O2O-TP: An Offline-to-Online Transformer-Based Reinforcement Learning Approach for Portfolio Management

Yuming Liao, Yuanxin Wei, Dan Huang, Yutong Lu*
School of Computer Science and Engineering
Sun Yat-Sen University, Guangzhou, China
{liaoym6, weiyx25}@mail2.sysu.edu.cn, {huangd79, luyutong}@mail.sysu.edu.cn

***Abstract*—Portfolio management allocates wealth across assets to maximize returns. Reinforcement learning (RL) is promising, but it must balance risk-controlled decisions with rapid adaptation to changing markets. Offline RL trains reliably on fixed historical data using supervised-style objectives, yet its performance is limited by dataset quality and coverage. Online RL can adapt through exploration, but it requires costly real-time interaction with the environment. We introduce O2O-TP, an offline-to-online Transformer RL framework that pretrains on historical data and then refines online with low computational overhead. O2O-TP embeds a Dirichlet policy head to produce simplex-constrained portfolio weights, enabling deterministic or stochastic execution. Across four stock markets, O2O-TP achieves cumulative wealth up to 3.3 times higher than established finance and RL baselines, with low variance between random seeds. Online refinement keeps efficient, accounting for only 14% to 23% of total runtime. These results indicate that the proposed framework enables effective and computationally efficient adaptation in dynamic financial markets.**

***Index Terms*—Reinforcement learning, decision transformer, online fine-tuning, portfolio management, financial trading**

I. Introduction

Reinforcement learning is well-suited for financial portfolio management. In this sequential decision-making setting, an RL agent repeatedly allocates capital among assets under uncertainty. The agent's performance is evaluated by both profit and risk. This approach supports policies that dynamically adjust portfolio weights in response to market fluctuations, rather than relying on a static allocation strategy.

Offline RL learns from fixed historical datasets, so its performance hinges on dataset coverage and label quality. Although expert trajectories can produce strong policies, noisy or suboptimal trajectories often lead to severe degradation. Moreover, training solely on static data leaves agents unable to adapt to evolving market conditions, where non-stationarity and volatility require continuous adjustment.

Online RL has been widely explored for financial decision-making, with backtesting results indicating potential profitability. Methods such as DeepScalper [1] and DeepTrader [2] integrate temporal convolution layers, a spatial attention mechanism, and graph convolution networks to enrich state representations. Other approaches, including EarnHFT [3] and MacroHFT [4], adopt hierarchical RL for multi-market adaptation. Multimodal designs such as PROFIT [5] and SARL [6] incorporate textual information from news and market sentiment to improve robustness. Despite these advances, a fundamental limitation remains: online RL depends on repeated interaction with financial environments, which is computationally costly, unstable, and often impractical in realistic settings.

To address the limitations of offline and online learning, we introduce a hybrid approach that combines offline pretraining with online refinement. Offline pretraining uses a causal Transformer to model fixed-length histories of past states, portfolio weight allocations, and rewards. It then outputs a valid portfolio weight vector. Online refinement uses a rolling window of recent rollouts and replays the stored experience to stabilize and improve updates, allowing the policy to track distribution shifts with minimal additional interaction. The Decision Transformer (DT) for offline sequence modeling was first introduced by Chen et al. [7]. DT is analogous to next-token prediction in language models, as it predicts an action based on the preceding trajectory context. Zheng et al. [8] extended DT with an online refinement stage and reported strong results on MuJoCo robotic control benchmarks, motivating an **offline-to-online Transformer pipeline for portfolio management**.

However, financial markets differ fundamentally from robotic control scenarios, hindering the direct application of DT-based RL to financial trading traces. In stock markets, rewards are noisy, immediate feedback is weak, and informative signals are often sparse. In portfolio management, allocation decisions span many assets in a non-stationary market structure [9]. Furthermore, portfolio actions must satisfy feasibility constraints on weights, including nonnegativity and a sum-to-one budget [10]. Therefore, models should enforce financial constraints and adapt to evolving market conditions.

We propose **O2O-TP**, an offline-to-online Transformer tailored to portfolio management, and make three contributions:

- A low-overhead online refinement scheme was developed to update the pretrained policy using recent experience. This method effectively tracks distribution shifts with

This research was supported by Guangdong S&T Program under Grant No. 2024B0101040005.
* Corresponding author.

