Secure Cooperative Offloading Across Domains With DIB-Enhanced Multi-Agent RL and Credence Mutual Information

Kexin Li, Changjian Xu, Hongwei Zhu, Wen Yin, Yu Liu, Yu Peng, *Member, IEEE*, Yong Wang, Xingwei Wang, and Schahram Dustdar, *Fellow, IEEE*

***Abstract*—Multi-access Edge Computing (MEC) supports latency-critical applications while coping with heterogeneous edge resources and stringent privacy requirements. In practice, task offloading is hindered by security risks in cross-domain transmission and by insufficient coordination among decentralized user devices. To address these challenges, we propose MITRA-DIB, a distributed computation-offloading framework based on multi-agent reinforcement learning (MARL) for *Domain-Secure MEC (DS-MEC)*. MITRA-DIB introduces DIB (Decision Information Bottleneck) to learn task-adaptive representations that preserve essential decision-making features while suppressing decision-irrelevant perturbations, thereby mitigating spurious correlations and reducing unnecessary inter-agent coupling. To promote effective cooperation under centralized training with decentralized execution (CTDE), we further design a *Credence Mutual Information (CMI)* auxiliary reward computed via neural Mutual Information estimators, and we incorporate quality-aware experience shaping with positive/negative buffers to selectively amplify beneficial interactions while attenuating detrimental ones. Moreover, we enforce a domain-security rule that constrains sensitive tasks to within-domain execution and secures cross-domain transmission of normal tasks through lightweight cryptography primitives. Extensive experiments show that MITRA-DIB consistently outperforms state-of-the-art baselines, improving the task success rate by up to 30.3% in complex scenarios while achieving lower system costs and faster convergence.**

***Index Terms*—Task offloading, multi-access edge computing, privacy security, multi-agent reinforcement learning.**

I. Introduction

THE explosive growth of the Internet of Things (IoT) and mobile smart terminals has catalyzed the emergence of numerous compute-intensive and latency-critical applications, such as augmented reality and in-vehicle real-time monitoring [1], [2]. These applications demand ultra-low latency and high energy efficiency at the network edge [3]. Multi-access Edge Computing (MEC) addresses this demand by offloading tasks from user devices (UDs) to nearby edge servers, thereby alleviating the computational burden on UDs and reducing reliance on remote clouds. However, practical MEC deployments are hindered by load imbalance arising from resource heterogeneity and security threats during cross-domain transmission [4], [5]. *Consequently, designing task-offloading strategies that improve efficiency while supporting reliable collaboration and practical security protection remains an important problem.*

Traditional mathematical-programming-based solutions are often difficult to scale in heterogeneous and time-varying MEC systems due to the need for accurate system modeling and repeated optimization [6]. In contrast, distributed task offloading via multi-agent reinforcement learning (MARL) offers scalability and low online overhead by enabling each UD to learn adaptive policies from interaction. Most existing studies adopt centralized training with decentralized execution (CTDE), where a centralized critic leverages global information during training while each agent executes using only local observations [7]. Despite encouraging progress, current MARL-based offloading methods remain insufficient for realistic MEC scenarios where inter-device communication is limited and privacy requirements are stringent. Specifically, three issues are particularly pronounced.

First, security is a first-class constraint in practical offloading. The open and heterogeneous nature of the edge/cloud infrastructure brings significant risks during cross-domain transmission, which necessitates additional protection services integrated into the offloading architecture [8]. Moreover, even if link-layer encryption is adopted, directly transmitting structured and redundant semantic/state features may still expose sensitive information through side channels or representation reversibility, enabling adversaries to infer private semantics or inject

Received 22 April 2026; revised 3 August 2026; accepted 13 August 2026. Date of publication 20 August 2026; date of current version 9 October 2026. This work was supported in part by Sichuan Science and Technology Program under Grant 2024ZDZX0011, Grant 2025ZNSFSC1472, Grant 2025ZHCG0002, Grant 2026-YF09-00044-SN and Grant 2026NSFSC1458, in part by the Postdoctoral Fellowship Program of CPSF under Grant GZC20251074, and in part by the Fundamental Research Funds for the Central Universities under Grant ZYGX2025XJ043. *(Corresponding author: Yu Peng.)*

