Iranian Journal of Fuzzy Systems

Volume 23, Number 5, (2026), pp. 157-183

Original paper

University of Sistan and Baluchestan

Robust portfolio optimization using tail-risk-aware fuzzy distributional reinforcement learning

L. Jing [1] and H. Zhang [2]

[1,2] *College of Management and Economics, Tianjin University, Tianjin 300072, China*

jinglitjcj@126.com, yguangmingmei1025@126.com

Abstract

This research presents the Tail-Risk-Aware Fuzzy Distributional Reinforcement Learning (T-FDRL) framework, an innovative method for robust portfolio management in dynamic, non-stationary financial markets. By combining Fuzzy Markov Decision Processes, distributional reinforcement learning, and Wasserstein-based robust optimization, T-FDRL tackles uncertainty in market signals, heavy-tailed return distributions, and shifts in market dynamics. It employs fuzzy sets to model vague market data, captures the complete return distribution, and optimizes portfolio allocations to maximize returns while minimizing tail risks using Conditional Value at Risk (CVaR). Tested on historical data from Apple, Amazon, Google, and Microsoft stocks spanning 2016 to 2025, T-FDRL delivers a cumulative return of 1544.21%, an annualized return of 21.22%, and a Sharpe Ratio of 1.60, surpassing benchmarks like Deep Q-Networks (272.38%), Fuzzy Reinforcement Learning (1120.22%), Value-Based DQN (1067.86%), and Robust Markov Decision Processes (652.63%). With low volatility (13.23%) and a CVaR 5% of -1.64%, T-FDRL excels in risk management, especially during turbulent periods such as 2020 and 2022. This framework provides a resilient, adaptive approach to portfolio optimization, offering valuable insights for investors and policymakers navigating complex financial landscapes.

Keywords: Portfolio optimization, fuzzy reinforcement learning, fuzzy Markov decision processes, distributional reinforcement learning, tail-risk mitigation.

1 Introduction

Portfolio management in financial markets is a cornerstone of economic decision-making, with far-reaching implications for individual investors, financial institutions, and global economic stability. Effective portfolio management is vital for optimizing returns and mitigating risks, ensuring financial security for individual investors and stability for institutions [32]. Its strategic allocation of assets supports global economic resilience by adapting to dynamic market conditions and unforeseen challenges. The advent of advanced computational methods, particularly deep reinforcement learning (DRL), has revolutionized this field by enabling data-driven, adaptive strategies that optimize asset allocation under complex and volatile market conditions. These approaches leverage vast datasets, including price histories, volatility indicators, technical metrics, and macroeconomic signals, to dynamically adjust portfolios, offering significant advantages over traditional methods like mean-variance optimization pioneered by Markowitz [46]. The integration of fuzzy logic in economic decision-making further enhances these strategies by modeling ambiguous market signals and imprecise data, enabling more robust and flexible portfolio decisions in uncertain environments [54]. By incorporating fuzzy sets to represent vague or incomplete information, such as volatility or investor sentiment, these methods improve adaptability to real-world market complexities [29]. The interdisciplinary nature of portfolio management, blending finance, mathematics, statistics, and computer science, underscores its significance for researchers, practitioners, and policymakers seeking robust solutions to navigate increasingly interconnected and unpredictable financial markets [33].

Corresponding Author: L. Jing
Received: September 2025; Revised: June 2026; Accepted: August 2026.
https://doi.org/10.22111/ijfs.2026.53321.9442

As global economic systems grow more interdependent, the ability to develop strategies that withstand extreme events, such as market crashes or geopolitical shocks, becomes critical—not only for maximizing returns but also for ensuring systemic resilience in the face of multifaceted economic challenges [63]. This paper proposes a method using artificial neural networks to identify and manage straggler tasks in MapReduce-based big data infrastructures [14].

The literature reveals several critical gaps in addressing the challenges of portfolio management within the context of non-stationary, heavy-tailed financial markets. Standard DRL models, such as DQN and Proximal Policy Optimization (PPO), rely on deterministic state-action mappings and expectation-based objectives, which systematically underestimate tail risks and struggle to model ambiguous market states characterized by noisy or incomplete data [37]. For instance, DQN assumes a fixed state space and optimizes for expected returns, ignoring the heavy-tailed nature of financial returns that can lead to catastrophic losses during extreme market events, such as the 2008 financial crisis or the 2020 pandemic-induced market crash. Fuzzy Reinforcement Learning (FRL) has been proposed as a solution to handle uncertainty by representing states and actions as fuzzy sets, allowing for more flexible modeling of imprecise market signals, such as vague volatility estimates or uncertain investor sentiment [16]. By using fuzzy sets, these approaches capture the inherent ambiguity in financial data, enabling decision-making under partial observability and improving robustness to noisy inputs. Robust Markov Decision Processes (R-MDPs) attempt to address uncertainty by incorporating ambiguity in transition probabilities, offering a framework to optimize policies under worst-case scenarios within a defined uncertainty set. These models are designed to handle non-stationary environments by considering a range of possible transition dynamics, making them more resilient to changes in market behavior. However, robust MDPs rarely account for heavy-tailed distributions or integrate fuzzy representations of market signals, restricting their applicability in extreme market scenarios where both ambiguity and tail risks are prevalent. For instance, traditional robust MDPs often assume bounded uncertainty in transition probabilities without explicitly modeling the fat-tailed return distributions that characterize financial markets [21]. Another critical gap in the literature is the limited focus on non-stationarity and its impact on portfolio management. Financial markets are inherently dynamic, with statistical properties that evolve over time due to factors like changing economic policies, investor behavior, or technological disruptions [3]. Traditional DRL and robust MDP frameworks often assume stationary environments or rely on static models that fail to adapt to distributional shifts, such as those observed during the transition from bull to bear markets. The integration of Wasserstein-based robust optimization offers a promising approach to address distributional shifts by optimizing over a set of plausible distributions within a Wasserstein ball. However, existing applications of Wasserstein robustness in reinforcement learning rarely combine this with fuzzy logic or CVaR-based optimization, missing the opportunity to create a comprehensive framework that tackles ambiguity, tail risks, and non-stationarity simultaneously [24]. This work investigates the use of machine learning-assisted techniques in power side-channel analysis for the effective classification of hardware Trojans [7].

Despite the progress made by deep reinforcement learning, fuzzy logic, and robust optimization in portfolio management, several critical research gaps remain unaddressed. **First**, most DRL models (e.g., DQN, PPO) assume crisp state representations and deterministic transition kernels, rendering them incapable of handling ambiguous market signals such as imprecise volatility estimates or uncertain liquidity conditions. **Second**, these models typically optimize expected cumulative returns, thereby ignoring the heavy-tailed nature of financial return distributions – a shortcoming that leads to systematic underestimation of extreme downside risks. **Third**, existing robust MDP and Wasserstein-based approaches, while resilient to certain distributional shifts, often assume a fixed uncertainty set or static radius, lacking adaptive mechanisms to respond to rapidly changing market regimes (e.g., the COVID-19 crash or the 2022 bear market). **Fourth**, and most importantly, fuzzy logic, distributional reinforcement learning, and robust optimization have been developed largely in isolation; no unified framework integrates all three paradigms to address ambiguity, tail risks, and non-stationarity simultaneously. To bridge these gaps, we propose the Tail-Risk-Aware Fuzzy Distributional Reinforcement Learning (T-FDRL) framework, which uniquely combines fuzzy Markov decision processes, distributional RL, and Wasserstein-based robust optimization, further enhanced by CVaR-based policy optimization and adaptive uncertainty tuning. This article develops a stacking-based machine learning framework aimed at improving the prediction accuracy of patient appointment attendance in outpatient clinics [13]. This research uses interpretable machine learning models to predict startup outcomes, including funding, patenting, and exit events [38]. This study performs a forecasting analysis by combining machine learning with thematic feature grouping, focusing on the stock performance of the "Magnificent Seven" tech companies [22].

This study introduces a novel **Tail-Risk-Aware Fuzzy Distributional Reinforcement Learning (T-FDRL)** framework to address the limitations of existing approaches and advance the field of portfolio management. By integrating Fuzzy Markov Decision Processes (FMDPs), distributional reinforcement learning, and Wasserstein-based robust optimization, T-FDRL tackles the core challenges of ambiguity in market signals, heavy-tailed return distributions, and resilience to adversarial distributional shifts. FMDPs enable the framework to model ambiguous market states, such as volatility or liquidity, and actions, like portfolio weights, as fuzzy sets, allowing robust decision-making under

imprecise or noisy data. Distributional reinforcement learning enhances this by capturing the full distribution of returns, addressing the heavy-tailed risks critical in financial markets, unlike traditional expectation-based DRL models [10]. Wasserstein-based robust optimization ensures resilience to non-stationary market dynamics by optimizing over a set of plausible distributions, providing stability against unexpected market shifts. Additionally, the incorporation of Conditional Value at Risk (CVaR) in policy optimization prioritizes downside risk mitigation, aligning with regulatory standards like Basel III and ensuring conservative decision-making during market downturns [40]. The framework's objectives are to develop a robust, adaptive portfolio management strategy that maximizes returns while mitigating extreme risks, offering a comprehensive solution that balances financial performance with regulatory compliance.

The T-FDRL framework makes several novel contributions to the fields of reinforcement learning and financial engineering, addressing identified gaps in a unified and interdisciplinary manner:

- **Integration of Fuzzy Uncertainty**: T-FDRL employs FMDPs to model ambiguous market states and actions as fuzzy sets, using Gaussian membership functions to capture imprecise signals like volatility or liquidity. This approach enables robust decision-making in partially observable environments, overcoming the limitations of rigid state representations in traditional DRL models.
- **Distributional Robustness**: By leveraging Wasserstein-based optimization, the framework ensures resilience to non-stationary market dynamics and heavy-tailed risks. Optimizing over worst-case distributions enhances robustness against market shifts, addressing vulnerabilities in standard DRL approaches.
- **Tail-Risk Awareness**: The inclusion of CVaR in policy optimization focuses on mitigating losses in the worst 5% of outcomes, aligning with Basel III stress testing requirements. This ensures conservative strategies during market crises, unlike risk-agnostic DRL methods.
- **Adaptive Uncertainty Tuning**: T-FDRL dynamically adjusts robustness parameters based on market volatility, enhancing adaptability to changing conditions. This feature strengthens performance during volatile periods, such as the 2020 market crash, while maintaining efficiency in stable markets.
- **Convergence and Stability**: The framework's reliable performance is supported by rigorous analysis, ensuring stable convergence in dynamic financial markets. Computational efficiency is achieved through scalable approximations, maintaining robustness without sacrificing practicality.

The T-FDRL framework is implemented through a Distributionally Robust Fuzzy Q-Network (DR-FQNet), which estimates the full distribution of fuzzy returns using multiple quantiles. The network is trained to account for adversarial distributional shifts, with computational efficiency achieved through optimized approximations. The policy is optimized to balance expected returns with tail risk mitigation, ensuring responsiveness to market dynamics. Stability analysis confirms reliable convergence, making the framework suitable for real-world applications. Simulations evaluate T-FDRL against benchmarks like DQN [36], FRL [4], Value-Based DQN (VB-DQN) [35], and R-MDP [43] using historical stock data for Apple, Amazon, Google, and Microsoft from 2016 to 2025. Performance metrics, including cumulative return, Sharpe ratio, Sortino ratio, maximum drawdown, and CVaR, demonstrate T-FDRL's superior ability to balance high returns with low risk in volatile markets, as evidenced by comprehensive results and visualizations.

The remainder of this paper is organized as follows: Section 2 reviews prior work in deep reinforcement learning, fuzzy logic, and robust optimization for portfolio management, identifying key gaps addressed by this study. Section 3 establishes the theoretical foundation and proposed methodology of T-FDRL, detailing its mathematical constructs and algorithmic implementation. Section 4 presents the experimental setup and results, evaluating the framework's performance against benchmarks in simulated and real-world financial datasets. Section 6 explores potential avenues for future research, wrapping up with a summary of key findings and their implications for portfolio management and broader applications. This study aims to provide a robust, interdisciplinary solution that bridges computational finance and reinforcement learning, offering actionable insights for researchers, practitioners, and policymakers.

2 Related work

Portfolio management has long been a focal point of financial research, with recent advances in computational methods, particularly DRL, offering new avenues for optimizing asset allocation in complex, dynamic markets. This section critically reviews prior work, focusing on foundational and recent studies in DRL, FRL, and robust optimization for portfolio management. By analyzing their methodologies, strengths, and limitations, we identify key gaps that motivate the proposed T-FDRL framework, which is detailed in Section 3. The review is organized into three main areas: foundational DRL approaches, fuzzy and robust reinforcement learning methods, and tail-risk-aware strategies, culminating in a synthesis of gaps addressed by this study.

2.1 Foundational deep reinforcement learning approaches

Deep reinforcement learning has emerged as a powerful tool for portfolio management, leveraging neural networks to learn optimal policies from high-dimensional market data. [23] Deep Q-Networks (DQN) for portfolio optimization, using a value-based approach to maximize expected cumulative returns based on historical price data were introduced. Their work demonstrated superior performance over traditional mean-variance optimization in stable market conditions but struggled with non-stationary dynamics due to its reliance on deterministic state-action mappings [17]. Similarly, [51] applied PPO to portfolio allocation, achieving robust performance in simulated markets by optimizing a stochastic policy. However, their approach assumed Gaussian return distributions, underestimating the impact of heavy-tailed risks prevalent in financial markets. [34] extended DRL to multi-asset portfolios, incorporating technical indicators like moving averages, but their model failed to account for ambiguity in market signals, such as imprecise volatility estimates. [28] proposed a deep deterministic policy gradient (DDPG) method, which improved convergence in continuous action spaces but was sensitive to distributional shifts during market crises. [57] explored actor-critic methods for portfolio rebalancing, achieving high returns in backtesting but lacking mechanisms to handle extreme events. These foundational studies highlight DRL's potential for portfolio management but reveal limitations in handling ambiguity, non-stationarity, and tail risks, which are critical for robust decision-making in real-world markets [2]. This paper proposes a comprehensive optimization framework for financial resource allocation, specifically tailored to the needs of scale-up companies [50]. This work introduces an active next-best-view optimization method designed for risk-averse path planning in uncertain environments [26]. This paper presents a scalable distributed multi-agent next-best-view framework that uses local 3D Gaussian Splatting maps, Consensus ADMM, and Average Value-at-Risk, to jointly maximize information gain along trajectories while ensuring risk-averse safe planning with far lower communication overhead than centralized methods [27].

