Tail-Risk-Sensitive Scheduling of Version Age of Information via Distributional Reinforcement Learning

Haoyuan Pan, *Member, IEEE,* Sizhao Chen, Zhaorui Wang, *Member, IEEE,* Tse-Tin Chan, *Member, IEEE*

***Abstract*—Ensuring timely and semantically accurate information delivery is critical in real-time wireless systems. While Age of Information (AoI) quantifies temporal freshness, Version Age of Information (VAoI) captures semantic staleness by tracking version evolution between transmitters and receivers. Existing VAoI scheduling approaches primarily optimize the long-term average VAoI, but fail to explicitly account for rare yet severe semantic staleness events that may potentially degrade downstream performance under stochastic packet arrivals and unreliable channels. This paper investigates risk-sensitive VAoI scheduling in a multi-user status update system under long-term transmission cost constraints. We formulate the problem as a constrained Markov decision process and adopt Conditional Value-at-Risk (CVaR) to characterize and control the tail behavior of the VAoI process. To solve the resulting problem, we develop a risk-sensitive deep distributional diffusion-based Soft Actor-Critic framework, termed RS-D3SAC, which employs quantile-based distributional critics to model the return distribution and enable CVaR-oriented optimization of the return distribution. In addition, a diffusion-based actor is incorporated as a complementary policy representation mechanism to improve learning stability. Extensive simulations demonstrate that RS-D3SAC consistently achieves lower CVaR of VAoI while maintaining competitive average VAoI performance under transmission cost constraints. Comparisons with D2SAC, a diffusion-policy baseline to isolate the contribution of diffusion-based policy representation, further reveal that the major tail-risk reduction originates from the distributional critic, whereas the diffusion-based actor provides supplementary performance gains. These findings demonstrate the effectiveness of distributional reinforcement learning for risk-sensitive VAoI optimization and highlight the complementary benefits of diffusion-based policy representations in wireless scheduling systems.**

***Index Terms*—Version age of information, diffusion model, distributional reinforcement learning, risk-sensitive scheduling, conditional value-at-risk.**

I. Introduction

In real-time wireless communication systems, maintaining timely and reliable information updates is essential for system perception, control, and decision-making. To quantitatively characterize information freshness, the Age of Information (AoI) has been widely adopted as a metric that measures the time elapsed since the generation of the most recently received update [1]. AoI effectively captures the temporal freshness of information and has been extensively studied in a wide range of communication scenarios, including networked control and cyber-physical systems [2], [3].

With the rapid development of semantic and task-oriented communication paradigms, characterizing information freshness solely from a temporal perspective is no longer sufficient. In many practical systems, information evolves not only over time but also across discrete versions, and the impact of outdated information depends on whether newer and potentially meaningful information states have been received. To capture such version-aware information, the Version Age of Information (VAoI) metric has been introduced to quantify the version discrepancy between transmitters and receivers [4]–[6]. Unlike conventional AoI, VAoI captures how many newer information versions are missing at the receiver, providing a useful freshness measure for systems in which version evolution corresponds to meaningful information updates. For example, in digital twin and intelligent control systems, updates may be generated when the monitored physical state changes or when a new validated state estimate becomes available. In such settings, missing newer versions implies that the destination continues to operate with outdated information, and a larger version gap indicates that more updated information has not yet been received. This version-aware notion of freshness can therefore provide a useful basis for scheduling when outdated information may adversely affect downstream decision-making.

Existing studies on AoI/VAoI optimization and scheduling have predominantly adopted a *risk-neutral* formulation that minimizes the long-term average semantic freshness [3], [7], [8]. While effective in improving mean performance, such formulations do not explicitly characterize the tail behavior of semantic staleness. In VAoI scheduling, rare but severe staleness events arise from the interaction of stochastic channel failures, resource-constrained scheduling, and the temporal coupling of VAoI evolution. In particular, consecutive transmission failures or prolonged non-scheduling intervals can result in persistent accumulation of semantic discrepancy, while transmission cost constraints may further defer updates under limited resources. Consequently, different scheduling policies may exhibit similar average VAoI while having substantially different risks of severe staleness. One policy may maintain moderate VAoI relatively consistently, whereas another may achieve lower VAoI in most slots but occasionally experience prolonged

This work was supported in part by the National Natural Science Foundation of China under Grants 62371302 and 62601121, in part by Shenzhen Science and Technology Program under Grant JCYJ20250604181549064, in part by the Guangdong Basic and Applied Basic Research Foundation (Joint Fund of Dongguan) Youth Project under Grant 2025A1515110318, in part by the Innovation Center for Nuclear Facilities Decommissioning and Radioactive Waste Management Technology, and in part by the Research Matching Grant Scheme from the Research Grants Council of Hong Kong. (*Corresponding author: Haoyuan Pan.*)

H. Pan and S. Chen are with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China (e-mails: hypan@szu.edu.cn, 2510103060@mails.szu.edu.cn).

Z. Wang is with the School of Computing and Information Technology, Great Bay University, Dongguan, China (e-mail: wangzhaorui@gbu.edu.cn).

T.-T. Chan is with the Department of Mathematics and Information Technology, The Education University of Hong Kong, Hong Kong, China (e-mail: tsetinchan@eduhk.hk).

service interruptions that lead to severe VAoI excursions. Compared with AoI where large values mainly reflect temporal delay, a large VAoI represents a sustained discrepancy between the source and destination versions and may therefore have a more direct impact on downstream inference or control. Thus, minimizing average VAoI alone may leave the system vulnerable to unfavorable freshness outcomes that are not reflected by the mean performance.

These observations motivate the development of a *risk-sensitive* VAoI scheduling framework that explicitly controls tail risk while maintaining satisfactory average performance. In this work, we adopt the Conditional Value-at-Risk (CVaR) as the tail-risk metric. CVaR characterizes the expected performance within the worst fraction of realizations and has been widely recognized as a principled and coherent measure for risk-sensitive optimization [9]. By directly accounting for unfavorable outcomes, CVaR provides a direct measure of semantic staleness risk and is particularly relevant to wireless applications in which severe information degradation should be limited [10]. Importantly, CVaR depends on the tail of the underlying performance distribution rather than its expectation alone. Therefore, expectation-based optimization does not retain sufficient information to distinguish different return distributions with similar mean performance but substantially different tail risks. This motivates explicitly characterizing the distribution of future performance, rather than only its expectation, as the basis for risk-sensitive VAoI scheduling.

From a methodological perspective, VAoI scheduling problems are further complicated by stochastic packet arrivals, random transmission outcomes, and long-term resource limitations. These factors induce highly uncertain system dynamics that are difficult to characterize accurately using analytical models. Classical model-based optimization approaches [11], [12] typically require complete system knowledge and often struggle to achieve scalability and robustness in such uncertain environments. In contrast, deep reinforcement learning (DRL) has recently demonstrated strong potential for wireless resource allocation and information freshness optimization due to its data-driven and model-free nature [13]. By learning policies through interactions with the environment, DRL provides a flexible framework for solving complex scheduling problems without requiring explicit system models.

Despite their empirical success, most existing DRL-based VAoI scheduling approaches remain expectation-oriented and therefore provide limited capability for controlling tail risk. Recently, distributional deep reinforcement learning (distributional DRL) has emerged as a promising framework that models the full distribution of future returns rather than only their expectation [14]. For a given state-action pair (s, a), let $Z(s, a)$ denote the corresponding conditional discounted return distribution. Conventional actor-critic methods characterize this distribution only through its mean, $Q(s, a) = \mathbb{E}[Z(s, a)]$, and therefore discard information about the dispersion and tail of future outcomes. In contrast, distributional RL approximates $Z(s, a)$ directly, thereby retaining the quantile information required to evaluate and optimize risk-sensitive objectives such as CVaR. Prior studies have demonstrated the potential of distributional DRL to improve robustness and reliability in wireless resource allocation and radio management under uncertainty [15]. These observations motivate the use of distributional DRL as a natural framework for risk-sensitive VAoI scheduling, where the upper tail of VAoI, equivalently the lower tail of the corresponding return, is of interest.

In addition to risk-sensitive learning, policy representation plays an important role in wireless scheduling problems. Although standard multilayer perceptron (MLP)-based policies can theoretically represent arbitrary distributions over discrete actions, learning effective policies in VAoI scheduling is challenging due to the strong temporal coupling between scheduling decisions and future semantic staleness. In particular, the scheduler may need to rapidly adjust priorities among users as their VAoIs evolve, while simultaneously accounting for stochastic transmission outcomes and long-term resource constraints. Unstable policy updates may therefore result in unfavorable service sequences, causing persistent VAoI accumulation and increasing the risk of severe tail outcomes. Therefore, more expressive policy representation can be beneficial for CVaR-oriented optimization. Recent studies suggest that by generating actions through an iterative denoising process, diffusion-based policy representations can provide smoother policy updates and improved learning stability in DRL tasks [16], [17]. Accordingly, we incorporate diffusion-based policy generation as a complementary mechanism to improve the stability and robustness of risk-sensitive VAoI scheduling.

This paper investigates risk-sensitive VAoI scheduling in a multi-user wireless status update system with stochastic packet arrivals and unreliable transmission channels. The problem is formulated as a constrained Markov decision process (CMDP) under long-term transmission cost constraints, where CVaR is adopted to characterize the tail behavior of the VAoI process. To solve the resulting problem, we develop a risk-sensitive deep distributional diffusion model-based Soft Actor-Critic (RS-D3SAC) algorithm. Specifically, RS-D3SAC incorporates quantile-based distributional critics inspired by QR-DQN [14] to explicitly model the return distribution and enable CVaR-oriented policy optimization. In addition, a diffusion-based actor is introduced as a complementary component to improve policy smoothness and learning stability through an iterative denoising process. RS-D3SAC combines a reward formulation based on semantic staleness and transmission cost constraints with a Lagrangian primal-dual update mechanism to enforce long-term resource constraints during learning. For comparison, we adopt D2SAC as a risk-neutral diffusion-policy baseline [18]. Relative to D2SAC, the proposed RS-D3SAC replaces scalar critics with quantile-based distributional critics, thereby enabling CVaR-oriented policy optimization while preserving the same diffusion-based policy architecture.

Extensive simulation results demonstrate that the proposed RS-D3SAC algorithm achieves lower CVaR of VAoI compared with standard SAC and other conventional DRL approaches, while maintaining competitive average VAoI performance under long-term transmission cost constraints. In addition, RS-D3SAC achieves average VAoI performance comparable to that of the Lyapunov optimization-based Drift-Plus-Penalty (DPP) benchmark [19], which leverages system model param-

eters to optimize average freshness. In contrast, RS-D3SAC achieves lower CVaR than DPP, demonstrating its advantage in suppressing severe VAoI realizations. Comparisons with D2SAC further reveal that the the distributional critic contributes more substantially to tail-risk control, whereas the the diffusion-based actor provides moderate but consistent performance improvements through enhanced policy stability and expressiveness.

The main contributions of this paper are summarized as follows:

- We put forth a risk-sensitive VAoI scheduling framework for multi-user wireless status update systems under long-term transmission cost constraints. Unlike conventional average-oriented VAoI optimization, the proposed framework adopts CVaR to explicitly characterize and control rare but extreme tail realizations of the VAoI process.
- We develop the RS-D3SAC algorithm, which integrates quantile-based distributional critics with a diffusion-based actor to enable CVaR-oriented policy optimization in constrained VAoI scheduling problems. The proposed framework combines distributional reinforcement learning for tail-risk control with a diffusion-based policy representation mechanism for improved policy smoothness and learning stability.
- Extensive simulations demonstrate that RS-D3SAC achieves substantial reductions in CVaR of VAoI while maintaining competitive average VAoI performance. The results further show that the distributional critic provides the larger marginal improvement in our ablation study, whereas the diffusion-based actor provides complementary performance improvements.

II. Related Work

A. Semantic Freshness Metrics and Scheduling Approaches

AoI has become a fundamental metric for characterizing information timeliness in wireless status update systems, measuring the elapsed time since the most recently received update was generated [1]. It has been extensively studied in scenarios where timely information delivery is critical. We refer readers to the monograph [3] and the references therein for comprehensive overviews of AoI theory and applications. To move beyond purely temporal freshness, several semantic- and task-aware metrics have been proposed. VAoI extends AoI by specifically accounting for version mismatches between transmitters and receivers, thus capturing semantic staleness in systems where information evolves across discrete versions [4]–[6]. Related metrics, such as Urgency of Information (UoI) [38] and Goal-Oriented Tensor (GoT) [39], further emphasize task-dependent or goal-oriented freshness requirements.

Under these metrics, a large body of work has focused on scheduling policies that minimize long-term average freshness. Model-based approaches have been developed for VAoI optimization in a variety of settings, including energy-harvesting gossip networks [4], [5], federated learning systems [7], and fading broadcast channels [12]. More recently, [29] incorporated CVaR into AoI optimization to mitigate rare but severe staleness events, while [30] proposed statistical AoI as a tunable metric interpolating between average and peak age. Although these methods provide valuable analytical insights and risk-sensitive performance guarantees, they typically rely on precise system models or simplified assumptions, which limit scalability and robustness in highly dynamic wireless environments. Moreover, to the best of our knowledge, no systematic risk-sensitive study of VAoI has targeted the semantic staleness tail behavior.

To address modeling limitations, DRL has been introduced as a model-free alternative for VAoI-aware scheduling. For example, deep-Q-learning-based methods have been applied to energy-harvesting IoT systems to learn adaptive scheduling policies under uncertainty [23]. However, existing learning-based approaches remain largely risk-neutral, focusing on minimizing average VAoI only. While a few studies [31], [32] have applied DRL to risk-sensitive AoI objectives, the fundamental distinction between AoI and VAoI prevents these methods from directly controlling rare but severe semantic staleness through version evolution. Such events can disproportionately affect reliability and safety in stochastic wireless systems, motivating the adoption of tail-risk-aware metrics such as CVaR for VAoI scheduling, as pursued in the current work.

B. Diffusion Model-Based Reinforcement Learning

Diffusion models have recently been incorporated into DRL as expressive generative policy representations that can improve policy smoothness and learning stability [24], [27]. In continuous action spaces, Diffusion Q-Learning (DQL) employs conditional diffusion models to represent policies and enables efficient policy improvement in offline reinforcement learning settings [24]. Beyond continuous control, diffusion-based policy generation has been extended to discrete and structured decision spaces relevant to communication networks. The diffusion-based policy usually operates in a latent continuous space and is later mapped to discrete scheduling decisions (via a softmax transformation) [17], [18]. The work in [18] integrated diffusion models into the SAC framework to generate optimal decisions for AI-generated content service providers. Diffusion-based policies have also been applied to cloud-edge collaborative scheduling driven by large language model (LLM) requests [25], as well as satellite edge computing, where diffusion-enhanced DRL improves decision quality and generalization [26].

