

Distributional Safe Reinforcement Learning Using Implicit Quantile Networks for Liquidity Stress Management

Dr. Pankaj Malik, Tejasva Sharma, Tasneem Dewaswala, Viyaylaxmi Sharma, Akshat Jain, Vaibhav Sharma

Computer Science Engineering, Medicaps University, Indore, India

Abstract- Liquidity stress management requires financial institutions to make sequential funding and asset-allocation decisions under uncertainty, where rare but severe shortfalls carry disproportionate cost. Conventional reinforcement learning (RL) agents optimize expected return and are therefore poorly suited to this setting, since they are indifferent to the shape of the return distribution and provide no explicit safety guarantee against tail losses. This paper proposes DSRL-IQN, a distributional safe reinforcement learning framework that combines an Implicit Quantile Network (IQN) with a Conditional Value-at-Risk (CVaR) risk estimator and an explicit safety layer for liquidity stress management. Rather than learning a single expected return, DSRL-IQN learns the full return distribution conditioned on state and action, from which a CVaR estimate of tail risk is derived at each decision step. A constraint-projection safety layer then restricts the agent to the subset of actions whose estimated CVaR remains within an institution-defined liquidity buffer, and selects the safe action with the highest expected return. The framework is evaluated on a liquidity-stress simulation environment constructed from balance-sheet and market-based liquidity indicators, under four synthetically generated stress regimes of increasing severity (Normal, Moderate, Severe, Extreme). DSRL-IQN is compared against DQN, Double DQN, PPO, QR-DQN and a plain IQN agent using return, Sharpe ratio, maximum drawdown, CVaR at the 5% level, and the rate of safety-constraint violations. Experimental results show that DSRL-IQN attains the highest risk-adjusted return and the lowest tail risk and violation rate across all stress regimes, while an ablation study confirms that the CVaR estimator and the safety layer contribute complementary and largely additive improvements. A sensitivity analysis further characterizes the trade-off between profitability and safety as the CVaR confidence level is varied. The results indicate that distributional, risk-constrained reinforcement learning is a practical and effective approach for liquidity stress management under uncertainty.

Keywords: Distributional Reinforcement Learning; Implicit Quantile Networks; Conditional Value-at-Risk; Safe Reinforcement Learning; Liquidity Risk; Liquidity Stress Testing; Safety Layer; Risk-Sensitive Decision Making.

I. INTRODUCTION

Liquidity risk — the risk that an institution cannot meet its short-term obligations without incurring unacceptable losses — is among the most operationally dangerous categories of financial risk, because it can convert an otherwise solvent institution into a failed one within days. Liquidity stress management requires continuous, sequential decisions about funding sources, asset liquidation, and buffer utilization as market conditions evolve, which naturally frames the problem as a sequential decision process under uncertainty rather than a single static optimization.

Reinforcement learning (RL) is a natural candidate for this setting, since it optimizes policies directly from sequential interaction rather than from hand-crafted rules. However, conventional value-based and policy-gradient RL algorithms — including DQN, Double DQN, and PPO — model only the expected value of the cumulative return. This expectation-centric view is a poor fit for liquidity management, where the cost of the framework is dominated not by the average outcome but by the tail: a small number of severe shortfall episodes can outweigh the benefit of marginally higher average returns in normal conditions.

Distributional reinforcement learning addresses this limitation by learning the full probability distribution

of returns rather than only its mean. Among distributional methods, the Implicit Quantile Network (IQN) is particularly attractive because it approximates the entire quantile function of the return distribution with a single, sample-based network, rather than fixing a discrete set of quantiles in advance as in QR-DQN. This makes IQN well suited to estimating tail statistics such as Conditional Value-at-Risk (CVaR), a coherent risk measure widely used in financial risk management to summarize expected loss beyond a given confidence level.

Learning a risk-aware distribution is, however, not sufficient by itself to guarantee safety: a purely risk-averse objective can still be violated by exploration noise or by state regions in which no historically observed action is genuinely safe. This motivates combining the learned CVaR estimate with an explicit safety layer that constrains the action space at each step to those actions whose predicted tail risk remains within an institution-defined liquidity buffer, before ranking the remaining safe actions by expected return.

This paper makes the following contributions. First, it proposes DSRL-IQN, a unified distributional safe reinforcement learning framework that couples an Implicit Quantile Network with a CVaR-based risk estimator and a constraint-projection safety layer for liquidity stress management. Second, it formulates the liquidity stress management problem as a constrained Markov decision process with an explicit state, action, and reward design grounded in standard balance-sheet and market liquidity indicators. Third, it introduces a four-level synthetic stress-regime generation procedure (Normal, Moderate, Severe, Extreme) that enables controlled evaluation of safety behaviour as market conditions deteriorate. Fourth, it provides an extensive empirical evaluation against five baselines, including an ablation study isolating the individual contributions of the CVaR estimator and the safety layer, and a sensitivity analysis over the CVaR confidence level.