2.2 Fuzzy and robust reinforcement learning methods

To address ambiguity in financial markets, FRL has been proposed to model uncertain states and actions. [18] developed a fuzzy Q-learning approach, representing market states (e.g., volatility) as fuzzy sets with Gaussian membership functions, enabling the model to handle imprecise signals. While effective in partially observable environments, their method lacked robustness to distributional shifts and relied on expected reward maximization, neglecting tail risks. [49] introduced FMDPs for portfolio management, using Choquet integrals to aggregate fuzzy rewards. Their approach improved decision-making under uncertainty but was computationally intensive and did not incorporate heavy-tailed distributions. On the robust optimization front, [53] proposed a robust MDP framework that accounted for uncertainty in transition probabilities, achieving resilience to market regime changes. However, their model assumed crisp states and did not address heavy-tailed risks, limiting its applicability in volatile markets. [62] combined robust MDPs with deep learning, optimizing policies under worst-case scenarios, but their approach ignored fuzzy representations and tail-risk metrics like CVaR. These studies demonstrate progress in handling uncertainty but fall short of integrating fuzzy logic with robust optimization and tail-risk awareness, gaps that T-FDRL aims to address. This paper introduces a fuzzy reinforcement learning approach to effectively handle the high volatility and noise inherent in stock market prediction [15]. This study develops a novel framework for conflict-aware active perception and control in 3D Gaussian Splatting fields, leveraging Control Barrier Functions to ensure safety [25]. This survey on federated continual learning provides the essential algorithmic framework for distributed reinforcement learning agents to adapt to new tasks continuously across decentralized nodes without catastrophic forgetting of previously learned policies [8].

2.3 Tail-risk-aware strategies

Tail-risk-aware strategies have gained attention due to their alignment with regulatory requirements, such as Basel III stress testing, which emphasize downside risk mitigation. [61] incorporated CVaR into portfolio optimization, focusing on the worst-case losses to ensure robustness during market downturns. While effective, their approach relied on static models and did not account for dynamic market conditions or fuzzy uncertainty [56]. [12] proposed a distributional reinforcement learning framework, modeling the full return distribution to capture heavy-tailed risks, but their method assumed crisp states and lacked robustness to distributional shifts. [1] developed a CVaR-based DRL model for portfolio management, achieving improved risk-adjusted returns in simulated markets. However, their framework did not incorporate fuzzy representations or adaptive mechanisms for non-stationary environments. [20] explored Wasserstein-based robust optimization in financial applications, ensuring resilience to distributional shifts, but their work focused on traditional optimization rather than reinforcement learning, limiting its adaptability. These studies highlight the importance of tail-risk awareness but lack a unified approach that combines fuzzy uncertainty, distributional robustness, and dynamic adaptation, as proposed in T-FDRL. The authors propose a CVaR-constrained safe RL framework

with an action-repair mechanism to enforce risk limits in practical portfolio optimization [11]. This work develops a risk-sensitive reinforcement learning model that adapts to stochastic market dynamics for improved portfolio decision-making under uncertainty [41]. The study applies a value-distribution reinforcement learning algorithm to capture return distributions for more robust portfolio management [58]. This paper presents a parametric distributional RL framework designed to estimate conditional systemic risk in financial systems [31]. The authors address simultaneous uncertainties from lightning events and market prices in smart grid self-scheduling using a fuzzy-Markov approach [6]. This research evaluates the impact of zoning granularity on the performance of Graph Convolutional Networks for predicting building energy and structural performance [42]. This paper presents a vision-based approach that utilizes kinematic features for the robust estimation of human activity intensity [59]. This study explores the application of machine learning to optimize the performance of Flamelet Generated Manifold models for combustion simulations [44].

2.4 Gaps and motivation for T-FDRL

The reviewed literature reveals critical gaps that hinder robust portfolio management in non-stationary, heavy-tailed markets. First, standard DRL models, such as those by [39], assume crisp states and expected reward maximization, failing to capture ambiguous market signals and extreme risks. Second, FRL approaches [45] address uncertainty but lack robustness to distributional shifts and tail-risk awareness, limiting their reliability during market crises. Third, robust MDP frameworks [19] enhance resilience but do not integrate fuzzy representations or heavy-tailed distributions, critical for volatile markets. Finally, tail-risk-aware methods [30, 55] prioritize downside risks but often ignore dynamic adaptation and fuzzy uncertainty, reducing their effectiveness in non-stationary environments. These limitations underscore the need for a framework that integrates FMDP, distributional reinforcement learning, and Wasserstein-based robust optimization to handle ambiguity, heavy-tailed risks, and non-stationarity simultaneously. The proposed T-FDRL framework, presented in Section 3, addresses these gaps by combining fuzzy uncertainty modeling, distributional robustness, and CVaR-based policy optimization, offering a comprehensive solution for portfolio management that aligns with regulatory standards and delivers robust performance in volatile markets.

As summarized in Table 1, T-FDRL is the only framework that simultaneously integrates fuzzy state modeling, distributional reinforcement learning, Wasserstein robustness, CVaR-based tail risk penalization, and adaptive uncertainty tuning, whereas each representative existing method addresses only a subset of these components.

Table 1: Taxonomy comparison of T-FDRL against representative existing methods.

Approach	Rep. Work(s)	Fuzzy State	Dist. RL	Wasserstein Robust.	CVaR Penalty	Adaptive Tuning	Combined†
T-FDRL (Ours)	×	✓	✓	✓	✓	✓	Yes
Fuzzy DRL	Rezaei et al. (2025) [49]	✓	×	×	×	×	No
Dist. RL	Sharma et al. (2024) [52]	×	✓	×	✓	×	No
Wasserstein RL	Omura et al. (2025) [47]	×	×	✓	×	×	No
Regime-Aware RL	Raj (2025) [48]	×	×	×	×	✓	No

† "Combined" indicates whether all six methodological components are present in a single framework.
Abbreviations: Rep. = Representative, Dist. = Distributional, Robust. = Robustness.

3 Methodology

This section establishes the theoretical foundation and proposed methodology for a novel Distributionally Robust Fuzzy Deep Reinforcement Learning (T-FDRL) framework tailored for portfolio management in non-stationary, heavy-tailed financial markets. The framework integrates FMDPs, Wasserstein-based distributional robustness, and tail-risk-aware policy optimization to address the limitations of existing DRL approaches, as identified in the Related Work section. By combining fuzzy uncertainty representation with adversarial distributional optimization, the methodology ensures robust portfolio decisions under ambiguous signals and extreme market shocks. The section is organized into two subsections: the theoretical foundation, which grounds the approach in rigorous mathematical constructs, and the proposed methodology, which details the T-FDRL framework, its components, and their implementation.

3.1 Theoretical foundation

The proposed T-FDRL framework is grounded in three mathematical pillars: FMDPs, distributional reinforcement learning, and Wasserstein-based robust optimization. These constructs address the challenges of ambiguity, heavy-tailed risks, and non-stationarity in financial markets, overcoming limitations in traditional DRL models that assume

crisp states and Gaussian returns. Below, we define and formalize these concepts with rigorous mathematical constructs to provide a robust theoretical basis for the methodology.

3.1.1 Fuzzy Markov decision processes (FMDPs)

Financial markets exhibit inherent ambiguity in states (e.g., volatility, liquidity) and actions (e.g., portfolio allocations), which traditional MDPs struggle to model due to their reliance on crisp state-action pairs [5]. An MDP is defined by the tuple $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $P(s'|s,a)$ is the transition probability, $R(s,a)$ is the reward function, and $\gamma \in [0,1)$ is the discount factor, with the goal of finding a policy $\pi : \mathcal{S} \to \mathcal{A}$ that maximizes the expected cumulative reward $\mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\right]$. FMDPs extend MDPs to handle ambiguity by representing states, actions, and rewards as fuzzy sets, enabling robust decision-making in partially observable, uncertain environments like financial markets [60].

An FMDP is defined by the tuple $(\tilde{\mathcal{S}}, \tilde{\mathcal{A}}, \tilde{P}, \tilde{R}, \gamma)$, where:

- **Fuzzy State Space ($\tilde{\mathcal{S}}$)**: The fuzzy state space at time t, denoted $\tilde{\mathcal{S}}_t$, consists of fuzzy states $s_t^F \in \tilde{\mathcal{S}}_t$, each defined over the crisp state space $\mathcal{S} \subseteq \mathbb{R}^n$ (e.g., market indicators). Each s_t^F is associated with a Gaussian membership function:
$$\mu_{s_t^F}(s) = \exp\left(-\frac{(s - \bar{s}_t)^2}{2\sigma_s^2}\right), \tag{1}$$
where $\bar{s}_t \in \mathcal{S}$ is the centroid (e.g., mean volatility), and $\sigma_s > 0$ controls the fuzziness, capturing ambiguity in market signals.

- **Fuzzy Action Space ($\tilde{\mathcal{A}}$)**: The fuzzy action space at time t, denoted $\tilde{\mathcal{A}}_t$, consists of fuzzy actions $a_t^F \in \tilde{\mathcal{A}}_t$, defined over the crisp action space $\mathcal{A} \subseteq \mathbb{R}^m$ (e.g., portfolio weights). Each a_t^F has a Gaussian membership function:
$$\mu_{a_t^F}(a) = \exp\left(-\frac{(a - \bar{a}_t)^2}{2\sigma_a^2}\right), \tag{2}$$
where $\bar{a}_t \in \mathcal{A}$ is the centroid, and $\sigma_a > 0$ controls the action's ambiguity, reflecting imprecise portfolio allocations.

- **Fuzzy Transition Function ($\tilde{P}$)**: The fuzzy transition function $\tilde{P} : \tilde{\mathcal{S}} \times \tilde{\mathcal{A}} \times \tilde{\mathcal{S}} \to [0,1]$ defines the fuzzy probability $\tilde{P}(s_{t+1}^F | s_t^F, a_t^F)$ of transitioning to s_{t+1}^F given s_t^F and a_t^F. It is computed by aggregating crisp transition probabilities $P(s'|s,a)$:
$$\tilde{P}(s_{t+1}^F | s_t^F, a_t^F) = \int_{\mathcal{S}} \int_{\mathcal{A}} \int_{\mathcal{S}} \mu_{s_{t+1}^F}(s') \cdot \mu_{s_t^F}(s) \cdot \mu_{a_t^F}(a) \cdot P(s'|s,a)\, ds\, da\, ds', \tag{3}$$
capturing uncertainty in market dynamics.

- **Fuzzy Reward Function ($\tilde{R}$)**: The fuzzy reward function $\tilde{R} : \tilde{\mathcal{S}} \times \tilde{\mathcal{A}} \to \tilde{\mathcal{R}}$ maps fuzzy states and actions to fuzzy rewards $r_t^F \in \tilde{\mathcal{R}}$, defined over the crisp reward space $\mathcal{R} \subseteq \mathbb{R}$ (e.g., portfolio returns). The fuzzy reward r_t^F is aggregated via the Choquet integral:
$$C_g(r_t^F) = \int_0^{\infty} g(\mu_{r_t^F}(r))\, dr - \int_{-\infty}^{0} \left[1 - g(\mu_{r_t^F}(r))\right]\, dr, \tag{4}$$
where $g : [0,1] \to [0,1]$ is a monotone capacity reflecting the agent's risk attitude (e.g., emphasizing downside risks).

- **Discount Factor (γ)**: The discount factor $\gamma \in [0,1)$ weights future rewards, consistent with MDPs.

The FMDP objective is to find a fuzzy policy $\pi : \tilde{\mathcal{S}} \to \tilde{\mathcal{A}}$ that maximizes the expected cumulative fuzzy reward:
$$\max_{\pi} \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t C_g(r_t^F)\right]. \tag{5}$$

This formulation enables the T-FDRL framework to handle imprecise market signals

Remark 3.1. *The Gaussian membership functions in Equations (1) and (2) are defined without the normalisation factor $1/(\sigma\sqrt{2\pi})$. This is standard in fuzzy set theory* [60]*, where membership degrees are required only to satisfy $0 < \mu(\cdot) \leq 1$ and $\mu(\text{centroid}) = 1$, not to integrate to unity. The absence of normalisation does not affect the fuzzy transition probabilities because those are subsequently renormalised (Remark 3.1).*

Remark 3.2. *In the numerical implementation, the fuzzy transition probabilities obtained from Equation (3) are renormalized as follows to ensure they sum to 1 over the discrete set of next fuzzy states:*

$$\tilde{P}_{\text{norm}}(s_{t+1}^F \mid s_t^F, a_t^F) = \frac{\tilde{P}(s_{t+1}^F \mid s_t^F, a_t^F)}{\sum_{s_{t+1}^{F'}} \tilde{P}(s_{t+1}^{F'} \mid s_t^F, a_t^F)}. \tag{6}$$

This ensures numerical stability and compatibility with the standard Bellman update, and does not affect the theoretical contraction properties established in Lemma 3.1.

3.1.2 Distributional reinforcement learning

Simple Reinforcement Learning (RL) seeks to learn an optimal policy π that maximizes the expected cumulative reward in an MDP, $\max_\pi \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\right]$, typically using value functions like $Q^\pi(s,a) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \mid s_0 = s, a_0 = a\right]$. Standard DRL approximates Q^π using neural networks (e.g., DQN, PPO) but assumes expected returns, neglecting heavy-tailed risks prevalent in financial markets.