Although these studies demonstrate the potential of diffusion-based policy representations, existing works primarily focus on average-performance optimization and do not address risk-sensitive objectives. In particular, their application to information freshness optimization, especially to VAoI-aware scheduling under tail-risk constraints, is not well investigated.

C. Distributional Deep Reinforcement Learning and Risk-Sensitive Wireless Optimization

Distributional DRL departs from classical DRL by modeling the entire return distribution rather than only its expectation, therefore providing a richer characterization of uncertainty in stochastic environments. Representative algorithms include C51 [28], which discretizes the return distribution support,

TABLE I: Comparison of Our Proposed Algorithms with Existing Works

References	Metric (AoI/VAoI)	RL-based	Distributional RL	Diffusion-based	Risk-Aware
Category 1: Risk-Neutral Approaches (Focusing on Average Performance)					
[20], [21]	AoI	×	×	×	×
[4]- [8]	VAoI	×	×	×	×
[22]	AoI	✓	×	×	×
[23]	VAoI	✓	×	×	×
[17], [18], [24], [25], [26], [27]	N/A	✓	×	✓	×
Category 2: Risk-Sensitive & Distributional Approaches (Focusing on Tail-Risk)					
[28], [14]	N/A	✓	✓	×	×
[29], [30]	AoI	×	×	×	✓
[31], [32]	AoI	✓	×	×	✓
[10], [33], [34]	N/A	✓	×	×	✓
[15], [35], [36], [37]	N/A	✓	✓	×	✓
Ours (RS-D3SAC)	**VAoI**	✓	✓	✓	✓

and QR-DQN [14], which parameterizes the distribution using quantile functions and offers improved numerical stability.

By explicitly capturing tail behavior, distributional DRL has shown clear advantages in risk-sensitive decision-making problems [36]. CVaR has emerged as a principled tail-risk metric and has been incorporated into DRL frameworks to improve robustness under adverse conditions [37]. Representative works include CVaR-optimized policy gradient methods [33] and constrained policy optimization with CVaR-aware objectives [10]. In wireless systems, these techniques have been applied to scenarios involving channel or interference uncertainty, and latency variability [15], [34], [35].

Despite these advances, the application of distributional reinforcement learning to VAoI-aware wireless scheduling remains largely unexplored. As summarized in Table I, most existing studies focus on either risk-neutral optimization or partially risk-aware formulations. To the best of our knowledge, none explicitly investigates CVaR-oriented distributional reinforcement learning for controlling the tail behavior of VAoI under long-term transmission cost constraints. To address this gap, we develop RS-D3SAC, a CVaR-oriented distributional DRL framework for VAoI scheduling that incorporates diffusion-based policy generation as a complementary mechanism to improve policy smoothness and learning stability. This framework establishes a connection between semantic freshness optimization and risk-sensitive distributional reinforcement learning.

TABLE II: Key Parameter and Symbol Descriptions

Symbol	Description
N, n	Total number of users and user index
r_n	Packet generation rate of user n
p	Packet transmission success probability
$G(t,n)$	Newest packet version index at the scheduler
$B(t,n)$	Delivered packet version index at the destination
$\mathcal{D}(t,n)$	Instantaneous VAoI
$\mathcal{D}_{max}$	Maximum truncated VAoI
$\mathcal{Z}$	Distribution of VAoI samples across users and time
$s_t, \mathcal{S}$	System state at slot t and state space
$a_t, \mathcal{A}$	Scheduling action at slot t and action space
$P, p(\cdot\|\cdot)$	State transition dynamics and probability
R, r_t	Reward function and immediate reward at slot t
c_t	Binary instantaneous transmission cost variable
η, η_{max}	Average transmission cost and its maximum budget
λ	Lagrange multiplier for the cost constraint
δ	Update step size for the Lagrange multiplier
γ	Discount factor for future rewards
K	Number of denoising steps for the diffusion model
$\mathrm{q}(\cdot\|\cdot)$	Forward diffusion process distributions
ϵ	Standard Gaussian noise term
$p_\theta(x_0)$	Generative distribution parameterized by θ
$\epsilon_\theta(\cdot)$	Learned denoising noise predictor network
π_θ	Stochastic policy distribution
$\theta^*, \theta, \hat{\theta}$	Actor network parameters (optimal, online, target)
$\phi^i, \hat{\phi}^i$	Scalar critic network parameters (in D2SAC) ($i = 1, 2$)
$\xi^i, \hat{\xi}^i$	Distributional critic network parameters ($i = 1, 2$)
$\mathcal{M}, \mathcal{B}, b$	Replay buffer, mini-batch, and batch size
$y^i, \hat{y}$	Critic i estimate and target Q-value (TD target)
$\mathcal{N}_q$	Number of quantiles in quantile regression
$\tau_1, \ldots, \tau_{\mathcal{N}_q}$	Uniform probability intervals (quantile fractions)
σ_j	Learned quantile value for the j-th interval
κ	Huber loss threshold in the quantile Huber loss
α	System-level CVaR confidence level
φ	Return-level lower-tail probability
Z	Random return distribution
δ_z	Dirac delta distribution centered at z

III. System Model and Performance Metrics

This section presents the system model and the VAoI metric used to quantify information staleness in multi-user status update systems. Building on these foundations, we formalize a system-level risk-sensitive metric, namely the CVaR of VAoI, which accounts for extreme staleness events. The key notations used throughout this paper are summarized in Table II.

A. System Model

We consider a discrete-time multi-user status update system, as illustrated in Fig. 1. A central scheduler receives state updates from N independent users and forwards them to a destination over a shared wireless channel. Time is slotted, and all slots have equal duration. At the beginning of slot t, user $n \in \{1, \ldots, N\}$ generates a new status packet with probability $r_n \in (0, 1]$. Packet generation is instantaneous, and newly generated packets arrive at the scheduler without delay or error. The scheduler maintains a dedicated buffer for each user, storing only the most recently generated packet (i.e., N buffers in total). Whenever a new packet arrives, the previous packet for that user is overwritten to prevent the accumulation of stale information.

The wireless channel from the scheduler to the destination is assumed to support at most one transmission per slot. We assume the packet duration occupies the entire slot. The scheduler must determine which user, if any, to transmit to. We define the scheduling action at slot t as $a_t \in \mathcal{A} = \{0, 1, \ldots, N\}$, where $a_t = 0$ represents idling and $a_t = n$ indicates that

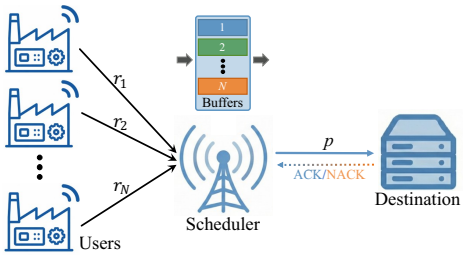


Fig. 1: System model for version-aware status updating: N users generate meaningful, versioned information updates, such as event-driven perception data and digital twin state updates, and transmit them to a destination over unreliable wireless channels. At most one user is scheduled per time slot.

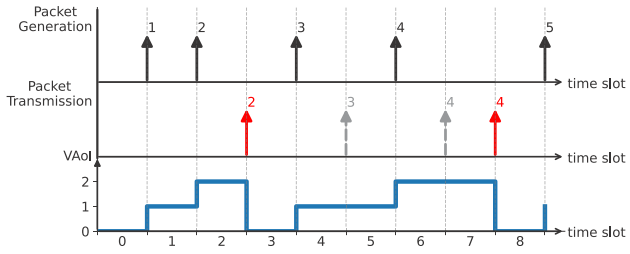


Fig. 2: Example of a single user's VAoI evolution. Black arrows indicate packet generation, red arrows indicate successful packet transmission, and gray dashed arrows indicate failed packet transmission.

the latest packet of user n is selected for transmission. A scheduled transmission succeeds with probability $p \in (0, 1]$.[1] At the end of each slot, a reliable acknowledgment (ACK) or negative ACK (NACK) is returned to the scheduler, enabling its real-time version tracking at the destination. This model captures key stochastic elements, i.e., random packet arrivals and unreliable wireless transmissions, that jointly determine information staleness in real-time systems.

B. Performance Metrics

To quantify the freshness of information delivered to the destination, we adopt the VAoI as our primary performance metric. Unlike classical AoI, which measures the elapsed time since the generation of the last successfully received update, VAoI tracks semantic version gaps between the scheduler and the destination. The VAoI metric therefore reflects the number of missing updates rather than the elapsed time since the last one. Let $G(t, n)$ denote the version index of user n's newest packet stored at the scheduler at the beginning of slot t. If user n generates a new packet at slot t, the version stored at the scheduler increments as $G(t, n) = G(t - 1, n) + 1$, and remains unchanged otherwise, i.e., $G(t, n) = G(t - 1, n)$.

Let $B(t, n)$ denote the version index of the most recently delivered packet for user n at the destination at the end of slot t. If user n is scheduled at slot t and the transmission succeeds, the version index at the destination is updated to match the one at the scheduler, $B(t, n) = G(t, n)$; otherwise, $B(t, n) = B(t - 1, n)$. Accordingly, the instantaneous VAoI of user n at slot t is defined as

$$\mathcal{D}(t, n) = G(t, n) - B(t - 1, n), \tag{1}$$

which follows the standard VAoI formulation in prior work and quantifies the version discrepancy between the latest update at the source and the information available at the destination [4], [12]. VAoI increases when newer versions become available at the scheduler and is reduced when the latest available version is successfully delivered.

Fig. 2 gives an illustrative example for the VAoI evolution for a single user, highlighting how packet generations, successful transmissions, and transmission failures jointly determine the version gap. In the figure, black arrows indicate packet generations that arrive at the scheduler at the beginning of a time slot. Since each scheduled transmission occupies the entire slot, red arrows represent successful deliveries to the destination at the end of the slot, while gray dashed arrows represent failed transmission attempts.

We observe in Fig. 2 that, unlike the classical AoI, which resets based on elapsed time since the generation of the last successful update, VAoI resets only when the version indices at the scheduler and the destination align. At the beginning of slots 1 and 2, new packets with versions 1 and 2 are generated, increasing the VAoI to 1 and 2, respectively. The packet with version 2 is scheduled and successfully delivered. Since no new packet is generated at the beginning of slot 3, the destination version matches the scheduler in slot 3, resetting the VAoI to 0. In slot 4, a new packet with version 3 is first generated and then scheduled for transmission, but this attempt fails. With no new generation occurring in slot 5, the VAoI remains at 1. In slot 6, a new packet with version 4 is generated, increasing the VAoI to 2. Although this packet is scheduled for transmission, the delivery again fails, so the destination version remains outdated. In slot 7, the version-4 packet is retransmitted successfully, aligning the scheduler and destination versions and resetting the VAoI to 0 at slot 8 (e.g., no new packet is generated at slot 8). This example shows that VAoI captures the semantic mismatch between the freshest available update at the scheduler and the most recently delivered update at the destination.

A commonly used risk-neutral performance measure is the long-term time-averaged average VAoI, which captures typical system freshness but does not explicitly account for rare and severe staleness events. To characterize tail behavior, such as severe version mismatches caused by bursty arrivals or consecutive transmission failures, we adopt the CVaR at system-level CVaR confidence level $\alpha \in (0, 1)$. Specifically, we collect all VAoI samples across users and time

$$\mathcal{Z} = \{\mathcal{D}(t, n) : t = 1, \ldots, T,\ n = 1, \ldots, N\} \tag{2}$$

and construct the empirical distribution with the cumulative distribution function (CDF) $F_{\mathcal{Z}}$. The system CVaR of the mixed empirical distribution is given by

$$\mathrm{CVaR}_\alpha(\mathcal{Z}) = \min_{z \in \mathbb{R}} \left\{ z + \frac{1}{(1-\alpha)TN} \sum_{t=1}^{T} \sum_{n=1}^{N} \left[\mathcal{D}(t, n) - z\right]_+ \right\}, \tag{3}$$

[1] This abstraction is widely used in the AoI and VAoI literature, as it enables clear evaluation of scheduling and decision-making mechanisms without introducing additional physical-layer complexities. The proposed CMDP and DRL framework can be readily extended to more realistic time-varying wireless channels, such as Rayleigh fading or Markov fading channels, due to its model-free nature.

where $[x]_+ = \max\{x, 0\}$ and z denotes the auxiliary Value-at-Risk (VaR) threshold. CVaR quantifies the expected VAoI among the worst $(1-\alpha)$ fraction of user-slot samples, providing a meaningful measure of tail risk for real-time status updates.[2]

In many practical applications, the scheduler must maintain fresh information under a transmission cost constraint. The average transmission cost is defined as

$$\eta = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{I}[a_t \neq 0], \tag{4}$$

where $\mathbb{I}[\cdot]$ is an indicator function that takes the value 1 if the condition inside holds and 0 otherwise. The scheduling policy must ensure $\eta \leq \eta_{\max}$, where $\eta_{\max} \in (0, 1]$ denotes the maximum allowable average transmission cost budget.

Let $\{a_1, \ldots, a_T\}$ denote the sequence of scheduling actions over horizon T. The resulting risk-sensitive scheduling problem is formulated as

$$\begin{aligned} \min_{a_1,\ldots,a_T} \quad & \mathrm{CVaR}_\alpha(\mathcal{Z}) \\ \text{s.t.} \quad & \eta \leq \eta_{\max} \end{aligned} \tag{5}$$

Under the above model and constraints, we develop DRL-based approaches for risk-sensitive VAoI scheduling in Sections IV and V. To facilitate a clear presentation of the proposed framework, we first introduce a constrained D2SAC algorithm, which employs a diffusion-based actor together with conventional scalar critics [18]. D2SAC serves as an intermediate framework that isolates the effect of diffusion-based policy representation and provides a reference point for evaluating the contribution of risk-sensitive learning.

Building upon this foundation, Section V introduces the proposed RS-D3SAC algorithm. Specifically, RS-D3SAC replaces the scalar critics in D2SAC with quantile-based distributional critics to explicitly model the return distribution and enable CVaR-oriented policy optimization. Since CVaR depends on the tail behavior of the return distribution rather than only its expectation, distributional reinforcement learning is essential for principled risk-sensitive control. This progressive construction also allows stable integration of diffusion-based policy generation and distributional value estimation within a unified learning framework.

IV. Diffusion-Based SAC Framework

In this section, we introduce a diffusion-based Soft Actor-Critic framework, termed D2SAC, which serves as the foundation for the proposed risk-sensitive RS-D3SAC algorithm. Specifically, D2SAC employs a diffusion-based actor together with conventional scalar critics to learn scheduling policies under long-term transmission cost constraints. This intermediate framework enables us to isolate the effect of diffusion-based policy representation before incorporating distributional critics for CVaR-oriented risk-sensitive optimization in Section V.