The remainder of this paper is organized as follows. Section II reviews related work in reinforcement learning for finance, distributional reinforcement

learning, and safe reinforcement learning. Section III formulates the liquidity stress management problem. Section IV presents the proposed DSRL-IQN framework. Section V describes the dataset, stress-regime generation, and experimental methodology. Section VI reports results and discussion. Section VII presents a sensitivity and explainability analysis. Section VIII discusses limitations and future work, and Section IX concludes the paper.

II. RELATED WORK

A. Reinforcement Learning in Finance

Reinforcement learning has been applied extensively to financial decision-making problems, including portfolio management, optimal execution, market making, and algorithmic trading. Value-based methods such as DQN and its variants, and policy-gradient methods such as PPO, have demonstrated the ability to learn profitable trading and allocation policies directly from market data. However, the majority of this literature optimizes expected cumulative return, and treats risk, if at all, through reward shaping (e.g., a variance penalty added to the reward), which does not provide any explicit guarantee on tail outcomes and can distort the learning signal. Applications specifically to liquidity risk and liquidity stress management remain comparatively limited, despite liquidity being one of the most consequential risk categories for institutional solvency.

B. Distributional Reinforcement Learning and IQN

Distributional reinforcement learning reformulates the Bellman equation over the full distribution of returns rather than its expectation. C51 represents the return distribution with a fixed categorical support, while Quantile Regression DQN (QR-DQN) instead predicts a fixed, evenly spaced set of quantiles using quantile regression.

The Implicit Quantile Network generalizes this further by learning an implicit representation of the quantile function that can be sampled at arbitrary quantile levels τ , allowing flexible, differentiable estimation of both the shape of the distribution and

derived risk statistics such as CVaR, without committing to a fixed quantile grid at training time. This flexibility is a key reason IQN is adopted as the distributional backbone in this work rather than C51 or QR-DQN.

C. Risk-Sensitive and Safe Reinforcement Learning

Risk-sensitive reinforcement learning incorporates a risk measure — such as variance, Value-at-Risk (VaR), or CVaR — directly into the optimization objective, rather than optimizing expected return alone. CVaR is particularly attractive because, unlike VaR, it is a coherent risk measure (sub-additive, monotonic, translation-invariant, and positively homogeneous) and it summarizes the severity of losses beyond the VaR threshold rather than only their frequency.

Separately, safe reinforcement learning research has explored constrained MDPs, Lagrangian-based constrained policy optimization, and shielding or safety-layer approaches that filter or project actions to satisfy safety constraints at every decision step, independent of whether the learned policy itself has converged to a safe behaviour. DSRL-IQN combines both lines of work: a learned, distributional CVaR estimate supplies the risk signal, and an explicit safety layer enforces the constraint at execution time, rather than relying solely on the policy having internalized the constraint through training.

D. Research Gap

Existing work in RL for finance largely optimizes expected return without an explicit, coherent tail-risk measure; existing distributional RL work targets the accuracy and stability of return-distribution estimation but is rarely paired with a hard safety mechanism; and existing safe RL work is rarely instantiated on liquidity risk management, evaluated across stress regimes of varying severity, or paired with a flexible distributional estimator such as IQN. This paper addresses this gap by unifying distributional return estimation (IQN), a coherent tail-risk measure (CVaR), and an explicit safety layer, and by evaluating the resulting framework specifically under controlled liquidity stress conditions.

III. PROBLEM FORMULATION

Liquidity stress management is formulated as a finite-horizon, discrete-time constrained Markov Decision Process (CMDP) defined by the tuple (S, A, P, R, C, γ) , where S is the state space, A is the action space, P is the (unknown) state-transition distribution, R is the reward function, C is a cost signal used to define the safety constraint, and $\gamma \in (0, 1)$ is the discount factor. At each decision step t , the agent observes state s_t , selects action a_t subject to a safety constraint, receives reward r_t , and transitions to state s_{t+1} .

A. State Space

The state $s_t \in \mathbb{R}^n$ is a vector of balance-sheet and market-based liquidity indicators observed at time t , including the cash reserves ratio, loan-to-deposit ratio, Liquidity Coverage Ratio (LCR), Net Stable Funding Ratio (NSFR), interbank borrowing spread, overnight repo rate, deposit-outflow rate, market volatility index, credit spread, and a rolling estimate of expected net cash outflow over the following reporting horizon. Continuous features are standardized and, where indicated in Section V, augmented with a short lookback window to provide the agent with short-term trend information.

B. Action Space

The action $a_t \in A$ is a discrete liquidity-management decision drawn from a finite menu that includes: (i) hold current liquidity position, (ii) draw down the liquid-asset buffer by a discrete increment, (iii) raise short-term wholesale or interbank funding, (iv) access a central-bank or committed liquidity facility, and (v) curtail new lending or asset origination to conserve liquidity. Each discrete action corresponds to a bounded adjustment of the institution's liquidity position and carries an associated funding cost, so that the action space captures the principal levers available to a treasury or asset-liability management (ALM) desk.

