验中表现较好。因此，我们依此方法展示 PPO 实际更新过程：

- Actor：梯度上升

$$\theta_{k+1}=\arg\max_\theta \frac{1}{\left|\mathscr{D}_k\right| T} \sum_{\tau\in D_k} \sum_{t=0}^{\mathrm{T}} \mathscr{L}^{\text{PPO-Clip}} \tag{2.29}$$

- Critic：梯度下降

$$\phi_{k+1}=\arg\min_\phi \frac{1}{\left|\mathscr{D}_k\right| T} \sum_{\tau\in\mathscr{D}_k} \sum_{t=0}^{\mathrm{T}}\left(V_\phi\left(S_t\right)-\hat{G}_t\right)^2 \tag{2.30}$$

其中：执行策略为 π_{θ_k}，轨迹集合 $\mathscr{D}_k=\left\{\tau_i\right\}$，$T$ 步奖励 $\hat{G}_t$。

2.2 交易数据处理

2.2.1 数据表征

本文主体是不同时间粒度的 OHLCV，通常表示一天的开盘价、最高价、最低价、收盘价和成交量。对于更高粒度的数据而言，需要由 LOB 聚合而成的，代表某个时间段的第一笔成交价格、最高成交价格、最低成交价格、最后一笔成交价格、当前时间段的成交量。有很多方法可以对时间序列进行特征表示。在此部分将详细讨论目前金融市场使用最广泛的编码方式，但本课题不包含对多模态数据的讨论，例如使用新闻和投资情绪等作为模型输入的一部分。

有下列几种处理原始数据的方式：

- **标准化**：由于不同资产的价格差异很大，我们通常需要在输入模型前对每一种资产的价格进行标准化处理。在金融市场经常使用回报比率 $\frac{v_t}{v_{t-1}} - 1$ 来进行一个缩放作为资产价格标准化处理。
- **动态**：将最后几个时间步连接为成为每个时间点的状态是强化学习中常见的一个技巧，通常窗口大小 $\boldsymbol{T}$ 设置成 $3, 5, 10$ 等值。
- **降噪**：一般使用各类技术指标（被动）或者 auto-encoder 之类的降维模型[34]（主动）降低数据噪声。常见的技术指标将在下一节进行介绍。
- **离散化**[35]：目的也是降低波动性，增强有效信号，可以在数据标准化后，设定阈值来实现功能。事实上，这压缩了时间序列并将其转换为基于价格-事件的序列，是一种简化时间序列的方法，使其与交易更相关，并消除了原始数据中存在的一些噪音。

2.2.2 技术指标

通常而言，技术指标实际上就是这样一种映射关系：

$$T_{i,t} = f_i\left(\begin{bmatrix} p_{1,t-N+1} & p_{1,t-N+2} & \cdots & p_{1,t} \\ p_{2,t-N+1} & p_{2,t-N+2} & \cdots & p_{2,t} \\ \vdots & \vdots & \ddots & \vdots \\ p_{m,t-N+1} & p_{m,t-N+2} & \cdots & p_{m,t} \end{bmatrix}\right) \tag{2.31}$$

其中：N 为窗口大小，t 为当前时间步，i 代表某一个指标，m 代表投资组合大小。通常而言，对于单资产的技术指标，$m = 1$。在表2-1，我们列举最常见的技术指标，并在后文中分条列举其功能与计算公式。

技术指标	中文名称
SMA	简单移动平均
EMA[36]	指数移动平均
MACD	指数平滑异同移动平均线
RSI[37]	相对强弱指标
turbulance[38]	波动率

表 2-1　技术指标对照表

简单移动平均

- 主要功能：通常用于时间序列数据，以平滑短期波动并突出长期趋势或周期。
- 数学解释：作为一个低通滤波器来过滤掉高频信号部分。
- 公式如下：

$$\begin{aligned} \mathrm{SMA}_k &= \frac{p_{n-k+1} + p_{n-k+2} + \cdots + p_n}{k} \\ &= \frac{1}{k} \sum_{i=n-k+1}^{n} p_i \end{aligned} \tag{2.32}$$

指数移动平均

- 主要功能：相较于简单移动平均增加了权重，权重按照时间指数衰减。
- 数学解释：作为一个低通滤波器来过滤掉高频信号部分。
- 公式如下：

$$\begin{aligned} \mathrm{EMA}_{\mathrm{Today}} = &\left(\mathrm{Value}_{\mathrm{Today}} * \left(\frac{\mathrm{Smoothing}}{1 + \mathrm{Days}} \right) \right) \\ &+ \mathrm{EMA}_{\mathrm{Yesterday}} * \left(1 - \left(\frac{\mathrm{Smoothing}}{1 + \mathrm{Days}} \right) \right) \end{aligned} \tag{2.33}$$

- 通常 Smoothing 可以设置成 2。

指数平滑异同移动平均线

- 主要功能：揭示价格趋势的强度、方向、动量和持续时间的变化。
- 公式如下：
  - 首先计算指数移动平均差值

$$\mathrm{MACD}_{\mathrm{line}} = \mathrm{EMA}_{12} - \mathrm{EMA}_{26} \tag{2.34}$$

  - 其次计算信号线

$$\text{signal line} = \mathrm{EMA}_9(\mathrm{MACD}_{\mathrm{line}}) \tag{2.35}$$

相对强弱指标[39]

- 主要功能：探测是否存在极端变化。
- 公式如下：
  - 首先需要计算：

$$\begin{aligned} D_t &= \frac{1}{N} \cdot \max\left(0, p_{t-1} - p_t\right) + \left(1 - \frac{1}{N}\right) \cdot D_{t-1} \\ U_t &= \frac{1}{N} \cdot \max\left(0, p_t - p_{t-1}\right) + \left(1 - \frac{1}{N}\right) \cdot U_{t-1} \end{aligned} \tag{2.36}$$

  - 其中：D_t 表示平均下行变化，U_t 表示平均上行变化，N 代表时间窗口大小。
  - 其次需要计算：

$$\mathrm{RSI}_t = 100\left(1 - \frac{D_t}{D_t + U_t}\right) \tag{2.37}$$

短期波动率

- 主要功能：探测是否存在极端变化。
- 数学解释：短期波动率可以用马氏距离进行建模。
- 公式如下：

$$d_t = \left(\mathbf{y_t} - \boldsymbol{\mu}\right)' \boldsymbol{\Sigma}^{-1} \left(\mathbf{y_t} - \boldsymbol{\mu}\right) \tag{2.38}$$

- 其中：$\mathbf{y_t}$ 代表 t 时刻的收益，$\boldsymbol{\mu}$ 代表采样阶段的历史收益平均值，$\boldsymbol{\Sigma}$ 代表采样阶段的历史收益的协方差矩阵。向量均为列向量，矩阵大小为 $n \times n$（n 代表组合内资产个数）。

2.2.3 风险评估指标

风险评估指标是强化学习系统奖励部分的重要组成要件，也是评估量化交易系统效果的重要组成部分。这里列举最常用的风险评估方式。

2.2.3.1 夏普比率与其变体

夏普比率[40]衡量一项投资在对其调整风险后，相对于无风险资产的表现。计算公式由下式给出：

$$S_a = \frac{E\left[R_a - R_b\right]}{\sqrt{\mathrm{var}\left[R_a - R_b\right]}} \tag{2.39}$$

其中：R_a 是目标组合的收益，R_b 是无风险资产的收益。分母则是超额部分的标准差。

索提诺比率[41]则是在计算风险波动率时替换成了下行风险波动的标准差

$$S = \frac{R - T}{DR} \tag{2.40}$$

其中：R 是目标组合的收益的平均收益，T 时目标收益（同时也被称为 MAR）。DR 为下行风险，如下所示。

$$DR = \sqrt{\int_{-\infty}^{T} (T - r)^2 f(r) dr} \tag{2.41}$$

$f(r)$ 为收益的概率密度函数。

2.2.3.2 在险价值

在险价值[42]描述了在市场处于正常波动状态下，对应于给定的置信度水平，投资组合或资产组合在未来特定的一段时间内所遭受的最大可能损失值，计算公式由下式给出：

$$\mathrm{VaR}_{\alpha}(X) = -\inf\left\{x \in \mathbb{R} : F_X(x) > \alpha\right\} = F_Y^{-1}(1 - \alpha) \tag{2.42}$$

其中：$F_X(x)$ 代表超额部分的累积分布函数。$\mathrm{VaR}_{\alpha}(X)$ 是在 α 水平下最小的 y，使得 $Y := -X$ 不超过 y 的概率至少为 $1 - \alpha$。也就是说，$\mathrm{VaR}_{\alpha}(X)$ 是 Y 的 $1 - \alpha$ 分位数。但很显然分布函数在现实中很难获得，只能靠采样近似。

2.2.3.3 最大回撤

最大回撤[43]是衡量下行风险的指标。它表示某个时期资产组合的价值从极大值到后续最小值的最大跌幅，计算公式由下式给出：

$$\mathrm{MDD}(T) = \max_{\tau \in (0,T)} D(\tau) = \max_{\tau \in (0,T)} \left[\max_{t \in (0,\tau)} X(t) - X(\tau) \right] \tag{2.43}$$

其中：T 代表目标时间，$X(t)$ 代表投资组合净值，$D(t)$ 代表在 t 时刻的最大跌幅。

2.3 量化交易过程

2.3.1 智能体与交易环境

智能体需要在市场环境中不断探索以获得较好的交易经验。本课题总结了常见的智能体模型、动作选择以及奖励反馈，以上需要在实验前预先设定。

2.3.1.1 环境设定

交易环境的数据来源主要还是依托在 OHLCV 原始数据以及一些技术指标。环境需要处理智能体的动作，对智能体所处的状态进行更新。由于每种动作代表的含义都有所不同，环境的处理流程也会有所差别。此外，环境也需要给予智能体一个奖励值，设定方式有很多，常见设定方式将在奖励设置部分进行介绍。

2.3.1.2 动作形式

动作主要可以分成离散形式和连续形式两大类，每一大类都有对应的模型可以解决。

- **离散型**：离散动作空间常使用 DQN 类模型，其对应的动作含义一般分为两类，一类交易动作仅包含看多，另一类则包含看空。而对应的调仓空间则同样可以分为两类，一类是仅支持满仓与空仓，另一类则是支持买卖特定数量的资产。
- **连续型**：连续动作空间常使用 DDPG、TD3、PPO、SAC 类模型，其对应的动作含义和离散型的一致，但对应的调仓空间则能扩展到连续空间上，交易可以变得更加自由。

2.3.1.3 奖励设置

奖励将会引导智能体向正确的方向进行学习。因此，好的奖励设计能够显著影响智能体的行为，使它朝向收益最大化的方向进行学习。常见的奖励函数如下：

- **风险度量类**：常见的集中在夏普比率类和最大回撤等指标。
- **累计收益类**：主要包括累计净收益、对数累计净收益。
- **短期收益类**：主要集中在买卖的市价差、交易损失。

2.3.2 强化学习量化交易特点

- **不断变化的交易时间序列数据分布**：每个资产的交易数据都是通过市场博弈得到的，而市场极易受到外部信号影响产生交易数据分布变化，且随时间动态连续改变，更准确表述是随实时交易而呈近似连续的变化。因此，量化交易策略不是去想办法预测资产价格趋势信号，而是在出现价格波动时探测出潜在交易时机，择机进行交易获得收益。
- **数据粒度显著影响泛化性**：由于加密货币交易频率很高，使用高精度数据进行分析需要极大的存储空间与计算量，而且也会产生大量噪声，因此一般使用小时级别或者日线级别数据一般也能发现较为明显的交易信号，而使用更高精度数据很有可能造成模型学习受过度噪声影响而产生泛化性。目前大部分深度强化学习模型还属于小模型范畴，过度采用超大数据很有可能导致在实际工作中效果不佳。