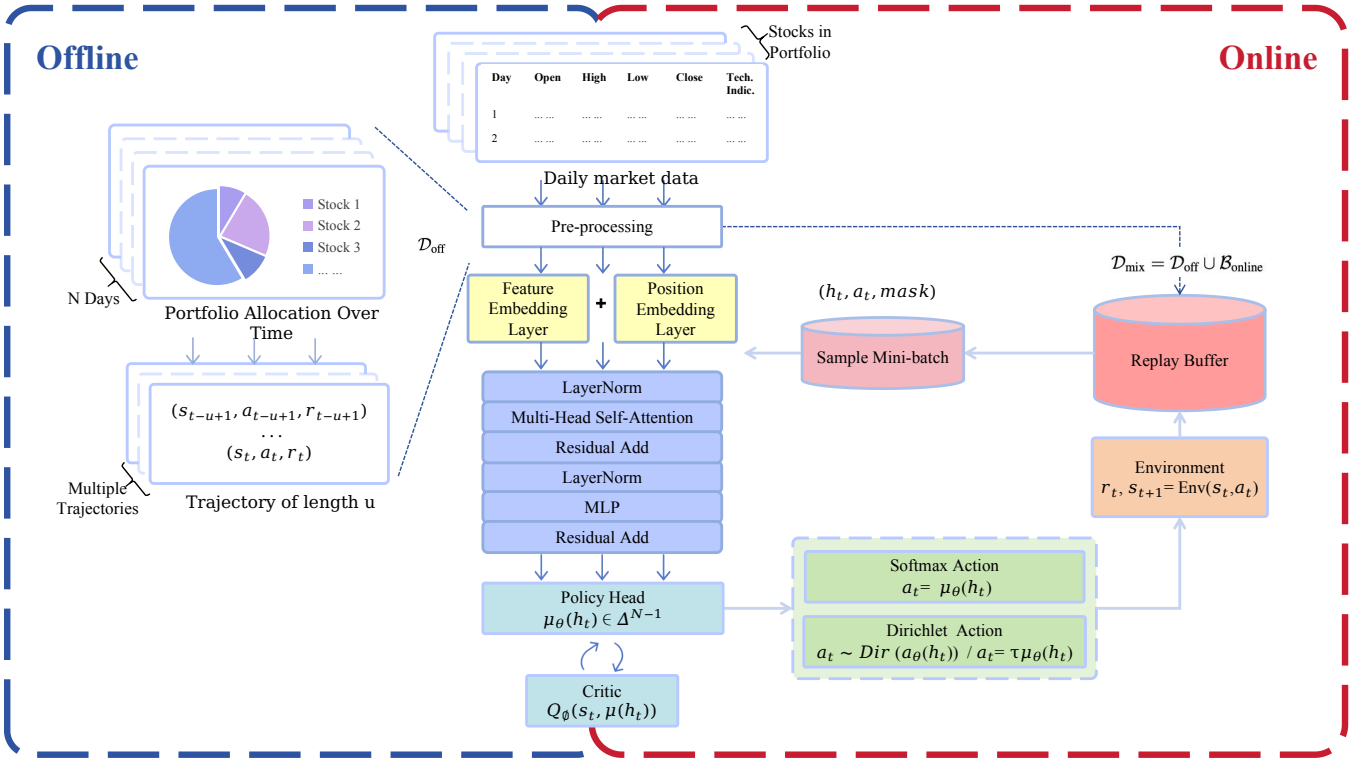


Fig. 1: Our framework consists of two stages: (1) *offline pretraining* on historical trajectories $\mathcal{D}_{\text{off}}$ to initialize the Transformer policy and critic; and (2) *online refinement* that interacts with the environment, appends new transitions to $\mathcal{B}_{\text{online}}$, and continues training by sampling from the mixed buffer $\mathcal{D}_{\text{mix}} = \mathcal{D}_{\text{off}} \cup \mathcal{B}_{\text{online}}$ using the same actor–critic update rules.

minimal computational resources, thereby addressing the non-stationarity of financial markets.

- A Dirichlet policy head outputs nonnegative, sum-to-one portfolio weights by parameterizing a Dirichlet distribution. This method enables feasible and uncertainty-aware allocations for both deterministic and stochastic execution.
- We design two deployment modes for online fine-tuning, **On-D** and **On-S**. **On-D** trains with stochastic Dirichlet actions but deploys deterministic mean allocations, whereas **On-S** samples portfolio weights from the Dirichlet distribution during inference. The relation between market structure and mode selection is further investigated, providing practical guidance for deployment.

Supplementary materials and implementation details are available.[1].

II. Method

A. Problem Formulation

We model portfolio trading as a data-driven finite-horizon decision process $(\mathcal{S}, \mathcal{A}, \mathcal{R}, T)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ the action space, $\mathcal{R}$ the per-step reward, and T the number of trading days. The transition dynamics are induced by the historical data stream and the portfolio update rule. At each step $t \in \{0, \ldots, T-1\}$, the state s_t is constructed from market information at day t, an allocation a_t is executed, and the next market observation at day $t+1$ becomes available. Portfolio variables update deterministically based on the selected allocation and realized price relatives. An offline preprocessing pipeline generates state–action–reward tuples by converting raw historical prices into standardized training trajectories, as shown in Fig. 1.

State: At step t, the state $s_t \in \mathbb{R}^d$ concatenates OHLC prices for each asset with technical indicators, where d denotes the state dimension. All features are standardized by z-score normalization, using the mean and standard deviation calculated at the dataset level.