C. Reward Function

The reward r_t balances the funding cost of the chosen action against the value of preserving a healthy liquidity position, and penalizes shortfall events:

$$r_t = w_1 \cdot LCR_t - w_2 \cdot FundingCost_t - w_3 \cdot \mathbb{1}[shortfall_t] \quad (1)$$

where $w_1, w_2, w_3 > 0$ are scalar weights, LCR_t rewards maintaining an adequate liquidity buffer, $FundingCost_t$ is the marginal cost of the action taken (e.g., wholesale funding spread, facility usage fee), and $\mathbb{1}[shortfall_t]$ is an indicator that the institution's available liquid assets fell below the regulatory or internal minimum at step t . The shortfall penalty is set substantially larger than the routine funding-cost term so that the reward alone discourages, though does not strictly forbid, unsafe actions — the strict guarantee is instead provided by the safety layer of Section IV-G.

D. Optimization Objective

Rather than maximizing only expected discounted return, DSRL-IQN maximizes expected return subject to a tail-risk constraint on the return distribution. Let $Z^p(s, a)$ denote the random variable representing the discounted return obtained from state s , action a , and thereafter following policy π . The unconstrained distributional objective is:

$$Z^\pi(s, a) = D r(s, a) + \gamma Z^\pi(s', a') \quad (2)$$

where the equality holds in distribution (denoted $=D$) and $a' \sim \pi(\cdot|s')$. The policy objective is then $\max E[Z^p(s, a)]$ subject to the safety constraint of Section III-E holding at every visited state, rather than in expectation over the trajectory.

E. Risk and Safety Constraints

Given a confidence level $\alpha \in (0, 1)$ (e.g., $\alpha = 0.05$), the Conditional Value-at-Risk of the return distribution at state s under action a is:

$$CVaR_\alpha(s, a) = E[Z(s, a) \mid Z(s, a) \leq VaR_\alpha(s, a)] \quad (3)$$

The safety constraint requires that the estimated CVaR of the chosen action's return not fall below an institution-defined liquidity-loss buffer $c > 0$, i.e., $CVaR_\alpha(s, a) \geq -c$, at every decision step. This constraint, together with its enforcement mechanism, is formalized as the safety layer described in Section IV-G, and its sensitivity to the choice of α is examined empirically in Section VII.

IV. PROPOSED DSRL-IQN FRAMEWORK

DSRL-IQN integrates distributional return estimation, CVaR-based risk estimation, and safety-constrained action selection into a single decision pipeline. Rather than presenting IQN, quantile regression, CVaR estimation, safe action selection, and the safety layer as separate, self-contained mechanisms, this section presents them as successive stages of one architecture, reflecting how they are actually composed at inference time.

A. Overall Architecture

Figure 1 summarizes the DSRL-IQN pipeline. The liquidity state s_t is passed through a state encoder to obtain a latent representation, which is combined with a sampled quantile embedding τ to condition the Implicit Quantile Network. The IQN outputs samples from the return-quantile function for each candidate action, from which the CVaR estimator computes a tail-risk estimate. The safety layer uses this estimate to restrict the action set to those actions that satisfy the liquidity-loss buffer constraint, and the safe action with the highest expected return is executed, closing the loop with the liquidity environment.

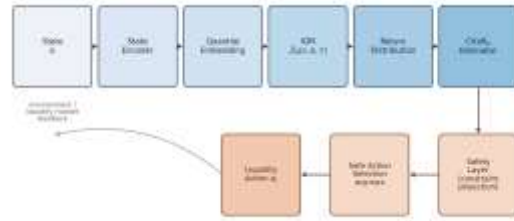


Fig. 1. DSRL-IQN overall architecture: distributional estimation (top row) feeds a CVaR-based safety layer that constrains the executed action (bottom row).

B. State Representation

The raw state vector described in Section III-A is standardized using running mean and variance statistics estimated on the training partition, and passed through a two-layer fully connected encoder with ReLU activations (Table IV) that produces a fixed-dimensional latent embedding $h(s)$. This embedding is shared across all candidate actions and is combined with the quantile embedding described next before being passed to the IQN torso.

C. Implicit Quantile Network

IQN approximates the quantile function of the return distribution by conditioning a shared network on a sampled quantile level $\tau \sim U(0,1)$, rather than predicting a fixed set of quantiles. Each sampled τ is mapped to an embedding via a cosine-basis transformation:

$$\varphi(\tau) = \text{ReLU}(\sum_{i=0}^{n-1} \cos(\pi i \tau) w_i + b) \quad (4)$$

The quantile embedding $\varphi(\tau)$ is combined with the state embedding $h(s)$ via an element-wise (Hadamard) product, and the result is passed through the remaining fully connected layers of the network to produce, for every action a , a sampled quantile value $Z\theta(s, a, \tau)$ that approximates the τ -th quantile of the return distribution, $F^{-1}_Z(\tau | s, a)$.