[2]It is worth noting that the adopted CVaR formulation characterizes the tail behavior of the empirical VAoI process rather than trajectory-level risk. Nevertheless, temporal coupling is implicitly captured by the MDP state-transition dynamics, since prolonged staleness accumulation can lead to repeated high-VAoI realizations that contribute to the empirical tail. Developing trajectory-aware risk measures for semantic freshness control is an interesting direction for future research.

A. CMDP Formulation

The VAoI scheduling problem is formulated as a constrained Markov decision process (CMDP) with long-term transmission cost constraints, represented by $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \mathcal{C})$ describing the state space, action space, state transition dynamics, reward function, and cost function, respectively. The system components are defined below.

State Space $\mathcal{S}$: The system state at slot t is the N-dimensional VAoI vector

$$s_t = \big(\mathcal{D}(t,1), \mathcal{D}(t,2), \ldots, \mathcal{D}(t,N)\big) \in \mathcal{S}, \tag{6}$$

where $\mathcal{D}(t,n) \in \{0, 1, \ldots, \mathcal{D}_{\max}\}$ is the VAoI of user n at slot t. To obtain a finite MDP state space and avoid unnecessary analytical and computational complexity, the VAoI is truncated at a sufficiently large value $\mathcal{D}_{\max}$. This truncation is also consistent with practical systems in which information beyond a sufficiently large freshness level may be regarded as effectively obsolete. The truncated VAoI values are used in the state representation, reward computation, and performance evaluation. In our experiments, the testing trajectories of the converged policies remain strictly below $\mathcal{D}_{\max}$, indicating that the truncation boundary does not affect the reported empirical performance or the observed tail behavior.[3]

Action Space $\mathcal{A}$: The action space is defined as a finite discrete set $\mathcal{A} = \{0, 1, \ldots, N\}$, where each element corresponds to a categorical scheduling decision. Specifically, $a_t = 0$ denotes that no user is scheduled for transmission at slot t, while $a_t = n$ for $n \in \{1, \ldots, N\}$ indicates that user n is selected for transmission. Therefore, the action space contains $N+1$ discrete actions in total. As will be presented next, although the action space is discrete, the diffusion model operates in a continuous latent space and generates a latent action representation via iterative denoising. This representation is then mapped to a categorical distribution over discrete actions using a softmax layer, from which the final action is sampled.

State Transition Dynamics $\mathcal{P}$: The state transition probability $p(s_{t+1}|s_t, a_t) \in \mathcal{P}$ characterizes the probabilistic evolution of the system state from s_t to s_{t+1} under action a_t. In each time slot t, user n generates a new status update with probability r_n, resulting in a random packet-generation event that determines the freshest version index $G(t,n)$ available at the scheduler. Meanwhile, if user n is scheduled for transmission at slot t, the update is successfully delivered with probability p, yielding a reception outcome and determining version index $B(t,n)$ at the destination.

As illustrated in Fig. 2, the resulting state transition from s_t to s_{t+1} is fully determined by the pair of stochastic processes $\{G(t,n)\}$ and $\{B(t,n)\}$, which are parameterized by the arrival probabilities r_n and the channel success probability p. In this work, both r_n and p are assumed to be unknown to the scheduler. Consequently, the state transition dynamics cannot be explicitly characterized, and both the risk-neutral and risk-sensitive VAoI scheduling problems are addressed

[3]In our experiments presented in Section VI, $\mathcal{D}_{\max}$ is set to 100. The maximum untruncated VAoI observed for the converged policies is only 33, i.e., no samples reach or exceed $\mathcal{D}_{\max}$.

using model-free reinforcement learning, which learns effective control policies directly through interactions with the environment.[4]

Reward Function $\mathcal{R}$ and Cost Function $\mathcal{C}$: The CMDP is equipped with a reward function $\mathcal{R}: \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, which assigns an immediate reward to guide policy learning

$$r_t = -\sum_{n=1}^{N} \mathcal{D}(t,n) - \lambda c_t, \tag{7}$$

The first term penalizes the system-wide VAoI at slot t, encouraging the policy to maintain information freshness. The second term penalizes the instantaneous transmission cost via the Lagrange multiplier $\lambda \geq 0$. The binary cost variable is defined as $c_t = 1$ if $a_t \neq 0$, and $c_t = 0$ otherwise, so that a transmission cost is incurred only when a packet is scheduled. To enforce $\eta \leq \eta_{\max}$, the dual variable λ is updated using the projected gradient rule [40]

$$\lambda \leftarrow [\lambda + \delta(\eta - \eta_{\max})]_+, \tag{8}$$

where $\delta > 0$ is a step size and η denotes the empirical running average transmission cost. The cumulative return is given by $R = \sum_{t=0}^{T} \gamma^t r_t$, where $\gamma \in (0,1]$ is a discount factor that weights future rewards relative to immediate ones, i.e., we adopt a discounted-return formulation and seek to maximize the expected discounted return under the CMDP framework. This formulation provides a tractable surrogate for the long-term average VAoI minimization problem, while enabling the incorporation of long-term transmission cost constraints.

Although the overall MDP dynamics remain structurally similar to conventional AoI scheduling models, the state variable in VAoI represents semantic version mismatch rather than purely temporal delay. Similar to AoI-based models, a successful transmission resets the corresponding users VAoI to zero, while unsuccessful or deferred transmissions lead to VAoI growth. The key distinction is that VAoI evolves according to the underlying information version process instead of a deterministic linear time increment. Consequently, large VAoI values indicate a large version discrepancy between the source and destination, which makes the tail behavior of the VAoI distribution particularly important and motivates the proposed risk-sensitive optimization framework.

Despite the conceptual clarity of the CMDP formulation, learning an effective scheduling policy remains challenging due to the intricate VAoI evolution and the non-stationarity induced by adaptive dual updates. Although the action space is discrete and low-dimensional, the induced scheduling policy exhibits highly structured state-dependent behavior arising from stochastic packet arrivals, random transmission failures, user competition, and long-term transmission cost constraints. In particular, the scheduler may adopt an aggressive strategy for users with severe or rapidly increasing semantic staleness, while remaining conservative for users whose freshness risk is relatively low. As the VAoI states and resource conditions evolve, these user-specific scheduling priorities can change sharply, resulting in substantial variations in action preferences across neighboring states. Although conventional neural policies can theoretically represent such mappings, learning these rapidly varying scheduling behaviors may lead to unstable policy updates. Therefore, we employ a diffusion-based actor as a complementary mechanism to promote more expressive policy representation through iterative denoising and facilitate stable learning. The resulting D2SAC framework is introduced in the following subsection.

[4]An alternative is to first estimate the unknown system parameters and then apply a model-based scheduler. While viable with an accurate system model, this estimate-then-optimize approach requires explicit model reconstruction. For CVaR-oriented optimization, the objective is determined by the tail of the long-term controlled performance distribution rather than only by first-order system statistics. Parameter estimation alone may not fully capture the distributional effects induced by the controlled dynamics and scheduling policy. In contrast, the proposed end-to-end distributional reinforcement learning framework directly learns the scheduling policy through environment interactions and optimizes the desired risk-sensitive objective.

B. Deep Diffusion Model-Based SAC (D2SAC)

Diffusion models have recently emerged as powerful generative frameworks for modeling complex and multimodal distributions via iterative denoising. Their ability to capture structured dependencies across dimensions makes them well-suited to wireless scheduling problems. Leveraging these advantages, we incorporate a denoising diffusion probabilistic model (DDPM) into the SAC framework to obtain an expressive and stable policy architecture with improved exploration. The resulting diffusion-based policy network and the overall D2SAC structure are illustrated in Figs. 3 and 4, respectively.

1) Policy Generation by the Diffusion Model

A diffusion model comprises two stochastic components: a *forward* process that gradually corrupts a clean sample with Gaussian noise, and a *reverse* process that reconstructs structured samples from pure noise. We next briefly describe the two processes.

Forward Process: As shown in Fig. 3, the forward process constructs a Markov chain that iteratively perturbs an initial clean sample $\boldsymbol{x}_0$ drawn from the unknown true distribution into Gaussian noise through K noise-injection steps

$$\mathrm{q}(\boldsymbol{x}_K|\boldsymbol{x}_0) = \prod_{k=1}^{K} \mathrm{q}(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}), \tag{9}$$

where each transition follows

$$\mathrm{q}(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) = \mathbf{N}\left(\boldsymbol{x}_k; \sqrt{1-\beta_k}\boldsymbol{x}_{k-1}, \beta_k \mathbf{I}\right), \tag{10}$$

where $\mathbf{N}(\cdot)$ represents Gaussian distribution, $\mathbf{I}$ is the identity matrix, and $\beta_k = 1 - e^{-\frac{\beta_{\min}}{K} - \frac{2k-1}{2K^2}(\beta_{\max}-\beta_{\min})}$ denotes the forward process variance governed by a variance preserving scheduler [24]. $\beta_{\min}$ and $\beta_{\max}$ are the minimum and maximum noise levels, respectively. For efficient sampling, the forward process can be expressed in closed form to generate a noisy sample $\boldsymbol{x}_k$ at any arbitrary time step k. Expanding the chain yields a closed-form noisy sample at any step k

$$\mathrm{q}(\boldsymbol{x}_k|\boldsymbol{x}_0) = \mathbf{N}\left(\boldsymbol{x}_k; \sqrt{\bar{\alpha}_k}\boldsymbol{x}_0, (1-\bar{\alpha}_k)\mathbf{I}\right), \tag{11}$$

where $\bar{\alpha}_k = \prod_{i=1}^{k} \alpha_i$ and $\alpha_k = 1 - \beta_k$. Thus, sampling can be performed by

$$\boldsymbol{x}_k = \sqrt{\bar{\alpha}_k}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_k}\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathbf{N}(\mathbf{0}, \mathbf{I}). \tag{12}$$

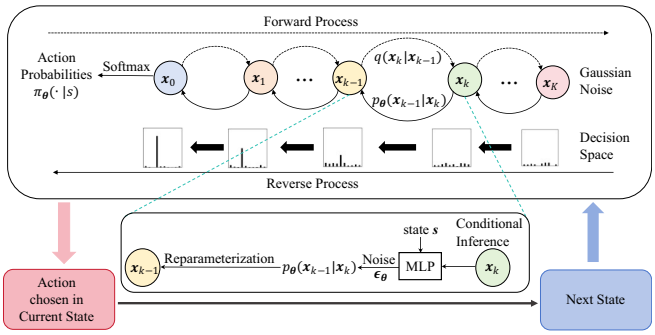


Fig. 3: Schematic overview of the diffusion-based policy network.

In contrast to generative modeling tasks, no dataset of optimal decisions is available in reinforcement learning. Hence, D2SAC does *not* execute the forward process during training; only the reverse process is used to generate policy samples.

Reverse Process: The reverse diffusion process iteratively reconstructs $\boldsymbol{x}_0$ from noise $\boldsymbol{x}_K \sim \mathbf{N}(\mathbf{0}, \mathbf{I})$, as shown in Fig. 3. Our goal is to recover the latent representation $\boldsymbol{x}_0$ of the optimal decision from a noisy sample, where $\boldsymbol{x}_0 = \left[x_0^0, x_0^1, \cdots, x_0^N\right]$, which is later mapped into an action distribution. The generative distribution of $\boldsymbol{x}_0$ can be written as

$$\mathrm{p}_{\boldsymbol{\theta}}(\boldsymbol{x}_0) = \mathrm{p}(\boldsymbol{x}_K) \prod_{k=1}^{K} \mathrm{p}_{\boldsymbol{\theta}}(\boldsymbol{x}_{k-1}|\boldsymbol{x}_k), \tag{13}$$

where $\mathrm{p}(\boldsymbol{x}_K)$ is a Gaussian noise distribution and $\boldsymbol{\theta}$ represents the parameters of the reverse model. The reverse transition is approximated by

$$\mathrm{p}_{\boldsymbol{\theta}}(\boldsymbol{x}_{k-1}|\boldsymbol{x}_k) = \mathbf{N}\left(\boldsymbol{x}_{k-1}; \boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_k, k, s), \tilde{\beta}_k \mathbf{I}\right), \tag{14}$$

with $\widetilde{\beta}_k = \frac{1-\bar{\alpha}_{k-1}}{1-\bar{\alpha}_k}\beta_k$ denoting a deterministically computable variance amplitude. The mean $\boldsymbol{\mu}_{\boldsymbol{\theta}}$ is parameterized by a neural network conditioned on both the diffusion step index k and the state s. Using the forward-process relationship and Bayes' theorem, the reverse mean can be expressed as

$$\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_k, k, s) = \frac{\sqrt{\alpha_k}\,(1-\bar{\alpha}_{k-1})}{1-\bar{\alpha}_k}\boldsymbol{x}_k + \frac{\sqrt{\bar{\alpha}_{k-1}}\beta_k}{1-\bar{\alpha}_k}\boldsymbol{x}_0. \tag{15}$$

The reconstruction of original sample $\boldsymbol{x}_0$ can be obtained according to (12)

$$\boldsymbol{x}_0 = \frac{1}{\sqrt{\bar{\alpha}_k}}\boldsymbol{x}_k - \sqrt{\frac{1}{\bar{\alpha}_k} - 1} \cdot \tanh\left(\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_k, k, s)\right), \tag{16}$$

where $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}$ is the learned denoising noise predictor. The $\tanh(\cdot)$ function here is applied to limit the magnitude of the estimated noise, preventing unstable reconstructions and preserving the structure of the underlying action distribution.

The mean $\boldsymbol{\mu}_{\theta}$ can thus be written based on (15) and (16)

$$\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_k, k, s) = \frac{1}{\sqrt{\alpha_k}}\left(\boldsymbol{x}_k - \frac{\beta_k}{\sqrt{1-\bar{\alpha}_k}}\tanh\left(\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_k, k, s)\right)\right). \tag{17}$$

To allow gradient backpropagation through stochastic sampling, the reparameterization trick is applied [41]

$$\boldsymbol{x}_{k-1} = \boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_k, k, s) + \sqrt{\tilde{\beta}_k}\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathbf{N}(\mathbf{0}, \mathbf{I}). \tag{18}$$

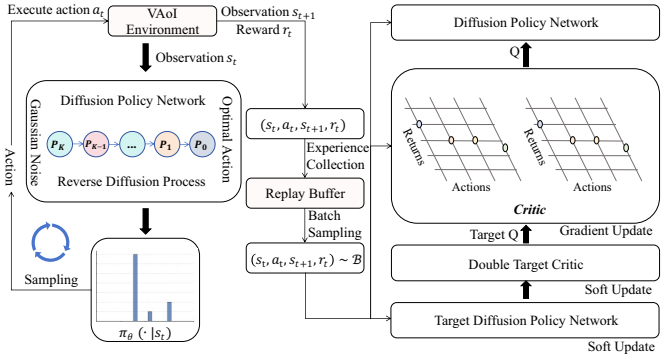


Fig. 4: Architecture of the D2SAC algorithm.