Action: The action $a_t \in \mathcal{A}$ is a portfolio weight vector $a_t = (a_t^{(1)}, \ldots, a_t^{(N)})$ with $a_t^{(i)} \geq 0$ and $\sum_{i=1}^{N} a_t^{(i)} = 1$, where N is the number of assets. Let u denote the context length and define the causal history as

$$h_t = \{(s_{t-u+1}, a_{t-u+1}, r_{t-u+1}), \ldots, (s_{t-1}, a_{t-1}, r_{t-1})\}. \tag{1}$$

We parameterize the policy as a Dirichlet distribution [11] over allocations conditioned on h_t:

$$a_t \sim \pi_\theta(\cdot \mid h_t) = \text{Dirichlet}\big(\alpha_\theta(h_t)\big), \tag{2}$$

$$\alpha_\theta(h_t) = \tau\, \mu_\theta(h_t),\ \mu_\theta(h_t) \in \Delta^{N-1},\ \tau > 0, \tag{3}$$

[1]https://github.com/theyml/O2O-TP

where Δ^{N-1} denotes the probability simplex. This enables calibrated stochastic exploration via the concentration parameter τ.

For comparison, we consider a deterministic baseline where the allocation $\mu_\theta(h_t) \in \Delta^{N-1}$ is parameterized by a softmax head.

Reward and portfolio update: Let $\mathbf{p}_t \in \mathbb{R}^N_{>0}$ denote the close-price vector at day t and define the element-wise gross returns as

$$\mathbf{x}_{t+1} = \mathbf{p}_{t+1} \oslash \mathbf{p}_t, \tag{4}$$

where $\oslash$ denotes element-wise division.

The per-step reward reflects the portfolio return net of transaction costs,

$$r_t = a_t^\top(\mathbf{x}_{t+1} - \mathbf{1}) - c\|a_t - a_{t-1}\|_1, \tag{5}$$

where $c \geq 0$ is the transaction-cost coefficient, and we initialize a_{t-1} as the uniform allocation at the beginning of each episode.

Wealth then evolves as

$$V_{t+1} = V_t(1 + r_t), \tag{6}$$

and episodes terminate on the final trading day.

Learning objective: We optimize the policy with two alternative losses depending on the parameterization.

Dirichlet objective:

$$\begin{aligned}\min_\theta \ \mathcal{L}_{\mathrm{Dir}}(\theta;\mathcal{D}) = \beta\, \mathbb{E}_{(h_t,a_t)\sim\mathcal{D}}\big[-\log \pi_\theta(a_t \mid h_t)\big] \\ + \ \eta\, \mathbb{E}_{h_t\sim\mathcal{D}}\big[-H\big(\pi_\theta(\cdot \mid h_t)\big)\big] \\ - \ \lambda\, \mathbb{E}_{h_t\sim\mathcal{D}}\big[Q_\phi(s_t, \mu_\theta(h_t))\big],\end{aligned} \tag{7}$$

where $\pi_\theta(\cdot \mid h_t) = \mathrm{Dirichlet}\big(\alpha_\theta(h_t)\big)$ is the allocation distribution and $\mu_\theta(h_t) \in \Delta^{N-1}$ denotes its mean. Here $\beta \geq 0$ weights the supervised imitation term and $\eta \geq 0$ controls entropy regularization; λ is defined in (11).

Softmax (MSE) objective:

$$\begin{aligned}\min_\theta \ \mathcal{L}_{\mathrm{MSE}}(\theta;\mathcal{D}) = \mathbb{E}_{(h_t,a_t)\sim\mathcal{D}}\left[\|\mu_\theta(h_t) - a_t\|_2^2\right] \\ - \lambda\, \mathbb{E}_{h_t\sim\mathcal{D}}\left[Q_\phi(s_t, \mu_\theta(h_t))\right],\end{aligned} \tag{8}$$

where $\mu_\theta(h_t) = \mathrm{softmax}(f_\theta(h_t)) \in \Delta^{N-1}$ is the deterministic allocation and a_t is the action stored in the buffer (offline demonstration actions and executed actions from online rollouts). When $\lambda = 0$, (8) reduces to pure behavioral cloning.

Critic (Q-function) learning: We learn an action-value function $Q_\phi(s, a)$ with a one-step temporal-difference (TD) objective using target networks, i.e., we regress the current estimate $Q_\phi(s_t, a_t)$ toward a bootstrap target formed by the immediate reward and the discounted next-step value.

For a transition $(s_t, a_t, r_t, s_{t+1}, m_t) \sim \mathcal{D}$ sampled from the training buffer, we form the bootstrap target

$$y_t = r_t + (1 - d_t)\gamma\, Q_{\bar\phi}(s_{t+1}, \mu_{\bar\theta}(h_{t+1}))\,, \tag{9}$$

where $\gamma \in (0, 1]$ is the discount factor, $m_t \in \{0, 1\}$ indicates episode termination, and $\bar\phi, \bar\theta$ denote target-network parameters.

We then minimize the squared TD error

$$\min_\phi \ \mathcal{L}_Q(\phi;\mathcal{D}) = \mathbb{E}\left[\big(Q_\phi(s_t, a_t) - y_t\big)^2\right]. \tag{10}$$

In our implementation, a_t in (10) is the action stored in the buffer (offline demonstration actions and executed actions from online rollouts), whereas the policy update queries $Q_\phi(s_t, \mu_\theta(h_t))$.