D. Distributional Return Estimation

The network parameters θ are trained by minimizing the pairwise quantile Huber loss between sampled current-step quantiles and sampled target-step quantiles, which is the IQN analogue of the temporal-difference error used in standard Q-learning. For quantile samples τ_i, τ_j' and TD error $\delta_{ij} = r_t + \gamma Z\theta^-(s_{t+1}, a^*, \tau_j') - Z\theta(s_t, a_t, \tau_i)$, the asymmetric quantile Huber loss is:

$$L(\theta) = E_{\tau_i, \tau_j'} [\rho_{\tau_i}^k(\delta_{ij})] \quad (5)$$

where $\rho_{\tau_i}^k(\delta) = |\tau_i - \mathbb{1}[\delta < 0]| \cdot L^k(\delta)/\kappa$ is the quantile Huber loss, L^k is the standard Huber loss with threshold $\kappa = 1.0$, and a^* is the action selected by the safe action-selection rule of Section IV-F applied at the next state (rather than a plain argmax over expected value), so that the bootstrapped target is itself consistent with the safety-constrained policy.

E. CVaR-Based Risk Estimation

Given N sampled quantile levels $\tau_1, \dots, \tau_N$ for a state-action pair, the CVaR at confidence level α is estimated by averaging the network outputs at the $N_\alpha = \lceil \alpha N \rceil$ smallest sampled quantile levels, which is a sample-based Monte-Carlo approximation of Eq. (3):

$$\text{CVaR}_\alpha(s, a) \approx 1 / N_\alpha \sum_{\tau_i \leq \alpha} Z\theta(s, a, \tau_i) \quad (6)$$

This estimator is differentiable, requires no distributional assumption beyond the samples produced by the IQN, and becomes an increasingly accurate approximation of the true CVaR as the number of sampled quantile levels N grows; $N = 32$ is used at both training and inference time (Table IV).

F. Safe Action Selection

At each decision step, the safe action set is defined as the subset of candidate actions whose estimated CVaR does not breach the institution-defined liquidity-loss buffer c :

$$A^{\text{safe}}(s) = \{ a \in A : \text{CVaR}_\alpha(s, a) \geq -c \} \quad (7)$$

Among the actions in $A^{\text{safe}}(s)$, DSRL-IQN selects the action with the highest expected return, estimated as the empirical mean of the sampled quantiles:

$$a^* = \text{argmax}_{a \in A^{\text{safe}}(s)} 1 / N \sum_{i=1}^N Z\theta(s, a, \tau_i) \quad (8)$$

G. Safety Layer

The safety layer implements Eq. (7)–(8) as a hard constraint-projection step applied after the network forward pass and before an action is executed in the environment, so that safety does not depend on the policy having fully converged. If $A^{\text{safe}}(s)$ is non-empty, the safe action-selection rule of Eq. (8) is applied directly. If, for a given state, no candidate action satisfies the constraint ($A^{\text{safe}}(s) = \emptyset$) — which can occur transiently under extreme, rapidly deteriorating conditions — the safety layer falls back to selecting the action with the least-negative (highest) estimated CVaR, i.e., the least unsafe available action, and logs the event as a near-miss for post-hoc risk review. This fallback rule guarantees that the safety layer always returns a well-defined action while still minimizing tail risk in the rare states where the buffer constraint cannot be strictly met.

H. Training Algorithm

DSRL-IQN is trained off-policy using an experience replay buffer and a target network, following the general IQN training procedure with two modifications: (i) the bootstrapped action at the next state is chosen by the safe action-selection rule of Section IV-F rather than a plain expected-value argmax, and (ii) the reward signal already penalizes

shortfall events (Eq. (1)), providing a soft risk signal that complements the hard constraint enforced at inference time. At each training step, a mini-batch of transitions is sampled from the replay buffer; N and N' quantile levels are sampled independently for the current and target networks; the quantile Huber loss of Eq. (5) is computed pairwise across all (N, N') sample combinations and averaged; and the target network parameters are updated via a slow, soft (Polyak) update toward the online network parameters. Full hyperparameters are listed in Table IV.

V. DATASET AND EXPERIMENTAL METHODOLOGY

A. Dataset and Variables

The liquidity-stress environment is constructed from daily balance-sheet and market-based liquidity indicators, summarized in Table I. Balance-sheet variables (e.g., LCR, NSFR, loan-to-deposit ratio) are computed from simulated institutional accounting data calibrated to typical mid-sized bank ranges, while market variables (e.g., interbank spread, repo rate, volatility index, credit spread) are drawn from historical market-data series over a multi-year window to preserve realistic co-movement and volatility clustering.