Repeating this denoising process from $k = K$ to 1 produces the clean sample $\boldsymbol{x}_0$, which is then transformed into an action distribution via the softmax function

$$\boldsymbol{\pi}_{\boldsymbol{\theta}}(\cdot|s) = \left\{\frac{e^{x_0^i}}{\sum_{j=0}^{N} e^{x_0^j}}, \forall i \in \mathcal{A}\right\}, \tag{19}$$

Elements in $\boldsymbol{\pi}_{\boldsymbol{\theta}}(\cdot|s)$ correspond to the probabilities of selecting each possible action in state s. In particular, the probability of choosing action a under state s is denoted by $\pi_{\boldsymbol{\theta}}(a|s)$.

2) Integrating Diffusion Model Into the SAC Framework

Soft Actor-Critic (SAC) is a state-of-the-art DRL algorithm grounded in the maximum-entropy RL framework. Its objective is to jointly maximize the expected long-term return and the entropy of the policy. The entropy term encourages the policy to remain sufficiently stochastic, thus promoting robust and persistent exploration in complex environments. Within SAC, the actor network learns a stochastic policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}(\cdot|s_t)$, while two critic networks estimate the state-action value by minimizing the temporal-difference (TD) error. The double-Q structure mitigates value overestimation and improves stability in policy evaluation and optimization.

Diffusion-Based Policy Network: Building upon the SAC framework, D2SAC replaces the conventional neural policy network (implemented by a multi-layer perceptron) with the diffusion-based policy model illustrated in Fig. 3. The global optimization objective remains unchanged, i.e., maximizing the sum of the expected discounted return and the policy entropy. The diffusion model is used to enhance the expressiveness of the policy by generating actions through a multi-step denoising process, enabling the model to capture complex or multimodal state-dependent action preferences that frequently arise in multi-user wireless scheduling.

Formally, the diffusion policy network is trained to determine the optimal parameters $\boldsymbol{\theta}^*$ according to

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t \left(r_t + \psi H(\boldsymbol{\pi}_{\boldsymbol{\theta}}(\cdot|s_t))\right)\right], \tag{20}$$

$$= \arg\min_{\boldsymbol{\theta}} -\mathbb{E}\left[\sum_{t=0}^{T} \gamma^t \left(r_t + \psi H(\boldsymbol{\pi}_{\boldsymbol{\theta}}(\cdot|s_t))\right)\right], \tag{21}$$

where γ is the discount factor and ψ is the temperature coefficient, and $H(\boldsymbol{\pi}_{\boldsymbol{\theta}}(\cdot|s_t)) \triangleq -\boldsymbol{\pi}_{\boldsymbol{\theta}}(\cdot|s_t)^T \log \boldsymbol{\pi}_{\boldsymbol{\theta}}(\cdot|s_t)$ denotes the policy entropy.

As shown in Fig. 4, the actor in D2SAC is realized by the diffusion policy network $\boldsymbol{\pi}_{\boldsymbol{\theta}}$, whose output distribution is generated through a K-step reverse diffusion process. Starting from an initial Gaussian noise vector $\boldsymbol{x}_K \sim \mathbf{N}(\mathbf{0}, \mathbf{I})$, the network iteratively denoises the latent representation to obtain a clean sample $\boldsymbol{x}_0$. This final representation is then transformed into an action distribution via the softmax operator in (19), producing the stochastic policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}(\cdot|s_t)$.

To accommodate this generative policy structure, we denote by a vector-valued Q-function $\boldsymbol{Q}_{\boldsymbol{\phi}}(s_t)$ that includes the Q-values for all actions under state s_t, and by $Q_{\boldsymbol{\phi}}(s_t, a_t)$ the scalar Q function for taking action a_t in state s_t. The actor's update objective is therefore

$$\min_{\boldsymbol{\theta}} \; -\Big[\boldsymbol{\pi}_{\boldsymbol{\theta}}(\cdot|s_t)^T \boldsymbol{Q}_{\boldsymbol{\phi}}(s_t) + \psi H(\boldsymbol{\pi}_{\boldsymbol{\theta}}(\cdot|s_t))\Big], \tag{22}$$

where $\boldsymbol{Q}_{\boldsymbol{\phi}}(s_t) = \min\big(\boldsymbol{Q}_{\boldsymbol{\phi}^1}(s_t), \boldsymbol{Q}_{\boldsymbol{\phi}^2}(s_t)\big)$ is the element-wise minimum of the two critic networks with parameters $\boldsymbol{\phi}^1$ and $\boldsymbol{\phi}^2$ (see the details below). This minimum operation is consistent with SAC's conservative value-estimation strategy and effectively suppresses Q-overestimation.

As indicated in Fig. 4, a replay buffer $\mathcal{M}$ stores transitions (s_t, a_t, s_{t+1}, r_t) collected during agent-environment interaction. Note that the instantaneous cost c_t is used only for online Lagrange multiplier updates, while the replay buffer stores the Lagrangian-shaped reward for policy training. During training, a mini-batch $\mathcal{B}$ of size b transitions is sampled from the replay buffer $\mathcal{M}$. The gradient of the actor loss with respect to $\boldsymbol{\theta}$ is computed as

$$\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) = \mathbb{E}_{s_t \sim \mathcal{B}}\Big[- \nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}(\cdot|s_t)^T \boldsymbol{Q}_{\boldsymbol{\phi}}(s_t) - \psi \nabla_{\boldsymbol{\theta}} H(\boldsymbol{\pi}_{\boldsymbol{\theta}}(\cdot|s_t))\Big]. \tag{23}$$

where $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ denote the policy and Q-function parameters, respectively. The first term drives the policy toward actions with high estimated value, while the second term promotes exploration by encouraging higher entropy (i.e., serving as an action entropy regularizer). The policy parameters are then updated via gradient descent based on (23)

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \omega_a \cdot \nabla_{\boldsymbol{\theta}} \mathcal{L}\big(\boldsymbol{\theta}\big), \tag{24}$$

where ω_a is the actor's learning rate. To stabilize training and avoid oscillations, a slowly-updated target policy network with parameters $\hat{\boldsymbol{\theta}}$ is maintained. Its parameters are updated periodically using a soft update mechanism

$$\hat{\boldsymbol{\theta}} \leftarrow \zeta \boldsymbol{\theta} + (1 - \zeta)\hat{\boldsymbol{\theta}}, \tag{25}$$

where $\zeta \in (0, 1)$ is the soft update coefficient, which controls the stability of the target network.

Double Critic Network: In the D2SAC architecture (Fig. 4), the critic networks aim to estimate the Q-values with high accuracy. To enhance robustness, SAC adopts a double-critic architecture with parameters $\boldsymbol{\phi}^1$ and $\boldsymbol{\phi}^2$. Each critic is updated independently and outputs a vector $\boldsymbol{Q}_{\boldsymbol{\phi}^i}(s_t)$, providing Q-values for all actions in state s_t. Consistent with the actor, each critic has a corresponding target network, with parameters $\hat{\boldsymbol{\phi}}^1$ and $\hat{\boldsymbol{\phi}}^2$.

Algorithm 1 D2SAC: Deep Diffusion Soft Actor-Critic

Initialize: Replay buffer $\mathcal{M}$; total training iterations E; diffusion actor network with parameters $\boldsymbol{\theta}$ and target actor with parameters $\hat{\boldsymbol{\theta}}$; two critic networks with parameters $\boldsymbol{\phi}^1$ and $\boldsymbol{\phi}^2$, and corresponding target critic networks with parameters $\hat{\boldsymbol{\phi}}^1$ and $\hat{\boldsymbol{\phi}}^2$; Lagrange multiplier $\lambda \geq 0$; empirical average cost estimate $\eta = 0$; global step counter $l = 0$.
for training iteration $e = 1$ to E **do**
  Observe current state s_1
  for environment step $t = 1$ to T **do**
    Sample initial diffusion noise $\boldsymbol{x}_K \sim \mathbf{N}(\mathbf{0}, \mathbf{I})$
    for denoising step $k = K$ down to 1 **do**
      Compute scaled denoising prediction $\tanh\big(\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_k, k, s_t)\big)$ using the diffusion actor network
      Compute reverse-process mean $\boldsymbol{\mu}_{\boldsymbol{\theta}}$ using (14)-(17)
      Sample $\boldsymbol{x}_{k-1}$ via reparameterization (18)
    end for
    Obtain action distribution from $\boldsymbol{x}_0$ using (19) and sample action $a_t \sim \boldsymbol{\pi}_{\boldsymbol{\theta}}(\cdot|s_t)$
    Execute a_t, observe next state s_{t+1}, reward r_t, and instantaneous cost c_t
    Store transition (s_t, a_t, s_{t+1}, r_t) in replay buffer $\mathcal{M}$
    $l \leftarrow l + 1$
    Update empirical average cost: $\eta \leftarrow \eta + \frac{1}{l}(c_t - \eta)$
    Set $s_t \leftarrow s_{t+1}$
  end for
  Update Lagrange multiplier λ according to (8)
  Sample minibatch $\mathcal{B} \subset \mathcal{M}$
  Update critic parameters $\boldsymbol{\phi}^1$, $\boldsymbol{\phi}^2$ by minimizing the double-Q loss in (28)
  Update diffusion actor parameters $\boldsymbol{\theta}$ using the policy gradient method (23) and (24)
  Soft-update target networks by (25) and (29)
end for
return Well-trained D2SAC policy $\boldsymbol{\pi}_{\boldsymbol{\theta}^*}$

During policy evaluation, the smaller output of the two Q-networks is used to construct the target, mitigating overestimation bias and improving convergence. The target value is

$$\hat{y} = r_t + \gamma\Big(\boldsymbol{\pi}_{\hat{\boldsymbol{\theta}}}(\cdot|s_{t+1})^T \min_{i=1,2} \boldsymbol{Q}_{\hat{\boldsymbol{\phi}}^i}(s_{t+1}) + \psi H(\boldsymbol{\pi}_{\hat{\boldsymbol{\theta}}}(\cdot|s_{t+1}))\Big), \tag{26}$$

where $\boldsymbol{\pi}_{\hat{\boldsymbol{\theta}}}$ and $Q_{\hat{\boldsymbol{\phi}}^i}$ denote the outputs of the target actor and the two target critic networks. The current Q-value generated by critic i is

$$y^i = Q_{\boldsymbol{\phi}^i}(s_t, a_t) \tag{27}$$

where $Q_{\boldsymbol{\phi}^i}(s_t, a_t)$ denotes the Q-value corresponding to the action a_t produced by the i-th critic $\boldsymbol{Q}_{\boldsymbol{\phi}^i}(s_t)$. The critic parameters are updated by minimizing the TD error between $\hat{y}$ and y^i

$$\min_{\boldsymbol{\phi}^1, \boldsymbol{\phi}^2} \mathbb{E}_{(s_t, a_t, s_{t+1}, r_t) \sim \mathcal{B}}\left[\sum_{i=1,2} \big(\hat{y} - y^i\big)^2\right]. \tag{28}$$

By minimizing the TD error in (28), we can continuously improve the estimation accuracy of the critics, thereby providing more reliable Q-value evaluations for policy improvement. As in the target actor network, the parameters of the target critics are updated using a soft update mechanism

$$\hat{\boldsymbol{\phi}}^i \leftarrow \zeta \boldsymbol{\phi}^i + (1-\zeta)\hat{\boldsymbol{\phi}}^i,\ i=1,2. \tag{29}$$

Algorithm 1 summarizes the overall D2SAC training procedure. In classical stochastic primal-dual algorithms for CMDPs, the primal variables (actor-critic parameters) are typically updated on a faster timescale than the dual variable λ. In Algorithm 1, we adopt a simplified implementation in which λ and the actor-critic parameters are each updated once per training iteration. Importantly, the dual update relies on a running average of accumulated constraint costs, whereas the actor-critic parameters are updated using stochastic minibatch gradients sampled from the replay buffer. Consequently, the dual variable λ evolves more smoothly in practice, resulting in an effective separation of timescales and stable constrained learning behavior. Moreover, although the replay buffer stores transitions collected under different values of λ, the gradual evolution of the dual variable, together with the experience replay mechanism and slowly updated target networks, mitigates the effect of the non-stationary reward signal induced by adaptive dual updates, thereby improving the stability of off-policy learning. Building upon D2SAC, we next develop a distributional extension that models the full return distribution and enables CVaR-oriented optimization for risk-sensitive VAoI scheduling.

V. Risk-Sensitive VAoI Scheduling with Deep Distributional Diffusion Model-Based SAC

This section develops a risk-sensitive extension of D2SAC tailored for multi-user VAoI scheduling. Unlike D2SAC, which remains risk-neutral because the critic estimates only the expected return, we present the RS-D3SAC framework, which integrates a distributional critic to enable explicit CVaR evaluation and tail-aware updates. Directly optimizing the CVaR of the stationary system-wide VAoI process is generally intractable in model-free settings because the stationary distribution induced by a policy is unknown and difficult to characterize. RS-D3SAC adopts a CVaR-oriented discounted-return formulation and provides a tractable risk-sensitive surrogate to mitigate extreme semantic staleness in VAoI.

A. D3SAC: Distributional D2SAC

In this work, we retain the CMDP structure of the risk-sensitive formulation in (5), including the state and action spaces, transition dynamics, reward function, and transmission cost, as defined in Section IV. This allows us to isolate the effect of explicitly incorporating tail risk while preserving the underlying VAoI scheduling problem. Specifically, rather than introducing a different instantaneous reward for risk-sensitive control, we retain the original reward formulation and modify the learning objective to account for the tail behavior of future performance.

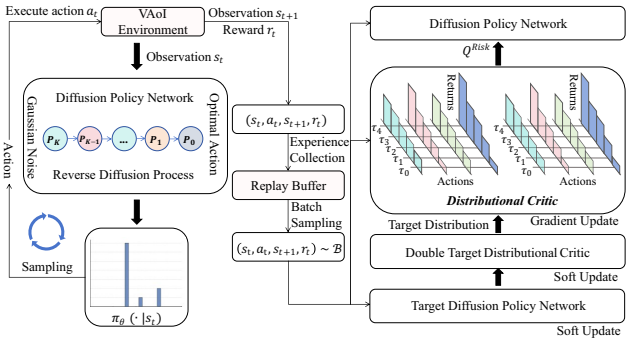


Fig. 5: Architecture of the proposed RS-D3SAC algorithm.