Kexin Li, Yu Peng, and Yong Wang are with the Laboratory of Intelligent Collaborative Computing, University of Electronic Science and Technology of China, Sichuan 611731, China, and also with the Ubiquitous Intelligence and Trusted Services Key Laboratory of Sichuan Province, Chengdu 611731, China (e-mail: ypeng@uestc.edu.cn).

Changjian Xu, Hongwei Zhu, and Wen Yin are with the Laboratory of Intelligent Collaborative Computing, University of Electronic Science and Technology of China, Chengdu 611731, China.

Yu Liu is with the Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100085, China.

Xingwei Wang is with the College of Computer Science and Engineering, Northeastern University, Shenyang 110819, China.

Schahram Dustdar is with Distributed Systems Group, TU Wien, 1040 Vienna, Austria, and also with ICREA, 08010 Barcelona, Spain.

The code is available at https://github.com/Cherlie-321/MITRA-DIB.

Digital Object Identifier 10.1109/TSC.2026.3725533

1939-1374 © 2026 IEEE. All rights reserved, including rights for text and data mining, and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: TU Wien Bibliothek. Downloaded on November 03,2026 at 09:06:24 UTC from IEEE Xplore. Restrictions apply.

perturbations that degrade learning stability and cooperative efficiency [9], [10].

Second, effective cooperation is challenging under decentralization. Even when the critic observes the full system state, if agents are optimized primarily with a global system-cost signal without explicitly accounting for others' actions, coordination failures may arise and lead to suboptimal joint performance [11]. This challenge is amplified in dynamic MEC environments, where coupling among agents should be encouraged only when necessary, rather than being an unintended byproduct of correlated noise or transient fluctuations.

Third, raw system states in multi-agent interactive environments often contain substantial high-variance perturbations that fluctuate rapidly over time but are only weakly related to task decisions. If policies inadvertently rely on such volatile information, they may learn spurious correlations, overfit to incidental patterns, and exhibit degraded robustness under environmental shifts or adversarial disturbances [12].

To address these challenges, we propose **MITRA-DIB** (Mutual-Information Trust Agents with DIB enhancement), a novel computation-offloading framework designed for *Domain-Secure Multi-access Edge Computing (DS-MEC)*. The design of MITRA-DIB follows a multi-tier optimization logic. (1) *System modeling with domain security:* we introduce a domain-security rule that enforces within-domain execution for sensitive tasks and imposes encryption and security-cost constraints on cross-domain transmission of normal tasks, while preserving device scalability. (2) *Decision-sufficient representation learning:* we develop a new bottleneck module, termed **DIB — Decision Information Bottleneck**, to encode raw system states into task-adaptive representations that retain factors sufficient for offloading decisions while suppressing random perturbations and redundant semantics. Such a bottleneck not only improves training stability, but also reduces unnecessary inter-agent coupling. From the perspective of a *robust action prior*, constraining the learned policy distribution to avoid excessive deviation from a prior action distribution can prevent extreme or brittle action biases under perturbations, improving adaptability to varying task demands. Furthermore, DIB enforces controlled compression to reduce the information exposure surface at the representation level, thereby mitigating privacy risks and mitigating spurious-correlation-driven instability. (3) *CMI-guided cooperation with quality-aware experience shaping:* based on DIB representations and joint actions, we construct a *Credence Mutual Information (CMI)* term as an auxiliary reward to enhance coordination under CTDE and preserve scalability as the number of agents increases. We employ neural mutual-information estimators to compute upper/lower bounds and use them to provide informative cooperative signals. To avoid uniformly amplifying both good and bad correlations, we further split experiences into high- and low-quality buffers and selectively amplify or attenuate their influence during learning, fostering cooperation without inducing excessive coupling. The main contributions are summarized as follows:

- We formulate a DS-MEC offloading model with a domain-security rule and integrate lightweight cryptography primitives to protect cross-domain transmissions while preserving scalability and practical deployability.
- We propose **DIB (Decision Information Bottleneck)**, a task-oriented representation bottleneck that suppresses high-variance, decision-irrelevant perturbations to mitigate spurious correlations and reduce unnecessary inter-agent coupling, thereby improving stability, robustness, and communication efficiency in cooperative offloading.
- We develop a CMI-guided MARL framework with quality-aware positive/negative experience buffers, enabling effective cooperation under CTDE while avoiding the amplification of suboptimal correlations; extensive experiments demonstrate that our approach reduces average system cost and improves task success rate by up to 30.3% in complex DS-MEC scenarios.

The rest of this article is organized as follows. Section II reviews related work. The system model and optimization problem of this paper are illustrated in Section III. Section IV describes MITRA-DIB in detail, covering the DIB encoder, the CMI-based cooperation reward with MINE/CLUB estimators, the positive/negative replay design, and the CTDE training algorithm. Section V reports simulation settings and evaluation results. Finally, Section VI concludes the article.

II. Related Work

In this section, we review the work related to this paper and summarize the existing studies.

A. MEC Task Offloading and Resource Allocation

MEC task offloading and resource allocation are formulated as mixed-integer nonlinear programming [13] or stochastic optimization problems [14]. The primary objective is to minimize total system costs under strict constraints on energy consumption, edge capacity, and queueing delays [15]. Traditionally, these complex formulations are addressed via mathematical programming or game-theoretical frameworks [16].

Notable approaches include gradient projection to manage partial offloading, and Lyapunov-based optimization for adaptive resource management under time-varying conditions [17]. While providing rigorous bounds, their effectiveness relies heavily on exact modeling and precise forecasting. In highly dynamic edge environments, iterative solvers cause significant computational bottlenecks. Consequently, limited scalability and vulnerability to environmental non-stationarity remain major deployment obstacles [3], [18], [19].

While meta-heuristic algorithms are frequently employed to circumvent these bottlenecks, their slow convergence fundamentally conflicts with the real-time demands of latency-critical tasks [20]. To bridge this gap, recent work increasingly adopts deep reinforcement learning, which shifts the heavy computational burden to offline training and facilitates rapid and model-free execution in dynamic environments [21], [22], [23].

Authorized licensed use limited to: TU Wien Bibliothek. Downloaded on November 03,2026 at 09:06:24 UTC from IEEE Xplore. Restrictions apply.

B. Security and Privacy-Aware Offloading

Beyond computational efficiency, security threats such as eavesdropping, tampering, and inference attacks represent critical constraints in Multi-access Edge Computing systems [24], [25]. These challenges become particularly acute during cross-domain interactions. The inherent openness of edge environments allows malicious entities to intercept offloaded tasks or inject perturbations easily. To address these threats, recent research has explored various security-aware offloading frameworks. For example, trusted execution environments provide powerful hardware-level isolation to guarantee the security of the execution process on the edge side. However, they rely on specific hardware architectures. Homomorphic encryption can also provide security guarantees, but it introduces huge computational overhead, which conflicts with the core goals of processing latency-sensitive MEC systems [26]. Other approaches utilize Physical Layer Security to defend against eavesdropping [27], [28] or integrate technologies such as blockchain and Digital Twins to ensure reliable, tamper-proof communication and robust authentication [29], [30], [31].

However, these existing security measures primarily focus on protecting communication channels or providing macro-level access control, whereas the task sensitivity constraints and representation-level information leakage risks in cross-domain offloading are less considered. Specifically, raw semantic or state features transmitted across domains can leak sensitive information through representation reversibility or side channels [9], [10]. This risk persists even when the underlying link is fully encrypted.