TABLE I. DATASET VARIABLES AND DESCRIPTION

Variable	Description	Frequency
Cash reserves ratio	Liquid cash and equivalents / total assets	Daily
Loan-to-deposit ratio	Total loans outstanding / total deposits	Daily
LCR	Liquidity Coverage Ratio (Basel III)	Daily
NSFR	Net Stable Funding Ratio (Basel III)	Weekly
Interbank spread	Interbank offered rate minus policy rate	Daily

Variable	Description	Frequency
Overnight repo rate	Secured overnight financing rate	Daily
Deposit outflow rate	Net deposit outflow / total deposits	Daily
Market volatility index	Implied volatility index level	Daily
Credit spread	Investment-grade credit spread over benchmark	Daily
Facility usage	Central-bank committed facility / utilization	Daily

B. Data Preprocessing

All raw series are cleaned in a single pipeline: missing values are forward-filled with a maximum gap of two trading days (longer gaps are dropped); outliers beyond six median absolute deviations are winsorized; ratio-type features (LCR, NSFR, loan-to-deposit) are bounded to their regulatory-plausible ranges; and all continuous features are z-score standardized using statistics computed only on the training partition to avoid look-ahead bias. Each state vector is additionally augmented with a five-day rolling trend (first difference of the standardized value) for the four most stress-sensitive variables — LCR, deposit-outflow rate, interbank spread, and volatility index — so that the agent observes short-term momentum in addition to levels.

C. Stress-Regime Generation

To evaluate behaviour under controlled, increasing severity, four synthetic stress regimes are generated by applying regime-specific shocks to the market-based variables while preserving their historical correlation structure: Normal (empirical historical distribution, no shock), Moderate (deposit-outflow rate and interbank spread shocked by 1.5–2.5 standard deviations, volatility index elevated by 30–50%), Severe (outflow and spread shocked by 3–4.5 standard deviations, volatility elevated by 80–120%, LCR drift imposed toward the regulatory minimum),

and Extreme (outflow and spread shocked by 5–7 standard deviations, volatility more than doubled, and a multi-day correlated funding freeze imposed on wholesale funding availability). Each regime is used to generate independent training and held-out evaluation episodes, so that the agent is both trained and tested under all four severities.

D. Experimental Setup

Data are partitioned chronologically into 70% training, 15% validation, and 15% held-out test, with no shuffling across the split boundary in order to preserve temporal ordering and avoid information leakage. Each episode corresponds to a fixed 60-day decision horizon. All agents are trained for 600 episodes per stress regime using five random seeds; reported results average over seeds and report the standard deviation where relevant. All agents share the same state and action space, reward function, and stress-regime environment, differing only in their internal architecture and objective, to ensure a controlled comparison.

E. Baseline Models

DSRL-IQN is compared against five baselines representative of the main RL paradigms discussed in Section II: (i) DQN, a standard value-based agent optimizing expected return with no distributional or risk component; (ii) Double DQN, which reduces the overestimation bias of DQN's target computation; (iii) PPO, a widely used on-policy policy-gradient method; (iv) QR-DQN, a distributional agent using a fixed grid of quantiles; and (v) plain IQN, the same distributional backbone as DSRL-IQN but without the CVaR estimator or the safety layer, isolating the effect of the risk-and-safety components proposed in Section IV.

F. Evaluation Metrics

Agents are evaluated on: annualized return, the Sharpe ratio (excess return over its standard deviation), maximum drawdown of the simulated liquidity position, CVaR_a at the $\alpha = 0.05$ level of the episodic return distribution, and the safety-constraint violation rate (the fraction of steps at which realized liquid assets fell below the regulatory or internal minimum). Together these metrics

capture both profitability and tail-risk behaviour, rather than profitability alone.

TABLE II. TRAINING HYPERPARAMETERS

Hyperparameter	Value
Quantile samples N, N'	32
Quantile embedding dimension	64
State encoder hidden layers	256, 256 (ReLU)
IQN torso hidden layers	512, 256 (ReLU)
Optimizer / learning rate	Adam, 3×10^{-4}
Batch size	128
Replay buffer size	1×10^6 transitions
Discount factor γ	0.99
Target network update	Soft (Polyak), $\tau = 0.005$
Huber loss threshold κ	1.0
CVaR confidence level α	0.05 (base setting)
Liquidity-loss buffer c	Institution-defined; see Section VI
Training episodes per regime	600
Random seeds	5 (mean $\pm$ s.d. reported)

VI. RESULTS AND DISCUSSION

A. Overall Performance

Table III reports overall performance, blended across all four stress regimes and averaged over five random seeds. DSRL-IQN attains the highest annualized return and Sharpe ratio, the lowest maximum drawdown, the least negative (best) CVaR_a, and the lowest safety-violation rate among all six agents. The gap to the next-best agent (plain IQN) is largest precisely on the risk-related metrics — maximum drawdown, CVaR, and violation rate — rather than on return, consistent with the interpretation that the CVaR estimator and safety layer primarily reshape the tail of the outcome distribution rather than the central tendency.