3、 实验方法

3.1 项目设计

整体流程包含三个模块组：第一模块组是数据与预处理模块、第二模块组是算法与模拟器模块、第三模块组是回测与可视化模块。图3-1为该项目设计图。

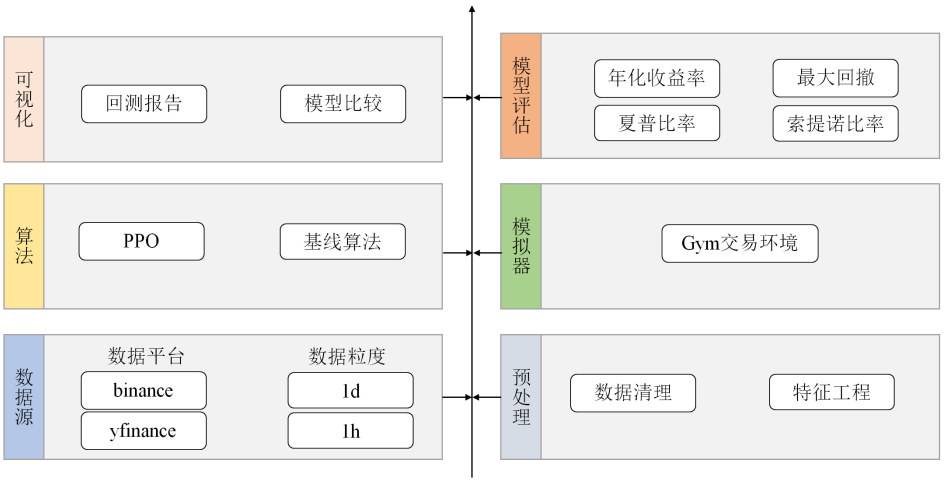


图 3-1 项目框架

3.1.1 数据与预处理模块

3.1.1.1 数据获取

可以通过YahooDownloader和BinanceDownloader进行数据获取。

- YahooDownloader：适合数据粒度为日线的交易数据。

```
df = YahooDownloader(start_date = TRAIN_START_DATE,
    end_date = TEST_END_DATE,
             ticker_list = CRYPTO_10_TICKER, time_interval
                = "1d").fetch_data()
```

- BinanceDownloader：适合数据粒度更小的交易数据。

```
df10d = BinanceDownloader().get_datas(start_date =
    TRAIN_START_DATE,
             end_date = TEST_END_DATE, tickers =
                CRYPTO_10b_TICKER, ts='1d')
```

请注意：BinanceDownloader需要用户给定key与secret_key。与账户绑定才能创建下载实例。

3.1.1.2 数据清理

虽然理论上加密货币交易是 24h 进行的，但每个交易所都会定期进行维护，产生数据的部分缺失。例如币安交易所在 2021.02.11 日维护时产生了一段时间的信息缺

失。因此在这几个小时的维护时间内需要清除一些异常数据点，并填补缺失数据。通常有两种方式：

- 找到异常数据点集，并且填补时间缺失的部分。
- 直接删除异常数据点，让模型认为这段数据是连续的。

这里采用的是第一种做法。

3.1.1.3 数据表征

由于模型不可能把所有历史信息都塞入状态空间，因此需要适当选择合适的技术指标作为模型观测的一部分。在这里，选择的技术指标如下：

```
INDICATORS = [
    "macd",
    "boll_ub",
    "boll_lb",
    'atr',
    "rsi",
    "close_7_sma",
    "close_30_ema",
    "close_365_ema",
    "log-ret"
]
```

3.1.1.4 状态空间

有了预处理数据，可以构造出一个智能体的状态空间。我们假设智能体可以从环境中观测到

- 每期的 OHLCV
- 智能体当前余额
- 智能体当前持仓
- 技术指标

那么我们可以构造图3-2的几个状态空间

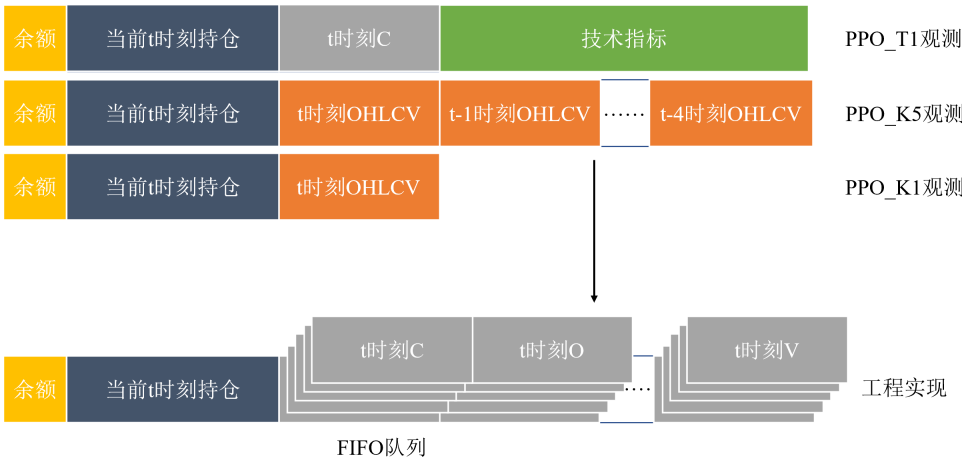


图 3-2　状态空间设计与工程实现思路

对于 PPO_T 观测可以由如下代码生成

```
def _initiate_state(self):
    if self.crypto_dim > 1:
        state = (
            [self.cash] + self.num_crypto_shares +
               self.data.close.values.tolist()
            + sum((self.data[tech].values.tolist() for tech in
               self.tech_indicator_list), [])
        )
    else:
        state = (
            [self.cash] + self.num_crypto_shares +
               [self.data.close]
            + sum(([self.data[tech]] for tech in
               self.tech_indicator_list), [])
        )
    return state
```

对于 PPO_K 观测可以由如下代码生成

```
def _initiate_state(self):
    if self.crypto_dim > 1:
        state = (
            [self.cash] + self.num_crypto_shares +
               self.data.close.values.tolist() +
               self.data.open.values.tolist() +
               self.data.high.values.tolist() +
               self.data.low.values.tolist() +
               self.data.volume.values.tolist()
        )
    else:
        state = (
```

```
                [self.cash] + self.num_crypto_shares +
                    [self.data.close] + [self.data.open] +
                    [self.data.high] + [self.data.low] +
                    [self.data.volume]
            )
        return state
```

3.1.2 算法与模拟器模块

3.1.2.1 动作空间

在传统的股票市场中，买入与卖出都是以“手”作为交易单位的。因此，买卖皆为整数倍的操作。但在加密货币市场上，则没有这个限制。因此使用持仓比例作为交易动作对象，这里可以假定仓位大小为当前资产价值。显然，这样的设定更符合加密货币交易。构造如下动作空间（使用了一个self.action_scaling超参数）

```
self.action_space = spaces.Box(low=-1, high=1,
    shape=(self.action_space,))
scaled_actions = actions * self.portfolio_value *
    self.action_scaling
```

本作设置self.action_scaling为 0.1 使得智能体的行动状态更加稳定。

3.1.2.2 奖励函数

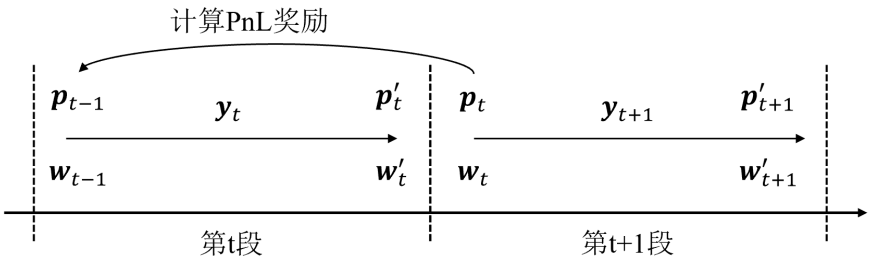


图 3-3　当期盈亏率计算

奖励函数的设置关乎到最终训练的效果。在此，我们选择了若干个常见奖励函数设计。记短期盈亏率系数为 α，盈亏率阈值为为 β，当期的交易时间为 t，短期交易时间段为 $T = 30$，当期投资组合净值为 p_t，最高投资组合净值为 $p_{\max}$，最大回撤比例 τ，数据粒度 ζ（如果是日线数据则为 1，小时数据则为 24）：

1. 当期盈亏率：$r_1 = (p_t - p_{t-1})/p_{t-1}$
2. 短期盈亏率：$r_2 = \alpha(p_t - p_{t-T+1})/p_{t-T+1} * T/\zeta$

3. 总盈亏率：$r_3 = (p_t - p_0)/p_0$
4. 超过最大回撤惩罚：$r_4 = \beta\tau 1_{p_t/p_{\max}<1-\tau}$，1 为示性函数

奖励部分的设计直接影响到训练的效果，这里有如下的考量：

- 如果只设置总盈亏率作为最后的奖励的话，奖励可能过于稀疏。但如果仅设置每期盈亏率作为每期的奖励，模型则会过于贪婪。因此，需要做一下权衡。在实现的过程中，我们不仅设置了总盈亏率和每期盈亏率，而且还动态设置了短期盈亏率。短期盈亏率作为模型的重要奖励，可以捕捉到较为长程的智能体行为关系。在实验中设置 α 为 0.1。
- 此外，如果出现超过最大回撤时，智能体应该立刻抛售作为止损，防止智能体在市场剧烈波动时行为过于激进。在实现中，设定了最大回撤比例 τ 为 0.05，β 为 100。
- 在实验中，pnl 代表使用了 1 和 3 的奖励，new 代表使用了 1-4 的所有奖励设计。

3.1.2.3 模拟器构造

根据 gym 环境构造一个简易的交易环境模拟，在附录中给出部分交易环境模拟代码：

- **智能体交互**
    - 计算奖励
    - 计算资产价值
    - 智能体行动，使用后面的买卖逻辑
    - 事务处理完成后记录当前状态
- **买/卖代币**
    - 买代币检查余额是否充足/卖代币检查是否持有
    - 根据动作买/卖代币
    - 更新余额以及当前的资产组合
    - 返回实际购买/卖出的金额

3.1.3 回测与可视化模块

使用下列三项指标作为回测与可视化模块的组成。在这里介绍介绍一下每个功能的设计初衷以及希望达成的效果。

- **盈利能力**。盈利能力是投资活动的核心目标。在金融市场中，投资者的主要目标是实现资本的保值增值。因此，强化学习模型在量化交易中的主要任务是通

过自主学习和决策，从而在市场中获取最大的收益。盈利能力作为衡量这一目标实现程度的重要指标，具有关键意义。盈利能力反映了策略的有效性。在量化交易中，投资策略的质量和有效性对投资收益具有直接影响。强化学习应用于量化交易，可以实现自主学习和调整投资策略，以适应市场环境的变化。因此，盈利能力作为衡量投资策略执行效果的指标，可以直接反映强化学习模型在量化交易中的有效性。