We set

$$\lambda = \kappa / \mathbb{E}[|Q_\phi(s_t, \mu_\theta(h_t))|]\,, \tag{11}$$

where $\kappa > 0$ is a scaling hyperparameter, and the expectation is computed over each minibatch.

Offline pretraining uses historical trajectories $\mathcal{D}_{\mathrm{off}}$, while online refinement draws from the mixed buffer $\mathcal{D}_{\mathrm{mix}} = \mathcal{D}_{\mathrm{off}} \cup \mathcal{B}_{\mathrm{online}}$, with $\mathcal{B}_{\mathrm{online}}$ collecting trajectories from rolling-window interactions.

B. Offline Learning Strategy

In the first stage, we pretrain the model on historical market trajectories without collecting additional interactions, as shown on the left side of Fig. 1. Each trajectory consists of a sequence of state-action-reward tuples. We use a GPT-style causal Transformer with positional embeddings [12], [13] to encode these sequences, capturing temporal dependencies within a fixed context length u. Raw input features are standardized based on statistics computed from the training set.

The policy head can be selected between Dirichlet and Softmax. And, both heads produce feasible portfolio weights that are nonnegative and sum-to-one. With the Dirichlet head, the network predicts concentration parameters that induce a distribution over portfolio allocations. And, actions are executed using the distribution mean. The Softmax head outputs logits and determines allocations by applying the softmax function. Supervised learning is employed to fit the policy to actions from offline trajectories. The training objective minimizes the negative log-likelihood (7) of observed actions under the Dirichlet distribution, with an optional entropy term. For numerical stability, target weights are ε-clipped and normalized, and Dirichlet concentrations are parameterized to remain strictly positive.

C. Online Learning Strategy

Building on the offline initialization, the online stage further refines the policy using rolling market windows, as illustrated in Fig. 1. During each online iteration, the agent interacts with a window of length H, which advances by a stride of S. The collected transitions are added to a bounded replay buffer, which is initialized with offline trajectories. Model updates sample mini-batches from this mixed buffer and optimizes the same loss function as used in offline pretraining. The data distribution gradually shifts as new experiences are accumulated.

We consider two Dirichlet-based execution modes: deterministic execution of a Dirichlet policy using its mean, and

TABLE I: Experimental hyperparameters and settings.

Parameter	Value
Transformer embedding dimension	128
Transformer layers	5
Dropout	0.1
Context length (u)	40
Optimizer	AdamW
Actor / Critic learning rate	10^{-4} / 10^{-6}
Weight decay	10^{-4}
Warm-up steps	5,000
Offline pretraining (iters / steps per iter)	10 / 2,000
Online refinement (iters / updates per iter)	10 / 1,000
Replay buffer size	1,000
Online rollout horizon (H)	40
Online trade stride (S)	20
Dirichlet concentration scale (τ)	50

stochastic execution by sampling from the Dirichlet distribution. Within the Dirichlet framework (3), τ modulates exploration by influencing the dispersion of sampled allocations. A smaller τ increases variance and encourages exploration, whereas a larger τ concentrates actions near the mean. Sampling may be centered on the current prediction or a uniform portfolio, and employing a convex combination of the two mitigates the risk of overly concentrated allocations.

D. Implementation Details and Hyperparameters

We employ a GPT-style causal Transformer as the actor and utilize an MLP-based critic network to facilitate stable TD learning.

We optimize the model with AdamW, incorporate weight decay, and implement a linear learning-rate warm-up schedule. Offline pretraining is conducted over multiple iterations, each with a fixed number of gradient updates, and is subsequently followed by online refinement utilizing a bounded replay buffer. Table I summarizes key hyperparameters.

III. Experiment

A. Datasets and Environment

We evaluate our method in standardized market environments from FinRL-Meta [14] across four international equity universes. Table II summarizes the market universes, the number of assets N, and the month-level train/trade splits.

At each decision time, the agent observes a state formed by concatenating OHLC prices with technical indicators, including SMA, EMA, momentum, rate of change, rolling volatility, ATR, RSI, CCI, DX, MACD, and Bollinger upper and lower bands [15], for all tradable assets. Reallocations incur a transaction cost of 0.25%, and features are standardized using training-split statistics.

B. Framework

a) Baseline:

All selected baselines are reproducible under the same portfolio management setting and benchmark pipeline, enabling a fair, controlled, and consistent evaluation across portfolio decision-making methods. Some newer approaches are discussed in related work but target different task formulations, making direct comparison not fully compatible with our setting.

- **MOM**: Equal-weights the three-month winners and applies a one-month skip [16]. This serves as a classical finance baseline.
- **DDQN**: A value-based online RL agent that learns Q-values from interaction and acts greedily at evaluation [17]. It represents a classical value-based RL approach for sequential portfolio decision-making.
- **EIIE**: An online model-free RL framework instantiated with CNN/RNN/LSTM policies, directly maps market features to portfolio weights [18]. It provides a classic end-to-end RL design for the portfolio management problem.
- **DeepTrader**: An online deep RL portfolio model incorporates macro market conditions and graph-structured inter-asset dependencies [2]. It serves as a market-aware deep RL baseline with richer representation modeling.

b) O2O-TP Configurations:

We report a *hierarchical* ablation in which variants are organized as incremental modifications to a simpler baseline: we start from offline-only MSE learning, then (i) switch the head/supervision from MSE to DIR, (ii) enable online refinement, and (iii) enable stochastic action sampling during online data collection.