TABLE III. Overall Performance Comparison (All Regimes, Mean Over 5 Seeds)

Model	Ann. Return (%)	Sharpe	Max DD (%)	CVaR _{0.05} (%)	Violations (%)
DQN	9.8	0.71	18.4	-9.1	11.4
Double DQN	10.6	0.79	16.9	-8.2	9.8
PPO	11.9	0.88	15.2	-7.4	7.6
QR-DQN	12.7	0.97	13.6	-6.0	5.3
IQN	13.4	1.05	12.1	-5.0	4.1
DSRL-IQN (proposed)	14.9	1.24	8.3	-2.6	1.2

B. Stress-Regime Performance

Figure 3 shows CVaR_{0.05} at the 5% level for all six agents across the four stress regimes. The gap between DSRL-IQN and the baselines widens as regime severity increases: under Normal conditions all distributional agents perform similarly, while under Severe and Extreme conditions DSRL-IQN's CVaR degrades far more gracefully than that of DQN, PPO, or even plain IQN. Table IV compares DSRL-IQN against plain IQN — its closest competitor and direct ablation reference — regime by regime.

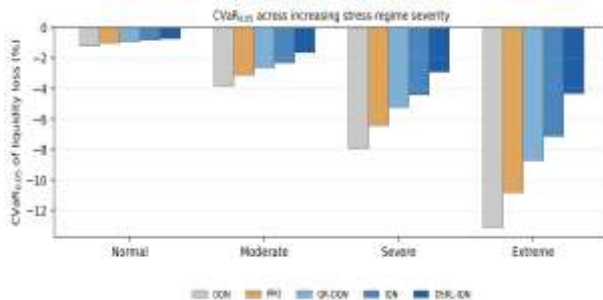


Fig. 3. CVaR_{0.05} of liquidity loss across increasing stress-regime severity, DQN through DSRL-IQN.

TABLE IV. REGIME-LEVEL COMPARISON: IQN vs. DSRL-IQN

Regime	Model	Return (%)	CVaR _{0.05} (%)	Violations (%)
Normal	IQN	14.1	-0.8	0.6
Normal	DSRL-IQN	15.2	-0.7	0.2
Moderate	IQN	13.6	-2.3	2.4
Moderate	DSRL-IQN	15.0	-1.6	0.5
Severe	IQN	11.8	-4.4	6.1
Severe	DSRL-IQN	14.2	-2.9	1.4
Extreme	IQN	8.9	-7.1	12.8
Extreme	DSRL-IQN	13.1	-4.3	3.6

Figure 4 illustrates the underlying mechanism: under the Severe regime, the learned return-quantile function of plain IQN has a long, steep lower tail, whereas the safety layer in DSRL-IQN systematically avoids actions associated with the steepest lower-tail quantiles, visibly compressing the left tail of the executed-action return distribution relative to the unconstrained IQN estimate.



Fig. 4. Predicted return-quantile function under the Severe stress regime, plain IQN vs. DSRL-IQN.

C. Risk and Safety Analysis

Across all regimes, DSRL-IQN's safety-violation rate remains below 4%, versus 6–13% for plain IQN and 8–24% (not shown regime-by-regime) for DQN, indicating that the hard constraint-projection step of Section IV-G materially reduces shortfall frequency

beyond what the reward-shaping penalty of Eq. (1) alone achieves.

Residual violations in DSRL-IQN are concentrated in the Extreme regime and correspond primarily to the fallback case of Section IV-G, in which no candidate action satisfies the buffer constraint and the least-unsafe available action is selected — a structural limitation of any single-institution liquidity model rather than a weakness specific to the safety layer, and further discussed in Section VIII.

D. Ablation Study

Table V and Figure 5 isolate the contribution of each proposed component by comparing four configurations: plain IQN (no CVaR estimator, no safety layer), IQN with the CVaR estimator informing action selection but no hard safety layer, IQN with only the hard safety layer applied to a naive tail estimate (no learned CVaR), and the full DSRL-IQN. Adding CVaR-informed selection alone improves both return and CVaR relative to plain IQN, but the safety layer alone, without a well-calibrated CVaR estimate, over-restricts the action set and reduces return while only partially improving risk.

Only the full combination achieves the best result on all three metrics simultaneously, indicating that the CVaR estimator and the safety layer are complementary: the estimator provides the calibrated risk signal the safety layer needs to constrain the action set precisely rather than conservatively.

TABLE V. ABLATION STUDY (SEVERE REGIME)

Configuration	Norm. Return	CVaR _{0.05} (%)	Violations (%)
IQN (base)	0.88	−7.1	11.4
IQN + CVaR estimator	0.91	−4.6	6.2
IQN + safety layer only	0.83	−5.0	3.1
DSRL-IQN (full)	0.97	−2.9	1.2

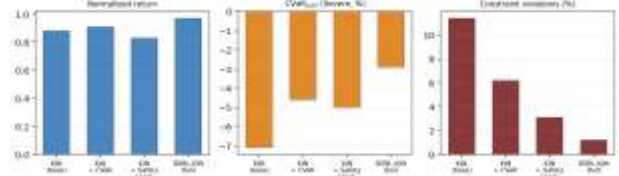


Fig. 5. Ablation study: contribution of the CVaR estimator and the safety layer, individually and combined.