Recall that in D2SAC, each critic outputs a vector-valued Q-function $\boldsymbol{Q}_{\boldsymbol{\phi}}(s_t)$ (we omit the index of the critic to improve clarity), where each scalar component $Q_{\boldsymbol{\phi}}(s_t, a_t)$ characterizes only the expected discounted return associated with action a_t in state s_t. Such a representation is sufficient for risk-neutral optimization, but it discards information about the distribution of future outcomes. Consequently, two actions may have similar expected returns while exhibiting substantially different risks of unfavorable future VAoI evolution.

To address this limitation, D3SAC replaces the scalar critic with a distributional quantile critic capable of modeling the entire return distribution, while the actor continues to employ the diffusion-based generative model. The additional distributional information enables the policy to distinguish between actions that have similar average performance but different risks of severe unfavorable outcomes. More specifically, instead of estimating the scalar value $Q_{\boldsymbol{\phi}}(s_t, a_t)$, D3SAC models the return as a random variable, $Z_{\boldsymbol{\xi}}(s_t, a_t)$, where $\boldsymbol{\xi}$ denotes the parameters of the distributional critic. The conventional Q-value is recovered as the mean of the learned distribution, $Q_{\boldsymbol{\xi}}(s_t, a_t) = \mathbb{E}[Z_{\boldsymbol{\xi}}(s_t, a_t)]$. By learning the full distribution, the policy becomes sensitive to rare but severe VAoI spikes, which often dominate performance in real-time or safety-critical applications.

The distributional critic satisfies the distributional Bellman equation [14]

$$Z_{\boldsymbol{\xi}}(s_t, a_t) \overset{\mathrm{D}}{=} r_t + \gamma\Big(Z_{\boldsymbol{\xi}}(s_{t+1}, a') - \psi \log \pi_{\hat{\boldsymbol{\theta}}}(a'|s_{t+1})\Big), \tag{30}$$

where $\overset{\mathrm{D}}{=}$ denotes equality in distribution, and $Z_{\boldsymbol{\xi}}(s_{t+1}, a')$ denotes the distribution of future returns under action a' in the next state s_{t+1}, $a' \sim \boldsymbol{\pi}_{\hat{\boldsymbol{\theta}}}(\cdot|s_{t+1})$. Directly modeling $Z_{\boldsymbol{\xi}}(s_t, a_t)$, rather than its expectation, enables the algorithm to penalize low-quantile returns associated with large VAoI realizations.

To approximate the return distribution, D3SAC adopts quantile-regression distributional critics as in QR-DQN. It approximates the CDF of the return by learning a set of quantiles. Let the cumulative probability of $Z_{\boldsymbol{\xi}}(s_t, a_t)$ be divided into $\mathcal{N}_q$ uniform probability intervals $\{\tau_1, \tau_2, ..., \tau_j, ..., \tau_{\mathcal{N}_q}\}$, where $\tau_j = j/\mathcal{N}_q$. For each interval (τ_{j-1}, τ_j), QR-DQN defines the midpoint $\hat{\tau}_j = (\tau_{j-1} + \tau_j)/2,\ j = \{1, 2, ..., \mathcal{N}_q\}$. The critic then learns the corresponding quantile values $\sigma_j(s_t, a_t; \boldsymbol{\xi})$, representing the $\hat{\tau}_j$-quantile of the return distribution. Under this

representation, the distribution $Z_{\boldsymbol{\xi}}(s_t, a_t)$ is approximated by a uniform mixture of Dirac masses at these learned quantiles

$$Z_{\boldsymbol{\xi}}(s_t, a_t) = \frac{1}{\mathcal{N}_q} \sum_{j=1}^{\mathcal{N}_q} \delta_{\sigma_j(s_t, a_t; \boldsymbol{\xi})}, \tag{31}$$

where δ_z denotes a Dirac distribution centered at z. This representation provides a flexible approximation suitable for modeling the skewed distributions with pronounced upper-tail behavior, often observed in semantic freshness metrics such as VAoI.

As in D2SAC, D3SAC employs a double-critic architecture to improve stability and reduce overestimation bias. Two independent distributional critics, parameterized by $\boldsymbol{\xi}^1$ and $\boldsymbol{\xi}^2$, generate independent estimates $Z_{\boldsymbol{\xi}^1}(s_t, a_t)$ and $Z_{\boldsymbol{\xi}^2}(s_t, a_t)$. Corresponding target networks are denoted by $\hat{\boldsymbol{\xi}^1}$ and $\hat{\boldsymbol{\xi}^2}$.

Given the next state s_{t+1}, the target action a' is obtained using the target diffusion-based actor network $\boldsymbol{\pi}_{\hat{\boldsymbol{\theta}}}(\cdot|s_{t+1})$ (as described in Section IV). The two target critics then produce the next-state return distributions $Z_{\hat{\boldsymbol{\xi}^1}}(s_{t+1}, a')$ and $Z_{\hat{\boldsymbol{\xi}^2}}(s_{t+1}, a')$. Following the double-critic principle, the target distribution, denoted by $Z_{\hat{\boldsymbol{\xi}}}(s_{t+1}, a')$, is obtained by taking the element-wise minimum of the quantile outputs

$$\begin{aligned} Z_{\hat{\boldsymbol{\xi}}}(s_{t+1}, a') &= \frac{1}{\mathcal{N}_q} \sum_{j=1}^{\mathcal{N}_q} \delta_{\sigma_j(s_{t+1}, a'; \hat{\boldsymbol{\xi}})} \\ &= \frac{1}{\mathcal{N}_q} \sum_{j=1}^{\mathcal{N}_q} \delta_{\min\{\sigma_j(s_{t+1}, a'; \hat{\boldsymbol{\xi}^1}), \sigma_j(s_{t+1}, a'; \hat{\boldsymbol{\xi}^2})\}} \end{aligned} \tag{32}$$

This element-wise minimum is adopted as a heuristic clipped-quantile approximation inherited from the double-critic principle; it is not intended to represent the exact distribution of the minimum [36].[5]

Each distributional critic is trained by minimizing the quantile regression Huber loss across all quantile pairs

$$\mathcal{L}(\boldsymbol{\xi}^i) = \frac{1}{\mathcal{N}_q} \sum_{j=1}^{\mathcal{N}_q} \sum_{j'=1}^{\mathcal{N}_q} \rho_{\hat{\tau}_j}^{\kappa}\left(\Delta_{jj'}^i\right), \quad i = 1, 2, \tag{33}$$

where $\Delta_{jj'}^i$ is the TD error corresponding to the quantile estimates, given by

$$\begin{aligned} \Delta_{jj'}^i = r_t + \gamma\Big(\sigma_{j'}(s_{t+1}, a'; \hat{\boldsymbol{\xi}}) - \psi \log \pi_{\hat{\boldsymbol{\theta}}}(a'|s_{t+1})\Big) \\ - \sigma_j(s_t, a_t; \boldsymbol{\xi}^i), \quad i = 1, 2 \end{aligned} \tag{34}$$

where $a' \sim \boldsymbol{\pi}_{\hat{\boldsymbol{\theta}}}(\cdot|s_{t+1})$ and $\sigma_j(\cdot)$ denotes the j-th quantile of the return distribution. The quantile Huber loss $\rho_\tau^\kappa(\cdot)$ is defined by

$$\rho_\tau^\kappa(u) = \begin{cases} |\tau - \mathbb{I}\{u < 0\}| \cdot \frac{1}{2}u^2, & |u| \le \kappa, \\ |\tau - \mathbb{I}\{u < 0\}| \cdot \kappa(|u| - \frac{1}{2}\kappa), & \text{otherwise}, \end{cases} \tag{35}$$

where $\mathbb{I}\{\cdot\}$ is the indicator function and κ denotes the Huber loss threshold between quadratic and linear loss regimes.

The total loss used for training is the sum of the two distributional critic losses

$$\mathcal{L}_{total} = \mathcal{L}(\boldsymbol{\xi}^1) + \mathcal{L}(\boldsymbol{\xi}^2). \tag{36}$$

By minimizing this objective, the two distributional critics provide a clipped approximation of the return distribution used for risk-sensitive value estimation, enabling D3SAC to capture both central tendencies and tail risks. After each update, the target distributional critic parameters are softly updated according to

$$\hat{\boldsymbol{\xi}^i} \leftarrow \zeta \boldsymbol{\xi}^i + (1 - \zeta)\hat{\boldsymbol{\xi}^i}, \ i = 1, 2. \tag{37}$$

B. RS-D3SAC for Risk-Sensitive VAoI Scheduling

We now present risk-sensitive D3SAC (RS-D3SAC) for CVaR-based VAoI scheduling. Directly optimizing the CVaR of the stationary system-wide mixed VAoI distribution expressed in (3) is generally intractable, since it requires explicit characterization of the stationary state distribution induced by a policy. Moreover, the VAoI process exhibits long-term temporal dependencies and cross-user coupling, making the stationary distribution difficult to analyze. To address this challenge, we adopt a discounted-return formulation and optimize the CVaR of the cumulative return, where the one-step reward is defined based on the negative VAoI and transmission cost penalty. Under this formulation, trajectories associated with persistently large VAoI values correspond to the lower tail of the return distribution. Consequently, maximizing the CVaR of the discounted return serves as a principled surrogate for mitigating extreme semantic staleness events.[6] This formulation is also consistent with standard approaches in risk-sensitive and distributional reinforcement learning [34], [37].

To obtain a tractable surrogate, we leverage the learned return distribution $Z_{\boldsymbol{\xi}}$ to reduce risk. For a return-level lower-tail probability $\varphi \in [0, 1]$, the VaR and CVaR of the return distribution $Z_{\boldsymbol{\xi}}$ are defined as

$$\mathrm{VaR}_\varphi[Z_{\boldsymbol{\xi}}] = \inf\{z \in \mathbb{R} : F_{Z_{\boldsymbol{\xi}}}(z) \ge \varphi\}, \tag{38}$$

$$\mathrm{CVaR}_\varphi[Z_{\boldsymbol{\xi}}] = \mathbb{E}\left[z \mid z \le \mathrm{VaR}_\varphi[Z_{\boldsymbol{\xi}}]\right], \tag{39}$$

where $F_Z(z)$ is the CDF of the return distribution. With distributional RL, the CVaR is readily computed by averaging quantile samples with $\hat{\tau}_j \le \varphi$. We therefore optimize $\mathrm{CVaR}_\varphi[Z_{\boldsymbol{\xi}}]$ as a practical risk-sensitive surrogate.

Since the one-step reward in (7) equals the negative VAoI minus transmission cost, the lower tail of the return distribution corresponds to the upper tail of the VAoI distribution. Consequently, maximizing the CVaR of the return effectively

[5]Quantile crossing is measured as the fraction of evaluated state-action samples for which at least one pair of adjacent quantile estimates violates the nondecreasing order. By this measure, quantile crossing occurs in 1.09% of evaluated samples during training, primarily in the early stages, and decreases to 0.02% for converged models. Thus, quantile crossing is rare after convergence, although monotonic ordering is not strictly guaranteed.

[6]We note that establishing a precise analytical relationship between the CVaR of the discounted return and the CVaR of the stationary VAoI distribution remains challenging due to the time-dependent and stochastic nature of the scheduling process. Therefore, the proposed approach should be interpreted as a principled and practically tractable surrogate for tail-risk control in VAoI scheduling.

Algorithm 2 RS-D3SAC: Risk-Sensitive Deep Distributional Diffusion Soft Actor-Critic

1: **Initialize:** Replay buffer $\mathcal{M}$; total training iterations E; diffusion actor network with parameters $\boldsymbol{\theta}$ and target actor with parameters $\hat{\boldsymbol{\theta}}$; two distributional critic networks with parameters $\boldsymbol{\xi}^1$ and $\boldsymbol{\xi}^2$, and corresponding target critic networks with parameters $\hat{\boldsymbol{\xi}^1}$ and $\hat{\boldsymbol{\xi}^2}$; Lagrange multiplier $\lambda \geq 0$; empirical average cost estimate $\eta = 0$; global step counter $l = 0$; number of quantile samples $\mathcal{N}_q$; system-level CVaR confidence level α; return-level lower-tail probability $\varphi = 1 - \alpha$.
2: **for** training iteration $e = 1$ to E **do**
3: Observe current state s_1
4: **for** environment step $t = 1$ to T **do**
5: Sample initial diffusion noise $\boldsymbol{x}_K \sim \mathbf{N}(\mathbf{0}, \mathbf{I})$
6: **for** denoising step $k = K$ down to 1 **do**
7: Compute scaled denoising prediction $\tanh\big(\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_k, k, s_t)\big)$ using the diffusion actor network
8: Compute reverse-process mean $\boldsymbol{\mu}_{\boldsymbol{\theta}}$ using (14)-(17)
9: Sample $\boldsymbol{x}_{k-1}$ via reparameterization (18)
10: **end for**
11: Obtain action distribution from $\boldsymbol{x}_0$ using (19) and sample action $a_t \sim \boldsymbol{\pi}_{\boldsymbol{\theta}}(\cdot|s_t)$
12: Execute a_t, observe next state s_{t+1}, reward r_t, and instantaneous cost c_t
13: Store transition (s_t, a_t, s_{t+1}, r_t) in replay buffer $\mathcal{M}$
14: $l \leftarrow l + 1$
15: Update empirical average cost: $\eta \leftarrow \eta + \frac{1}{l}(c_t - \eta)$
16: Set $s_t \leftarrow s_{t+1}$
17: **end for**
18: Update Lagrange multiplier λ according to (8)
19: Sample minibatch $\mathcal{B} \subset \mathcal{M}$
20: Update distributional critics $\boldsymbol{\xi}^1, \boldsymbol{\xi}^2$ by minimizing the quantile Huber loss (36)
21: Compute the risk-sensitive vector-valued Q-function $\boldsymbol{Q}_{\boldsymbol{\xi}}^{\text{Risk}}$, where each CVaR-based Q-value $Q_{\boldsymbol{\xi}}^{\text{Risk}}$ is obtained by (40)
22: Update diffusion actor parameters $\boldsymbol{\theta}$ using the policy gradient method (23) and (24) by replacing $\boldsymbol{Q}_\phi$ with $\boldsymbol{Q}_{\boldsymbol{\xi}}^{\text{Risk}}$
23: Soft-update target networks using (25) and (37)
24: **end for**
25: **return** Well-trained RS-D3SAC policy $\boldsymbol{\pi}_{\boldsymbol{\theta}^*}$

reduces high-VAoI outcomes, encouraging the policy to mitigate unfavorable VAoI realizations. Formally, we define the risk-sensitive Q-value as

$$Q_{\boldsymbol{\xi}}^{\text{Risk}}(s_t, a_t) = \text{CVaR}_\varphi[Z_{\boldsymbol{\xi}}(s_t, a_t)]. \tag{40}$$

Given this definition, let us denote the risk-sensitive vector-valued Q-function $\boldsymbol{Q}_{\boldsymbol{\xi}}^{\text{Risk}}(s_t)$ that includes the risk-sensitive Q-values for all actions under state s_t. The risk-sensitive actor update objective becomes

$$\min_{\boldsymbol{\theta}} \; -\Big[\boldsymbol{\pi}_{\boldsymbol{\theta}}(\cdot|s_t)^T \boldsymbol{Q}_{\boldsymbol{\xi}}^{\text{Risk}}(s_t) + \psi H(\boldsymbol{\pi}_{\boldsymbol{\theta}}(\cdot|s_t))\Big], \tag{41}$$

while the remaining steps of the actor training pipeline remain identical to those in Section IV, except that $\boldsymbol{Q}_{\boldsymbol{\phi}}$ is replaced by $\boldsymbol{Q}_{\boldsymbol{\xi}}^{\text{Risk}}$. The formulation of RS-D3SAC shifts the policy from expectation-oriented to tail-risk-oriented, directing learning to mitigate extreme staleness events.