- **风险控制**。在金融市场中，投资者在追求收益的同时，必须承担一定程度的风险。高收益往往伴随着高风险，因此在制定和执行投资策略时，投资者需要在收益与风险之间取得平衡。强化学习应用于量化交易，需要在追求盈利能力的同时，关注风险控制能力的提升。风险控制能力体现了策略的稳定性。在投资领域，稳定性是衡量投资策略长期可持续性的重要因素。强化学习模型在量化交易中需要具备较强的风险控制能力，以确保在不同市场环境下，投资策略能够保持稳定的表现，减少短期波动对投资组合的影响。在金融市场中，投资者信心对市场稳定性和投资决策具有重要意义。较强的风险控制能力意味着投资策略在市场波动时能够更好地保持稳定，从而提高投资者对策略的信心，降低市场恐慌和投资者情绪波动的影响。
- **鲁棒性**。深度强化学习方法在性能上对超参数高度敏感，并对许多因素（例如随机种子和随时间变化的市场平稳性转移也非常敏感。这种敏感导致策略不鲁棒，对于量化交易等高风险应用可能是较为危险的。相较于传统方法，可能效果很好，但风险控制很难闭环导致没有实际投资者敢于使用该方法作为交易方式。因此，需要建立一系列演示以表征该方法的在通常的投资上是稳健的，且不需要特别复杂的调参过程。

3.2 工程实现

3.2.1 量化交易过程模拟

3.2.1.1 交易假设

1. 有 m 个可通过流动资金投资的资产和一种流动资金。
2. 流动资金的价值永远是 1 个单位，参考的是 USDT。
3. 所有的交易都是在每一期末尾进行操作，操作完即可进入新的一期。
4. 所有操作都是立刻执行，不产生交易滑点。
5. 每次投资标的均以市场价成交。默认市场价为每一期 close 价格。
6. 每次投资标的的成交金额占总成交量极少，对市场无影响。
7. 不要求最后一期折算成流动资金，只需要让投资组合的价值最大。

8. 只做多不做空，并且不包含杠杆，无需缴纳保证金。

3.2.1.2 投资组合分配过程

- 相对交易价格向量

$$\boldsymbol{y}_t := \boldsymbol{v}_t \oslash \boldsymbol{v}_{t-1} = \left(1, \frac{v_{1,t}}{v_{1,t-1}}, \frac{v_{2,t}}{v_{2,t-1}}, \ldots, \frac{v_{m,t}}{v_{m,t-1}}\right)' \tag{3.1}$$

- 初始权重向量

$$\boldsymbol{w}_0 = (1, 0, \ldots, 0)' \tag{3.2}$$

- 投资组合价值（设 p_0 为初始资金）

$$p_t = \boldsymbol{y}_t \cdot \boldsymbol{w}_t \tag{3.3}$$

- 每期的末尾权重变动计算

$$\boldsymbol{w}_t' = \frac{\boldsymbol{y}_t \odot \boldsymbol{w}_{t-1}}{\boldsymbol{y}_t \cdot \boldsymbol{w}_{t-1}} \tag{3.4}$$

- 给定投资组合价值变化比率 μ_t

$$p_t = \mu_t p_t' \tag{3.5}$$

- 一期的回报比率与对数回报比率

$$\begin{aligned} \rho_t &= \frac{p_t}{p_{t-1}} - 1 = \frac{\mu_t p_t'}{p_{t-1}} - 1 = \mu_t \boldsymbol{y}_t \cdot \boldsymbol{w}_{t-1} - 1 \\ r_t &= \ln \frac{p_t}{p_{t-1}} = \ln\left(\mu_t \boldsymbol{y}_t \cdot \boldsymbol{w}_{t-1}\right) \end{aligned} \tag{3.6}$$

- 最终投资组合价值

$$p_f = p_0 \exp\left(\sum_{t=1}^{t_f+1} r_t\right) = p_0 \prod_{t=1}^{t_f+1} \mu_t \boldsymbol{y}_t \cdot \boldsymbol{w}_{t-1} \tag{3.7}$$

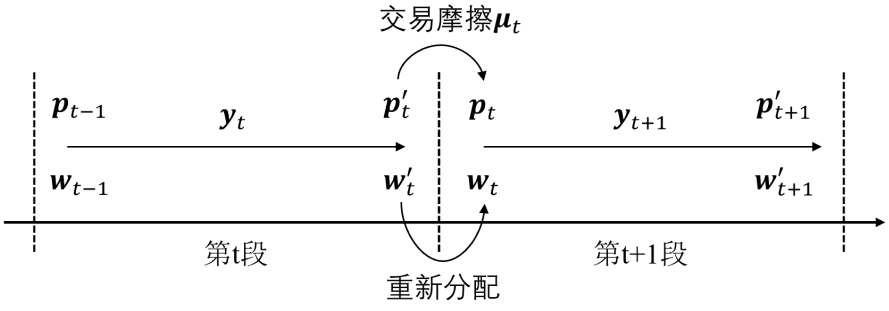


图 3-4　多阶段交易过程示意图

多阶段交易过程（图3-4）中存在交易摩擦效应。在时期 t 内，市场的价格相对向量 y_t 表示，驱动投资组合价值和投资组合权重从 p_{t-1} 和 w_{t-1} 到 p'_t 和 w'_t。在时刻 t 进行交易，并将资金重新分配，因市场需要对交易行为收取手续费，交易过程由此产生所谓的交易摩擦，反复交易过程将投资组合的价值缩小到 p_t。我们想要求解的最优交易过程，这个最优交易过程所受的影响仅跟投资组合权重和市场的价格相对向量有关联。[44]。

3.2.2 数据概览

这里简单介绍一下数据集信息：

数据集名称	起始时间	终止时间	时间粒度
crypto_t10_d	2021-01-01	2023-04-01	1d
crypto_t10_h	2021-01-01	2023-04-01	1h

表 3-1　数据集

代币信息如下：

```
CRYPTO_10b_TICKER = [
 'BTCUSDT',
 'ETHUSDT',
 'BNBUSDT',
 'XRPUSDT',
 'ADAUSDT',
 'DOGEUSDT',
 'SOLUSDT',
 'LTCUSDT',
 'TRXUSDT',
 'ETCUSDT'
]
```

这里，我们展示一下小时级别数据的**价格波动曲线**。（对于其他粒度的数据类似，这里不重复展示）

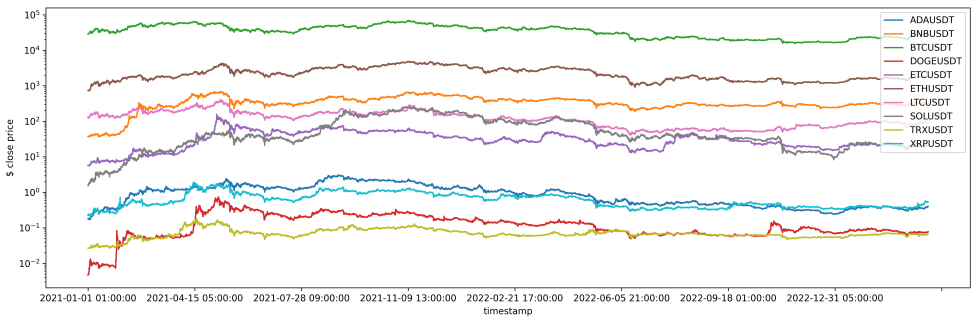


图 3-5　十种代币的价格曲线（小时级别），纵坐标为对数坐标。数据来源：Binance

对数收益率曲线

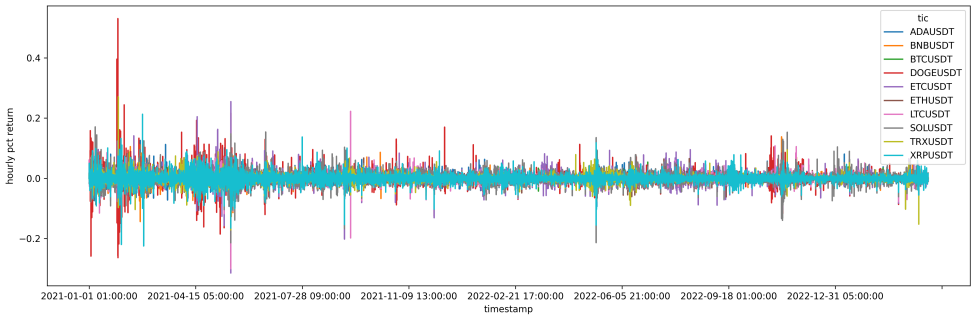


图 3-6　十种代币的对数收益（小时级别）。数据来源：Binance

累计收益曲线

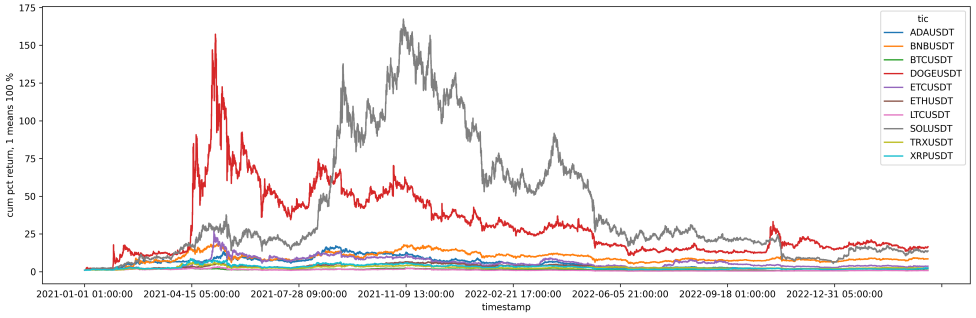


图 3-7　十种代币的累计收益（小时级别），记初始价格为 1。数据来源：Binance

3.2.3　数据集划分

实验中，我们将采用类似于文献[45]提出的数据集划分方法。首先，我们使用 2021 年 1 月 1 日至 2022 年 6 月 30 日的数据进行训练和验证，然后使用 3 个月的数据进行测试。接下来，我们按照下图所示，每 3 个月推移一次，逐个作为测试集进行回测比较。最终，我们使用 2023 年 1 月 1 日至 2023 年 4 月 1 日的数据进行回测。一共使用三段测试集进行结果分析。