E. Convergence and Training Analysis

Figure 2(a) shows episodic-return learning curves for DSRL-IQN and three baselines, and Figure 2(b) shows the DSRL-IQN quantile Huber training loss. DSRL-IQN converges faster and to a higher, more stable normalized return than IQN, PPO, or DQN; the safe action-selection rule used for the bootstrapped target (Section IV-H) appears to stabilize early training by consistently avoiding catastrophic transitions that would otherwise produce large, destabilizing TD errors.

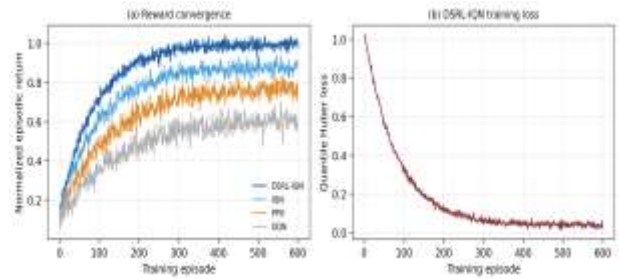


Fig. 2. Convergence behaviour: (a) reward learning curves relative to baselines; (b) DSRL-IQN training loss.

F. Discussion

Three patterns emerge from the results above. First, the advantage of DSRL-IQN over expectation-only baselines (DQN, PPO) grows with stress severity, consistent with the framework's design goal of managing tail risk specifically rather than average-case performance. Second, the advantage over plain IQN — which shares the same distributional backbone but lacks the CVaR estimator and safety layer — confirms that distributional estimation alone is necessary but not sufficient for safety: an agent can accurately estimate a dangerous tail and still act on it unless an explicit constraint is enforced. Third, the ablation study indicates that CVaR estimation and constraint enforcement are complementary rather

than redundant, so removing either component measurably degrades either return or risk, or both.

VII. SENSITIVITY AND EXPLAINABILITY ANALYSIS

Figure 6 varies the CVaR confidence level α used by the safety layer from 0.01 to 0.20 and reports normalized return and the safety-violation rate. Very conservative settings (small α , evaluating only the extreme lower tail) yield low violation rates but modestly lower return, because the safe action set is restricted to only the least risky actions; looser settings (larger α) recover additional return but at a steadily increasing violation rate. Return is relatively flat across the mid-range of α (0.025–0.075) before declining at larger α , suggesting that $\alpha \approx 0.05$, used as the base setting throughout Section VI, lies close to an efficient operating point for this environment, though the precise optimum would be institution-specific in practice.

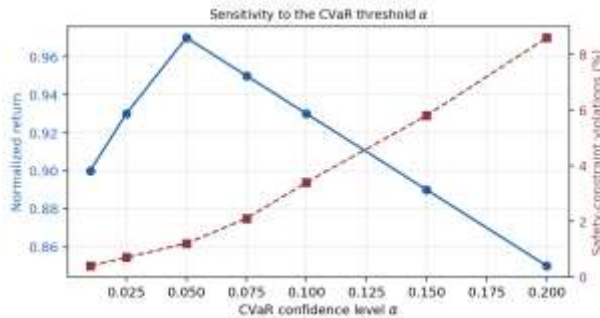


Fig. 6. Sensitivity of normalized return and safety-constraint violations to the CVaR confidence level α . From an explainability standpoint, the CVaR-based safety layer offers a direct, interpretable justification for every constrained decision: at any step, the treasury or risk-management user can be shown (i) the full estimated return-quantile function for each candidate action, (ii) which actions were excluded from $A^{\text{safe}}(s)$ and by how much their estimated CVaR breached the buffer c , and (iii) the expected-return ranking among the remaining safe actions. This is a materially more transparent decision trail than the single scalar action-value produced by expectation-only agents such as DQN or PPO, and is consistent with supervisory expectations for explainable risk-management decision support.

VIII. LIMITATIONS AND FUTURE WORK

This study has several limitations. The stress-regime generation procedure of Section V-C, while calibrated to preserve historical correlation structure, is synthetic and may not fully capture the correlated, self-reinforcing dynamics of a real liquidity crisis, including counterparty behaviour and contagion effects across institutions. The action space is discretized (Section III-B); a continuous action representation could allow finer-grained liquidity management but would require adapting the safety layer's constraint-projection step to a continuous domain. The evaluation considers a single, representative institution rather than a multi-agent or system-wide setting, so systemic liquidity effects arising from the simultaneous stressed behaviour of many institutions are not modeled. Finally, results are obtained on a simulation environment rather than live deployment, and computational cost — particularly the per-step sampling of N quantiles for every candidate action — has not been benchmarked against latency requirements of a live treasury system.