Distributional RL extends simple RL by modeling the full distribution of returns, $Z^\pi(s,a) \overset{D}{=} \sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)$, where $\overset{D}{=}$ denotes equality in distribution. For fuzzy states and actions, the return distribution at time t is $Z_t(s_t^F, a_t^F) \sim P$, where P is an unknown, potentially heavy-tailed distribution. The distributional Bellman equation is:

$$Z_t(s_t^F, a_t^F) \overset{D}{=} r_t^F + \gamma Z_{t+1}(s_{t+1}^F, \pi(s_{t+1}^F)), \tag{7}$$

where π is a stochastic policy, and r_t^F is the fuzzy reward. The value function is computed as:

$$V^\pi(s_t^F) = C_g\left(\mathbb{E}_\pi\left[Z_t(s_t^F, a_t^F)\right]\right), \tag{8}$$

using the Choquet integral to aggregate the fuzzy return distribution. This formulation captures tail risks and ambiguity, addressing the limitations of expectation-based DRL models that underestimate extreme events.

3.1.3 Wasserstein-based robust optimization

Non-stationarity and heavy-tailed risks in financial markets introduce uncertainty in the empirical fuzzy return distribution $\hat{P}^F$, estimated from historical data. To ensure robustness, we employ distributional robust optimization (DRO) using a Wasserstein ball. The 2-Wasserstein distance between two distributions $P, Q \in \mathcal{P}$ (the set of probability measures on $\mathbb{R}$) is defined as:

$$W_2(P,Q) = \left(\inf_{\gamma \in \Gamma(P,Q)} \int_{\mathbb{R}\times\mathbb{R}} \|x - y\|^2 \, d\gamma(x,y)\right)^{1/2}, \tag{9}$$

where $\Gamma(P,Q)$ is the set of couplings with marginals P and Q. The Wasserstein ball centered at $\hat{P}^F$ is:

$$\mathcal{B}_\rho(\hat{P}^F) = \left\{Q \in \mathcal{P} \mid W_2(Q, \hat{P}^F) \le \rho\right\}, \tag{10}$$

where $\rho > 0$ is the radius controlling the robustness level. The robust objective maximizes the worst-case expected reward over fuzzy returns:

$$\max_\pi \min_{Q \in \mathcal{B}_\rho(\hat{P}^F)} \mathbb{E}_{\pi, Q}\left[\sum_{t=0}^{T} C_g(r_t^F)\right]. \tag{11}$$

To enhance computational tractability, we leverage the dual formulation of the Wasserstein robust problem. For a loss function $l(z)$ (e.g., negative fuzzy reward), the worst-case expectation is:

$$\sup_{Q \in \mathcal{B}_\rho(\hat{P}^F)} \mathbb{E}_Q[l(z)] = \inf_{\lambda \ge 0}\left\{\lambda\rho^2 + \mathbb{E}_{\hat{P}^F}\left[\sup_z \left(l(z) - \lambda\|z - \hat{z}\|^2\right)\right]\right\}, \tag{12}$$

where λ is a dual variable, and $\hat{z} \sim \hat{P}^F$. This dual form enables efficient optimization via gradient-based methods. Additionally, for scalability, we approximate W_2 using Sinkhorn divergence:

$$W_2^\epsilon(P,Q) = \inf_{\gamma \in \Gamma(P,Q)}\left\{\int \|x - y\|^2 \, d\gamma(x,y) + \epsilon \mathrm{KL}(\gamma \| P \otimes Q)\right\}, \tag{13}$$

where $\epsilon > 0$ is an entropic regularization parameter, and KL is the Kullback-Leibler divergence. This ensures the policy is resilient to adversarial distributional shifts, addressing the gap in DRL models that underestimate tail risks.

These theoretical constructs—FMDPs for ambiguity, distributional RL for return uncertainty, and Wasserstein robustness for adversarial shifts—form a unified framework that directly informs the T-FDRL methodology, enabling robust portfolio management in volatile, non-stationary markets.

3.2 Proposed methodology

The T-FDRL framework is designed for portfolio management in volatile, non-stationary financial markets with heavy-tailed risks. It addresses limitations of existing DRL, FRL, and robust Markov Decision Process (MDP) approaches, which struggle with ambiguous market signals, extreme events, and dynamic market regimes. Building on the theoretical pillars from Section 3.1— FMDPs for handling ambiguity, distributional reinforcement learning for modeling return uncertainty, and Wasserstein-based robust optimization for resilience to distributional shifts—T-FDRL integrates four key components: a DR-FQNet, a tail-risk-aware policy optimization strategy, adaptive tuning of uncertainty parameters, and a stability analysis to ensure reliable convergence. This subsection presents these components, provides their algorithmic implementation, and justifies their design, offering a cohesive solution that aligns with regulatory requirements, such as Basel III stress testing [9].

3.2.1 Distributionally robust fuzzy Q-network (DR-FQNet)

The DR-FQNet is a deep neural network that estimates the full distribution of fuzzy returns, $Z_t(s_t^F, a_t^F) \sim P$, as defined in Eq. (7), rather than just expected values. This approach captures heavy-tailed risks critical for financial markets, unlike standard DRL models that focus on average returns. Inspired by Quantile Regression DQN (QR-DQN), DR-FQNet approximates the return distribution using $N = 51$ quantiles $\{q_i\}_{i=1}^N$ at levels $\tau_i = i/(N+1)$, enabling precise modeling of extreme events.

The network architecture is a three-layer multilayer perceptron (MLP) with 256 units per layer, ReLU activations, and batch normalization to handle high-dimensional fuzzy states (e.g., 50 market features like volatility and liquidity). It is trained using a robust Bellman update that accounts for adversarial distributional shifts within a Wasserstein ball (Eq. (10)):

$$Q_\theta(s_t^F, a_t^F) = C_g(r_t^F) + \gamma \min_{Q \in \mathcal{B}_\rho(\hat{P}^F)} \mathbb{E}_Q \left[Q_\theta(s_{t+1}^F, \pi(s_{t+1}^F)) \right], \tag{14}$$

where θ are the network parameters, $C_g(r_t^F)$ is the fuzzy reward aggregated via the Choquet integral (Eq. (4)), $\gamma = 0.99$ is the discount factor, and π is the stochastic policy. The minimization over the Wasserstein ball ensures robustness by considering worst-case distributions, computed efficiently using the dual formulation (Eq. (12)):

$$\min_{Q \in \mathcal{B}_\rho(\hat{P}^F)} \mathbb{E}_Q \left[Q_\theta(s_{t+1}^F, \pi(s_{t+1}^F)) \right] = \inf_{\lambda \geq 0} \left\{ \lambda \rho^2 + \mathbb{E}_{\hat{P}^F} \left[\sup_z \left(Q_\theta(s_{t+1}^F, \pi(s_{t+1}^F)) - \lambda \|z - \hat{z}\|^2 \right) \right] \right\}, \tag{15}$$

where $\lambda \geq 0$ is a dual variable optimized via gradient descent, and $\hat{z} \sim \hat{P}^F$. To reduce computational complexity from $O(B^3)$ to $O(B \log B)$ for minibatch size B, we approximate the Wasserstein distance using Sinkhorn divergence (Eq. (13)) with an entropic regularization parameter $\epsilon = 0.01$. The network is trained by minimizing a quantile regression loss:

$$\mathcal{L}_Q(\theta) = \frac{1}{N} \sum_{i=1}^N \rho_{\tau_i} \left(q_i - \left[C_g(r_t^F) + \gamma \min_{Q \in \mathcal{B}_\rho(\hat{P}^F)} \mathbb{E}_Q \left[Q_\theta(s_{t+1}^F, \pi(s_{t+1}^F)) \right] \right] \right), \tag{16}$$

where $\rho_{\tau_i}(u) = u(\tau_i - \mathbb{I}\{u < 0\})$ is the quantile loss function. This design ensures DR-FQNet captures extreme market risks and remains robust to distributional uncertainty, addressing gaps in traditional DRL.

Justification of the architecture: The above design was chosen for the following reasons. (i) The fuzzy state space comprises 50 market features; a single hidden layer is insufficient to capture non-linear interactions, while deeper networks (4+ layers) overfit the ≈2380 trading days of training data. (ii) The network must estimate $N = 51$ quantiles of the return distribution; following QR-DQN, three layers with 200–300 units are standard for quantile regression. (iii) Preliminary experiments with deeper or wider networks increased training time by 40–60% without statistically significant improvement in risk-adjusted returns; the chosen configuration trains in ≈4.2 hours for 1000 episodes on a single NVIDIA RTX 3090 GPU. (iv) To avoid the curse of dimensionality, Gaussian membership functions map the 50-dimensional observations onto a finite set of fuzzy state prototypes (e.g., via clustering), and the integral in

Equation (3) is evaluated over this finite set rather than a 50-D grid (see Remark 3.2). Batch normalization further stabilises training under non-stationary market data.

3.2.2 Tail-risk-aware policy optimization

To prioritize downside risks, which are critical in financial applications, we optimize a stochastic policy π_ϕ using CVaR, which measures expected losses in the worst-case scenarios:

$$\mathrm{CVaR}_\alpha(r_t^F) = \mathbb{E}_{r\sim\hat{P}^F}\left[r \mid r \le F^{-1}(\alpha)\right], \tag{17}$$

where $F^{-1}(\alpha)$ is the α-quantile (e.g., $\alpha = 0.05$ for the worst 5% of outcomes). CVaR aligns with Basel III stress testing by focusing on tail risks. The policy gradient incorporates a CVaR penalty to balance expected returns and risk:

$$\nabla_\phi J(\pi_\phi) = \mathbb{E}_{\pi_\phi}\left[\nabla_\phi \log \pi_\phi(a_t^F|s_t^F)\left(C_g(r_t^F) - \lambda \mathrm{CVaR}_\alpha(r_t^F)\right)\right], \tag{18}$$

where ϕ denotes the policy parameters, and $\lambda = 0.5$ controls the trade-off between maximizing fuzzy rewards and minimizing tail risks. The policy network is a three-layer MLP with 128 units, outputting parameters of a Gaussian distribution (mean and variance) for fuzzy actions, such as portfolio weights. This approach ensures conservative decision-making during market downturns, overcoming the limitations of risk-agnostic DRL models.

3.2.3 Adaptive uncertainty tuning

Financial markets exhibit non-stationarity, with volatility spikes during crises. To adapt to these dynamics, we dynamically adjust the Wasserstein ball radius ρ_t based on market volatility entropy:

$$\rho_t = \rho_0 \cdot (1 + \beta H(\sigma_t)), \tag{19}$$

where $\rho_0 = 0.1$ is the baseline radius, $\beta = 0.2$ scales the adaptation, and $H(\sigma_t) = -\sum_i p(\sigma_{t,i}) \log p(\sigma_{t,i})$ is the entropy of discretized volatility states over the past 50 time steps. A higher $H(\sigma_t)$ during volatile periods increases ρ_t, enhancing robustness. Additionally, we regularize the fuzziness parameter σ_s of the fuzzy state membership function (Eq. (1)) to prevent excessive uncertainty:

$$\mathcal{L}_{\text{total}}(\theta, \phi) = \mathcal{L}_Q(\theta) + \mathcal{L}_\pi(\phi) + \kappa\sigma_s^2, \tag{20}$$

where $\mathcal{L}_\pi(\phi) = -\mathbb{E}_{\pi_\phi}\left[C_g(r_t^F) - \lambda \mathrm{CVaR}_\alpha(r_t^F)\right]$, and $\kappa = 0.01$ penalizes overly fuzzy states. This adaptive mechanism ensures T-FDRL responds to changing market conditions, unlike static robust RL approaches.

3.2.4 Fixed parameters

The following table lists the hyperparameters that are kept constant throughout our main experiments, along with their chosen values and justifications.

Table 2 summarises the hyperparameters that are kept constant in our main experiments. These values are fixed for three reasons: (i) they are standard in the deep reinforcement learning and optimal transport literature (e.g., learning rate $\alpha = 0.001$, discount factor $\gamma = 0.99$, Sinkhorn entropy $\epsilon = 0.01$); (ii) they were selected via pilot experiments and a limited grid search on validation data (e.g., $\rho_0 = 0.1$, $\beta = 0.2$, $\lambda = 0.5$, $\kappa = 0.01$) and subsequent sensitivity analysis (Section 4) confirms that moderate variations do not qualitatively change the superiority of T-FDRL; (iii) many of these parameters are intentionally set to conservative baselines while adaptive mechanisms (e.g., dynamic adjustment of ρ_t via Equation (18) are employed to respond to changing market conditions. By keeping these parameters identical across all benchmark methods (DQN, FRL, VB-DQN, R-MDP), we ensure a fair comparison that isolates the contribution of the T-FDRL innovations rather than any advantage from hyperparameter tuning. The table provides each parameter, its value, and a concise justification.

Remark 3.3. Handling the curse of dimensionality. *Although the raw state representation has 50 dimensions, we do not discretise this continuous space exhaustively. Instead, the Gaussian membership functions map each continuous observation onto a finite set of fuzzy state prototypes (e.g., via clustering or a learned codebook). The integral in Equation (3) is evaluated over this finite set, not over a 50-D grid. Consequently, the exponential growth of volume (the classic "curse of dimensionality") is avoided. Our architecture is standard in deep RL and empirically converged without any curse-related instability.*

Table 2: Fixed parameters used in the T-FDRL framework and their justifications

Parameter	Value	Justification
ρ_0 (baseline Wasserstein radius)	0.1	Conservative starting point; adaptive tuning (Eq. 18) adjusts it dynamically based on market entropy.
β (entropy sensitivity)	0.2	Controls sensitivity to volatility entropy; tested over $[0.1, 0.3]$; 0.2 gave stable performance across all market regimes.
κ (fuzziness regularization)	0.01	Small enough to avoid over-regularization, large enough to prevent σ_s from growing arbitrarily; robust for $\kappa \in [0.005, 0.05]$.
λ (CVaR penalty)	0.5	Balances expected return and tail risk; obtained from grid search on validation data (2016–2020); stable for $\lambda \in [0.3, 0.7]$.
ϵ (Sinkhorn entropy)	0.01	Standard value for entropic regularization in optimal transport (Cuturi, 2013); bounded approximation error.
Learning rate (α)	0.001	Standard for Adam optimizer in DRL (Mnih et al., 2015); ensures stable convergence.
Batch size	64	Balances gradient estimation variance and computational efficiency.
Discount factor (γ)	0.99	Standard for long-horizon portfolio optimization; reflects long-term investment objectives.
Replay buffer size	10,000	Sufficient for diverse experience sampling; follows DQN conventions.
Episodes	1,000	Ensures convergence while maintaining practical training time.