In contrast, our DS-MEC framework addresses these vulnerabilities through a multi-tiered security design. At the macro and link layers, we strictly confine highly sensitive tasks to within-domain execution, while securing the cross-domain transmission of normal tasks using lightweight cryptography. Crucially, at the representation level, we introduce a bottleneck module inspired by Information Bottleneck principles [32]. By actively compressing transmitted state features to retain only decision-sufficient information, our approach effectively reduces the information exposure surface, mitigating inference and reconstruction risks that bypass standard encryption.

C. Multi-Agent RL and Coordination Mechanisms.

To overcome traditional optimization bottlenecks, MARL has been widely adopted for decentralized MEC offloading [33], [34], [35], [36]. While algorithms like MADDPG [37], FACMAC [38], HAPPO [39] facilitate autonomous decision-making, achieving robust coordination remains challenging. Mutual-information-based MARL methods have attracted increasing attention [40]. However, existing methods often estimate mutual information directly from raw states and generally assume that higher mutual information corresponds to stronger cooperation. In highly dynamic and safety-sensitive MEC scenarios, such methods have two limitations: mutual information only reflects statistical dependence and cannot distinguish beneficial cooperation from low-return coupling [41], [42]; moreover, directly estimating mutual information from raw states containing random channel fluctuations and redundant information may cause the model to learn spurious correlations [43], [44]. Therefore, MITRA-DIB employs a DIB jointly constrained by policy and value learning to extract decision-relevant representations, thereby reducing the influence of environmental noise and redundant information. Subsequently, a quality-aware CMI mechanism selectively enhances beneficial cooperation while suppressing harmful coupling.

TABLE I
KEY NOTATIONS IN THE SYSTEM MODEL

Notation	Description
$\mathcal{M}$, m	Set/index of user devices (UDs)
$\mathcal{E}$, $\mathcal{C}$	Sets of edge servers (ESs) and cloud servers (CSs)
$D(\cdot)$	Domain mapping function of an entity
$\mathcal{N}$, n	Set/index of time slots, $\mathcal{N} = \{1, \ldots, N\}$
$\mathcal{T}_n^m$	Task generated by UD m at slot n
h_n^m	Task input size (bits)
u^{com}	CPU cycles per bit for task computing
u^{enc}	CPU cycles per bit for encryption/decryption
$t^{\max}$	Maximum tolerable task latency
ι_n^m	Task security flag (1: sensitive, 0: normal)
Γ_n^m	Feasible offloading destination set for $\mathcal{T}_n^m$
Π_n^m	Cross-domain encryption indicator (1: encrypted transmission, 0: otherwise)
a_n^m	Offloading action of UD m at slot n
$\alpha_n^m, \beta_n^m, \gamma_n^m$	Task proportions for local/edge/cloud execution
$e_n^{m,\text{dest}}$	Selected edge destination (if $\beta_n^m \neq 0$)
$c_n^{m,\text{dest}}$	Selected cloud destination (if $\gamma_n^m \neq 0$)
s_n	Global system state at slot n
$Q_n^{\text{es}}, Q_n^{\text{cs}}$	Task queue lengths at ES/CS at slot n
$f^{\text{ud}}, f^{\text{es}}, f^{\text{cs}}$	Computing capacities of UD/ES/CS
υ_n^m	Remaining energy budget of UD m at slot n
B	Sub-channel bandwidth
p^{tran}	Transmit power of UD m (D2E or D2C)
$g_n^{m,\cdot}$	Channel gain (D2E or D2C)
σ^2	Noise power
$r_n^{m,\cdot}$	Achievable transmission rate (D2E or D2C)
$t_n^{m,\text{tran}}$	Transmission time for offloading
$t_n^{m,\text{wait}}$	Queueing delay at the selected server
$t_n^{m,\text{exe}}$	Execution time at the selected server
$t_n^{m,\text{back}}$	Result backhaul time
$t_n^{m,\text{enc}}, t_n^{m,\text{dec}}$	Encryption/decryption time for cross-domain tasks
$e_n^{m,\text{ud}}, e_n^{m,\text{D2E}}, e_n^{m,\text{D2C}}$	Energy consumption for local/edge/cloud processing
$\chi_n^{m,\mathcal{T}}$	Security cost induced by cross-domain transmissions
$\Lambda_s, \Lambda_t, \Lambda_e$	Weights for security/latency/energy costs
$Z_n^{m,\mathcal{T}}$	Total cost of task $\mathcal{T}_n^m$ at slot n
c	Overall system cost averaged over slots and UDs