In practice, we set $\varphi = 1 - \alpha$ to remain consistent with the definition of system CVaR of VAoI at confidence level α, i.e., $\text{CVaR}_\alpha(\mathcal{Z})$. Algorithm 2 summarizes the complete RS-D3SAC procedure. Compared to D2SAC in Algorithm 1, the main differences in Algorithm 2 are

(1) Initialization of the number of quantiles $\mathcal{N}_q$ and the CVaR confidence level α (Line 1).
(2) Replacement of D2SAC's scalar critics with double distributional critics trained using the quantile Huber loss (Line 20).
(3) Computation of CVaR-based risk-sensitive Q-values to guide policy improvement (Lines 21 and 22), allowing the actor to prioritize severe tail VAoI outcomes.

Compared with D2SAC, RS-D3SAC introduces additional computational overhead only in the critic update during training. Specifically, with $\mathcal{N}_q$ quantile outputs, the forward and backward passes of the critic scale linearly with $\mathcal{N}_q$, while the quantile Huber loss incurs an additional $\mathcal{O}({\mathcal{N}_q}^2)$ cost due to pairwise quantile comparisons. CVaR computation requires $O(N_q)$ operations when the learned quantiles are maintained in ordered form. In practice, $\mathcal{N}_q$ is chosen to be moderate (e.g., $\mathcal{N}_q = 32$ or 64), and these costs are negligible compared to the diffusion-based actor which dominates the overall runtime through K denoising steps. Therefore, RS-D3SAC achieves improved tail-risk control with only a modest increase in computational complexity. As demonstrated next, RS-D3SAC significantly improves the CVaR of VAoI performance, enabling the system to maintain semantic freshness effectively.

VI. Performance Evaluation

This section evaluates the proposed RS-D3SAC algorithm in terms of CVaR of VAoI. For completeness, we also report average VAoI performance in selected experiments. Extensive simulation results demonstrate that RS-D3SAC consistently outperforms existing DRL baselines. In addition, we conduct ablation studies to isolate the contributions of the diffusion-based actor and the distributional critic, clarifying how each component affects performance under the CVaR metric.

A. Simulation Setup

All implementations and simulations are carried out using PyTorch and executed on a high-performance computing server equipped with an Intel Xeon Gold 6226R CPU, an NVIDIA GeForce RTX 4090 GPU, and 512 GB of RAM, running Ubuntu Linux. The detailed configurations of the actor, scalar critic, and distributional critic networks for D2SAC and RS-D3SAC are summarized in Table III.

In our algorithms, the actor is implemented as a diffusion-based generative policy network. The actor network implements a time-conditioned denoising policy that maps Gaussian noise and the current system state to a denoised action distribution. At each diffusion step, a sinusoidal positional

TABLE III: Network Architectures of the Actor, Scalar Critic, and Distributional Critic with N users

Networks	Layer	Activation	Units
Actor	SinusoidalPosEmb	–	16
	FullyConnect	Mish	32
	FullyConnect	–	16
	Concatenation	–	–
	FullyConnect	Mish	256
	FullyConnect	Mish	256
	FullyConnect	Tanh	$N+1$
Scalar Critic	FullyConnect	Mish	256
	FullyConnect	Mish	256
	FullyConnect	–	$N+1$
Distributional Critic	FullyConnect	Mish	256
	FullyConnect	Mish	256
	FullyConnect	–	$(N+1)\times\mathcal{N}_q$

TABLE IV: Training Parameters.

Symbol	Description	Value
ω_a	Learning rate of the actor network	2×10^{-4}
ω_c	Learning rate of the critic network	2×10^{-3}
ψ	Temperature of action entropy regularization	0.05
ζ	Soft update coefficient	0.005
b	Batch size of $\mathcal{B}$	512
γ	Discount factor	0.95
K	Number of diffusion denoising steps	5
$\lvert\mathcal{M}\rvert$	Replay buffer capacity	5×10^{6}
E	Total training iterations	5000
T	Transitions per training iteration	100
κ	Huber loss threshold	1
δ	Lagrange update step size	5
$\mathcal{N}_q$	Number of quantiles in distributional critic	64
α	CVaR confidence level ($\varphi=1-\alpha$ by default)	0.75
$\mathcal{D}_{\max}$	Maximum truncated VAoI	100

embedding encodes the denoising timestep. It is integrated into the actor network to condition its computation on the current stage of the diffusion process. The actor backbone consists of three fully connected layers with Mish activations, followed by a final layer with a Tanh activation to constrain the outputs within a bounded range. During policy execution, the same time-conditioned actor network is applied iteratively for K denoising steps, with each forward pass corresponding to one denoising operation that progressively refines the action.

In the double-critic architecture, each critic consists of three fully connected layers with Mish activations followed by a scalar output layer that estimates the state-action value. In RS-D3SAC, the distributional critics share the same backbone architecture but replace the scalar output with multiple quantile outputs per action, following the QR-DQN framework. This distributional representation captures return uncertainty and tail behavior, enabling the computation of CVaR-based risk-sensitive Q-values to guide policy optimization.

All actor, scalar critic, and distributional critic networks are trained using the Adam optimizer. The learning rates are set to $\omega_a = 2\times10^{-4}$ for the actor network and $\omega_c = 2\times10^{-3}$ for both scalar and distributional critics. Target networks share the same architectures as their corresponding online networks and are updated via soft updates with coefficient $\zeta = 0.005$. A complete list of training parameters is provided in Table IV.

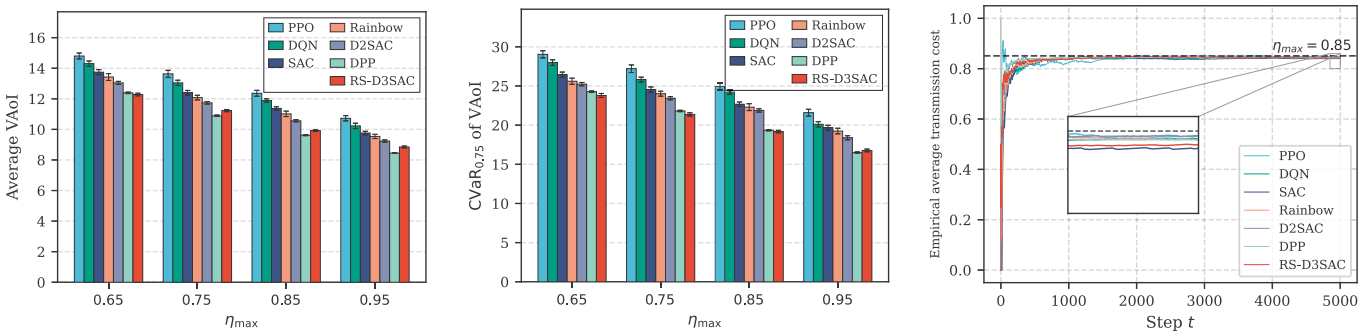
For performance comparison, we consider the following representative DRL baseline methods:

- **Proximal Policy Optimization (PPO)** [42]: A widely adopted policy-gradient-based algorithm that learns a stochastic policy for action selection. PPO employs a clipped surrogate objective to constrain policy updates to improve training stability and sample efficiency.
- **Deep Q Network (DQN)**: A canonical value-based DRL algorithm that approximates the action-value function using deep neural networks. At each decision epoch, DQN selects actions by maximizing the estimated Q-values corresponding to the current system state.
- **Rainbow** [43]: An extension of DQN that combines multiple algorithmic improvements, including dueling and double Q-learning, prioritized experience replay, and multi-step bootstrapping, to improve learning stability and performance. Notably, Rainbow incorporates the C51 distributional RL component. C51 models the return distribution as a categorical distribution over a fixed set of discrete support atoms and estimates only the probability mass for each atom, thus limiting its ability to accurately capture distributions with substantial tail mass.
- **Soft Actor-Critic (SAC)**: The baseline algorithm without the diffusion-based actor or the distributional critic.

All DRL baselines considered in this work are implemented within a Lagrangian primal-dual framework to handle the long-term transmission cost constraint. Specifically, a dynamically updated Lagrange multiplier is incorporated into the reward function to penalize constraint violations. Therefore, these baselines correspond to constrained variants rather than their standard unconstrained implementations.

In addition to DRL methods, we incorporate a classical model-based scheduling baseline based on the Drift-Plus-Penalty (DPP) framework. DPP is a well-established Lyapunov optimization approach for constrained stochastic control and has been widely applied to AoI/VAoI-related scheduling problems. As presented in [19], DPP transforms the constrained VAoI scheduling problem into a queue stability problem, where stabilizing the queue approximately enforces the long-term transmission cost constraint. The implementation of DPP assumes knowledge of system statistics, including packet arrival rates and channel success probabilities. That is, unlike DRL-based methods, DPP exploits analytical system structure and does not rely on value function approximation or data-driven learning. It is important to note that DPP is primarily designed for average-oriented optimization and does not directly account for tail-risk metrics such as CVaR. Nevertheless, for completeness, we also evaluate the CVaR performance achieved by the DPP policy to assess its robustness under rare but severe semantic staleness events. Overall, the considered baselines collectively represent two major paradigms for constrained scheduling under long-term resource constraints: model-free constrained DRL methods and model-based Lyapunov optimization approaches.

Unless otherwise stated, all evaluations are conducted under a unified system configuration to ensure fair and reproducible comparisons. The system consists of $N = 20$ users. Each user generates packets according to a Bernoulli process with rate $r_n = 0.75$, and each transmission succeeds independently with probability $p = 0.9$. The cost for each transmission is $c_t = 1$, subject to a long-term average transmission cost constraint $\eta_{\max} = 0.85$. During training, policy checkpoints



(a) (b) (c)

Fig. 6: Performance comparison and transmission cost constraint verification for different scheduling algorithms. (a) System average VAoI under different transmission cost constraints. (b) System $\mathrm{CVaR}_{0.75}$ of VAoI under different transmission cost constraints. (c) Empirical average transmission cost during testing under $\eta_{\max} = 0.85$.

are saved every 1000 environment steps, i.e., 500 checkpoints in total. At each checkpoint, the corresponding policy is fixed and evaluated in the testing environment. All reported results are averaged over 5 independent training runs with different random seeds, with each checkpoint evaluated using 10 independent test rollouts of 5000 time slots (i.e., 50 test rollouts in total). Except Fig. 11 below, the results are obtained at the 500-th checkpoint. The error bars and shaded regions denote 95% confidence intervals. The resulting VAoI trajectory is recorded and used to compute the empirical CVaR metrics.

B. Simulation Results

1) Benchmark Evaluation of D2SAC and RS-D3SAC

We first evaluate the performance of D2SAC and RS-D3SAC against the four representative DRL baselines and the DPP policy under identical training and testing conditions. Figs. 6(a) and 6(b) report the resulting average VAoI and $\mathrm{CVaR}_{0.75}$ of VAoI, respectively, under different long-term transmission cost constraints $\eta_{\max}$. As $\eta_{\max}$ increases, i.e., as the transmission constraint is relaxed, the average VAoI and the $\mathrm{CVaR}_{0.75}$ of VAoI generally decrease for all algorithms due to the increased availability of transmission opportunities.

As shown in Fig. 6(a), D2SAC achieves lower average VAoI than standard SAC and other conventional DRL approaches. This result suggests that the diffusion-based policy representation can improve policy learning in the considered scheduling problem, possibly due to smoother policy updates and improved exploration quality, despite the low-dimensional discrete action space.

Although RS-D3SAC is primarily designed to mitigate tail risk rather than optimize the mean performance, its average VAoI is also reported in Fig. 6(a). As observed, RS-D3SAC achieves a slightly lower average VAoI than D2SAC, indicating that introducing risk sensitivity does not come at the expense of mean performance. Instead, suppressing extreme VAoI realizations can also yield modest improvements in average performance. Moreover, the average VAoI achieved by RS-D3SAC closely approaches that of the model-based DPP benchmark, with only a marginal performance gap when $\eta_{\max}$ becomes relatively large. This demonstrates that the proposed DRL framework can effectively learn competitive average-oriented scheduling performance without requiring prior knowledge of system statistics.

As shown in Fig. 6(b), RS-D3SAC achieves a clear and consistent advantage in suppressing extreme VAoI values across different transmission cost constraints, compared with all baseline algorithms. The comparison between average VAoI and CVaR reveals an important distinction between average-oriented and risk-sensitive scheduling. In particular, DPP exploits the known system statistics to optimize average freshness performance and therefore can achieve competitive, or even lower, average VAoI. However, its advantage in average performance does not necessarily translate into superior tail behavior. In contrast, RS-D3SAC explicitly incorporates the lower-tail information of the return distribution into policy optimization and consequently achieves substantially lower CVaR. This difference highlights that minimizing typical freshness degradation and suppressing severe unfavorable freshness events are not equivalent scheduling objectives.

Notably, the performance improvement of RS-D3SAC over D2SAC is more significant for CVaR than for average VAoI. This highlights the critical role of the distributional critic in risk-sensitive scheduling. By explicitly learning multiple quantiles of the return distribution, the distributional critic captures not only the central tendency but also the tail behavior of return distribution induced by VAoI-related rewards. This enables the policy optimization process to account for extreme adverse events and avoid high-risk scheduling actions, thereby achieving substantially improved tail-risk control. D2SAC and the other baseline DRL methods focus only on the mean of the return distribution. As a result, they lack explicit risk-sensitive awareness of tail behavior and are inherently limited in their ability to suppress extreme VAoI outcomes under stochastic packet arrivals and transmission uncertainty.