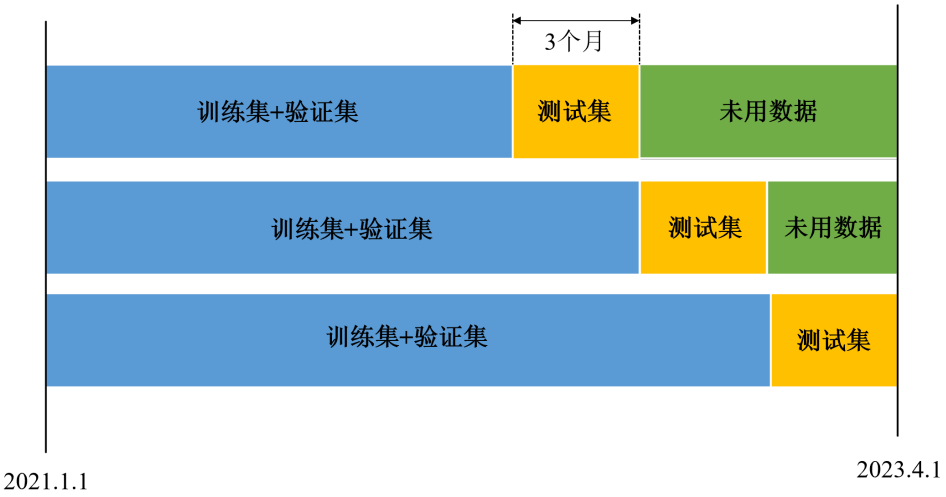


图 3-8　数据集划分

这里展示一下测试集的数据形态，因为直接显示价格曲线容易导致对低价格代币的忽视，因此选择了对数收益曲线作为展示内容。

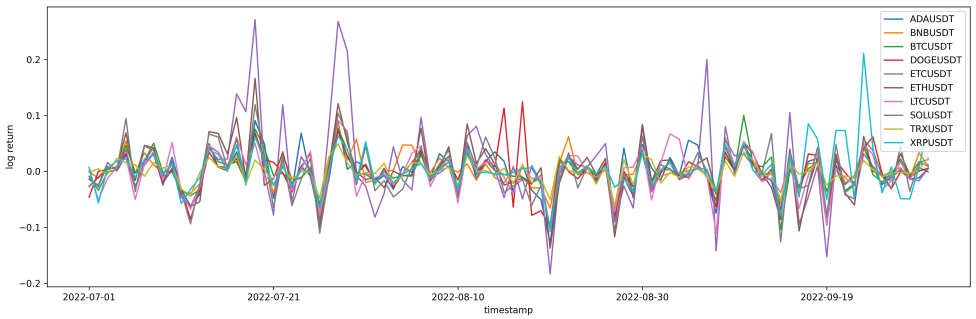


图 3-9　日线第一段测试集

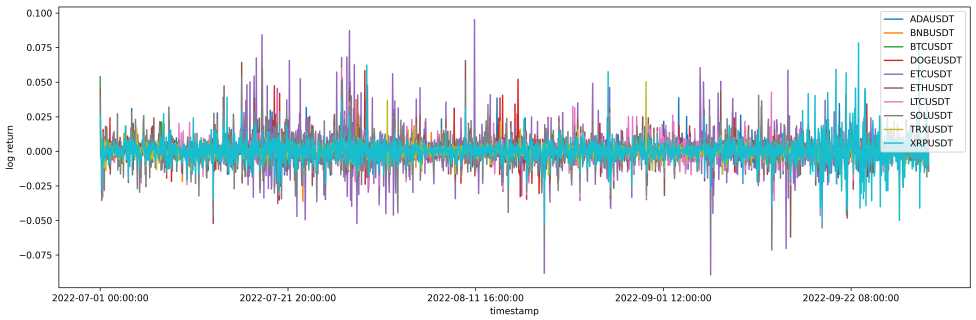


图 3-10　小时线第一段测试集

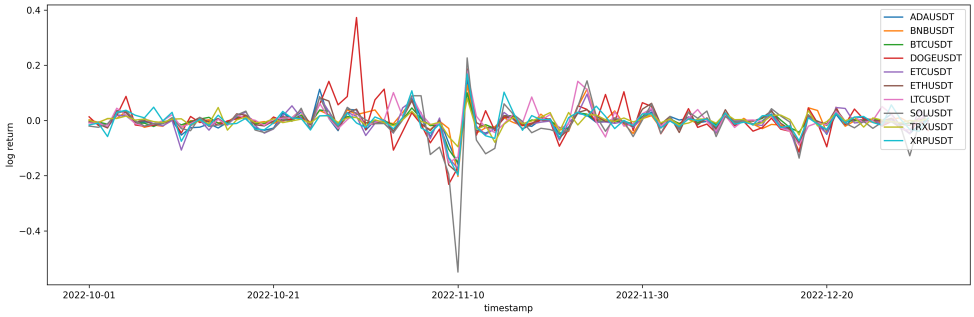


图 3-11　日线第二段测试集

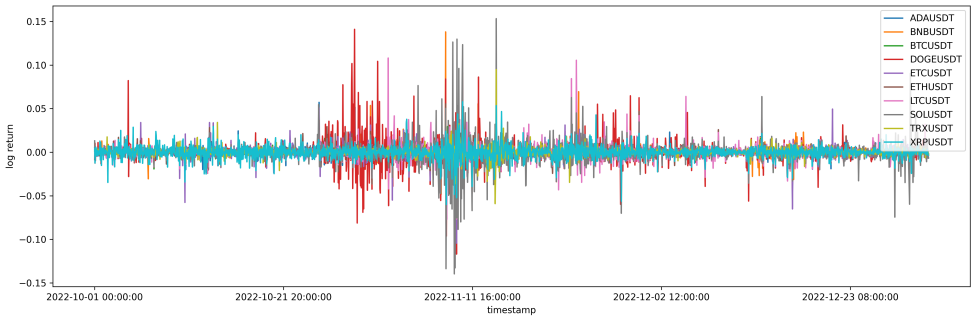


图 3-12　小时线第二段测试集

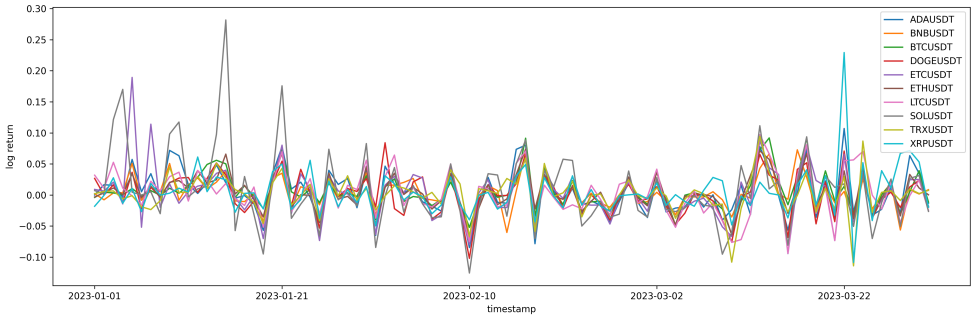


图 3-13　日线第三段测试集

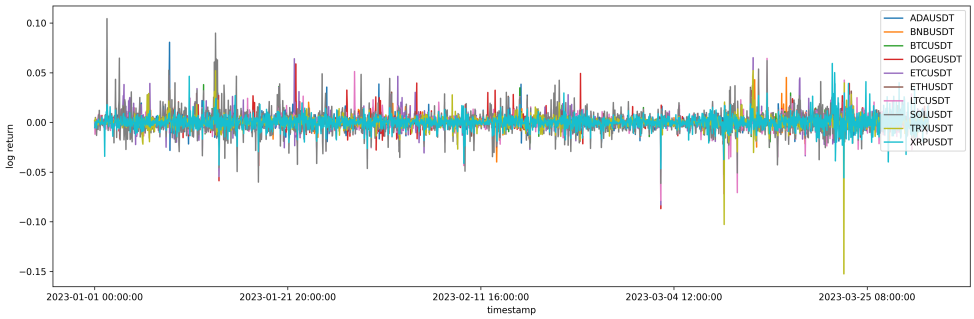


图 3-14　小时线第三段测试集

3.2.4 基线交易策略

基线策略用来对智能体策略评估。对于资产组合管理优化的经典策略，可以参考[46]，这里采用一些经典的优化方案。

- **买入并持有策略**：这是在加密货币市场交易时的经典基准。因为这是最简单的交易策略，因此购买资产并持有它直到测试期结束这一操作提供了一些信息丰富的基准利润衡量标准。如果显著超过此标准，说明使用强化学习进行交易的方法至少是有效的。
- **持续再平衡策略**[47]：不断通过调整资产仓位使得投资组合始终保持在初始的权重配比。

我们在测试时中只考虑买入并持有策略，具体而言为等权重买入并持有策略。

3.2.5 测试方式与超参数选择

对于深度强化学习模型，选择能够输出连续动作的 PPO 作为实验的核心算法。针对于观测，采用了三种观测作为实验观测候选。结合日线数据和小时线数据，采用不同的奖励函数设计，在三段测试集上验证有效性。实验超参数设置如表3-2、3-3所示：

参数名称	参数记号	设置值
总时间步	total_timestamp	3M
初始资金	cash	100K
手续费	fee	0.1%

表 3-2　环境超参数

参数名称	参数记号	设置值
种子	seed	42
PPO 更新步数	n_steps	2048
PPO 学习率	learning_rate	0.001
PPO 损失计算的熵系数	ent_coef	0.01
PPO 批量大小	batch_size	256

表 3-3　PPO 超参数

对于超参调整，我们遵循了[48]的建议，其他的超参数设置均保持 stable baseline 3 的默认超参数设置。

此外，实验中采用了以下记号，在表3-4予以说明。

记号	意义
tech/TI	采用技术指标观测的测试数据
stack_1	采用最近一期 OHLCV 观测的测试数据
stack_5	采用最近五期 OHLCV 观测的测试数据
pnl	采用传统的奖励函数设计
new	采用传统的奖励函数基础上加入风险的奖励函数设计