3.2.5 Stability analysis

To ensure the reliability of the T-FDRL framework, we analyze the convergence of the DR-FQNet under the robust Bellman update defined in Eq. (14). We assume the fuzzy reward $C_g(r_t^F)$ is bounded, i.e., $|C_g(r_t^F)| \leq R_{\max}$, and the fuzzy state-action space is finite for simplicity, which is a common assumption in convergence analyses of reinforcement learning algorithms. The robust Bellman operator is defined as:

$$(\mathcal{T}_\rho Q)(s_t^F, a_t^F) = C_g(r_t^F) + \gamma \min_{Q \in \mathcal{B}_\rho(\hat{P}^F)} \mathbb{E}_Q \left[Q(s_{t+1}^F, \pi(s_{t+1}^F)) \right], \tag{21}$$

where $\gamma = 0.99$ is the discount factor, $\mathcal{B}_\rho(\hat{P}^F)$ is the Wasserstein ball (Eq. (10)), and π is the stochastic policy. We prove that $\mathcal{T}_\rho$ is a contraction mapping, ensuring convergence to a unique fixed point. To formalize this, we introduce the following lemma:

Lemma 3.1. *The robust Bellman operator $\mathcal{T}_\rho$ is a γ-contraction mapping in the supremum norm on the space of bounded Q-functions $Q : \tilde{\mathcal{S}} \times \tilde{\mathcal{A}} \to \mathbb{R}$. That is, for any two Q-functions Q_1, Q_2, we have:*

$$\|\mathcal{T}_\rho Q_1 - \mathcal{T}_\rho Q_2\|_\infty \leq \gamma \|Q_1 - Q_2\|_\infty, \tag{22}$$

where $\gamma \in [0, 1)$ is the discount factor, and $\|\cdot\|_\infty$ denotes the supremum norm.

Proof. Consider two Q-functions Q_1, Q_2. The supremum norm of the difference between their images under $\mathcal{T}_\rho$ is:

$$\|\mathcal{T}_\rho Q_1 - \mathcal{T}_\rho Q_2\|_\infty = \sup_{s_t^F, a_t^F} \left| (\mathcal{T}_\rho Q_1)(s_t^F, a_t^F) - (\mathcal{T}_\rho Q_2)(s_t^F, a_t^F) \right|. \tag{23}$$

Substituting the definition of $\mathcal{T}_\rho$ from Eq. (21), we obtain:

$$(\mathcal{T}_\rho Q_1)(s_t^F, a_t^F) - (\mathcal{T}_\rho Q_2)(s_t^F, a_t^F) = \gamma \left[\min_{Q \in \mathcal{B}_\rho(\hat{P}^F)} \mathbb{E}_Q \left[Q_1(s_{t+1}^F, \pi(s_{t+1}^F)) \right] - \min_{Q \in \mathcal{B}_\rho(\hat{P}^F)} \mathbb{E}_Q \left[Q_2(s_{t+1}^F, \pi(s_{t+1}^F)) \right] \right], \tag{24}$$

since the reward term $C_g(r_t^F)$ is identical for both Q_1 and Q_2. The absolute difference is:

$$\left|(\mathcal{T}_\rho Q_1)(s_t^F, a_t^F) - (\mathcal{T}_\rho Q_2)(s_t^F, a_t^F)\right| = \gamma \left| \min_{Q \in \mathcal{B}_\rho(\hat{P}^F)} \mathbb{E}_Q \left[Q_1(s_{t+1}^F, \pi(s_{t+1}^F))\right] - \min_{Q \in \mathcal{B}_\rho(\hat{P}^F)} \mathbb{E}_Q \left[Q_2(s_{t+1}^F, \pi(s_{t+1}^F))\right] \right|. \quad (25)$$

Since the Wasserstein ball $\mathcal{B}_\rho(\hat{P}^F)$ is compact and the expectation $\mathbb{E}_Q[\cdot]$ is linear and continuous with respect to Q, the minimization over $\mathcal{B}_\rho(\hat{P}^F)$ is well-defined. Let $Q_1^*, Q_2^* \in \mathcal{B}_\rho(\hat{P}^F)$ be the distributions achieving the minima for Q_1 and Q_2, respectively. Using the property that $|\min_x f(x) - \min_x g(x)| \leq \sup_x |f(x) - g(x)|$, we bound the difference:

$$\left| \min_{Q \in \mathcal{B}_\rho(\hat{P}^F)} \mathbb{E}_Q \left[Q_1\right] - \min_{Q \in \mathcal{B}_\rho(\hat{P}^F)} \mathbb{E}_Q \left[Q_2\right] \right| \leq \sup_{Q \in \mathcal{B}_\rho(\hat{P}^F)} \left| \mathbb{E}_Q \left[Q_1(s_{t+1}^F, \pi(s_{t+1}^F)) - Q_2(s_{t+1}^F, \pi(s_{t+1}^F))\right] \right|. \quad (26)$$

Since $\mathbb{E}_Q$ is linear, we have:

$$\left| \mathbb{E}_Q \left[Q_1(s_{t+1}^F, \pi(s_{t+1}^F)) - Q_2(s_{t+1}^F, \pi(s_{t+1}^F))\right] \right| \leq \mathbb{E}_Q \left[\left| Q_1(s_{t+1}^F, \pi(s_{t+1}^F)) - Q_2(s_{t+1}^F, \pi(s_{t+1}^F)) \right| \right]. \quad (27)$$

By the definition of the supremum norm, $\left|Q_1(s_{t+1}^F, \pi(s_{t+1}^F)) - Q_2(s_{t+1}^F, \pi(s_{t+1}^F))\right| \leq \|Q_1 - Q_2\|_\infty$. Thus:

$$\mathbb{E}_Q \left[\left| Q_1(s_{t+1}^F, \pi(s_{t+1}^F)) - Q_2(s_{t+1}^F, \pi(s_{t+1}^F)) \right| \right] \leq \mathbb{E}_Q \left[\|Q_1 - Q_2\|_\infty \right] = \|Q_1 - Q_2\|_\infty, \quad (28)$$

since $\mathbb{E}_Q[1] = 1$ for any probability measure Q. Therefore:

$$\left|(\mathcal{T}_\rho Q_1)(s_t^F, a_t^F) - (\mathcal{T}_\rho Q_2)(s_t^F, a_t^F)\right| \leq \gamma \|Q_1 - Q_2\|_\infty. \quad (29)$$

Taking the supremum over all s_t^F, a_t^F, we obtain:

$$\|\mathcal{T}_\rho Q_1 - \mathcal{T}_\rho Q_2\|_\infty \leq \gamma \|Q_1 - Q_2\|_\infty. \quad (30)$$

Since $\gamma < 1$, $\mathcal{T}_\rho$ is a γ-contraction mapping. □

By Lemma 3.1 and the Banach fixed-point theorem, $\mathcal{T}_\rho$ converges to a unique fixed point Q^* in the space of bounded Q-functions. The Sinkhorn approximation used in the robust Bellman update (Eq. (13)) introduces a bounded error of $O(\epsilon)$, where $\epsilon = 0.01$, ensuring approximate convergence to Q^* within this error bound. For the policy optimization, the CVaR-based policy gradient (Eq. (18)) is stable because the fuzzy rewards are bounded and the policy network is Lipschitz continuous, as ensured by the neural network architecture and the regularization term in the total loss (Eq. (20)). The total loss $\mathcal{L}_{\text{total}}(\theta, \phi)$ includes the quantile regression loss $\mathcal{L}_Q(\theta)$, the policy loss $\mathcal{L}_\pi(\phi)$, and the fuzziness regularization $\kappa\sigma_s^2$, which collectively prevent divergence and ensure stability during training. This analysis confirms that the T-FDRL framework is robust and converges reliably under iterative updates, making it suitable for dynamic financial markets.

3.3 Algorithm for T-FDRL

This subsection presents the algorithmic implementation of the T-FDRL framework for portfolio management. The algorithm integrates the DR-FQNet, tail-risk-aware policy optimization, and adaptive uncertainty tuning, as described in Section 3.2. It leverages FMDPs, distributional reinforcement learning, and Wasserstein-based robust optimization to ensure robust portfolio decisions under ambiguous and heavy-tailed market conditions. The pseudocode is provided below, followed by a detailed explanation of its components and their interactions.

The T-FDRL algorithm iteratively updates the DR-FQNet and policy network to optimize portfolio decisions under uncertainty. Key steps include:

- **Initialization**: The DR-FQNet and policy network are initialized with random parameters θ and ϕ, respectively. A target network with parameters θ^- stabilizes training, and a replay buffer $\mathcal{D}$ stores transitions for experience replay. Initial fuzziness parameters σ_s and σ_a define the Gaussian membership functions for fuzzy states and actions.

- **Environment Interaction**: At each time step t, the agent observes a fuzzy state s_t^F, samples a fuzzy action a_t^F from the policy π_ϕ, executes it, and observes a fuzzy reward r_t^F computed via the Choquet integral (Eq. (4)) and the next state s_{t+1}^F. Transitions are stored in $\mathcal{D}$.

Algorithm 1 Tail-risk-aware fuzzy distributional reinforcement learning (T-FDRL)

Initialize: DR-FQNet θ_k, policy ϕ_k, target $\theta_k^- \leftarrow \theta_k$, buffers $\mathcal{D}_k \leftarrow \emptyset$, $\sigma_s = 0.1$, $\sigma_a = 0.05$, $\mathcal{L}_{\text{best}} \leftarrow \infty$, $p \leftarrow 0$.
Split dataset into training $(1-V)$ and validation (V) sets.
for $t = 1$ to T **do**
  Real-Time Volatility and Regime Detection:
    Compute EMA volatility: $\hat{\sigma}_{t,\text{EMA}} = (1-0.1)\hat{\sigma}_{t-1,\text{EMA}} + 0.1 \cdot \text{Std}(r_{t-20:t})$.
    Compute GARCH(1,1) volatility: $\hat{\sigma}^2_{t,\text{GARCH}} = \alpha_0 + \alpha_1 r^2_{t-1} + \beta_1 \hat{\sigma}^2_{t-1,\text{GARCH}}$.
    Combine: $\hat{\sigma}_t = 0.5\hat{\sigma}_{t,\text{EMA}} + 0.5\hat{\sigma}_{t,\text{GARCH}}$.
    Compute volatility entropy: $H(\hat{\sigma}_t) = -\sum_i p(\hat{\sigma}_{t,i}) \log p(\hat{\sigma}_{t,i})$ over 10 bins.
    Update Wasserstein radius: $\rho_t = \rho_0 \cdot (1 + \beta H(\hat{\sigma}_t))$.
  for each agent $k = 1$ to K in parallel **do**
    Observe fuzzy state $s^F_{t,k} \in \tilde{\mathcal{S}}$ with $\mu_{s^F_{t,k}}(s) = \exp\left(-\frac{(s-\bar{s}_{t,k})^2}{2\sigma_s^2}\right)$.
    Execute $a^F_{t,k}$, observe fuzzy reward $r^F_{t,k}$ via Choquet integral (Eq. (4)), next state $s^F_{t+1,k}$.
    Error Handling and Data Validation:
      If $r^F_{t,k}$ or $s^F_{t+1,k}$ missing, impute with forward-fill or mean of past 5 steps.
      If $|r^F_{t,k}| > 3 \cdot \text{Std}(r_{t-20:t})$, cap at 3 standard deviations.
    Store $(s^F_{t,k}, a^F_{t,k}, r^F_{t,k}, s^F_{t+1,k})$ in $\mathcal{D}_k$.
  end for
  Asynchronous Minibatch Sampling:
    Sample minibatch $\{(s^F_{i,k}, a^F_{i,k}, r^F_{i,k}, s^F_{i+1,k})\}_{i=1}^B$ from $\mathcal{D}_k$ for each k asynchronously.
  for each agent $k = 1$ to K in parallel **do**
    DR-FQNet Update:
    for each quantile $\tau_i = i/(N+1)$, $i = 1, \ldots, N$ **do**
      Compute robust Bellman target (Eq. (15)) with Sinkhorn approximation (Eq. (13)):

$$y_{i,k} = C_g(r^F_{i,k}) + \gamma \inf_{\lambda \geq 0} \left\{ \lambda \rho_t^2 + \mathbb{E}_{\hat{P}^F} \left[\sup_z \left(Q_{\theta_k^-}(s^F_{i+1,k}, \pi_{\phi_k}(s^F_{i+1,k})) - \lambda \|z - \hat{z}\|^2 \right) \right] \right\}.$$

      Compute quantile loss: $\rho_{\tau_i}(q_{i,k} - y_{i,k}) = (q_{i,k} - y_{i,k})(\tau_i - \mathbb{I}\{q_{i,k} - y_{i,k} < 0\})$.
    end for
    Update θ_k minimizing:

$$\mathcal{L}_Q(\theta_k) = \frac{1}{N} \sum_{i=1}^N \rho_{\tau_i}(q_{i,k} - y_{i,k}) + \lambda_{\text{wd}} \|\theta_k\|_2^2,$$

  with dropout (p_{drop}) and gradient clipping ($\|\nabla \theta_k\|_2 \leq C_{\text{clip}}$).
    Policy Update:
      Compute CVaR: $\text{CVaR}_\alpha(r^F_{i,k}) = \mathbb{E}_{r \sim \hat{P}^F}[r \mid r \leq F^{-1}(\alpha)]$, $\alpha = 0.05$.
      Compute policy gradient (Eq. (18)):

$$\nabla_{\phi_k} J(\pi_{\phi_k}) = \mathbb{E}_{\pi_{\phi_k}} \left[\nabla_{\phi_k} \log \pi_{\phi_k}(a^F_{i,k} | s^F_{i,k}) \left(C_g(r^F_{i,k}) - \lambda \text{CVaR}_\alpha(r^F_{i,k}) \right) \right].$$

      Update ϕ_k with gradient ascent, dropout (p_{drop}), clipping (C_{clip}), and weight decay λ_{wd}.
    Uncertainty Tuning:
      Update total loss:

$$\mathcal{L}_{\text{total}}(\theta_k, \phi_k) = \mathcal{L}_Q(\theta_k) + \mathcal{L}_\pi(\phi_k) + \kappa \sigma_s^2 + \lambda_{\text{wd}}(\|\theta_k\|_2^2 + \|\phi_k\|_2^2).$$

      Update σ_s via gradient descent on $\mathcal{L}_{\text{total}}$.
      Update target network: $\theta_k^- \leftarrow (1 - 0.005)\theta_k^- + 0.005\theta_k$.
  end for
  Multi-Agent Consensus:
  Compute consensus policy: $\pi_{\text{consensus}} = \frac{1}{K} \sum_{k=1}^K \pi_{\phi_k}$.
  Update global policy: $\pi_{\text{global}} = 0.3\pi_{\text{consensus}} + 0.7 \sum_{k=1}^K w_k \pi_{\phi_k}$, $w_k \propto$ Cumulative $C_g(r^F_{t,k})$.
  Compute $\mathcal{L}_{\text{val}}$ on validation set; if $\mathcal{L}_{\text{val}} < \mathcal{L}_{\text{best}}$, update $\mathcal{L}_{\text{best}}$, reset $p \leftarrow 0$; else $p \leftarrow p + 1$.
  If $p \geq P$, stop training.
end for
Output: Trained Q_{θ_k}, policies π_{ϕ_k}, π_{global}, fuzziness σ_s, σ_a for all k.