Fig. 6(c) illustrates the evolution of the empirical average transmission cost η during the testing phase for all evaluated algorithms when $\mathrm{CVaR}_{0.75}$ of VAoI is considered, where the black dashed line denotes the prescribed upper bound $\eta_{\max} = 0.85$. Each curve represents the average empirical transmission cost over the 50 test rollouts. As shown, the averaged transmission-cost curves remain below the prescribed threshold throughout the testing period, indicating that all algorithms satisfy the constraint on average. Across individual rollouts, two rollouts (4%) exceed the prescribed threshold. The mean empirical transmission cost across all rollouts is 0.842, and the maximum is 0.853, corresponding to a small

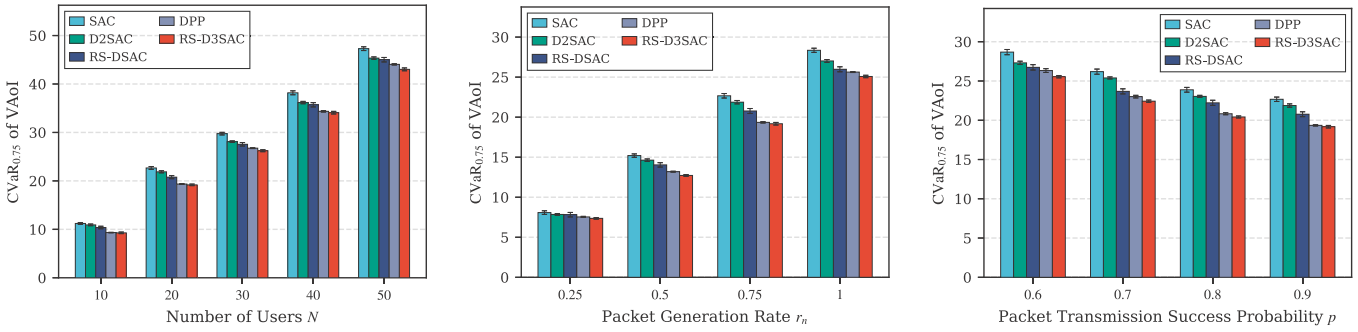


Fig. 7: Ablation study of the system-wide $\mathrm{CVaR}_{0.75}$ of VAoI performance under different system conditions. The results compare SAC, D2SAC, RS-DSAC, and RS-D3SAC to highlight the role of diffusion-based policy generation and distributional value modeling in tail-risk control. Subfigures illustrate the effect of (a) number of users, (b) packet generation rate, and (c) packet transmission success probability on tail-risk performance.

excess of 0.003 above the threshold. These results show that the average transmission cost remains within the prescribed budget, although a small fraction of rollouts exceed the threshold due to finite-horizon evaluations. Additional training-phase experiments, including the evolution of the episode return, λ, and the running transmission cost, empirically confirm that the proposed procedure exhibits stable learning behavior and gradual convergence of the constraint-related quantities (omitted due to page limit). Overall, RS-D3SAC achieves the tail-risk reduction reported in Fig. 6(b) without requiring additional resource consumption beyond the prescribed budget: the improved $\mathrm{CVaR}_{0.75}$ performance is attributed to more effective risk-sensitive optimization of the return distribution within the feasible policy set.

2) Component-wise Ablation Analysis

To isolate and quantify the individual contributions of the diffusion-based actor and the distributional critic, we report in Fig. 7 the $\mathrm{CVaR}_{0.75}$ performance of four algorithmic variants: (i) SAC, serving as the baseline actor-critic method; (ii) D2SAC, which augments SAC with a diffusion-based actor; (iii) RS-DSAC, a risk-sensitive SAC equipped with a quantile-regression distributional critic to compute risk-sensitive Q-values, but without the diffusion-based actor; and (iv) RS-D3SAC, which integrates both the diffusion-based actor and the risk-sensitive distributional critic. We evaluate these algorithms along three system dimensions: (a) number of users, (b) packet generation rate, and (c) packet transmission success probability. We also plot the results of DPP in Fig. 7. It shows that since DPP does not explicitly optimize tail risk, it exhibits inferior CVaR performance compared with RS-D3SAC. In contrast, RS-D3SAC effectively reduces extreme VAoI events by modeling the return distribution and performing CVaR-oriented policy optimization.

Fig. 7 shows that in most configurations, RS-DSAC achieves substantially larger improvements over SAC than D2SAC does, clearly demonstrating the importance of modeling the return distribution for risk-sensitive optimization. Since $\mathrm{CVaR}_{0.75}$ targets the worst 25% of outcomes, it cannot be optimized through expected returns alone. Despite its expressive diffusion-based actor, D2SAC remains mean-oriented and insensitive to tail behavior. In contrast, the distributional critic in RS-DSAC models multiple quantiles of the return distribution, enabling the policy to identify and suppress extreme VAoI events. Fig. 7 thus demonstrates the practical benefit of explicitly modeling the return distribution for CVaR-oriented optimization. RS-D3SAC combines the strengths of both components and consistently achieves the best CVaR performance; however, the distributional critic contributes more substantially to the observed CVaR improvement than the diffusion actor, with the diffusion-based actor providing complementary refinement.

Fig. 7(a) examines performance under varying numbers of users. As the user population increases, system load intensifies, resulting in more complex scheduling decisions and heavier-tailed VAoI distributions. Consequently, the relative advantage of RS-DSAC and RS-D3SAC over SAC and D2SAC becomes more significant in terms of CVaR. This trend indicates that value distribution modeling becomes increasingly important as system scale grows. In addition, the diffusion actor provides moderate but consistent improvements compared with the standard SAC baseline. The gains remain stable across different system scales, confirming the robustness and complementary benefits of the diffusion-based policy representation. Fig. 7(b) illustrates the impact of packet generation rate. At low generation rates, the system operates in a lightly loaded regime, and performance differences among algorithms are relatively small. As the generation rate increases, the benefit of distributional modeling becomes more evident. Quantitatively, the CVaR reduction achieved by RS-D3SAC compared with SAC increases from 8.9% to 15.4% as r_n increases from 0.25 to 0.75, further highlighting the effectiveness of distributional critics under high-load conditions in which extreme VAoI events occur more frequently. Moreover, Fig 7(c) examines the impact of packet transmission success probability. RS-D3SAC consistently outperforms the other methods. For instance, when $p = 0.8$, RS-D3SAC achieves a 14.5% reduction in $\mathrm{CVaR}_{0.75}$ compared with SAC, demonstrating robust performance under stochastic channel conditions.

Effects of Different CVaR Confidence Levels: Fig. 8 illustrates the tail-risk performance of the SAC-based algorithms for varying system-level CVaR confidence levels. As α increases, the CVaR of VAoI rises accordingly, since larger confidence levels place greater emphasis on rare but severe VAoI realizations. Across all confidence levels, RS-DSAC consistently outperforms D2SAC, demonstrating the advantage of learning the full return distribution rather than relying on

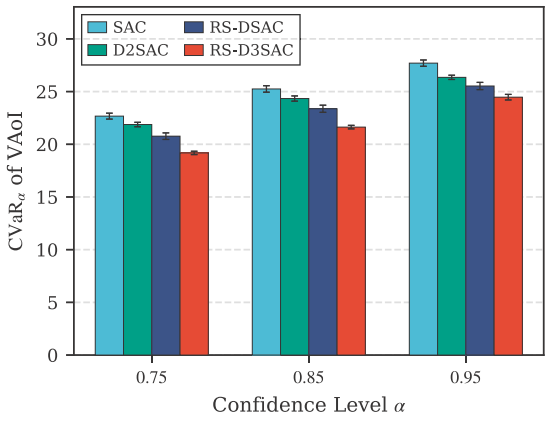
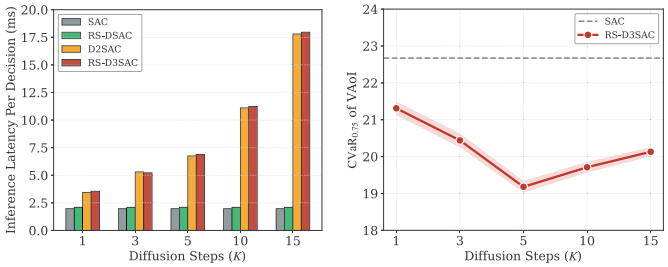


Fig. 8: Performance comparison of SAC, D2SAC, RS-DSAC, and RS-D3SAC under different CVaR confidence level α.

expectation-based value estimates. Thanks to the synergistic integration of the diffusion-based actor and the distributional critic, RS-D3SAC remains effective across a broad spectrum of risk sensitivities, maintaining superior performance even under stringent tail-risk requirements, such as $\alpha = 0.95$. These results further confirm that RS-D3SAC provides a robust and scalable solution for risk-sensitive VAoI scheduling, demonstrating its effectiveness in controlling extreme semantic staleness events under stringent risk requirements.

Effects of the Number of Denoising Steps K: To evaluate the practical feasibility of the proposed diffusion-based policy, we measure the *inference latency per decision*, defined as the time required to generate an action given the current system state during online deployment. Since policy parameters are fixed during deployment, inference latency represents the primary computational overhead in real-time VAoI scheduling systems. In our experiments, we did not observe substantial differences in convergence speed or training stability across the tested values of K. Fig. 9 shows the relationship between inference latency and K, together with the corresponding CVaR performance of VAoI.

As shown in Fig. 9(a), the inference latency of RS-D3SAC increases moderately with K, reaching approximately 3.5 ms, 6.9 ms, and 11.2 ms for $K = 1$, 5, and 10, respectively, while the SAC baseline maintains a latency of approximately 2 ms. D2SAC exhibits a similar trend, indicating that the additional computational overhead primarily originates from the diffusion-based actor. In contrast, RS-DSAC exhibits inference latency comparable to SAC, suggesting that the distributional critic introduces negligible online computational overhead. Fig. 9(b) shows that the system-level CVaR of VAoI decreases as K increases from 1 to 5, with the best performance achieved at $K = 5$. Further increasing K slightly degrades CVaR performance while incurring substantially higher inference latency, indicating that a moderate number of denoising steps captures most of the benefit of diffusion-based policy generation, consistent with observations in the diffusion-policy literature [18]. At $K = 5$, RS-D3SAC reduces the system-level CVaR of VAoI by 7.66% relative to RS-DSAC, while increasing the inference latency from approximately 2 ms to 6.9 ms per decision on the tested platform. This comparison



(a) (b)

Fig. 9: Impact of the denoising step K on inference latency and CVaR performance. Subfigures illustrate (a) the inference latency per decision and (b) the CVaR of VAoI under different values of K.

indicates that the distributional critic provides the dominant improvement in tail-risk control, whereas the diffusion-based actor offers an additional, smaller reduction at the cost of increased inference latency. RS-DSAC provides an alternative that retains the principal benefit of distributional risk-sensitive learning with lower inference latency. Since absolute latency depends on the hardware platform, these results characterize the relative computational overhead and performance trade-off rather than establishing universal real-time feasibility. Further acceleration techniques may be adopted for resource-constrained implementations.

3) Sensitivity to the Return-Level Tail Probability and Empirical Validation of the Surrogate Objective

The system-level objective in (5) is defined in terms of the empirical CVaR_α of VAoI, whereas RS-D3SAC optimizes the CVaR_φ of the discounted-return distribution. These two quantities are not generally equivalent. Since directly optimizing the stationary distribution of the VAoI process is difficult in a sequential stochastic CMDP, we use the return-level CVaR as a tractable risk-sensitive surrogate. To examine the empirical validity of this surrogate, we conduct two sets of experiments: a sensitivity analysis with respect to the return-level lower-tail probability φ, and a checkpoint-based analysis of the relationship between the return-level and system-level CVaR.

Fig. 10 illustrates the empirical $\mathrm{CVaR}_{0.75}$ of VAoI and the average VAoI achieved by RS-D3SAC for different values of φ. As φ increases, the average VAoI reduces in our experiments, indicating a gradual shift toward risk-neutral policy behavior as a larger fraction of the return distribution is considered. Meanwhile, the lowest empirical $\mathrm{CVaR}_{0.75}$ is obtained when $\varphi = 1 - \alpha = 0.25$. This result suggests that aligning the risk level used for return-level optimization with the tail fraction considered in the system-level evaluation provides the most effective tail-risk control in the considered setting. The results also illustrate the trade-off between average freshness and tail-risk mitigation as the risk level varies.

Fig. 11 examines the relationship between the $\mathrm{CVaR}_{0.25}$ of the return and the empirical $\mathrm{CVaR}_{0.75}$ of VAoI at different training checkpoints. As noted earlier, every 1000 environment steps, the current policy is fixed and evaluated over 5000 time slots in the testing environment. Because the reward is defined as the negative VAoI plus the cost penalty, a larger return CVaR corresponds to less severe lower-tail return outcomes. As training progresses, the $\mathrm{CVaR}_{0.25}$ of the return increases, while the empirical $\mathrm{CVaR}_{0.75}$ of VAoI decreases

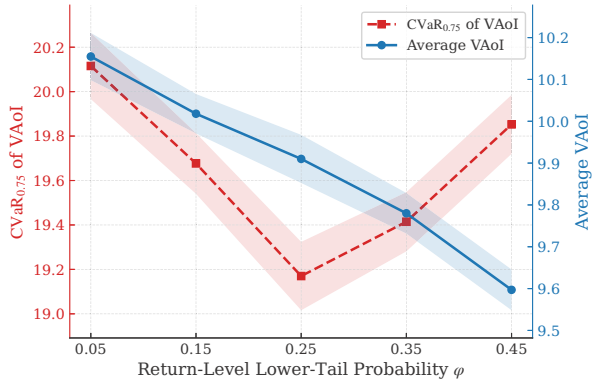


Fig. 10: Performance of RS-D3SAC under different return-level lower-tail probabilities φ when $\alpha = 0.75$.