表 3-4　实验记号说明

4、 实验结果与分析

4.1 实验结果

首先给出训练时间步在 3M 处各智能体投资组合价值变动曲线。

4.1.1 第一段测试集投资组合价值变动曲线

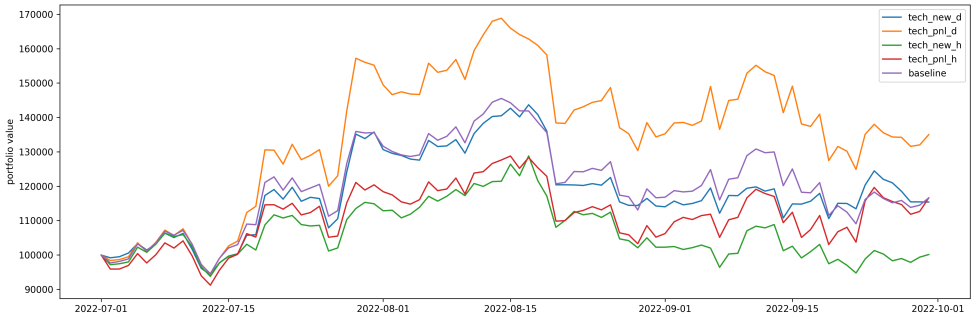


图 4-1　第一段测试集（采用技术指标观测）

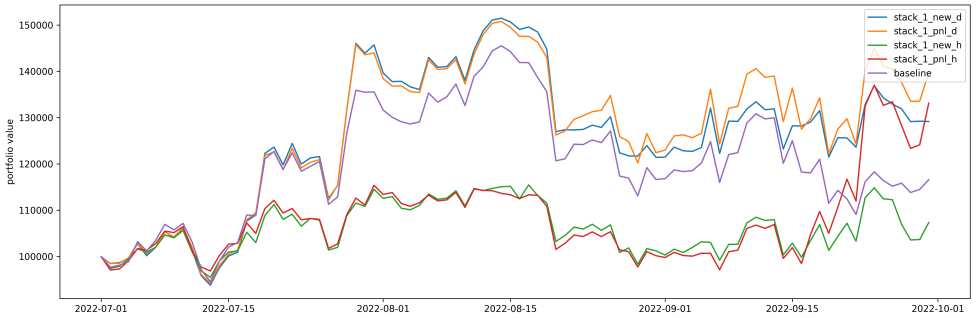


图 4-2　第一段测试集（采用一期 OHLCV 观测）

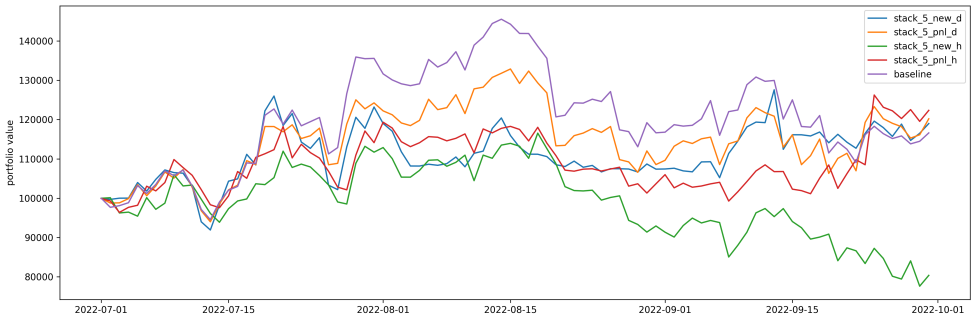


图 4-3　第一段测试集（采用五期 OHLCV 观测）

4.1.2 第二段测试集投资组合价值变动曲线

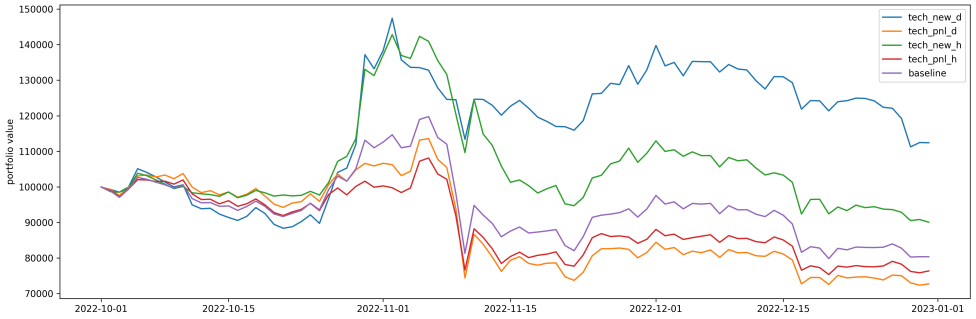


图 4-4 第二段测试集（采用技术指标观测）

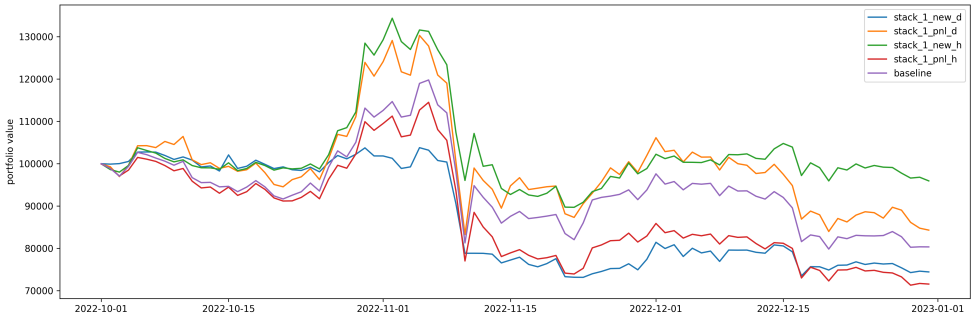


图 4-5 第二段测试集（采用一期 OHLCV 观测）

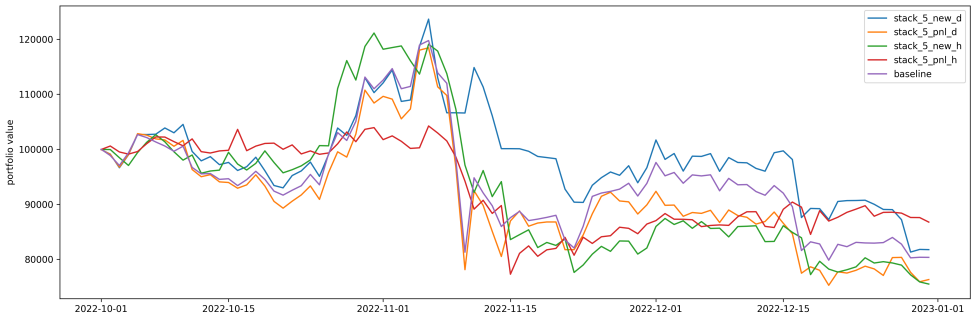


图 4-6 第二段测试集（采用五期 OHLCV 观测）

4.1.3 第三段测试集投资组合价值变动曲线

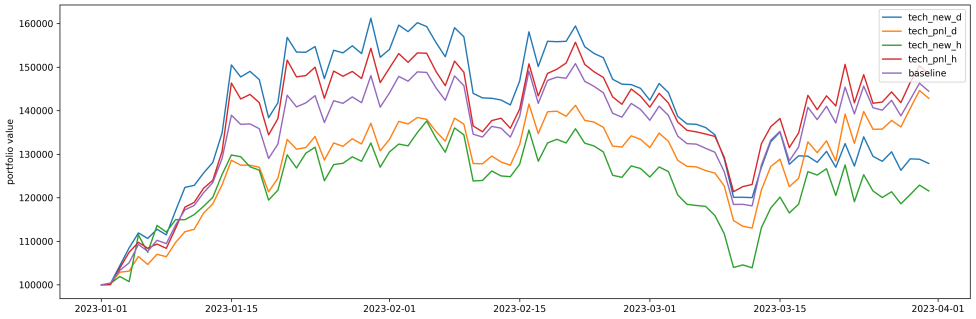


图 4-7 第三段测试集（采用技术指标观测）

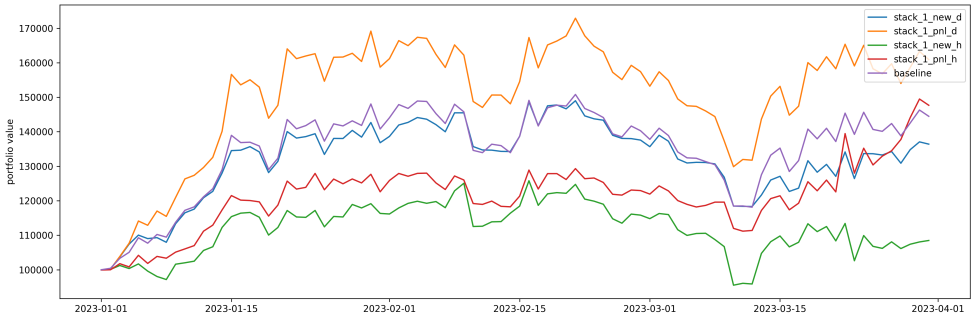


图 4-8 第三段测试集（采用一期 OHLCV 观测）

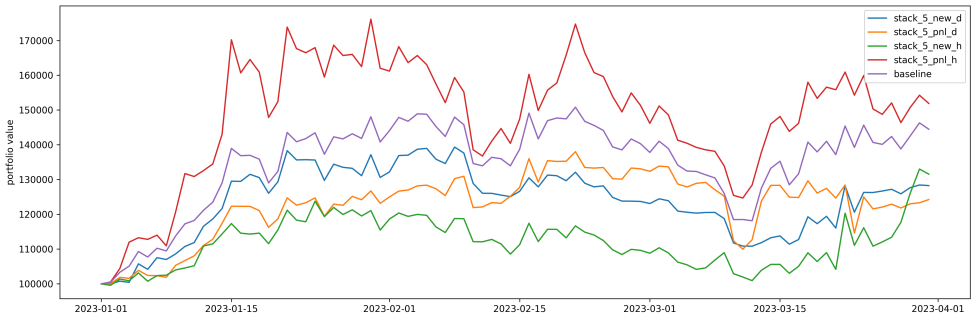


图 4-9 第三段测试集（采用五期 OHLCV 观测）

在本文中，我们通过对三个不同策略的实验结果进行细致的分析，发现了一些有趣的规律。这三种策略分别是 PPO_K1、PPO_K5 和 PPO_T1。对于这三个策略的资产净收益和回撤的表现，我们发现它们在相似的位置表现出相似的趋势。更具体地说，我们可以看到，在各自的策略执行过程中，三者的资产净收益都在相似的位置呈现出了上升的趋势。这可能说明在这些特定的位置，三者的策略选择可能具有某种共通性，这种共通性导致了在资产净收益上的相似表现。

类似的，我们也在相近的位置发现了三个策略的回撤现象。这可能意味着在这些特定的位置，三个策略面临的风险和挑战可能有一定的共性。这种共性可能在一定程度上影响了这三个策略的表现，并导致了回撤的出现。需要强调的是，这种回撤并不代表策略的失败，而是在投资过程中常见的风险表现。因此，对于投资者来说，理解并适应这种回撤现象是非常必要的。

总体而言，我们发现 PPO_K1 的效果最好。这可能是因为 PPO_K1 在策略选择上具有一定的优势，使得其在资产净收益和回撤的表现上都表现出了优秀的稳健性。然而，这并不意味着其他两种策略没有价值。相反，它们在某些特定的情况下可能会表现出更好的效果。因此，对于投资者来说，选择哪种策略并不是一个简单的问题，而需要根据自身的投资目标和风险承受能力来决定。

4.2 稳定性与收益对比分析

对每个智能体选择了训练步数在 2M-3M 处共 10 组数据计算均值和标准差作为对比分析对象。实验设计了四个测量指标：年化收益率用于衡量投资组合在测试时期折算到一年的总收益率；最大回撤是投资组合在最坏情况下可能面临的最大损失比例；夏普比率表示投资组合风险调整后的收益，即收益率与波动性之比；索提诺比率则是风险调整后收益的另一种指标，关注收益与下行波动性之比。实验数据如下：

模型		年化收益率	最大回撤	夏普比率	索提诺比率
小时线 pnl	PPO_K1	0.55±0.64	-0.19±0.03	0.87±0.81	1.48±1.41
	PPO_K5	1.10±0.76	-0.23±0.08	1.26±0.64	2.50±1.35
	PPO_T1	0.50±0.30	-0.23±0.02	0.95±0.28	1.40±0.46
日线 pnl	PPO_K1	1.23±0.46	-0.22±0.03	1.49±0.34	2.36±0.60
	PPO_K5	0.66±0.00	-0.20±0.00	1.16±0.00	1.71±0.00
	PPO_T1	1.38±0.18	-0.24±0.01	1.56±0.09	2.50±0.17
小时线 new	PPO_K1	-0.08±0.20	-0.19±0.02	-0.11±0.57	-0.12±0.75
	PPO_K5	-0.12±0.32	-0.34±0.04	-0.00±0.56	0.03±0.84
	PPO_T1	-0.05±0.18	-0.28±0.03	0.14±0.32	0.21±0.47
日线 new	PPO_K1	1.75±1.14	-0.20±0.03	1.60±0.51	2.77±1.36
	PPO_K5	0.63±0.02	-0.19±0.00	1.12±0.02	1.71±0.03
	PPO_T1	0.24±0.17	-0.23±0.01	0.66±0.28	0.99±0.42
买入并持有		0.52	-0.25	1.01	2.09

表 4-1　第一段测试集平均对比分析

模型		年化收益率	最大回撤	夏普比率	索提诺比率
小时线 pnl	PPO_K1	-0.62±0.20	-0.42±0.16	-1.22±0.62	-0.42±0.16
	PPO_K5	-0.38±0.19	-0.31±0.06	-0.83±0.46	-1.04±0.60
	PPO_T1	-0.56±0.06	-0.36±0.03	-1.07±0.30	-1.39±0.36
日线 pnl	PPO_K1	-0.36±0.04	-0.35±0.01	-0.32±0.14	-0.45±0.18
	PPO_K5	-0.49±0.03	-0.34±0.02	-0.71±0.07	-0.98±0.09
	PPO_T1	-0.67±0.04	-0.41±0.02	-1.35±0.17	-1.72±0.21
小时线 new	PPO_K1	-0.04±0.62	-0.29±0.08	-0.32±0.92	-0.24±1.53
	PPO_K5	-0.67±0.11	-0.41±0.06	-1.90±0.40	-2.29±0.44
	PPO_T1	-0.16±0.15	-0.42±0.07	0.04±0.34	0.13±0.63
日线 new	PPO_K1	-0.07±0.78	-0.30±0.10	-0.70±1.57	-0.42±2.44
	PPO_K5	-0.60±0.11	-0.37±0.03	-1.81±0.64	-2.17±0.67
	PPO_T1	-0.00±0.38	-0.32±0.08	0.14±0.72	0.35±1.21
买入并持有		-0.45	-0.33	-0.68	-0.92