- **MSE$^{\text{off}}$**: Offline-only training with an MSE objective.
- **DIR$^{\text{off}}$**: Replaces the policy head in MSE$^{\text{off}}$ with a Dirichlet head, and executes actions deterministically via the mean allocation.
- **MSE$^{\text{on}}$**: Adds an online refinement to MSE$^{\text{off}}$ and executes online actions deterministically.
- **DIR$^{\text{on-D}}$**: Refines DIR$^{\text{off}}$ online and executes the mean allocation deterministically during online interaction.
- **DIR$^{\text{on-S}}$**: Extends DIR$^{\text{on-D}}$ by sampling actions from the Dirichlet policy during online data collection.

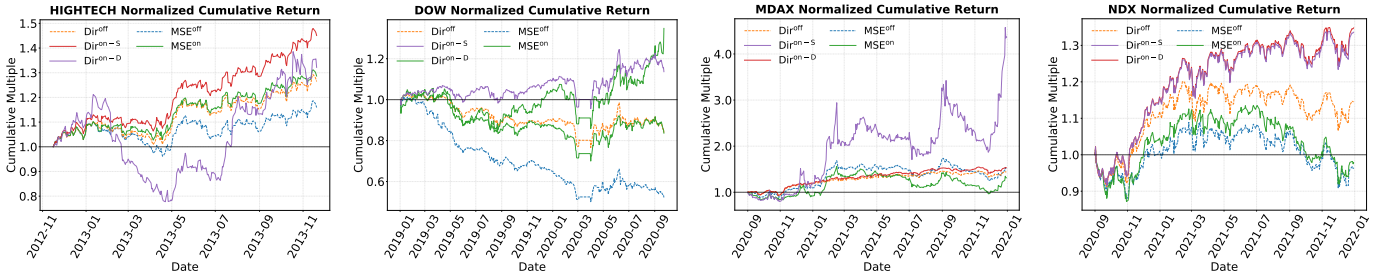
TABLE II: Dataset splits (month-level).

Market	N	Train (YYYY-MM)	Trade (YYYY-MM)
DOW	30	2009-01 $\sim$ 2018-12	2019-01 $\sim$ 2020-09
Hightech	9	2006-10 $\sim$ 2012-11	2012-11 $\sim$ 2013-11
MDAX	30	2009-01 $\sim$ 2020-08	2020-09 $\sim$ 2021-12
NASDAQ	30	2009-01 $\sim$ 2020-08	2020-09 $\sim$ 2021-12

C. Metrics

To provide a holistic assessment of profitability and risk, we report both return-based and risk-adjusted metrics that capture total growth, volatility, downside risk, and drawdown behavior across market regimes.

Specifically, we evaluate Cumulative Wealth (CW) for overall capital growth, Annualized Percentage Yield (APY) for



(a) Hightech (b) DOW (c) MDAX (d) NASDAQ

Fig. 2: Cumulative returns on four markets in the trading period.

TABLE III: Wall-clock time breakdown (mean $\pm$ std over $K=3$ seeds) between offline pretraining and online refinement. Online refinement requires low overhead and represents only a small portion of the total runtime.

Dataset	Offline total (s)	Online total (s)	Offline/Online
DOW	4210 ± 3882	814 ± 514	4.60 ± 1.37
Hightech	1640 ± 40	422 ± 12	3.89 ± 0.06
MDAX	5391 ± 312	910 ± 190	6.17 ± 1.78
NASDAQ	5109 ± 339	1209 ± 131	4.28 ± 0.79

annualized returns, Certainty Equivalent Return (CER) as a utility-based risk-adjusted return, Annualized Sharpe Ratio (ASR) as return per unit of total volatility, Sortino Ratio (SoR) as return per unit of downside volatility, Calmar Ratio (CR) as return relative to maximum drawdown, Maximum Drawdown (MD) as the worst peak-to-trough loss, and Annualized Volatility (AVO) as the overall risk exposure.

Among these metrics, CW is treated as the primary indicator because it most directly reflects overall financial gain. ASR and MD serve as the main supporting metrics for evaluating performance from the perspectives of risk-adjusted return and downside risk, respectively, while the remaining metrics provide complementary views of return and risk.

D. Main Results

On trading data, we evaluate all methods within the TradeMaster framework. Across all four markets, O2O-TP variants outperform MOM, DDQN, EIIE, and DeepTrader in both returns and risk-adjusted metrics. Table IV and Figure 2 present these results, including ablations of the policy head (MSE versus Dirichlet), the learning pipeline (offline versus offline-to-online), and the online deployment mode (sampling-based versus deterministic).