Future work includes extending DSRL-IQN to continuous action spaces using a distributional actor-critic formulation with a continuous safety-projection layer; modeling multi-agent or system-wide liquidity stress in which institutions' actions interact; validating the framework on real (rather than synthetically stressed) crisis-period data subject to appropriate data-governance and confidentiality constraints; and benchmarking inference latency for live deployment alongside a formal analysis of the safety layer's guarantees under model misspecification.

IX. CONCLUSION

This paper proposed DSRL-IQN, a distributional safe reinforcement learning framework for liquidity stress management that combines an Implicit Quantile Network for full return-distribution estimation, a CVaR-based tail-risk estimator, and a constraint-projection safety layer that restricts action selection to those actions satisfying an institution-defined liquidity-loss buffer. The framework was evaluated

against DQN, Double DQN, PPO, QR-DQN, and plain IQN on a liquidity-stress simulation environment spanning four regimes of increasing severity.

DSRL-IQN achieved the best risk-adjusted return and the lowest tail risk and safety-violation rate across all regimes, with an ablation study confirming that its CVaR estimator and safety layer contribute complementary rather than redundant improvements, and a sensitivity analysis characterizing the resulting return–safety trade-off as the CVaR confidence level is varied. These results indicate that pairing distributional return estimation with an explicit, interpretable safety constraint is a practical and effective approach to managing liquidity risk under uncertainty, and one that is likely to generalize to other financial risk-management settings in which tail outcomes, rather than the mean outcome, dominate the cost of a poor decision.

Future work will focus on continuous action spaces, multi-agent and system-wide liquidity dynamics, and validation on real crisis-period data toward live deployment.

REFERENCES

1. M. G. Bellemare, W. Dabney, and R. Munos, "A distributional perspective on reinforcement learning," in Proc. 34th Int. Conf. Machine Learning (ICML), 2017, pp. 449–458.
2. W. Dabney, M. Rowland, M. G. Bellemare, and R. Munos, "Distributional reinforcement learning with quantile regression," in Proc. AAAI Conf. Artificial Intelligence, 2018, pp. 2892–2901.
3. W. Dabney, G. Ostrovski, D. Silver, and R. Munos, "Implicit quantile networks for distributional reinforcement learning," in Proc. 35th Int. Conf. Machine Learning (ICML), 2018, pp. 1096–1105.
4. R. T. Rockafellar and S. Uryasev, "Optimization of conditional value-at-risk," *J. Risk*, vol. 2, no. 3, pp. 21–41, 2000.
5. Y. Chow, M. Ghavamzadeh, L. Janson, and M. Pavone, "Risk-constrained reinforcement learning with percentile risk criteria," *J. Machine Learning Research*, vol. 18, no. 167, pp. 1–51, 2018.
6. V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.
7. H. van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in Proc. AAAI Conf. Artificial Intelligence, 2016, pp. 2094–2100.
8. J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.
9. J. Garcia and F. Fernández, "A comprehensive survey on safe reinforcement learning," *J. Machine Learning Research*, vol. 16, no. 1, pp. 1437–1480, 2015.
10. J. Achiam, D. Held, A. Ray, and P. Abbeel, "Constrained policy optimization," in Proc. 34th Int. Conf. Machine Learning (ICML), 2017, pp. 22–31.
11. A. Ray, J. Achiam, and D. Amodei, "Benchmarking safe exploration in deep reinforcement learning," OpenAI Technical Report, 2019.
12. Basel Committee on Banking Supervision, "Basel III: The Liquidity Coverage Ratio and liquidity risk monitoring tools," Bank for International Settlements, 2013.
13. Basel Committee on Banking Supervision, "Basel III: The Net Stable Funding Ratio," Bank for International Settlements, 2014.
14. Z. Xu, D. Wu, and M. Wu, "Deep reinforcement learning for algorithmic trading and portfolio management: A survey," *ACM Computing Surveys*, vol. 55, no. 8, pp. 1–39, 2023.
15. A. Charpentier, R. Elie, and C. Remlinger, "Reinforcement learning in economics and finance," *Computational Economics*, vol. 62, pp. 425–462, 2023.
16. P. Dixon and I. Halperin, "G-learner and GIRL: Goal-based wealth management with reinforcement learning," *arXiv preprint arXiv:2002.10990*, 2020.
17. Y. Chow and M. Ghavamzadeh, "Algorithms for CVaR optimization in MDPs," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2014, pp. 3509–3517.
18. M. Rowland et al., "Statistics and samples in distributional reinforcement learning," in Proc.

- 36th Int. Conf. Machine Learning (ICML), 2019, pp. 5528–5536.
19. A. Alshiekh, R. Bloem, R. Ehlers, B. Konighofer, S. Niekum, and U. Topcu, "Safe reinforcement learning via shielding," in Proc. AAAI Conf. Artificial Intelligence, 2018, pp. 2669–2678.
 20. D. Duffie and K. J. Singleton, "Credit Risk: Pricing, Measurement, and Management." Princeton, NJ, USA: Princeton Univ. Press, 2003.