表 4-2　第二段测试集平均对比分析

模型		年化收益率	最大回撤	夏普比率	索提诺比率
小时线 pnl	PPO_K1	1.35±0.43	-0.19±0.03	2.01±0.35	3.19±0.78
	PPO_K5	0.71±0.69	-0.26±0.06	1.09±0.83	1.71±1.29
	PPO_T1	2.18±0.92	-0.22±0.04	2.32±0.16	3.92±0.04
日线 pnl	PPO_K1	2.79±0.00	-0.25±0.00	2.53±0.00	4.44±0.00
	PPO_K5	0.91±0.11	-0.20±0.00	1.59±0.12	2.33±0.22
	PPO_T1	1.89±0.25	-0.19±0.02	2.46±0.16	4.20±0.36
小时线 new	PPO_K1	0.64±0.67	-0.25±0.04	1.05±0.57	1.61±1.04
	PPO_K5	1.22±0.70	-0.21±0.02	1.59±0.79	3.01±1.55
	PPO_T1	0.37±0.27	-0.32±0.04	0.81±0.40	1.26±0.63
日线 new	PPO_K1	2.06±0.90	-0.22±0.01	2.31±0.37	4.14±0.95
	PPO_K5	0.75±0.21	-0.20±0.01	1.56±0.26	2.54±0.50
	PPO_T1	1.17±0.32	-0.23±0.01	1.84±0.23	2.99±0.45
买入并持有		1.80	-0.22	2.28	3.80

表 4-3　第三段测试集对比分析

- 首先对比**模型收益情况**。
  - **采用技术指标观测和最近一期 OHLCV 观测模型结合日线数据获益较好。**使用日线传统 pnl 奖励的结果分别为 1.38，-0.67，1.89 和 1.23，-0.36，2.79。对比买入与持有获得的超额部分分别为 0.86，-0.22，0.09 和 0.71，0.09，0.90。针对日线的平均超额部分分别为（针对所有模型和奖励样本取平均值）14% 和 61% 的超额年化收益率。
  - **采用最近五期的 OHLCV 观测或者小时线数据获益较差。**采用小时线的平均收益为 0.32，-0.41，1.08，而采用日线的平均收益为 0.98，-0.24，1.60。五期的 OHLCV 观测除了在小时线 pnl 和小时线 new 下小幅超过一期的 OHLCV 观测，在其余时间都是略逊于一期的观测。
  - 出现上述现象的是因为：第一，采用五期的观测空间过大，模型输入端的特征解析能力不够所致。第二，小时线数据量较大，需要更长时间的模型训练才能得到收敛解。
- 其次对比**模型风险控制情况**。
  - 使用加入风险机制奖励的智能体，**收益率在单边下行市场偏高**。使用日线数据为例，在第二段买入并持有出现-0.45 的负年化收益时，使用日线 new 奖励结合技术指标观测和最近一期 OHLCV 观测模型的结果时–0.00，0.07，相比之下，pnl 的结果则是-0.67，-0.36。
  - **在单边上行市场获益显著低于传统 pnl 奖励**。使用日线数据为例，在第三段买入并持有出现 1.80 的大的正年化收益时，使用日线 new 奖励结合技术指标观测和最近一期 OHLCV 观测模型的结果时 1.17，2.06，相比之下，pnl 的结果则是 1.89，2.79。
  - **最大回撤比例相较于 pnl 类似，都与基线水平接近**。这种情况可能与参数设定有关，如果对 5% 的熔断系数调低，并且对熔断之后的行为加以控制，其最大回撤比例可能会有显著改善。
  - 出现上述现象的是因为：第一，熔断系数起效，但不够全面刻画波动市场的行为，造成的最大回撤比例没有显著减少，但在单边下行的市场上还是起效的。第二，由于对于风险行为进行了控制，在利好的背景下，收益显著下降。因此，需要择机判断市场形态，选择合适的奖励作为训练素材已针对不同形态的市场。
- 最后，该模型行动表现提示这是一种**指数增强型交易策略**。相较于普通买入与持有交易而言，在适合的条件下，有一定超额年化收益优势。

4.3 训练加速比分析

在计算机科学的实践领域，硬件选择对于模型训练效率的影响是一种常见的关注点。近期的研究发现，在一些特定情境下，采用图形处理单元（Graphics Processing Unit，GPU）进行训练可能并非最优选择。相反，采用高带宽的 Macintosh（Mac）中央处理单元（Central Processing Unit，CPU）有可能带来更好的效果。本文在两台机器上分别进行了测试，测试参数如下：

组件	对应型号	参数
内存	Windows	32G DDR4-2666
	MacOS	16G LPDDR5-6400
cpu	Windows	i7-10700
	MacOS	m2pro
显卡	Windows	RTX2060S
	MacOS	m2pro

为了全面理解这一现象，我们需要深入探讨模型训练中的几个核心要素：数据吞吐、计算量和硬件性能。通常，GPU 在处理大规模并行计算任务时表现优秀，这使得其在许多机器学习任务中，尤其是深度学习任务中，成为首选。然而，这一观念并不适用于所有情境。特别地，在强化学习的设置下，这种观点的例外尤为明显。在这样的设定下，数据吞吐，而非计算量，成为影响效率的主要瓶颈。强化学习的特性使得数据生成过程与模型训练过程紧密耦合，这导致在数据处理和传输阶段可能出现瓶颈。在这种情况下，较高的数据吞吐率可以显著提高训练效率。图4-10[1]很好地展示了这一点。

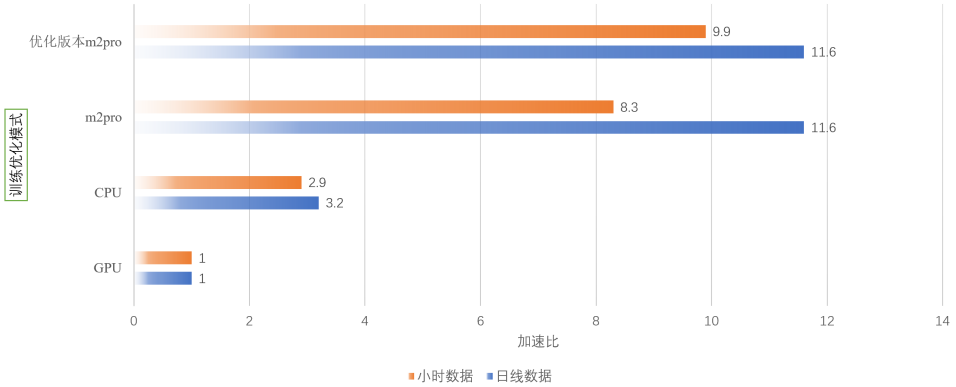


图 4-10　实验加速比

1　优化版本将会在内存保存一个专用的列表存放缓存数据

此时，一个高带宽的 Mac CPU 可以发挥其优势。Mac CPU 具有高速的内存接口，能够在短时间内处理和传输大量数据，从而有效地缓解数据吞吐瓶颈。此外，由于强化学习的训练过程并未涉及大规模的并行计算任务，因此，相较于 GPU，CPU 的计算性能并未成为影响训练效率的限制因素。

总体来说，根据数据吞吐和计算量的需求选择合适的硬件配置是至关重要的。尽管 GPU 在处理大规模并行计算任务上具有优势，但在特定的 RL 环境下，高带宽的 Mac CPU 由于其优异的数据处理和传输能力，可能会成为更好的选择。这一发现进一步强调了根据具体任务需求选择和配置硬件的重要性，也展现了在计算机科学领域，简单的“一刀切”策略往往无法满足所有情况的需求。

4.4 讨论

1. 关于**交易结果**：日线的平均年化收益率相比基线高出约 46%、6%、-20%，针对于表现较好的技术指标观测 PPO_{K1}，相比基线高出约 97%、23%、62%。而小时线的年化收益率则显著低于日线水平。针对于奖励函数设计，使用添加了风险指标（最大回撤与短期回报）的奖励效果不及没有添加风险指标，说明该算法对于添加风险指标的奖励不敏感。
2. 关于**模型训练**：本课题的所有实验都在 3M 处结束训练，比较分析选择 2M 到 3M 处的连续 100K 训练步的智能体。训练表明在 3M 处时，模型依旧还没有完成充分学习，震荡依旧比较大，预期在 10M 以上的训练步后，模型有可能得到更稳定和鲁棒的结果。
3. 关于**人为主观判断**：

   (a) **在基于数据驱动模型算法下，依旧会出现大回撤事件。**而在涨势强劲的第三段测试集上，使用该算法**没有实现**相较于买入和持有更高的收益率。因此该算法如果想获得更高收益，还需要结合人为判断和干预，因此相较与单边市场，目前构建的模型可能更适合于震荡形式市场。

   (b) 虽然从目前结果看，直接使用强化学习进行量化交易就类似一场无法控制的赌博，因为智能体是数据驱动的，学习到的策略并不完全受到控制，但在高频量化交易中，深度强化学习相比于传统方法还是**具有一定的优势**。具体表现在以下几点：

      i. **交易时延短，推理速度快**：深度强化学习算法通常可以在短时间内做出决策，非常适合用于高频交易。因为在这种交易方式中，需要在极短时间内对大量数据进行分析并做出快速决策。目前的深度强化学习模型都是“小模型”，推理时间较短，因此适合于低时延的高频交易。

ii. **较强的自适应能力**：强化学习一个优势是能够在动态环境中自适应地进行决策。量化交易是非常复杂且不断变化的博弈过程，强化学习算法能够在这种迅速变化环境中不断学习和调整策略。

iii. **丰富的数据驱动**：强化学习是一种数据驱动的自我完善方法，通过大量数据学习并不断优化策略。在量化交易中数据非常丰富，强化学习能够充分利用这些数据来调整投资策略的表现。

iv. **大量开源项目经验借鉴**：深度强化学习作为一种研究热点，已经在多个领域取得显著成果。随着研究的深入，有可能解决目前解释性较差和难训练等问题，不断提高深度强化学习在量化交易中的应用价值。

5、 结论

通过对加密货币交易市场波动性和不确定性现状的研究分析，本文提出了一种基于深度强化学习的投资组合管理算法策略，并构建了交易智能体，通过模拟交易过程验证算法策略在加密货币市场交易的可行性。实验结果表明：在三个使用日线数据的测试集和十个代表性代币构建的投资组合上，对比等权重买入与持有策略平均获得约 11% 的超额年化收益率；当智能体处理最近一期 OHLCV 数据时，智能体表现最好，取得了约 61% 的平均超额年化收益率，总体而言，本课题所采用的投资组合策略在震荡市场上比单边市场表现更好。综上所述，随着技术发展和研究深入，基于深度强化学习的交易模式可能成为投资组合管理领域的重要方法。