III. SYSTEM MODEL AND PROBLEM DEFINITION

This section presents the system model of the DS-MEC framework and formulates the corresponding optimization problem. We first introduce the multi-domain architecture with heterogeneous user devices, edge servers, and cloud resources. We then establish the communication and multi-tier computation models, incorporating security-aware task classification and cryptographic overhead. Finally, we formulate a joint optimization problem to minimize the overall system cost, including latency, energy consumption, and cross-domain security risk. The main notations are listed in Table I.

A. Model Overview

We consider two tiers of computational infrastructure: *edge servers* (ESs) and *cloud servers* (CSs). An ES is a

Authorized licensed use limited to: TU Wien Bibliothek. Downloaded on November 03,2026 at 09:06:24 UTC from IEEE Xplore. Restrictions apply.

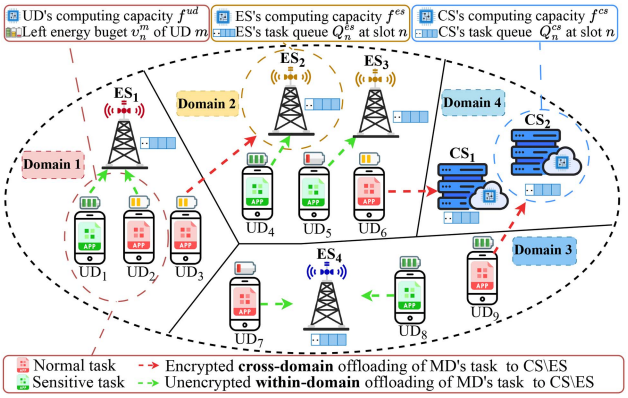


Fig. 1. An illustration of DS-MEC system.

computing node deployed at the network edge (e.g., co-located with access points/base stations or within a local administrative domain), which provides low-latency access but has comparatively limited computing capacity and may experience higher load variability. A CS is a remote cloud computing node (e.g., in centralized data centers), which typically offers much higher computing capacity and more stable resources at the cost of longer end-to-end communication latency. To capture cross-domain security requirements, we further model the infrastructure as multiple *domains*. As illustrated in Fig. 1, we consider a Domain-Secure Multi-access Edge Computing (DS-MEC) system consisting of edge servers (ESs) $\mathcal{E} = \{\mathrm{ES}_1, \ldots, \mathrm{ES}_{k_s}\}$, cloud servers (CSs) $\mathcal{C} = \{\mathrm{CS}_1, \ldots, \mathrm{CS}_{k_c}\}$, and user devices (UDs) $\mathcal{M} = \{\mathrm{UD}_1, \ldots, \mathrm{UD}_{k_{ud}}\}$.

In particular, all CSs are assumed to reside in a dedicated *cloud domain* that contains no UDs and is typically reachable through long-haul communications (e.g., Domain 4 in Fig. 1). In contrast, each *edge domain* contains a set of UDs and one or multiple co-located ESs, enabling low-latency and trusted intra-domain interactions (e.g., Domain 1-3 in Fig. 1). At each slot, UD m generates a task $\mathcal{T}_n^m$ with a security flag $\iota_n^m \in \{0, 1\}$, where $\iota_n^m = 1$ indicates a *sensitive* task and $\iota_n^m = 0$ indicates a *normal* task. Sensitive tasks are restricted to *within-domain* execution (i.e., local processing at the UD or offloading to ESs with the same domain identity), so that raw sensitive data never traverses domain boundaries. Normal tasks may be executed locally or offloaded to ESs/CSs *across domains*; however, whenever the selected destination is outside the UD's domain, the transmission is protected by lightweight cryptographic encapsulation (ECDH-based key establishment and AES encryption), and the resulting cryptographic overhead and cross-domain security risk are explicitly accounted for in the system-cost model.