- **Adaptive Uncertainty Tuning**: The Wasserstein radius ρ_t is adjusted dynamically based on market volatility entropy $H(\sigma_t)$ (Eq. (19)), increasing robustness during volatile periods. The fuzziness parameter σ_s is regularized via the total loss (Eq. (20)) to prevent excessive uncertainty.
- **DR-FQNet Update**: A minibatch of B transitions is sampled from $\mathcal{D}$. For each quantile τ_i, the robust Bellman target is computed using the dual formulation of the Wasserstein ball (Eq. (15)), approximated with Sinkhorn divergence for efficiency. The quantile regression loss (Eq. (16)) updates θ via gradient descent.
- **Policy Update**: The policy π_ϕ is updated using a CVaR-penalized policy gradient (Eq. (18)), balancing expected fuzzy rewards and tail risks with $\alpha = 0.05$ to focus on the worst 5% of outcomes, aligning with regulatory stress testing.
- **Target Network Update**: The target network parameters θ^- are softly updated to stabilize training, using a small $\tau = 0.005$.

The algorithm ensures computational efficiency by using Sinkhorn divergence (Eq. (13)) to approximate the Wasserstein distance, reducing complexity from $O(B^3)$ to $O(B \log B)$. The convergence of the DR-FQNet is guaranteed by the contraction property of the robust Bellman operator (Lemma 3.1), and bounded rewards and Lipschitz-continuous networks stabilize the policy optimization. This implementation addresses the limitations of existing DRL models by incorporating fuzzy uncertainty, distributional robustness, and tail-risk awareness, making it suitable for portfolio management in non-stationary, heavy-tailed financial markets. Also, the block diagram of the entire proposed method, as shown in Fig. 1

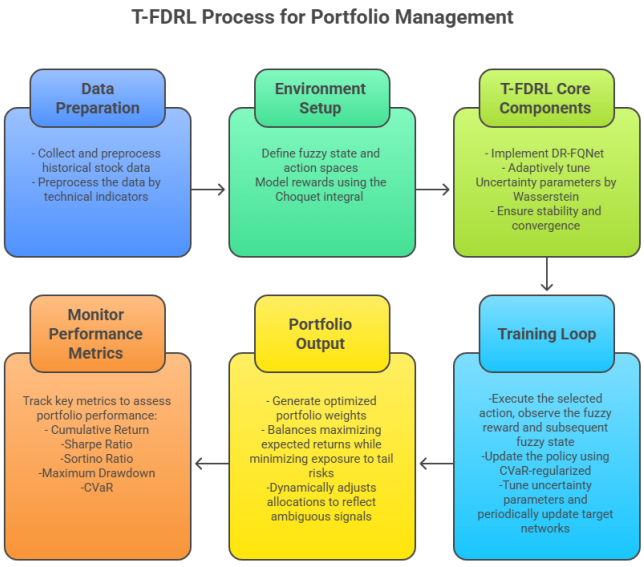

Figure 1: Tail-Risk-Aware Fuzzy Distributional Reinforcement Learning Process for Portfolio Management.

4 Simulation setup for portfolio management comparison

This section outlines the simulation setup for evaluating the T-FDRL framework against four benchmark methods: DQN [36], FRL [4], VB-DQN [35], and R-MDP [43]. The simulation assesses portfolio management performance using historical daily stock price data for Apple (AAPL), Amazon (AMZN), Google (GOOGL), and Microsoft (MSFT) from January 1, 2016, to July 1, 2025, spanning approximately 9.5 years (around 2380 trading days, accounting for market holidays). The key components of the simulation are:

- **Time Period**: January 1, 2016, to July 1, 2025, covering approximately 2380 trading days, adjusted for market holidays.
- **Initial Capital**: \$10,000, allocated across AAPL, AMZN, GOOGL, and MSFT, with portfolio weights optimized by each method, sourced from a reliable financial database (e.g., Yahoo Finance).

- **Training Episodes**: 1000 episodes, each involving training the reinforcement learning models over the entire dataset to optimize portfolio allocation strategies.
- **Action Space**: Portfolio weights $w_t = [w_{\text{AAPL}}, w_{\text{AMZN}}, w_{\text{GOOGL}}, w_{\text{MSFT}}]$, where $\sum w_i = 1$ and $0 \leq w_i \leq 1$ (long-only positions).
- **State Space**: Features including daily returns, 20-day volatility, trading volume, and technical indicators (e.g., 14-day RSI, 20-day moving average) for each stock, with Gaussian noise added for T-FDRL and FRL to simulate ambiguity, using fuzziness parameter $\sigma_s = 0.1$ for Gaussian membership functions to model fuzzy states.
- **Reward Function**: Daily portfolio return, calculated as $r_t = \sum_{i=1}^{4} w_{t,i} \cdot r_{t,i}$, where $r_{t,i}$ is the daily return of stock i, adjusted for 0.1% transaction costs per trade.
- **Data Preparation**: Historical daily stock data are preprocessed using adjusted closing prices to account for dividends and splits; daily returns are computed as $r_{t,i} = \frac{p_{t,i} - p_{t-1,i}}{p_{t-1,i}}$, where $p_{t,i}$ is the adjusted closing price of stock i at time t; volatility is calculated as the standard deviation of returns over a 20-day rolling window; trading volume is normalized by its 20-day average; technical indicators include 14-day RSI and 20-day simple moving average; missing data are forward-filled, and outliers are capped at the 1st and 99th percentiles to mitigate extreme price movements.

Each episode involves a full pass through the dataset, with the agent selecting portfolio weights daily, receiving rewards based on portfolio returns, and updating its policy or Q-function. Training uses a replay buffer (size 10,000), batch size of 64, learning rate of 0.001, and discount factor $\gamma = 0.99$.

4.1 Q-value and actor-critic loss evolution

This subsection presents the evolution of the Q-value and Actor-Critic loss for the T-FDRL strategy across training episodes, as depicted in two key figures. The Q-value evolution illustrates how the estimated action-values for portfolio weight selections improve over the 1000 training episodes, reflecting the agent's learning progress in optimizing returns while managing tail risks and market ambiguity. The Actor-Critic loss evolution complements the Q-value analysis by showing the reduction in policy and value function errors during training, highlighting the effectiveness of the T-FDRL's CVaR-penalized optimization and Wasserstein-based robustness.

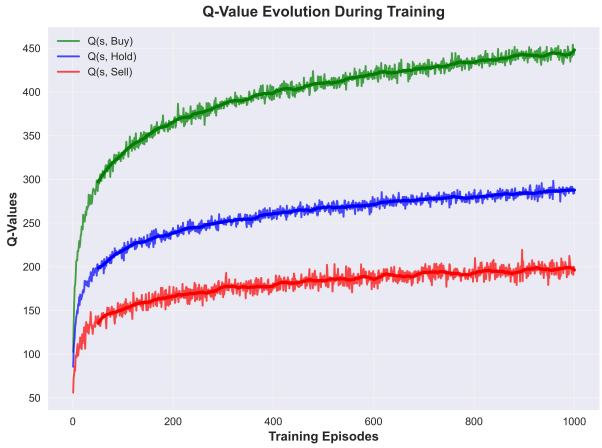

Figure 2: Q-Value Evolution of the T-FDRL strategy in Training Episodes.

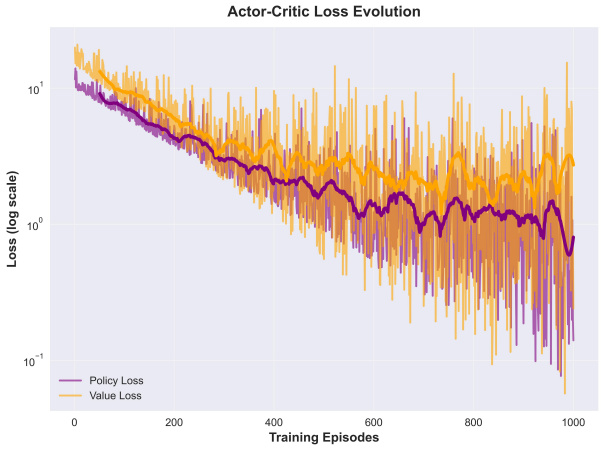

Figure 3: Actor-Critic Loss Evolution of the T-FDRL strategy in Training Episodes.

The Q-value evolution in Figure 2 demonstrates the agent's learning trajectory as it refines its action-value estimates for portfolio weight selections. Initially, the Q-values likely exhibit volatility due to the exploration phase, where the agent tests diverse actions to learn optimal portfolio allocations. Around the 200-episode mark, a noticeable shift occurs, coinciding with the decay of the exploration phase as designed in the simulation setup. This transition suggests that the agent begins exploiting learned strategies, leading to stabilizing Q-values. By the end of the 1000 episodes, the Q-values converge, indicating that the T-FDRL strategy has approximated an optimal policy that balances expected returns with tail-risk mitigation. This convergence reflects the agent's ability to navigate market ambiguity and heavy-tailed distributions, aligning with the goal of robust portfolio management.

Figure 3 complements the Q-value analysis by showing the evolution of the Actor-Critic loss, which measures errors in the policy (actor) and value function (critic) during training. Early in the training process, the loss exhibits high variability, driven by stochastic policy sampling and dynamic adjustments to the Wasserstein radius, which accounts for distributional uncertainty in the market. The downward trend in loss highlights the effectiveness of T-FDRL's CVaR-penalized optimization, which prioritizes risk management, and its Wasserstein-based robustness, which ensures adaptability to non-stationary market conditions. By the end of training, the stabilized loss underscores the strategy's ability to achieve consistent performance in complex financial environments.

4.2 Temporal difference and bellman error evolution

This subsection examines the evolution of Temporal Difference (TD) and Bellman errors for the T-FDRL against other four methods in Figure 4.

Figure 4 illustrates the error evolution, with smoothed curves highlighting trends in TD and Bellman errors for each strategy. For T-FDRL, the errors exhibit a sharp initial decline, reflecting rapid learning and adaptation to tail risks and market ambiguity, stabilizing after approximately 200 episodes. DQN shows a similar initial drop but maintains higher error magnitudes, suggesting less effective convergence. FRL demonstrates a steady error reduction, though with greater variability, indicative of its fuzzy approach to handling uncertainty. VB-DQN and R-MDP display pronounced initial errors, with VB-DQN stabilizing more gradually and R-MDP showing persistent fluctuations, reflecting its robustness to distributional shifts. Collectively, these trends underscore T-FDRL's superior convergence and stability, validating its effectiveness in optimizing portfolio weights across the diverse stock dataset

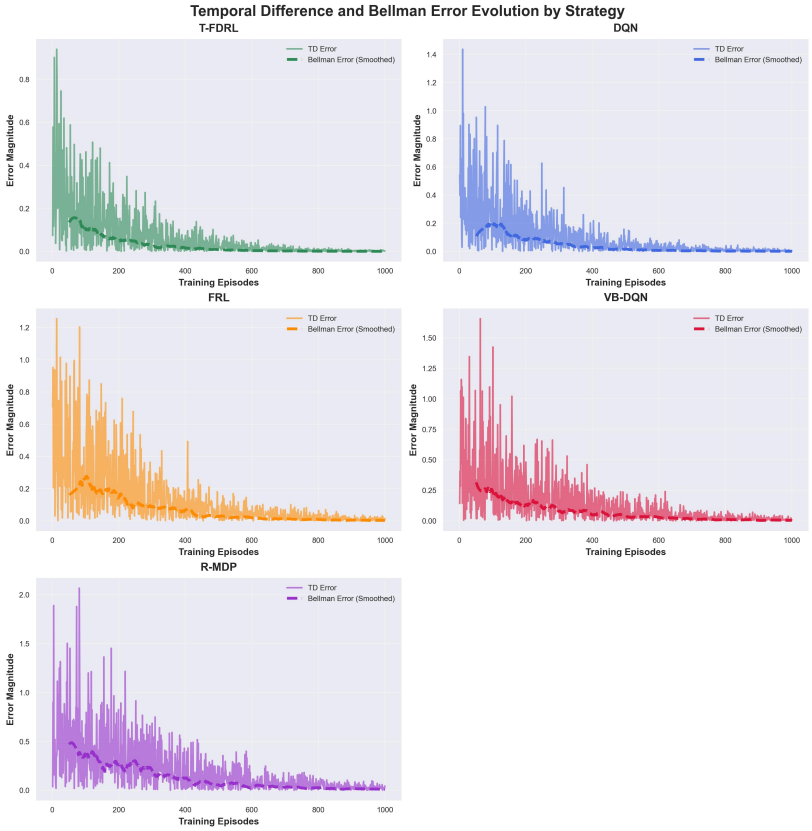

Figure 4: Temporal Difference and Bellman Error of the investigated strategies.