6、 不足与展望

6.1 不足

1. **数据和计算资源相对稀缺**是本课题在量化交易任务中应用强化学习的主要挑战。虽然量化交易领域的数据量相比于其他金融领域而言产生更多，也方便直接拿来分析，但本课题仅采用日线和小时线作为参考，其中很大原因就是更小粒度的数据获取成本很高。而且，即使解决了数据问题，计算资源稀缺又会带来很大困扰，计算资源不足是本文比较大的缺憾。
2. **量化交易的关键目标是在最大化利润和最小化风险之间取得平衡，但强化学习本身的奖励是单一的**。虽然多目标强化学习技术似乎提供了一种利润和风险之间平衡的手段，但现有的数据无法证明其能够真正意义上在任何交易情况下都能降低风险，在今后研究中训练具有不同风险容忍度的多样化交易策略并集成将会是是一个有趣的方向。
3. **需要更加真实的交易模拟**。高保真度的模拟是强化学习方法成功的关键基础。尽管本文考虑了实际交易费用，但普遍的市场影响还是被忽视了，它指的是一个交易者的行为对其他交易者的影响。因此，仅有历史市场数据模拟是不够的，还需要着眼于处理市场影响。构建高保真度的市场模拟器是一个非常具有挑战性但又非常重要的研究方向，在现阶段研究中还很难处理市场影响因素。
4. **强化学习算法很难收敛，或者学习很难达到预期**。由于强化学习算法是一边获取经验（或者在经验池内获取），一边训练有效的策略，因此在初始阶段，强化学习的样本效率很低，而且难以学习到真正有用的经验。在后期，强化学习交易的波动性很大，有时也很难挑选出最优的智能体。
5. 使用不同 A2C 算法后，PPO 是**唯一能够得到正常交易序列**的 A2C 算法。原因可能是其他离线采样算法容易在交易初期陷入一个局部最优解，模型在实测中容易较早就停止学习过程。
6. 由于**缺乏可解释性**，人们可能根本无法根据强化学习的决策进行实际交易，这也给模型应用带来较为致命的影响。但如果能采用集合人类视觉和自然语言的大模型，似乎可以来解释强化学习的行动策略，更好说明模型决策过程。

6.2 展望

现有的工作[7][9]已经展示了强化学习方法在量化交易任务中的成功和不足。本节将指出一些未来研究方向，并详细阐述几个关键的未解决问题和潜在解决方案。

- **行业需要研究自动机器学习方法**

  由于金融数据的低信噪比特性，以及强化学习方法对各种超参的普遍敏感性，基于强化学习的量化交易模型，其成功高度依赖于强化学习组件（例如奖励函数）和超参数的精心设计。因此，对于没有深入了解强化学习的行业工作人员，如经济学家和专业交易员，采用基于强化学习的交易策略将面临较大技术困难，但如果能引入自动机器学习方法，目前存在的技术壁垒就能克服。对于特征工程而言，自动机器学习也可以通过选择和收集有意义的特征自动构建。对于超参数调整，自动机器学习可以搜索适当的超参数，如超参数和奖励函数等进行调整。总之，将来在自动机器学习辅助下，即使对强化学习没有深入了解，基于强化学习的量化交易模型也会更易使用。

- **行业需要设立一个类似于 ImageNet 一样的标准化评测**

  新的基于强化学习的量化交易方法，一般需要与一些金融数据集上现有技术水平基线进行比较。目前，基线和数据集的选择还比较随意，这会产生操控数据以达到超过现有方法的行为发生。因此，目前在基于强化学习的量化交易方法排名上还没有广泛地形成共识，对新的强化学习算法进行基准测试变得极具现实意义。标准化的评估数据集和设立现有技术水平基线可能是解决这个问题的途径。至于评估标准，大多数现有的工作只用金融指标（如总收益）来评估强化学习算法的盈利能力，忽略了一些关键的方面，例如风险控制、可解释性、多样性、可靠性和普适性。实际上，由于金融市场低信噪比性质，强化学习方法仅依靠金融历史数据的高收益，很可能会过度拟合噪声，并在实际部署中失效。此外，一些作者选择了较低的基线作为比较对象，但花费更多的时间去调整超参数与网络结构，报告收益可信性就会因此降低。令人欣喜的是，有工作正在往标准化评测方向发展，通过采用标准数据集、多种基线实现和完整评估方案，基于强化学习的量化交易研究可以取得更好的发展。

参考文献

[1] CHAN E P. Quantitative trading: how to build your own algorithmic trading business[M]. John Wiley & Sons, 2021.

[2] MARKOWITZ H. PORTFOLIO SELECTION*[J/OL]. The Journal of Finance, 1952, 7(1): 77-91. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1540-6261.1952.tb01525.x. https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-6261.1952.tb01525.x. DOI: https://doi.org/10.1111/j.1540-6261.1952.tb01525.x.

[3] SHARPE W F. Capital asset prices: A theory of market equilibrium under conditions of risk[J]. The journal of finance, 1964, 19(3): 425-442.

[4] FAMA E F, FRENCH K R. Common risk factors in the returns on stocks and bonds[J]. Journal of financial economics, 1993, 33(1): 3-56.

[5] ZOU J, ZHAO Q, JIAO Y, et al. Stock Market Prediction via Deep Learning Techniques: A Survey [J]. arXiv preprint arXiv:2212.12717, 2022.

[6] JIANG W. Applications of deep learning in stock market prediction: recent progress[J]. Expert Systems with Applications, 2021, 184: 115537.

[7] MILLEA A. Deep reinforcement learning for trading—A critical survey[J]. Data, 2021, 6(11): 119.

[8] 梁天新杨小平. 基于强化学习的金融交易系统研究与发展[J]. 软件学报, 2019, 30(845-864). DOI: 10.13328/j.cnki.jos.005689.

[9] SUN S, WANG R, AN B. Reinforcement learning for quantitative trading[J]. arXiv preprint arXiv:2109.13851, 2021.

[10] DENG Y, BAO F, KONG Y, et al. Deep direct reinforcement learning for financial signal representation and trading[J]. IEEE transactions on neural networks and learning systems, 2016, 28(3): 653-664.

[11] JIANG Z, XU D, LIANG J. A deep reinforcement learning framework for the financial portfolio management problem[J]. arXiv preprint arXiv:1706.10059, 2017.

[12] LUCARELLI G, BORROTTI M. A deep Q-learning portfolio management framework for the cryptocurrency market[J]. Neural Computing and Applications, 2020, 32: 17229-17244.

[13] LIU X Y, YANG H, CHEN Q, et al. FinRL: A deep reinforcement learning library for automated stock trading in quantitative finance[J]. arXiv preprint arXiv:2011.09607, 2020.

[14] YANG H, LIU X Y, ZHONG S, et al. Deep reinforcement learning for automated stock trading: An ensemble strategy[C]//Proceedings of the first ACM international conference on AI in finance. 2020: 1-8.

[15] GUAN M, LIU X Y. Explainable Deep Reinforcement Learning for Portfolio Management: An Empirical Approach[Z]. 2021. arXiv: 2111.03995 [q-fin.PM].

[16] SUN S, QIN M, WANG X, et al. PRUDEX-Compass: Towards Systematic Evaluation of Reinforcement Learning in Financial Markets[J/OL]. Transactions on Machine Learning Research, 2023. https://openreview.net/forum?id=JjbsIYOuNi.

[17] KORATAMADDI P, WADHWANI K, GUPTA M, et al. Market sentiment-aware deep reinforcement learning approach for stock portfolio allocation[J]. Engineering Science and Technology, an International Journal, 2021, 24(4): 848-859.

[18] NAN A, PERUMAL A, ZAIANE O R. Sentiment and knowledge based algorithmic trading with deep reinforcement learning[C]//Database and Expert Systems Applications: 33rd International Conference, DEXA 2022, Vienna, Austria, August 22–24, 2022, Proceedings, Part I. 2022: 167-180.

[19] 齐岳黄硕华. 基于深度强化学习 DDPG 算法的投资组合管理[J]. 计算机与现代化, 2018, No.273(93-99).

[20] 傅丰王康. 基于深度强化学习 SAC 算法的投资组合管理[J]. 现代计算机, 2020, No.681(45-48).

[21] 左晓冬. 基于投资者情绪和深度强化学习的股票投资组合优化研究[J]. 中国管理信息化, 2023, 26(130-133).

[22] SCHNAUBELT M, FISCHER T G, KRAUSS C. Separating the signal from the noise–financial machine learning for twitter[J]. Journal of Economic Dynamics and Control, 2020, 114: 103895.

[23] SUTTON R S, BARTO A G. Reinforcement learning: An introduction[M]. MIT press, 2018.

[24] HAO DONG S Z, Zihan Ding. Deep Reinforcement Learning: Fundamentals, Research, and Applications[M]. Ed. by HAO DONG S Z, Zihan Ding. Springer Nature, 2020.

[25] HAUSKNECHT M, STONE P. Deep recurrent q-learning for partially observable mdps[J]. arXiv preprint arXiv:1507.06527, 2015.

[26] ZHOU L, QIN K, TORRES C F, et al. High-frequency trading on decentralized on-chain exchanges [C]//2021 IEEE Symposium on Security and Privacy (SP). 2021: 428-445.

[27] GOODFELLOW I, BENGIO Y, COURVILLE A. Deep learning[M]. MIT press, 2016.

[28] ZHANG A, LIPTON Z C, LI M, et al. Dive into Deep Learning[J]. arXiv preprint arXiv:2106.11342, 2021.

[29] SILVER D, LEVER G, HEESS N, et al. Deterministic policy gradient algorithms[C]// International conference on machine learning. 2014: 387-395.

[30] SCHULMAN J, MORITZ P, LEVINE S, et al. High-dimensional continuous control using generalized advantage estimation[J]. arXiv preprint arXiv:1506.02438, 2015.

[31] MNIH V, BADIA A P, MIRZA M, et al. Asynchronous Methods for Deep Reinforcement Learning [J/OL]. CoRR, 2016, abs/1602.01783. arXiv: 1602.01783. http://arxiv.org/abs/1602.01783.

[32] SCHULMAN J, LEVINE S, ABBEEL P, et al. Trust region policy optimization[C]//International conference on machine learning. 2015: 1889-1897.

[33] SCHULMAN J, WOLSKI F, DHARIWAL P, et al. Proximal policy optimization algorithms[J]. arXiv preprint arXiv:1707.06347, 2017.

[34] YASHASWI K. Deep reinforcement learning for portfolio optimization using latent feature state space (lfss) module[J]. arXiv preprint arXiv:2102.06233, 2021.

[35] SADIGHIAN J. Extending deep reinforcement learning frameworks in Cryptocurrency market making[J]. arXiv preprint arXiv:2004.06985, 2020.

[36] HUNTER J S. The exponentially weighted moving average[J]. Journal of quality technology, 1986, 18(4): 203-210.

[37] BELAFSKY P C, POSTMA G N, KOUFMAN J A. Validity and reliability of the reflux symptom index (RSI)[J]. Journal of voice, 2002, 16(2): 274-277.

[38] KRITZMAN M, LI Y. Skulls, financial turbulence, and risk management[J]. Financial Analysts Journal, 2010, 66(5): 30-41.

[39] MAREK P, SEDIVA B. Optimization and Testing of RSI[C]//11th International Scientific Conference on Financial Management of Firms and Financial Institutions. 2017.

[40] SHARPE W F. Mutual fund performance[J]. The Journal of business, 1966, 39(1): 119-138.

[41] SORTINO F A, PRICE L N. Performance measurement in a downside risk framework[J]. the Journal of Investing, 1994, 3(3): 59-64.

[42] JORION P. Value at risk: the new benchmark for managing financial risk[M]. The McGraw-Hill Companies, Inc., 2007.