On Hightech, DIR$^{\text{on-S}}$ achieves the highest cumulative wealth and the best Sharpe ratio, demonstrating early growth and maintaining the strongest performance curve. On DOW, DIR$^{\text{on-D}}$ yields the smoothest wealth trajectory and achieves the highest Sharpe, Sortino, and Calmar ratios. On MDAX, DIR$^{\text{on-D}}$ attains the highest cumulative wealth but experiences larger drawdowns. DIR$^{\text{on-S}}$ offers a strong balance between return and risk. On NASDAQ, DIR$^{\text{on-S}}$ delivers the strongest risk-adjusted performance, with cumulative wealth comparable to DIR$^{\text{on-D}}$.

In addition to performance, online refinement is low-overhead. Table III shows that it accounts for only 14%–23% of total runtime across markets. Additionally, DIR$^{\text{on-S}}$ is robust across random seeds, exhibiting low mean$\pm$std variance over markets and metrics, as shown in Table V.

IV. Discussion

A. Ablation Discussion

The ablation results demonstrate that incorporating online refinement into offline initialization produces the most substantial improvements. Offline-only variants MSE$^{\text{off}}$ and DIR$^{\text{off}}$ consistently underperform relative to their offline-to-online counterparts. These findings suggest that offline pretraining alone is inadequate in non-stationary markets, whereas online refinement effectively bridges the gap between historical training data and the trading period. Furthermore, Dirichlet-head variants consistently outperform MSE-head variants, indicating that a distributional, constraint-aware parameterization enhances portfolio actions. The Dirichlet head generates feasible nonnegative, sum-to-one weights and introduces controlled stochasticity, thereby capturing uncertainty more effectively and improving robustness to noisy market feedback.

The low runtime required for the online stage substantiates the primary claim that online refinement can be conducted with low computational overhead, thereby enabling frequent updates in non-stationary markets. Refinement over rolling market windows allows the policy to continuously integrate recent trajectories, enhancing adaptivity with few refinement iterations and limited additional computation.

In addition, DIR$^{\text{on-S}}$ exhibits low seed variance, indicating stable stochastic action execution during online data collection. This stability results from the controlled stochasticity of Dirichlet sampling, regulated by the concentration parameter, which encourages exploration without introducing excessive randomness that could destabilize learning.

B. Execution Mode Analysis

Table VI shows that the optimal online execution mode for DIR depends on the market. The deterministic deployment mode produces higher cumulative wealth on DOW and

TABLE IV: Standard baselines and O2O-TP variants, including both MSE and Dirichlet head settings, are reported. Offline-only variants function as ablation controls to evaluate the benefit of online refinement. Arrows indicate directionality: ↑ denotes that larger values are preferable, while ↓ denotes that smaller values are preferable.

		Hightech								**DOW**							
Type	**Framework**	CW↑	APY↑	CER↑	MD↓	AVO↓	ASR↑	CR↑	SoR↑	CW↑	APY↑	CER↑	MD↓	AVO↓	ASR↑	CR↑	SoR↑
Baseline	MOM	0.987	-0.013	-0.112	0.313	0.314	0.115	-0.042	0.131	1.112	0.063	-0.023	0.317	0.290	0.356	0.199	0.366
	DDQN	1.315	0.311	0.222	0.149	0.222	1.333	2.093	2.122	0.989	-0.006	-0.074	0.286	0.260	0.106	-0.021	0.145
	EIIE	0.965	-0.036	-0.086	0.190	0.223	-0.052	-0.189	-0.070	0.511	-0.327	-0.540	0.611	0.379	-0.854	-0.536	-1.090
	DeepTrader	0.709	-0.022	-0.022	0.291	0.003	-7.467	-0.076	-3.070	0.479	-0.024	-0.024	0.524	0.021	-1.135	-0.046	-1.121
Ours	MSE^{off}	1.175	0.174	0.131	0.133	0.169	1.032	1.308	1.653	0.498	-0.332	-0.452	0.524	0.219	-1.729	-0.635	-2.267
	MSE^{on}	1.292	0.290	0.229	0.067	0.159	1.677	4.335	2.659	0.839	-0.097	-0.134	0.257	0.180	-0.474	-0.376	-0.580
	DIR^{off}	1.265	0.262	0.207	0.085	0.162	1.515	7.000	2.368	1.137	0.077	0.042	0.141	0.179	0.505	0.546	0.614
	$\text{DIR}^{\text{on-S}}$	**1.450**	**0.445**	**0.343**	**0.060**	**0.159**	**2.375**	**7.469**	**3.811**	1.138	0.078	0.043	0.141	0.179	0.510	0.552	0.619
	$\text{DIR}^{\text{on-D}}$	1.322	0.319	0.174	0.359	0.322	1.022	0.889	1.732	**1.348**	**0.189**	**0.109**	**0.201**	**0.253**	**0.810**	**0.940**	**1.031**
		MDAX								**NASDAQ**							
Type	**Framework**	CW↑	APY↑	CER↑	MD↓	AVO↓	ASR↑	CR↑	SoR↑	CW↑	APY↑	CER↑	MD↓	AVO↓	ASR↑	CR↑	SoR↑
Baseline	MOM	1.323	0.231	0.172	0.145	0.189	1.194	1.592	1.538	1.204	0.149	0.095	0.166	0.211	0.767	0.901	0.917
	DDQN	1.441	0.311	0.219	0.123	0.228	1.302	2.525	1.930	0.886	-0.087	-0.174	0.190	0.289	-0.171	-0.458	-0.243
	EIIE	0.795	-0.161	-0.244	0.335	0.264	-0.529	-0.480	-0.560	0.910	-0.070	-0.131	0.189	0.240	-0.184	-0.372	-0.230
	DeepTrader	0.830	-0.008	-0.008	0.430	0.066	-0.092	-0.019	-0.127	0.543	-0.027	-0.028	0.495	0.057	-0.457	-0.055	-0.470
Ours	MSE^{off}	1.496	0.349	0.198	0.278	0.317	1.101	1.255	1.909	0.963	-0.028	-0.071	0.165	0.206	-0.035	-0.169	-0.054
	MSE^{on}	1.315	0.225	0.018	0.381	0.430	0.686	0.591	1.097	0.978	-0.016	-0.058	0.191	0.203	0.020	-0.086	0.031
	DIR^{off}	1.437	0.309	0.236	0.112	0.183	1.566	2.761	2.216	1.146	0.108	0.067	0.113	0.188	0.640	0.955	0.948
	$\text{DIR}^{\text{on-S}}$	1.530	0.372	0.283	**0.109**	**0.182**	**1.828**	3.414	2.586	**1.348**	**0.252**	**0.189**	**0.104**	**0.188**	**1.290**	**2.419**	**1.921**
	$\text{DIR}^{\text{on-D}}$	**4.371**	**1.994**	**0.351**	0.455	0.857	1.695	**4.385**	**2.999**	1.336	0.244	0.187	0.104	0.188	1.255	2.343	1.867