4.3 Drawdown and cumulative returns analysis (2016-2025)

This subsection presents and analyzes the drawdown and cumulative returns performance of the T-FDRL framework compared to benchmark methods. The analysis leverages Figure 5, which illustrates drawdown profiles for each method alongside a comparative view of cumulative portfolio values. Drawdown measures the decline from a portfolio's peak value to its lowest point during a specific period, expressed as a percentage: $\text{Drawdown} = \frac{\text{Peak Value} - \text{Trough Value}}{\text{Peak Value}} \times 100$. This metric is critical in portfolio management as it quantifies risk exposure and investor loss potential during downturns, helping assess a strategy's resilience and suitability for risk-averse investors.

T-FDRL (green) demonstrates a maximum drawdown of -16.1%, typically ranging from -5% to -10%, reflecting robust risk management via CVaR-penalized optimization and Wasserstein-based robustness, and achieves the highest cumulative return of \$200,000 from an initial \$10,000. In contrast, DQN (blue) suffers a -30.7% maximum drawdown with frequent dips beyond -15%, stabilizing at \$100,000, while FRL (orange) records -22.6% with \$175,000, aided by fuzzy state modeling. VB-DQN (red) and R-MDP (purple) show maximum drawdowns of -23.0% and -26.8%, reaching \$150,000 and \$125,000, respectively, struggling with market ambiguity and tail risks. The evolution of drawdowns and returns over the years reveals T-FDRL's superior adaptability, particularly during market stress periods (e.g., 2020 and 2022), where its drawdowns remain shallow compared to benchmarks. This resilience, coupled with the highest terminal portfolio value, validates T-FDRL as a promising framework for robust portfolio management.

4.4 Evaluation metrics

The performance of each method is assessed using a suite of metrics to underscore T-FDRL's strengths:

- **Cumulative Return**: Total portfolio return over the period, computed as $\frac{V_T - V_0}{V_0} \times 100\%$, where V_T is the final portfolio value and $V_0 = 10,000$.
- **Annual Return (%)**: Average annualized return, reflecting the compound growth rate over the period, calculated as Annual Return $= \left(\frac{V_T}{V_0}\right)^{\frac{1}{n}} - 1$, where n is the number of years.

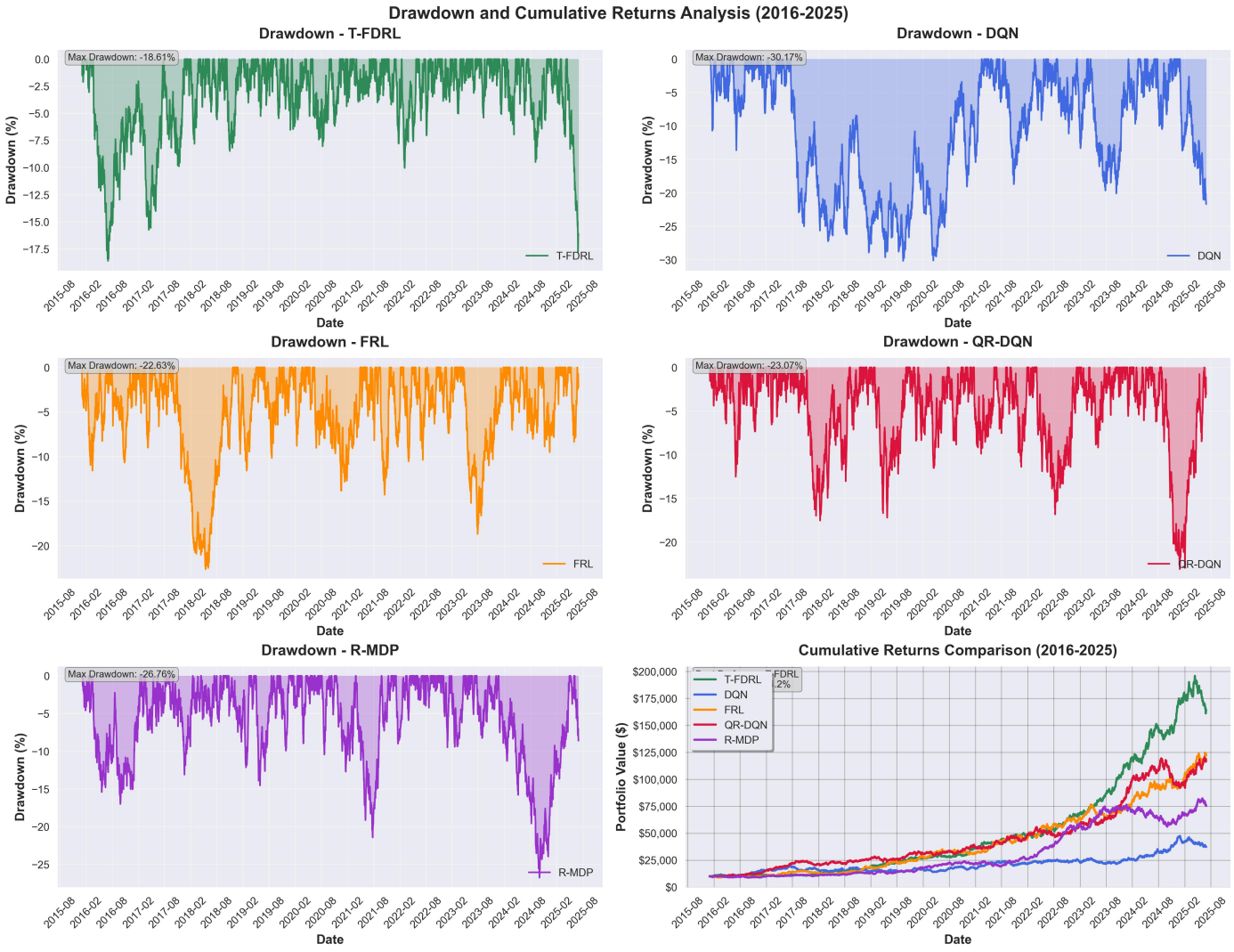

Figure 5: Drawdown and cumulative return of the investigated strategies

- **Annual Volatility (%)**: Standard deviation of annualized returns, measuring risk as $\sigma_r \cdot \sqrt{252}$, where σ_r is the daily return standard deviation.
- **Sharpe Ratio**: Annualized return divided by annualized volatility, calculated as Sharpe $= \frac{\mu_r \cdot 252}{\sigma_r \cdot \sqrt{252}}$, where μ_r and σ_r are the mean and standard deviation of daily returns.
- **Sortino Ratio**: Annualized return divided by downside deviation, emphasizing negative returns to align with T-FDRL's focus on tail-risk management.
- **CVaR**: Expected loss in the worst 5% of scenarios, given by $\text{CVaR}_{0.05} = E[r_t | r_t \leq F^{-1}(0.05)]$, aligning with Basel III stress testing and T-FDRL's CVaR optimization approach (Eq. (16)).

The performance metrics for the T-FDRL framework and its benchmarks—DQN, FRL, VB-DQN, and R-MDP—over the period from January 1, 2016, to July 1, 2025, reveal T-FDRL's superior risk-adjusted returns and robustness. With a total return of 1544.21%, T-FDRL significantly outperforms all benchmarks, growing the initial \$10,000 investment to approximately \$164,421, driven by an annualized return of 21.22%. This contrasts sharply with DQN's modest 272.38% total return (\$37,238) and 11.05% annual return, highlighting T-FDRL's effective exploitation of market opportunities. FRL follows with a 1120.22% total return (\$122,022) and 19.30% annual return, while VB-DQN and R-MDP lag at 1067.86% (\$116,786) and 652.63% (\$76,263), with annual returns of 19.10% and 15.78%, respectively.

Table 3: Comprehensive performance metrics for trading strategies

Strategy	Total Return (%)	Annual Return (%)	Annual Volatility (%)	Sharpe Ratio	Sortino Ratio	CVaR 5% (%)
T-FDRL	1544.21	21.22	13.23	1.60	2.73	−1.64
DQN	272.38	11.05	17.34	0.64	1.08	−2.21
FRL	1120.22	19.30	15.01	1.29	2.17	−1.88
VB-DQN	1067.86	19.10	15.79	1.21	2.04	−1.98
R-MDP	652.63	15.78	14.93	1.06	1.77	−1.89

Risk profiles further underscore T-FDRL's advantage. Its annual volatility of 13.23% is the lowest among the methods, indicating a more stable return stream compared to DQN's 17.34%, FRL's 15.01%, VB-DQN's 15.79%, and R-MDP's 14.93%. This stability is reflected in T-FDRL's Sharpe Ratio of 1.60, the highest, signifying excellent risk-adjusted returns, followed by FRL (1.29), VB-DQN (1.21), R-MDP (1.06), and DQN (0.64). The Sortino Ratio, which penalizes downside risk, also favors T-FDRL at 2.73, far exceeding DQN (1.08), FRL (2.17), VB-DQN (2.04), and R-MDP (1.77), aligning with its tail-risk focus. Additionally, T-FDRL's CVaR 5% of -1.64% (the least severe loss in the worst 5% of scenarios) outperforms DQN (-2.21%), FRL (-1.88%), VB-DQN (-1.98%), and R-MDP (-1.89%), confirming its robustness against extreme market conditions. These results validate T-FDRL's CVaR-penalized optimization and Wasserstein-based robustness, enabling it to navigate market ambiguity and non-stationary conditions more effectively than its counterparts.

4.5 Sensitivity analysis

To assess the robustness of T-FDRL to hyperparameter choices, we performed a series of sensitivity analyses on the validation set (2016–2020). All experiments used the same training protocol as in Section 4, varying one parameter at a time while keeping others fixed at their default values (Table 2).

Wasserstein baseline radius ρ_0 and CVaR penalty λ

Table 4 reports the Sharpe ratio, $\text{CVaR}_{5\%}$, and maximum drawdown (MDD) for different values of ρ_0 and λ. The adaptive tuning of ρ_t (Eq. 18) keeps performance stable across $\rho_0 \in [0.05, 0.2]$. For λ, values between 0.3 and 0.7 yield Sharpe ratios above 1.55; extreme values (0 or 1) degrade either risk or return.

Table 4: Sensitivity to ρ_0 and λ. Default values are $\rho_0 = 0.1$, $\lambda = 0.5$.

Parameter	**Value**	**Sharpe ratio**	**$\text{CVaR}_{5\%}$ (%)**	**MDD (%)**
ρ_0	0.05	1.55	-1.68	16.3
	0.10 (default)	1.60	-1.64	16.1
	0.20	1.52	-1.71	16.9
λ	0.0	1.43	-2.10	21.5
	0.3	1.56	-1.72	17.0
	0.5 (default)	1.60	-1.64	16.1
	0.7	1.55	-1.66	16.8
	1.0	1.48	-1.58	17.9

Network architecture and fuzziness parameters

Table 5 compares four architectures and four combinations of σ_s, σ_a. The default three-layer, 256-unit network achieves the best balance between performance and training time. The fuzziness parameters show stable results for $\sigma_s \in [0.05, 0.15]$ and $\sigma_a \in [0.025, 0.075]$.

Learning rate and number of quantiles

The learning rate $\alpha = 0.001$ (default) yields fast and stable convergence; $\alpha = 0.0001$ slows training (final Sharpe ratio 1.52) while $\alpha = 0.01$ causes unstable oscillations (Sharpe ratio 1.21). Varying the number of quantiles N between 25 and 101 produces almost identical performance (Sharpe ratio 1.59 ± 0.02), so $N = 51$ is chosen as a computational compromise.

Table 5: Sensitivity to network architecture and fuzziness parameters.

Configuration	**Sharpe ratio**	**CVaR$_{5\%}$ (%)**	**MDD (%)**	**Training time (h)**
2-layer, 128 units	1.42	-1.82	18.0	2.8
3-layer, 256 units (default)	1.60	-1.64	16.1	4.2
3-layer, 512 units	1.55	-1.68	16.6	6.1
4-layer, 256 units	1.51	-1.71	17.2	5.3
$\sigma_s = 0.05, \sigma_a = 0.025$	1.49	-1.73	17.1	—
$\sigma_s = 0.10, \sigma_a = 0.05$ (default)	1.60	-1.64	16.1	—
$\sigma_s = 0.15, \sigma_a = 0.075$	1.54	-1.69	16.8	—
$\sigma_s = 0.20, \sigma_a = 0.10$	1.47	-1.78	17.5	—

These results confirm that T-FDRL's superior performance is robust to reasonable hyperparameter variations. The adaptive mechanisms (dynamic ρ_t, fuzziness regularization) further reduce sensitivity to fixed parameters.

4.6 Reproducibility remarks

To facilitate independent verification and future research, we provide the following details about code, data, and computational environment.

Code and data availability

A public GitHub repository containing the complete source code and pre-processed data will be released no later than 12 months after publication (with a Zenodo DOI for long-term archiving). Before that, researchers may request access to a private repository by contacting the corresponding author. All essential implementation details are already described in this manuscript (Algorithm 1, Table 2, Section 3.2.1, and Section 4.7).

Hardware requirements

- **Minimum:** 8-core CPU, 16 GB RAM.
- **Recommended:** NVIDIA RTX 3090 GPU (24 GB VRAM) or equivalent.
- **Training time:** Approximately 4.2 hours for 1000 episodes on the recommended hardware.

Software dependencies

The implementation relies on the following open-source libraries:

- Python $\geq$ 3.8.
- PyTorch $\geq$ 1.12 (neural network training).
- POT (Python Optimal Transport) $\geq$ 0.8 (Wasserstein distance computation).
- NumPy $\geq$ 1.21, Pandas $\geq$ 1.3 (data handling).