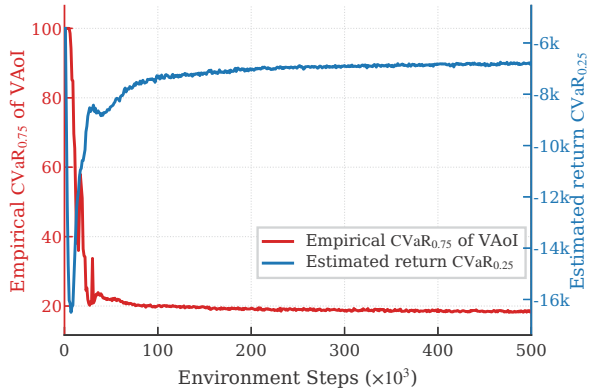


Fig. 11: Evolution of $\mathrm{CVaR}_{0.25}$ of the return and empirical $\mathrm{CVaR}_{0.75}$ of VAoI over environment steps.

correspondingly. To quantify the relationship between the two risk measures, we calculate Spearman's rank correlation coefficient between them across evaluated checkpoints, separately for each independent training run. Across independent runs, the mean correlation coefficient is -0.837, with a standard deviation of 0.021, indicating a strong negative monotonic association between the two quantities during training. The observed co-movement between the two quantities shown in Fig. 11 and the quantitative correlation support the empirical effectiveness of return-level CVaR as a surrogate objective for system-level VAoI tail-risk reduction.

VII. Conclusion

We investigate risk-sensitive VAoI scheduling in multi-user wireless status update systems with stochastic packet arrivals, unreliable transmission channels, and long-term transmission cost constraints. In particular, we put forth RS-D3SAC, a risk-sensitive deep distributional diffusion-based SAC algorithm that integrates a distributional critic with a diffusion-based actor to enable effective tail-risk-aware control.

To account for rare but severe semantic staleness events, the scheduling problem is formulated as a CMDP, and CVaR is adopted as a risk-sensitive performance metric to characterize the tail behavior of the VAoI process. The proposed RS-D3SAC approach employs quantile-based distributional critics to model the return distribution and enable CVaR-oriented policy optimization, while a diffusion-based actor is incorporated as a complementary mechanism to improve policy smoothness and learning stability. In addition, a Lagrangian primal-dual mechanism is adopted to enforce long-term transmission cost constraints during learning.

Extensive simulation results demonstrate that RS-D3SAC consistently achieves lower CVaR of VAoI compared with conventional DRL baselines while maintaining competitive average VAoI performance. Additional comparisons show that the distributional critic contributes more substantially to the CVaR reduction, whereas the diffusion-based actor provides additional and consistent improvements in performance and robustness. These results highlight the effectiveness of distributional reinforcement learning for risk-sensitive semantic information freshness optimization and demonstrate the complementary benefits of diffusion-based policy representations. Overall, the proposed framework provides a practical and principled approach for risk-aware semantic freshness management in future wireless scheduling systems.

References

[1] R. D. Yates, Y. Sun, D. R. Brown, S. K. Kaul, E. Modiano, and S. Ulukus, "Age of information: An introduction and survey," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 5, pp. 1183–1210, May 2021.

[2] A. Kosta, N. Pappas, V. Angelakis *et al.*, "Age of information: A new concept, metric, and tool," *Foundations and Trends® in Networking*, vol. 12, no. 3, pp. 162–259, 2017.

[3] N. Pappas, M. A. Abd-Elmagid, B. Zhou, W. Saad, and H. S. Dhillon, *Age of Information: Foundations and Applications*. Cambridge University Press, 2023.

[4] R. D. Yates, "The age of gossip in networks," in *Proc. IEEE ISIT*, Jul. 2021, pp. 2984–2989.

[5] B. Buyukates, M. Bastopcu, and S. Ulukus, "Version age of information in clustered gossip networks," *IEEE J. Sel. Areas Inf. Theory*, vol. 3, no. 1, pp. 85–97, Mar. 2022.

[6] E. Delfani and N. Pappas, "Version age-optimal cached status updates in a gossiping network with energy harvesting sensor," *IEEE Trans. Commun.*, vol. 73, no. 4, pp. 2344–2360, Apr. 2025.

[7] X. Hu, N. Pappas, and H. H. Yang, "Version age-based client scheduling policy for federated learning," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. Workshops (ICASSPW)*, Apr. 2024, pp. 695–699.

[8] I. Hasan and A. Arafa, "Version age of information with contact mobility in gossip networks," *arXiv preprint arXiv:2509.15184*, 2025.

[9] R. Rockafellar and S. Uryasev, "Conditional value-at-risk for general loss distributions," *Journal of Banking & Finance*, vol. 26, no. 7, pp. 1443–1471, 2002.

[10] Q. Zhang, S. Leng, X. Ma, Q. Liu, X. Wang, B. Liang, Y. Liu, and J. Yang, "Cvar-constrained policy optimization for safe reinforcement learning," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 36, no. 1, pp. 830–841, Jan. 2025.

[11] E. Delfani and N. Pappas, "Semantics-aware status updates with energy harvesting devices: Query version age of information," in *Proc. WiOpt*, Oct. 2024, pp. 177–184.

[12] G. Karevvanavar, H. Pable, O. Patil, R. Vishweshwar Bhat, and N. Pappas, "Version age of information minimization over fading broadcast channels," *IEEE Trans. Wireless Commun.*, vol. 24, no. 2, pp. 1620–1634, Feb. 2025.

[13] N. C. Luong, D. T. Hoang, S. Gong, D. Niyato, P. Wang, Y.-C. Liang, and D. I. Kim, "Applications of deep reinforcement learning in communications and networking: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3133–3174, 2019.

[14] W. Dabney, M. Rowland, M. Bellemare, and R. Munos, "Distributional reinforcement learning with quantile regression," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, Apr. 2018.

[15] E. Eldeeb and H. Alves, "Offline and distributional reinforcement learning for wireless communications," *IEEE Commun. Mag.*, vol. 63, no. 8, pp. 71–76, Aug. 2025.

[16] H. Du, R. Zhang, Y. Liu, J. Wang, Y. Lin, Z. Li, D. Niyato, J. Kang, Z. Xiong, S. Cui, B. Ai, H. Zhou, and D. I. Kim, "Enhancing deep reinforcement learning: A tutorial on generative diffusion models in network optimization," *IEEE Commun. Surveys Tuts.*, vol. 26, no. 4, pp. 2611–2646, 2024.

[17] Z. Chen, X. Long, L. Zhang, and W. Cai, "Toward diffusion-based deep reinforcement learning for discrete decision-making: Methods and evaluations," *IEEE Trans. Netw. Sci. Eng.*, vol. 13, pp. 4406–4421, 2026.

[18] H. Du, Z. Li, D. Niyato, J. Kang, Z. Xiong, H. Huang, and S. Mao, "Diffusion-based reinforcement learning for edge-enabled AI-generated content services," *IEEE Trans. Mob. Comput.*, vol. 23, no. 9, pp. 8902–8918, Sept. 2024.

[19] M. Zou and H. Pan, "Minimizing version age of information in resource-constrained multi-source systems via Lyapunov and learning based scheduling," in *International Conference on Information and Communication Networks (ICICN)*, 2025, pp. 444–449.

[20] G. Zhang, C. Shen, Q. Shi, B. Ai, and Z. Zhong, "Aoi minimization for wsn data collection with periodic updating scheme," *IEEE Trans. Wireless Commun.*, vol. 22, no. 1, pp. 32–46, 2023.

[21] H. Hu, K. Xiong, H.-C. Yang, Q. Ni, B. Gao, P. Fan, and K. B. Letaief, "Aoi-minimal online scheduling for wireless-powered iot: A lyapunov optimization-based approach," *IEEE Trans. Green Commun. Netw.*, vol. 7, no. 4, pp. 2081–2092, 2023.

[22] C. Zhou, H. He, P. Yang, F. Lyu, W. Wu, N. Cheng, and X. Shen, "Deep rl-based trajectory planning for aoi minimization in uav-assisted iot," in *2019 11th International Conference on Wireless Communications and Signal Processing (WCSP)*, 2019, pp. 1–6.

[23] E. Delfani and N. Pappas, "Optimizing version AoI in energy-harvesting IoT: Model-based and learning-based approaches," *arXiv preprint arXiv:2510.00904*, 2025.

[24] Z. Wang, J. J. Hunt, and M. Zhou, "Diffusion policies as an expressive policy class for offline reinforcement learning," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2023.

[25] Z. Yao, Z. Tang, W. Yang, and W. Jia, "Enhancing LLM QoS through cloud-edge collaboration: A diffusion-based multi-agent reinforcement learning approach," *IEEE Trans. Serv. Comput.*, vol. 18, no. 3, pp. 1412–1427, May-June, 2025.

[26] Z. Rao, Z. Zhu, Y. Yao, Y. Xu, Y. Cheng, and H. Du, "Computation-offloading optimization for satellite edge computing via diffusion and lyapunov-based deep reinforcement learning," *IEEE Internet Things J.*, vol. 12, no. 18, pp. 37 747–37 762, Sept. 2025.

[27] X. Cheng, W. Wu, Y. Wang, Z. Mao, Y. Lu, and P. Dong, "Dependency-aware cav task scheduling via diffusion-based reinforcement learning," *IEEE Trans. Netw. Sci. Eng.*, vol. 13, pp. 4797–4814, 2026.

[28] M. G. Bellemare, W. Dabney, and R. Munos, "A distributional perspective on reinforcement learning," in *Int. Conf. Mach. Learn.* PMLR, 2017, pp. 449–458.

[29] B. Zhou, W. Saad, M. Bennis, and P. Popovski, "Risk-aware optimization of age of information in the internet of things," in *Proc. IEEE ICC*, Jun. 2020, pp. 1–6.

[30] Y. Xiao, Q. Du, and G. K. Karagiannidis, "Statistical age of information: A risk-aware metric and its applications in status updates," *IEEE Trans. Wireless Commun.*, vol. 24, no. 3, pp. 2325–2340, Mar. 2025.

[31] H. Farag, M. Gidlund, and Č. Stefanović, "A deep reinforcement learning approach for improving age of information in mission-critical IoT," in *Proc. IEEE GCAIoT*, Dec. 2021.

[32] W. de Sombre, A. Ortiz, F. Aurzada, and A. Klein, "Risk-sensitive optimization and learning for minimizing age of information in point-to-point wireless communications," in *Proc. IEEE ICC Workshops*, May 2023, pp. 1161–1166.

[33] C. Ying, X. Zhou, H. Su, D. Yan, N. Chen, and J. Zhu, "Towards safe reinforcement learning via constraining conditional value-at-risk," *arXiv preprint arXiv:2206.04436*, 2022.

[34] W. Zhang, M. Derakhshani, G. Zheng, and S. Lambotharan, "Constrained risk-sensitive deep reinforcement learning for eMBB-URLLC joint scheduling," *IEEE Trans. Wireless Commun.*, vol. 23, no. 9, pp. 10 608–10 624, Sept. 2024.

[35] J. Liu, D. Li, and Y. Xu, "Deep distributional reinforcement learning-based adaptive routing with guaranteed delay bounds," *IEEE/ACM Trans. Netw.*, vol. 32, no. 6, pp. 4692–4706, Dec. 2024.

[36] X. Ma, J. Chen, L. Xia, J. Yang, Q. Zhao, and Z. Zhou, "DSAC: Distributional soft actor-critic for risk-sensitive reinforcement learning," *Journal of Artificial Intelligence Research*, vol. 83, 2025.

[37] S. H. Lim and I. Malik, "Distributional reinforcement learning for risk-sensitive policies," *Advances in Neural Information Processing Systems*, vol. 35, pp. 30 977–30 989, 2022.

[38] X. Zheng, S. Zhou, and Z. Niu, "Urgency of information for context-aware timely status updates in remote control systems," *IEEE Trans. Wireless Commun.*, vol. 19, no. 11, pp. 7237–7250, Nov. 2020.

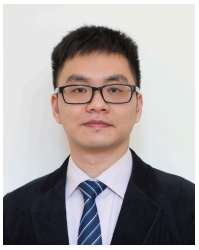
[39] A. Li, S. Wu, S. Sun, and J. Cao, "Goal-oriented tensor: Beyond age of information toward semantics-empowered goal-oriented communications," *IEEE Trans. Commun.*, vol. 72, no. 12, pp. 7689–7704, Dec. 2024.

[40] E. Altman, *Constrained Markov Decision Processes*. CRC Press, 1999, vol. 7.

[41] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020, pp. 6840–6851.

[42] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[43] M. Hessel, J. Modayil, H. Van Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. Azar, and D. Silver, "Rainbow: Combining improvements in deep reinforcement learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, no. 1, 2018.



Haoyuan Pan (Member, IEEE) received the B.E. and Ph.D. degrees in Information Engineering from The Chinese University of Hong Kong (CUHK), Hong Kong, in 2014 and 2018, respectively. From 2018 to 2020, he was a Postdoctoral Fellow with the Department of Information Engineering, CUHK. Since 2020, he has been with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China, where he is currently an Associate Professor. He received the Best Paper Award at the 2024 IEEE Wireless Communications and Networking Conference (WCNC). His research interests include wireless communications, information freshness and reliability, risk-sensitive reinforcement learning, and sequential decision-making under uncertainty.



Sizhao Chen received the B.S. degree from Jimei University, Xiamen, China, in 2025. He is currently working toward the M.E. degree in Shenzhen University, Shenzhen, China. His research interests include age of information (AoI), diffusion models, and reinforcement learning.



Zhaorui Wang (Member, IEEE) received the Ph.D. degree from The Chinese University of Hong Kong (CUHK) in 2019, and the B.S. degree from University of Electronic Science and Technology of China (UESTC) in 2015. He is now an Assistant Professor at the School of Computing and Information Technology, Great Bay University. He was a Research Assistant Professor at The Chinese University of Hong Kong (Shenzhen) from 2022 to 2025, a Postdoctoral Research Associate at CUHK from 2021 to 2022, and a Postdoctoral Research Associate at The Hong Kong Polytechnic University from 2019 to 2020. He is a recipient of the Hong Kong PhD Fellowship from 2015 to 2018. He has been selected in the post of "Pengcheng Peacock Plan" (Type C) since 2022. His research interests focus on super-resolution sensing and massive random access in low-altitude networks, as well as the implementation of corresponding hardware-software prototype systems.



Tse-Tin Chan (Member, IEEE) received the B.Eng. and Ph.D. degrees in information engineering from The Chinese University of Hong Kong (CUHK), Hong Kong, China, in 2014 and 2020, respectively. From 2020 to 2022, he was an Assistant Professor with the Department of Computer Science, The Hang Seng University of Hong Kong (HSUHK), Hong Kong. He is currently an Assistant Professor with the Department of Mathematics and Information Technology, The Education University of Hong Kong (EdUHK), Hong Kong. His research interests include wireless communications and networking, age of information (AoI), and AI-native wireless systems. He has served as a Guest Editor for the IEEE Open Journal of the Communications Society. He has also served on the organizing committees for workshops at IEEE ICC 2026 and IEEE VTC 2026-Spring, and as a TPC member for various flagship conferences such as IEEE ICC and IEEE GLOBECOM.