TABLE V: Seed variance of $\text{DIR}^{\text{on-S}}$ across markets. The mean ± standard deviation over $K=3$ random seeds is reported. Arrows indicate whether higher or lower values are preferred. The consistently low variance across seeds indicates that the online sampling scheme is robust to random initialization and data-collection variability.

Dataset	CW↑	APY(%)↑	CER(γ=3)↑	MD(%)↓	AVO↓	ASR↑	CR↑	SoR↑
DOW	1.137 ± 0.001	7.749 ± 0.061	0.043 ± 0.001	14.140 ± 0.037	0.179 ± 0.000	0.507 ± 0.004	0.548 ± 0.006	0.616 ± 0.005
Hightech	1.448 ± 0.005	44.412 ± 0.474	0.342 ± 0.003	5.791 ± 0.025	0.161 ± 0.000	2.364 ± 0.018	7.669 ± 0.109	3.787 ± 0.031
MDAX	1.540 ± 0.013	37.919 ± 0.910	0.288 ± 0.006	10.746 ± 0.009	0.182 ± 0.000	1.854 ± 0.034	3.517 ± 0.077	2.620 ± 0.046
NDX	1.346 ± 0.006	25.080 ± 0.342	0.189 ± 0.003	10.310 ± 0.036	0.188 ± 0.000	1.286 ± 0.020	2.433 ± 0.044	1.918 ± 0.028

MDAX, while the stochastic deployment mode outperforms on Hightech and NASDAQ.

This pattern reflects a key difference in market structure during the trading period. The DOW demonstrates extremely high tail turbulence and the most frequent adverse shocks. In these conditions, stochastic execution introduces additional action noise, amplifying portfolio turnover and increasing exposure to compounding drawdowns, which ultimately erodes long-term returns. In contrast, deterministic mean execution reduces unnecessary reallocations and stabilizes the trading trajectory, resulting in improved performance.

The MDAX, by comparison, is characterized by high baseline volatility and greater cross-sectional dispersion, along with notable tail turbulence but infrequent extreme shocks. In this regime, the main risk associated with stochastic execution is not frequent shock amplification but rather weight jitter under high volatility, which increases turnover and transaction-cost drag. Deterministic execution reduces this sensitivity to sampling noise, leading to more stable allocations and enhanced risk-adjusted performance.

Hightech, by contrast, shows rapid leadership changes but only moderate tail risk. In this environment, the exploratory nature of stochastic execution enables faster reallocation to emerging winners, yielding higher returns with minimal down-

TABLE VI: **Market structure and preferred online execution** mode for DIR. The "Preferred" column indicates whether deterministic or stochastic execution yields greater cumulative wealth in each market. Market statistics are calculated on the trading split. Within each column, darker shading represents higher values.

Dataset	**Preferred**	Vol_{50}	Disp_{50}	Turb_{95}	Rot_{mean}	$\text{Shock}_{\text{freq}}$
DOW	$\text{DIR}^{\text{on-D}}$	0.129	0.011	339.1	0.245	0.037
Hightech	$\text{DIR}^{\text{on-S}}$	0.158	0.013	41.3	0.282	0.005
MDAX	$\text{DIR}^{\text{on-D}}$	0.162	0.017	167.5	0.239	0.001
NDX	$\text{DIR}^{\text{on-S}}$	0.158	0.017	165.9	0.242	0.004

Market statistics. Vol50/Disp50: 50th-percentile (median) rolling volatility and cross-sectional dispersion; Turb95: 95th-percentile turbulence; Rot_{mean} : mean leader-rotation rate; $\text{Shock}_{\text{freq}}$: mean shock frequency ($|r_m| > 3\sigma_m$) under a rolling window.

side risk. The NASDAQ demonstrates a similar pattern. Although it is volatile, the lower frequency of extreme shocks reduces the cost of exploration, allowing stochastic execution to remain effective.