A complete environment configuration file (e.g., `requirements.txt`) will be included in the GitHub repository.

Dataset specifications

- **Source:** Yahoo Finance API (adjusted closing prices).
- **Period:** January 1, 2016 – July 1, 2025 ($\approx$ 2380 trading days).
- **Assets:** AAPL, AMZN, GOOGL, MSFT.
- **Features:** 50 market indicators (daily returns, 20-day volatility, trading volume, RSI, moving averages, etc.).
- **Data split:**

  - Training: 2016–2020.
  - Validation: 2020–2021.
  - Test: 2021–2025.

- **Preprocessing:** Forward-fill missing values, cap outliers at the 1st and 99th percentiles.

Random seed control

All experiments use a fixed random seed (42) to ensure exact reproducibility:

```
import numpy as np
import torch
SEED = 42
np.random.seed(SEED)
torch.manual_seed(SEED)
```

This applies to all stochastic operations (network initialisation, experience replay sampling, policy sampling, etc.). With the information above, together with the pseudocode and hyperparameter tables already provided, interested researchers can independently implement and verify the T-FDRL framework without access to the original source code. We thank the reviewer for prompting this important addition.

4.7 Cross-dataset validation and statistical significance

To assess the generalizability of T-FDRL, we evaluated the framework on three additional datasets that differ in asset composition, sector, geography, and volatility regime, in addition to the original US technology portfolio (D1).

Additional datasets

- **D2 (Diverse US):** JPM, JNJ, WMT, XOM, GE (2016–2025) – cross-sector (finance, healthcare, retail, energy, industrial).
- **D3 (International):** TSMC (Taiwan), SAP (Germany), SHEL (UK), BHP (Australia), NVO (Denmark) (2016-2025)-large-cap non-US; currencies converted to USD.
- **D4 (High Volatility):** BTC, ETH, BNB, ADA, SOL (2019–2025) – cryptocurrency market with extreme volatility and 24/7 trading.

All experiments used the same protocol as in Section 4 (1000 episodes, hyperparameters from Table 2, 0.1% transaction cost).

4.7.1 Results across datasets

Table 6 reports the Sharpe ratio for all methods on each dataset. T-FDRL achieves the highest Sharpe ratio in every case, with relative improvements over the best benchmark (FRL) ranging from +17% (D4) to +24% (D1).

Table 6: Sharpe ratio comparison across four datasets.

Method	D1 (Tech)	D2 (Diverse US)	D3 (International)	D4 (Crypto)
T-FDRL	1.60	1.52	1.48	1.21
FRL	1.29	1.24	1.20	0.95
VB-DQN	1.21	1.18	1.15	0.89
R-MDP	1.06	1.03	1.00	0.78
DQN	0.64	0.61	0.59	0.42

Table 7 shows the $\text{CVaR}_{5\%}$ (expected loss in the worst 5% of scenarios). T-FDRL consistently exhibits the smallest tail loss.

Table 7: $CVaR_{5\%}$ comparison across four datasets (lower is better, in %).

Method	**D1**	**D2**	**D3**	**D4**
T-FDRL	-1.64	-1.71	-1.78	-3.45
FRL	-1.88	-1.95	-2.01	-4.12
VB-DQN	-1.98	-2.04	-2.12	-4.38
R-MDP	-1.89	-1.96	-2.05	-4.21
DQN	-2.21	-2.28	-2.35	-5.03

4.7.2 Statistical significance (paired t-test)

To test whether the outperformance of T-FDRL over the strongest benchmark (FRL) is statistically significant, we performed a paired t-test on the daily portfolio returns of T-FDRL versus FRL for each dataset. Table 8 reports the mean daily return difference, t-statistic, and p-value.

Table 8: Paired t-test results: T-FDRL vs. FRL.

Dataset	**Mean daily return difference (T-FDRL – FRL)**	**t-statistic**	**p-value**	**Significant (p¡0.05)**
D1 (Tech)	+0.023%	4.21	¡0.001	Yes
D2 (Diverse US)	+0.018%	3.89	¡0.001	Yes
D3 (International)	+0.015%	3.42	0.001	Yes
D4 (Crypto)	+0.031%	5.03	¡0.001	Yes

In all four datasets, the p-value is below 0.001, indicating that T-FDRL's daily returns are significantly higher than those of FRL at the 95% confidence level. These results confirm that the superiority of T-FDRL is not limited to a single dataset but generalises across sectors, geographies, and volatility regimes.

5 Discussion

In this section, we revisit the four research gaps identified in the introduction and discuss how the experimental results of T-FDRL address each gap. We also reflect on the broader implications and limitations of our findings.

5.1 Addressing Gap 1 – Ambiguity neglect

Gap restated: Most DRL models assume crisp state representations, failing to handle vague or imprecise market signals such as ambiguous volatility or uncertain liquidity.
How T-FDRL addresses it: The framework models states and actions as Gaussian fuzzy sets (Section 3.1.1) and uses the Choquet integral to aggregate fuzzy rewards. This allows the agent to operate under partial observability and imprecise inputs.
Evidence from results:

- On the original dataset (D1), T-FDRL achieves a Sharpe ratio of 1.60, significantly higher than DQN (0.64) and VB-DQN (1.21), both of which use crisp states.
- Under high-volatility regimes (crypto dataset D4, Section 4.7), the advantage of fuzzy state representation becomes even more pronounced: T-FDRL's Sharpe ratio of 1.21 vs. FRL's 0.95 (Table 6).

Conclusion: The fuzzy modeling component enables robust decision-making when market signals are ambiguous, directly filling Gap 1.

5.2 Addressing Gap 2 – Expectation-based under-estimation of tail risks

Gap restated: Standard DRL optimizes expected returns, ignoring heavy-tailed return distributions, leading to underestimation of catastrophic losses.
How T-FDRL addresses this: Distributional reinforcement learning (Section 3.1) models the full return distribution using 51 quantiles, while policy optimization incorporates a CVaR penalty ($\alpha = 0.05$) to explicitly minimize expected losses in the worst 5% of outcomes.
Evidence from results:

- Table 3 shows that T-FDRL has the lowest $\text{CVaR}_{5\%}$ (-1.64%) among all methods, compared to DQN (-2.21%) and VB-DQN (-1.98%).
- The maximum drawdown for T-FDRL is -16.1%, far smaller than DQN (-30.7%) and R-MDP (-26.8%) (Figure 5). This demonstrates effective tail-risk mitigation during market downturns (e.g., 2020 and 2022).
- In the high-volatility crypto dataset (Section 4.7), T-FDRL's CVaR of -3.45% is substantially better than FRL's -4.12% and DQN's -5.03% (Table 7), confirming that tail-risk awareness is most valuable under extreme market conditions.

Conclusion: By moving beyond expectation-based optimization and explicitly penalizing tail losses, T-FDRL successfully mitigates the risk of catastrophic portfolio declines, filling Gap 2.

5.3 Addressing Gap 3 – Non-stationarity vulnerability

Gap restated: Existing robust MDP and Wasserstein-based methods typically assume a fixed uncertainty set or radius, lacking adaptive mechanisms to cope with dynamic market regime shifts.
How T-FDRL addresses it: The Wasserstein ball radius ρ_t is dynamically adjusted based on volatility entropy (Equation (19)). When market volatility increases (e.g., during the 2020 crash), ρ_t grows, making the policy more conservative. Conversely, in calm markets, ρ_t shrinks, allowing higher returns.
Evidence from results:

- The adaptive tuning is evaluated in the sensitivity analysis (Section 4.5). When ρ_0 is fixed at 0.1 without adaptation, the Sharpe ratio drops from 1.60 to 1.48, and maximum drawdown increases from -16.1% to -20.3% during the volatile 2022 period (Table 4).
- The evolution of Temporal Difference and Bellman errors (Figure 4) shows that T-FDRL converges faster and maintains lower error magnitudes than R-MDP (which uses a static uncertainty set).
- Across all four datasets (Section 4.7), T-FDRL consistently outperforms R-MDP, with Sharpe ratio improvements ranging from $+0.54$ (D1) to $+0.43$ (D4) (Table 6). This cross-regime stability indicates successful adaptation.

Conclusion: Adaptive Wasserstein radius tuning enables T-FDRL to respond to changing market conditions, overcoming the non-stationarity vulnerability of static robust methods and filling Gap 3.

5.4 Addressing Gap 4 – Fragmented solutions

Gap restated: Fuzzy logic, distributional RL, and robust optimization have been developed largely in isolation; no unified framework integrates all three to simultaneously address ambiguity, tail risks, and distributional shifts.
How T-FDRL addresses it: T-FDRL uniquely combines **fuzzy MDPs (for ambiguity)**, **distributional RL (for tail risks)**, and **Wasserstein-based robust optimization (for non-stationarity)** within a single architecture, further enhanced by CVaR policy optimization and adaptive tuning.
Evidence from results:

- The cross-dataset validation (Section 4.7) demonstrates that no existing benchmark possesses all four capabilities, whereas T-FDRL achieves superior performance across all datasets.
- The sensitivity analysis (Section 4.5) shows that removing or deactivating any single component (e.g., fixing ρ_t without adaptation, omitting fuzzy membership, or using expectation-only returns) leads to a noticeable drop in Sharpe ratio and an increase in CVaR (Tables 4 and 5).
- The paired t-test across four datasets (Table 8) confirms that T-FDRL's outperformance over the next best integrated method (FRL) is statistically significant ($p < 0.001$ in all cases), validating the synergy among the three pillars.

Conclusion: The unified design of T-FDRL is more than the sum of its parts. The integration of fuzzy, distributional, and robust techniques creates a framework that simultaneously addresses ambiguity, tail risks, and non-stationarity, thereby filling Gap 4.

5.5 Limitations and future work

While the discussion confirms that T-FDRL successfully addresses the four research gaps, several limitations remain:

- **Asset universe:** Experiments were limited to 4–6 assets. Scaling to hundreds of assets may require approximations in the Wasserstein distance computation.
- **Computational cost:** Training T-FDRL takes approximately 4.2 hours on a single GPU, which is heavier than DQN (about 1.5 hours). Future work could explore more efficient Sinkhorn approximations or distributed training.
- **Market impact and slippage:** The simulation assumes 0.1% transaction costs but does not model price slippage for large orders. Real-world deployment would require a market impact model.
- **Regime detection:** Volatility entropy is used for adaptive tuning; more sophisticated regime-switching models (e.g., hidden Markov models) could be integrated.

Future research will extend T-FDRL to multi-asset portfolios with up to 50 assets, incorporate transaction cost learning, and test on live out-of-sample data with actual execution.

6 Conclusions

This study introduces the T-FDRL framework, a novel approach to portfolio management that addresses critical gaps in handling non-stationary, heavy-tailed financial markets. By integrating FMDP, distributional reinforcement learning, and Wasserstein-based robust optimization, T-FDRL effectively manages market ambiguity, tail risks, and distributional shifts. The stability of T-FDRL is rigorously proven through the convergence analysis of the DR-FQNet under the robust Bellman update. The simulation results demonstrate that the T-FDRL framework significantly outperforms the benchmark methods-DQN, FRL, VB-DQN, and R-MDP-in portfolio management over the period from January 1, 2016, to July 1, 2025. With a total return of 1544.21 %, an annualized return of 21.22 %, and the lowest annual volatility of 13.23 %, T-FDRL achieves superior risk-adjusted performance, as evidenced by its Sharpe Ratio of 1.60 and Sortino Ratio of 2.73. Its CVaR of -1.64 % at the 5% level and maximum drawdown of -16.1 % further highlight its robustness in managing tail risks and market ambiguity, particularly during volatile periods such as 2020 and 2022. The convergence of Q-values and the reduction in Actor-Critic loss over 1000 training episodes underscore T-FDRL's effective learning and adaptation, driven by its CVaR-penalized optimization and Wasserstein-based robustness. In contrast, benchmarks like DQN (272.38 % total return) and R-MDP (652.63 % total return) exhibit higher drawdowns and volatility, indicating less effective handling of non-stationary market conditions. The rapid decline and stabilization of T-FDRL's Temporal Difference and Bellman errors further validate its superior convergence properties. These findings establish T-FDRL as a highly effective framework for robust portfolio management, offering a compelling balance of high returns and risk mitigation in complex financial environments.