[43] MAGDON-ISMAIL M, ATIYA A F. Maximum drawdown[J]. Risk Magazine, 2004, 17(10): 99-102.

[44] ORMOS M, URBÁN A. Performance analysis of log-optimal portfolio strategies with transaction costs[J]. Quantitative Finance, 2013, 13(10): 1587-1597.

[45] CHAN PHOOI M’NG J, MEHRALIZADEH M. Forecasting East Asian indices futures via a novel hybrid of wavelet-PCA denoising and artificial neural network models[J]. PloS one, 2016, 11(6): e0156338.

[46] LI B, HOI S C. Online portfolio selection: A survey[J]. ACM Computing Surveys (CSUR), 2014, 46(3): 1-36.

[47] COVER T M. Universal portfolios[J]. Mathematical finance, 1991, 1(1): 1-29.

[48] HENDERSON P, ISLAM R, BACHMAN P, et al. Deep reinforcement learning that matters[C]// Proceedings of the AAAI conference on artificial intelligence: vol. 32: 1. 2018.

附录

部分量化环境模拟

```
class CryptoTradingEnv(gym.Env):
    def _initiate_state(self):
        if self.crypto_dim > 1:
            state = (
                [self.cash] + self.num_crypto_shares +
                   self.data.close.values.tolist() +
                   self.data.open.values.tolist() +
                   self.data.high.values.tolist() +
                   self.data.low.values.tolist() +
                   self.data.volume.values.tolist()
            )
        else:
            state = (
                [self.cash] + self.num_crypto_shares +
                   [self.data.close] + [self.data.open] +
                   [self.data.high] + [self.data.low] +
                   [self.data.volume]
            )
        return state


    def _update_state(self):
        if self.crypto_dim > 1:
            state = (
                [self.state[0]] + list(self.state[1 :
                   (self.crypto_dim + 1)]) +
                   self.data.close.values.tolist() +
                   self.data.open.values.tolist() +
                   self.data.high.values.tolist() +
                   self.data.low.values.tolist() +
                   self.data.volume.values.tolist()
            )
        else:
            state = (
                [self.state[0]] + list(self.state[1 :
                   (self.crypto_dim + 1)]) + [self.data.close] +
                   [self.data.open] + [self.data.high] +
                   [self.data.low] + [self.data.volume]
            )
        return state


    def _calculate_portfolio_value(self, cash, crypto_prices,
       num_crypto_shares):
        return cash + np.sum(num_crypto_shares * crypto_prices)
```

```
def _sell_share(self, index, action):
    if self.state[index + 1] > 0:
        sell_num_shares = min(abs(action) / self.state[index +
            self.state_interval * self.crypto_dim + 1],
            self.state[index + 1])
        sell_amount = (self.state[index + self.state_interval *
            self.crypto_dim + 1] * sell_num_shares * (1 -
            self.sell_cost_pct[index]))
        self.state[0] += sell_amount
        self.state[index + 1] -= sell_num_shares
        self.cum_sell_cost += (self.state[index +
            self.state_interval * self.crypto_dim + 1] *
            sell_num_shares * self.sell_cost_pct[index])
        self.sell_trades += 1
    else:
        sell_amount = 0

    return sell_amount


def _buy_share(self, index, action):
    if self.state[0] > 0:
        buy_amount = min(self.state[0], action)
        buy_num_shares = buy_amount / (self.state[index +
            self.state_interval * self.crypto_dim + 1] * (1 +
            self.buy_cost_pct[index]))
        self.state[0] -= buy_amount
        self.state[index + 1] += buy_num_shares
        self.cum_buy_cost += (self.state[index +
            self.state_interval * self.crypto_dim + 1] *
            buy_num_shares * self.buy_cost_pct[index])
        self.buy_trades += 1
    else:
        buy_amount = 0

    return buy_amount

def _reward_setting(self, terminal):
    if terminal: reward =
       np.log(self.portfolio_values_memory[-1] /
       self.portfolio_values_memory[0]) * self.timestamp /
       self.data_granularity
    elif self.timestamp == self.state_interval - 1: reward = 0
    else:
        reward = np.log(self.portfolio_values_memory[-1] /
            self.portfolio_values_memory[-2])
        if self.timestamp > self.eval_time_interval and
            self.risk_control:
             short_term_reward =
                np.log(self.portfolio_values_memory[-1] /
```

```
                    self.portfolio_values_memory[-
                    self.eval_time_interval + self.state_interval])
                    * self.alpha * self.eval_time_interval /
                    self.data_granularity
                max_downturn_flag =
                    (self.portfolio_values_memory[-1] /
                    self.max_portfolio_value) < (1 -
                    self.max_downturn_threshold)
                reward += short_term_reward
                reward -= max_downturn_flag *
                    self.max_downturn_threshold * self.beta
                if (max_downturn_flag): self.immediate_sell = 1
                if (self.max_portfolio_value <
                    self.portfolio_values_memory[-1]):
                    self.max_portfolio_value =
                    self.portfolio_values_memory[-1]
        return reward


    def step(self, actions):

        self.state = self._update_state()
        self.terminal = self.timestamp >= self.max_timestamp
        if self.timestamp == self.state_interval - 1:
            self.rewards_memory = []
            self.actions_memory = []
            self.valid_actions_memory = []
            self.states_memory = []
            self.portfolio_values_memory = []

        if self.immediate_sell == 1:
            actions = np.zeros(self.crypto_dim) -
                self.portfolio_value
            self.immediate_sell = 0
            self.max_portfolio_value =
                self.portfolio_values_memory[-1]
        self.actions_memory.append(actions)
        self.portfolio_value =
            self._calculate_portfolio_value(self.state[0],
            np.array(self.state[(self.state_interval *
            self.crypto_dim + 1) : ((self.state_interval + 1) *
            self.crypto_dim + 1)]), np.array(self.state[1 : 1 +
            self.crypto_dim]))
        scaled_actions = actions * self.portfolio_value *
            self.action_scaling
        argsort_actions = np.argsort(actions)
        sell_index = argsort_actions[: np.where(actions <
            0)[0].shape[0]]
        buy_index = argsort_actions[::-1][: np.where(actions >
            0)[0].shape[0]]
        for index in sell_index: scaled_actions[index] =
```

```
            self._sell_share(index, scaled_actions[index]) * (-1)
        for index in buy_index: scaled_actions[index] =
            self._buy_share(index, scaled_actions[index])
        self.valid_actions_memory.append(scaled_actions /
            (self.portfolio_value * self.action_scaling))
        self.portfolio_value =
            self._calculate_portfolio_value(self.state[0],
            np.array(self.state[(self.state_interval *
            self.crypto_dim + 1) : ((self.state_interval + 1) *
            self.crypto_dim + 1)]), np.array(self.state[1 : 1 +
            self.crypto_dim]))
        self.portfolio_values_memory.append(self.portfolio_value)
        self.reward = self._reward_setting(self.terminal)
        self.rewards_memory.append(self.reward)
        self.states_memory.append(self.state)

        if self.terminal:
            if self.episode % self.print_verbosity == 0 and
               self.model_name != "":
                self.print_verbose()
        else:
            self.timestamp += 1
            self.data = self.df.loc[self.timestamp -
               self.state_interval + 1 : self.timestamp]

        return self.state, self.reward, self.terminal, {}


    def reset(self):
        self.timestamp = self.state_interval - 1
        self.data = self.df.loc[self.timestamp -
            self.state_interval + 1 : self.timestamp]
        self.state = self._initiate_state()
        self.portfolio_value =
            self._calculate_portfolio_value(self.cash,
            np.array(self.state[self.state_interval *
            self.crypto_dim + 1 : (self.state_interval + 1) *
            self.crypto_dim + 1]), np.array(self.num_crypto_shares))
        self.cum_buy_cost = 0
        self.cum_sell_cost = 0
        self.buy_trades = 0
        self.sell_trades = 0
        self.terminal = False
        self.immediate_sell = 0
        self.max_portfolio_value = self.portfolio_value
        self.episode += 1

        return self.state
```

致谢

时光荏苒，四年的大学本科生活即将结束，回首这段时光，许多美好的回忆仿佛又历历在目，感谢陪伴我成长的老师、学长和同学，在我最美好的一段青春岁月里给予我最无私的帮助，在即将分别之际，向你们表达我最真挚的感激。

感谢室友杨浩然和陈越，我们都是在大学二年级通过插班生考试一起踏入华师大校园，春华秋实、夏涵冬蕴，我们朝夕相处一起为更好的梦想努力；栉风沐雨、玉汝于成，我们额冠相庆一起分享成功的喜悦和快乐。志同道合、惺惺相惜，在这个秋天我们将分赴交复读研，愿我们友谊长存，地久天长！

感谢李云帆、徐涵钰、周辛娜、周子彦、雷超晶和张春贤这几位好友，感谢你们在学习上一直陪伴我左右，你们敢于质疑，勇于探索，鞭策我不断进步。在生活中，我们相互关心，守望相助，感谢你们已经成为我生命中不可或缺的存在！

感谢倪葹老师！从《概率论与数理统计》到《统计方法与机器学习》，倪老师带我入门统计学的殿堂，真正激发我学习的兴趣，在课余时间，还指导我参加美赛和双创项目，热心推荐我攻读交大安泰管理科学与工程，感谢您的悉心教导，我将一定不负所望！

感谢兰韵诗老师的《深度学习》与陆雪松老师的《计算机视觉》，让我在大三就能感受到深度学习的浩瀚海洋，领略了自然语言处理与视觉技术的无穷奥妙。同时也感谢黄波老师的《事业启航》，让我在面试的时候能把握自信，在大三暑期获得云从科技的实习岗位。

感谢毕业设计指导李翔老师！论文写作过程中，李老师给予我很大的帮助，他严谨的治学态度和扎实的专业知识，为我树立了很好的学术榜样，虽然李老师工作繁忙，仍然不厌其烦、耐心指导，感谢李老师的悉心教诲，让我在毕业设计中收获满满。

感谢尊敬的钱院长！至今还记得院长亲自给我们上了第一门专业课《计算机系统与云计算》，那段时间我收获了在数据学院求学的方法论，更加深入理解了数据科学与系统设计构造的关联，感谢钱院长的谆谆教导，我永远会铭记数据科学与工程学院是我学术启航的地方。

感谢辅导员杨兴龙老师！在我华师大求学旅程中，杨老师一直默默地帮助我们处理各种学院事务，无论是大事还是小事，他都会耐心地帮我们解决。他关心我们的生活，关注我们的成长，在我们遇到困难时，总是第一时间伸出援手。在这几年里，杨老师无微不至的照顾让我感受到了家人般的温暖。

感谢我的父母、外婆和爷爷奶奶！无论何时何地，你们都默默地支持着我。在我遇到学习的难关和生活的困境时，总是给予我鼓励和关爱。你们的陪伴和信任，让我勇敢地面对挑战，迈过一个又一个难关。家人的支持是我前进路上永恒的动力，你们永远是我心灵的避风港。

感谢我从小的伙伴张乐辰，我们从小相伴成长，一起学钢琴骑单车，那是一段无忧无虑的童年，乐宸现在密西根大学安娜堡分校攻读数据科学方向的研究生，虽然我们现在相隔万里，但纯真友谊永远不会改变，衷心祝你学有所成，身体健康！

最后，我要感谢所有曾经帮助过我的老师、同学和朋友们！在我求学路上，你们的关心、鼓励与支持是我前进的动力，感谢你们为我的成长付出的时间和精力，你们的善意和关爱会永远珍藏在我心里，也道一声彼此珍重，好人一生平安！