This analysis is based on aggregate market statistics and does not establish causality. More rigorous validation would require sub-period confidence intervals and controlled ablation studies, such as varying the Dirichlet concentration or online update frequency while maintaining a fixed offline policy.

V. Conclusion

This study presents O2O-TP, a Transformer-based portfolio management framework that combines offline pretraining with efficient online refinement. Experimental results show improved risk-adjusted cumulative returns and significant gains in the Sharpe ratio relative to competitive baselines. The low-overhead refinement process enables frequent updates, making the approach well-suited for non-stationary markets.

References

[1] S. Sun et al., "DeepScalper: A risk-aware reinforcement learning framework to capture fleeting intraday trading opportunities," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management (CIKM)*, 2022, pp. 1858–1867.

[2] Z. Wang, B. Huang, S. Tu, K. Zhang, and L. Xu, "DeepTrader: a deep reinforcement learning approach for risk-return balanced portfolio management with market conditions embedding," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 1, pp. 643–650, 2021.

[3] M. Qin, S. Sun, W. Zhang, H. Xia, X. Wang, and B. An, "EarnHFT: efficient hierarchical reinforcement learning for high frequency trading," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 13, pp. 14669–14676, 2024.

[4] C. Zong et al., "MacroHFT: Memory augmented context-aware reinforcement learning on high frequency trading," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2024, pp. 4712–4721.

[5] Y. Ye et al., "Reinforcement-learning based portfolio management with augmented asset movement prediction states," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 1, pp. 1112–1119, 2020.

[6] R. Sawhney, A. Wadhwa, S. Agarwal, and R. Shah, "Quantitative day trading from natural language using reinforcement learning," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2021, pp. 4018–4030.

[7] L. Chen et al., "Decision transformer: reinforcement learning via sequence modeling," *Advances in Neural Information Processing Systems*, 2021, pp. 15084–15097.

[8] Q. Zheng, A. Zhang, and A. Grover, "Online decision transformer," in *Proceedings of the 39th International Conference on Machine Learning*, 2022, pp. 27042–27059.

[9] K. Xu, Y. Zhang, D. Ye, P. Zhao, and M. Tan, "Relation-aware transformer for portfolio policy learning," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI-20)*, 2020, pp. 4647–4653.

[10] S. Sun, R. Wang, and B. An, "Reinforcement learning for quantitative trading," *ACM Transactions on Intelligent Systems and Technology*, vol. 14, no. 3, pp. 1–29, 2023.

[11] J. E. Mosimann, "On the compound multinomial distribution, the multivariate beta-distribution, and correlations among proportions," *Biometrika*, vol. 49, no. 1/2, pp. 65–82, 1962.

[12] A. Radford et al., "Language models are unsupervised multitask learners," OpenAI blog, 2019.

[13] T. Wolf et al., "Transformers: state-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2020, pp. 38–45.

[14] X. Liu et al., "FinRL-Meta: Market environments and benchmarks for data-driven financial reinforcement learning," *Advances in Neural Information Processing Systems*, 35, 2022, pp.1835–1849.

[15] R. W. Colby, *The encyclopedia of technical market indicators*, 1988.

[16] N. Jegadeesh and S. Titman, "Returns to buying winners and selling losers: implications for stock market efficiency," *The Journal of Finance*, vol. 48, pp. 65–91, 1993.

[17] H. van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, 2016.

[18] Z. Jiang, D. Xu, and J. Liang, "A deep reinforcement learning framework for the financial portfolio management problem," arXiv preprint arXiv:1706.10059, 2017.

[19] Y. Li et al., "Cardinality and bounding constrained portfolio optimization using safe reinforcement learning," in *2024 International Joint Conference on Neural Networks (IJCNN)*, 2024, pp. 1–8.

[20] Z. Li and V. Tam, "Developing an attention-based ensemble learning framework for financial portfolio optimisation," in *2024 International Joint Conference on Neural Networks (IJCNN)*, 2024, pp. 1–8.

[21] R. Lin, Z. Xing, M. Ma, and R. S. T. Lee, "Dynamic portfolio optimization via augmented ddpg with quantum price levels-based trading strategy," in *2023 International Joint Conference on Neural Networks (IJCNN)*, 2023, pp. 1–8.

[22] J. Jeon, J. Park, C. Park, and U. Kang, "Frequant: A reinforcement-learning based adaptive portfolio optimization with multi-frequency decomposition," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 1211–1221.

[23] H. Niu, S. Li, and J. Li, "MetaTrader: An reinforcement learning approach integrating diverse policies for portfolio optimization," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, pp. 1573-1583.