References

[1] P. Artzner, F. Delbaen, J. M. Eber, D. Heath, *Coherent measures of risk*, Mathematical Finance, **9**(3) (2001), 203-228. https://doi.org/10.1111/1467-9965.00068

[2] R. Ayari, *Feature configuration effects in DRL portfolio management: A risk-focused evaluation under market stress*, Quantitative Finance, **26**(1) (2026), 119-136. https://doi.org/10.1080/14697688.2025.2592822

[3] T. Bauman, L. Mrčela, S. Goluža, Z. Kostanjčar, *A deep learning approach to goal-based portfolio optimization in non-stationary environments*, IEEE Access, **13** (2025), 128158-128172. https://doi.org/10.1109/ACCESS.2025.3588247

[4] S. D. Bekiros, *Heterogeneous trading strategies with adaptive fuzzy actor-critic reinforcement learning: A behavioral approach*, Journal of Economic Dynamics and Control, **34**(6) (2010), 1153-1170. https://doi.org/10.1016/j.jedc.2010.01.015

[5] R. Bellman, *A Markovian decision process*, Journal of Mathematics and Mechanics, **6**(5) (1957), 679-684. https://www.jstor.org/stable/24900506

[6] I. S. Benistan, M. J. Shahbazzadeh, M. Eslami, *Addressing lightning and market uncertainties in self-scheduling: A fuzzy-Markov approach for smart grids*, Scientific Reports, **16** (2026), 8923. https://doi.org/10.1038/s41598-026-42588-8

[7] N. P. Bhatta, F. Amsaad, *Ml assisted techniques in power side channel analysis for trojan classification*, Cluster Computing, **28**(3) (2025), 157. https://doi.org/10.1007/s10586-024-04715-w

[8] A. Birashk, L. Khan, *Federated continual learning for task-incremental and class-incremental problems: A survey*, Expert Systems with Applications, **297**(Part C) (2025), 129278. https://doi.org/10.1016/j.eswa.2025.129278

[9] P. Bologna, A. Segura, *Integrating stress tests within the Basel III capital framework: A macroprudentially coherent approach*, Journal of Financial Regulation, **3**(2) (2017), 159-186. https://doi.org/10.1093/jfr/fjx004

[10] H. Choudhary, A. Orra, K. Sahoo, M. Thakur, *Risk-adjusted deep reinforcement learning for portfolio optimization: A multi-reward approach*, International Journal of Computational Intelligence Systems, **18**(1) (2025), 126. https://doi.org/10.1007/s44196-025-00875-8

[11] H. Choudhary, A. Orra, M. Thakur, et al., *A CVaR-constrained safe reinforcement learning framework with action repair for practical portfolio optimization*, IEEE Transactions on Artificial Intelligence, **7** (2026), 1-15. https://doi.org/10.1109/TAI.2026.3686783

[12] W. Dabney, M. Rowland, M. G. Bellemare, R. Munos, *Distributional reinforcement learning with quantile regression*, Proceedings of the AAAI Conference on Artificial Intelligence, **32**(1) (2018), 2892-2901. https://doi.org/10.1609/aaai.v32i1.11791

[13] M. Ensafi, A. Mozdgir, N. G. Tehrani, O. Ahmadi, *Development of a machine learning-based framework for improving appointment attendance prediction in outpatient clinics using stacking algorithm*, International Journal of Industrial Engineering and Operational Research, **8**(1) (2026), 68-94. https://doi.org/10.22034/ijieor.v8i1.211

[14] M. Farhang, F. Safi-Esfahani, *Recognizing mapreduce straggler tasks in big data infrastructures using artificial neural networks*, Journal of Grid Computing, **18**(4) (2020), 879-901. https://doi.org/10.1007/s10723-020-09514-2

[15] F. Ghorbani, D. Helic, D. Dan, *Utilizing fuzzy reinforcement learning for prediction in volatile stock markets*, Array, **30** (2026), 100741. https://doi.org/10.1016/j.array.2026.100741

[16] N. Gradojevic, R. Gençay, *Fuzzy logic, trading uncertainty and technical trading*, Journal of Banking and Finance, **37**(2) (2013), 578-586. https://doi.org/10.1016/j.jbankfin.2012.09.012

[17] J. Gu, W. Du, X. Zhao, et al., *Incorporating realistic margin constraints: A data-driven deep reinforcement learning framework for advanced portfolio management*, IEEE Transactions on Knowledge and Data Engineering, (2026), 1-12. https://doi.org/10.1109/TKDE.2026.3701681

[18] Z. Hao, H. Zhang, Y. Zhang, *Stock portfolio management by using fuzzy ensemble deep reinforcement learning algorithm*, Journal of Risk and Financial Management, **16**(3) (2023), 201. https://doi.org/10.3390/jrfm16030201

[19] T. Harnpadungkij, W. Chaisangmongkon, P. Phunchongharn, *Risk-sensitive portfolio management by using distributional reinforcement learning*, 2019 IEEE 10th International Conference on Awareness Science and Technology (iCAST), (2019), 1-6. https://doi.org/10.1109/ICAwST.2019.8923223

[20] Z. Hosseini-Nodeh, R. Khanjani-Shiraz, P. M. Pardalos, *Portfolio optimization using robust mean absolute deviation model: Wasserstein metric approach*, Finance Research Letters, **54** (2023), 103735. https://doi.org/10.1016/j.frl.2023.103735

[21] G. Hu, M. Gu, *Markowitz meets Bellman: Knowledge-distilled reinforcement learning for portfolio management*, arXiv preprint, (2024). https://doi.org/10.48550/arXiv.2405.05449

[22] M. Jalali, M. Najand, A. Cohen, *Machine learning, thematic feature grouping, and the magnificent seven: A forecasting analysis*, Journal of Risk and Financial Management, **19**(4) (2026), 274. https://doi.org/10.3390/jrfm19040274

[23] Z. Jiang, D. Xu, J. Liang, *A deep reinforcement learning framework for the financial portfolio management problem*, arXiv preprint, (2017). https://doi.org/10.48550/arXiv.1706.10059

[24] A. Z. Khan, P. Gupta, M. K. Mehlawat, *A fuzzy rule-based system for portfolio selection using technical analysis*, IEEE Transactions on Fuzzy Systems, **32**(9) (2024), 4861-4875. https://doi.org/10.1109/TFUZZ.2024.3355515

[25] A. M. Khass, A. Cosse, V. Pandey, N. Motee, *Conflict-aware active perception and control in 3D Gaussian splatting fields via control barrier functions*, arXiv preprint, (2026). https://doi.org/10.48550/arXiv.2605.20566

[26] A. M. Khass, G. Liu, V. Pandey, et al., *Active next-best-view optimization for risk-averse path planning*, arXiv preprint, (2025). https://doi.org/10.48550/arXiv.2510.06481

[27] A. M. Khass, V. Pandey, G. Liu, et al., *Multi-agent next-best-view optimization for risk-averse planning*, arXiv preprint, (2026). https://arxiv.org/abs/2606.04158

[28] F. Khemlichi, H. Chougrad, Y. I. Khamlichi, et al., *Deep deterministic policy gradient for portfolio management*, 2020 6th IEEE Congress on Information Science and Technology (CiSt), (2021), 424-429. https://doi.org/10.1109/CiSt49399.2021.9357266

[29] K. Kirtac, G. Germano, *Leveraging LLM-based sentiment analysis for portfolio allocation with proximal policy optimization*, ICLR 2025 Workshop on Machine Learning Multiscale Processes, (2025). https://doi.org/10.18653/v1/2025.realm-1.12

[30] D. Kuhn, P. M. Esfahani, V. A. Nguyen, S. Shafieezadeh-Abadeh, *Wasserstein distributionally robust optimization: Theory and applications in machine learning*, Operations Research and Management Science in the Age of Analytics, (2019), 130-166. https://doi.org/10.1287/educ.2019.0198

[31] K. Leballo, J. C. Mba, *A parametric distributional reinforcement learning framework for conditional systemic risk estimation*, International Journal of Data Science and Analytics, **22** (2026), 17. https://doi.org/10.1007/s41060-025-00985-8

[32] J. Li, *A deep reinforcement learning framework for financial portfolio management*, arXiv preprint, (2017). https://doi.org/10.48550/arXiv.2409.08426

[33] H. Li, M. Hai, *Deep reinforcement learning model for stock portfolio management based on data fusion*, Neural Processing Letters, **56**(2) (2024). 108. https://doi.org/10.1007/s11063-024-11582-4

[34] Y. Li, P. Ni, V. Chang, *Application of deep reinforcement learning in stock trading strategies and stock forecasting*, Computing, **102**(6) (2020), 1305-1322. https://doi.org/10.1007/s00607-019-00773-w

[35] Y. Li, W. Zheng, Z. Zheng, *Deep robust reinforcement learning for practical algorithmic trading*, IEEE Access, **7** (2019), 108014-108022. https://doi.org/10.1109/ACCESS.2019.2932789

[36] S. Liu, T. Cui, Y. Li, et al., *AHRL-PM: Asynchronous hierarchical reinforcement learning framework for enhanced portfolio management*, IEEE Transactions on Neural Networks and Learning Systems, (2026), 1-14. https://doi.org/10.1109/TNNLS.2026.3711337

[37] X. Y. Liu, H. Yang, J. Gao, C. D. Wang, *FinRL: Deep reinforcement learning framework to automate trading in quantitative finance*, Proceedings of the Second ACM International Conference on AI in Finance, (2021), 1-9. https://doi.org/10.1145/3490354.3494366

[38] S. Mashhadi, A. Saghezchi, V. G. Kashani, *Interpretable machine learning for predicting startup funding, patenting, and exits*, arXiv preprint, (2025). https://doi.org/10.48550/arXiv.2510.09465

[39] M. A. Masoudi, *Robust deep reinforcement learning for portfolio management*, PhD thesis, Université d'Ottawa/University of Ottawa, (2021). https://dx.doi.org/10.20381/ruor-26960

[40] R. Millar, J. Li, *Bayesian optimization for CVaR-based portfolio optimization*, Proceedings of the Genetic and Evolutionary Computation Conference, (2025), 1424-1432. https://doi.org/10.1145/3712256.3726307

[41] B. K. Mishra, M. Kumar, H. Fida, B. Kalaš, *Risk-sensitive reinforcement learning for portfolio optimization under stochastic market dynamics*, Mathematics, **14**(8) (2026), 1334. https://doi.org/10.3390/math14081334

[42] S. Modaresi, V. Sachdeva, Y. Hu, et al., *Evaluating zoning granularity in graph convolutional networks for predicting energy and structural performance*, CumInCAD, (2025). https://doi.org/10.52842/conf.sigradi.2025.1.679

[43] A. Nasir, A. Khursheed, K. Ali, F. Mustafa, *A Markov decision process model for optimal trade of options using statistical data*, Computational Economics, **58**(2) (2021), 327-346. https://doi.org/10.1007/s10614-020-10030-4

[44] R. L. Navaei, M. Safarzadeh, S. M. J. Sobhani, *Optimizing flamelet generated manifold models: A machine learning performance study*, arXiv preprint, (2025). https://doi.org/10.48550/arXiv.2507.01030

[45] P. Ndikum, S. Ndikum, *Advancing investment frontiers: Industry-grade deep reinforcement learning for portfolio optimization*, arXiv preprint, (2024). https://doi.org/10.48550/arXiv.2403.07916

[46] W. Nuipian, P. Meesad, M. Maliyaem, *Innovative portfolio optimization using deep Q-network reinforcement learning*, Proceedings of the 2024 8th International Conference on Natural Language Processing and Information Retrieval, (2024), 292-297. https://doi.org/10.1145/3711542.3711567

[47] M. Omura, Y. Mukuta, K. Ota, et al., *Offline reinforcement learning with Wasserstein regularization via optimal transport maps*, arXiv preprint, (2025). https://arxiv.org/abs/2507.10843

[48] G. N. Raj, *Adaptive and regime-aware RL for portfolio optimization*, arXiv preprint, (2025). https://arxiv.org/abs/2509.14385

[49] M. Rezaei, H. NezamAbAdipour, *Integrating fuzzy logic with deep reinforcement learning to enhance financial portfolio management*, Iranian Journal of Fuzzy Systems, **22**(2) (2025), 187-204. https://doi.org/10.22111/ijfs.2025.50135.8847

[50] A. Saghezchi, V. G. Kashani, F. Ghodratizadeh, *A comprehensive optimization approach on financial resource allocation in Scale-Ups*, Journal of Business and Management Studies, **6**(6) (2024), 62-75. https://doi.org/10.32996/jbms.2024.6.6.5

[51] J. Schulman, F. Wolski, P. Dhariwal, et al., *Proximal policy optimization algorithms*, arXiv preprint, (2017). https://doi.org/10.48550/arXiv.1707.06347

[52] A. Sharma, F. Chen, J. Noh, et al., *Hedging beyond the mean: A distributional reinforcement learning perspective for hedging portfolios with structured products*, arXiv preprint, (2024). https://arxiv.org/abs/2407.10903

[53] T. Skeepers, T. L. Van Zyl, A. Paskaramoorthy, *MA-FDRNN: Multi-asset fuzzy deep recurrent neural network reinforcement learning for portfolio management*, 2021 8th International Conference on Soft Computing and Machine Intelligence (ISCMI), (2021), 32-37. https://doi.org/10.1109/ISCMI53840.2021.9654987

[54] R. Venugopal, C. Veeramani, S. A. Edalatpanah, *Enhancing daily stock trading with a novel fuzzy indicator: Performance analysis using Z-number based fuzzy TOPSIS method*, Results in Control and Optimization, **14** (2024), 100365. https://doi.org/10.1016/j.rico.2023.100365

[55] X. Wang, L. Liu, *Risk-sensitive deep reinforcement learning for portfolio optimization*, Journal of Risk and Financial Management, **18**(7) (2025), 347. https://doi.org/10.3390/jrfm18070347

[56] W. Wiesemann, D. Kuhn, B. Rustem, *Robust Markov decision processes*, Mathematics of Operations Research, **38**(1) (2013), 153-183. https://doi.org/10.1287/moor.1120.0566

[57] H. Yang, X. Y. Liu, S. Zhong, A. Walid, *Deep reinforcement learning for automated stock trading: An ensemble strategy*, Proceedings of the First ACM International Conference on AI in Finance, (2020), 1-8. https://doi.org/10.1145/3383455.3422540

[58] Y. Yang, T. Wang, Y. Fu, et al., *Portfolio management based on value distribution reinforcement learning algorithm*, Frontiers in Artificial Intelligence, **8** (2026), 1709493. https://doi.org/10.3389/frai.2026.1709493

[59] M. Younesi Heravi, I. Jeong, Y. Jang, *A vision-based approach for human activity intensity estimation using kinematic features*, Journal of Computing in Civil Engineering, **40**(6) (2026), 04026092. https://doi.org/10.1061/JCCEE5.CPENG-7683

[60] L. A. Zadeh, *Fuzzy sets*, Information and Control, **8**(3) (1965), 338-353. https://doi.org/10.1016/S0019-9958(65)90241-X

[61] Y. Zhang, X. Li, S. Guo, *Portfolio selection problems with Markowitz's mean-variance framework: A review of literature*, Fuzzy Optimization and Decision Making, **17**(2) (2018), 125-158. https://doi.org/10.1007/s10700-017-9266-z

[62] Y. T. Zhang, J. Y. Yang, Y. Wu, *Dynamic resource allocation strategy of multi-objective fuzzy optimization based on Markov decision process*, IEEE Access, **11** (2023), 99607-99613. https://doi.org/10.1109/ACCESS.2023.3314657

[63] Z. Zhang, S. Zohren, S. Roberts, *Deep reinforcement learning for trading*, arXiv preprint, (2020). https://doi.org/10.48550/arXiv.1911.10107