We introduce the notion of *domain* to capture heterogeneous administrative boundaries and security requirements. Let $D(\cdot)$ denote the domain identity of an entity. If a UD can communicate with a specific ES (or a gateway to a CS) through a trusted and locally secured channel (e.g., within the same administrative network), we consider them to be in the same domain, i.e., $D(\mathrm{UD}) = D(\mathrm{ES})$ (and similarly for CS access if applicable).

The time horizon is equally divided into $\mathcal{N} = \{1, 2, \ldots, N\}$ slots. At each slot n, each UD $m \in \mathcal{M}$ (with computational capability f^{ud} and energy budget υ^m) generates a security-classified computational task $\mathcal{T}_n^m = \{h_n^m, u^{\mathrm{com}}, u^{\mathrm{enc}}, t^{\max}, \iota_n^m\}$. Here, h_n^m denotes the task size, u^{com} and u^{enc} denote the required CPU cycles per bit for computing and encrypting, $t^{\max}$ denotes the maximum tolerable latency, and ι_n^m denotes the security classification flag.

The CS and ES maintain task queues Q^{cs} and Q^{es}, respectively, with computing capacities f^{cs} and f^{es}. These capacities follow a natural hierarchy: $f^{\mathrm{ud}} \ll f^{\mathrm{es}} \ll f^{\mathrm{cs}}$.

Task $\mathcal{T}_n^m$ is executed locally via proportion α_n^m, or offloaded remotely via proportions β_n^m and γ_n^m to a destination within the feasible remote set Γ_n^m. Depending on the security flag ι_n^m, the remote set Γ_n^m equals $\{\mathrm{ES}_{k_{es}} \in \mathcal{E} \mid D(\mathrm{ES}_{k_{es}}) = D(m)\}$ for sensitive tasks with $\iota_n^m = 1$, and $\mathcal{E} \cup \mathcal{C}$ for normal tasks with $\iota_n^m = 0$. Let $dest \in \{\epsilon_n^{m,dest}, c_n^{m,dest}\}$ denote the selected remote destination. The encryption indicator Π_n^m equals 1 if and only if $\iota_n^m = 0$ and $D(dest) \neq D(m)$, and 0 otherwise.

We adopt partial task offloading. The offloading action for $\mathcal{T}_n^m$ is defined as $a_n^m = \{\alpha_n^m, \beta_n^m, \gamma_n^m, \epsilon_n^{m,\mathrm{dest}}, c_n^{m,\mathrm{dest}}\}$, where α_n^m, β_n^m, and γ_n^m denote the task proportions assigned to local, edge, and cloud execution, respectively, and $\epsilon_n^{m,\mathrm{dest}}$ and $c_n^{m,\mathrm{dest}}$ denote the selected destinations. These variables satisfy the following constraints: $\alpha_n^m + \beta_n^m + \gamma_n^m = 1, \epsilon_n^{m,\mathrm{dest}} \in \mathcal{E}$, $c_n^{m,\mathrm{dest}} \in \mathcal{C}$, $\gamma_n^m \beta_n^m = 0$. Accordingly, the system state at slot n is $s_n = \{Q_n^{\mathrm{cs}}, Q_n^{\mathrm{es}}, \upsilon_n^m, \mathcal{T}_n^m\}$.

B. Communication Model

D2C ($\gamma_n^m \neq 0$) and D2E ($\beta_n^m \neq 0$) links utilize OFDMA with sub-channel bandwidth B to mitigate interference [45]. The transmission rate for mode $\cdot \in \{\mathrm{D2C}, \mathrm{D2E}\}$ is:

$$r_n^{m,\cdot} = B \log_2(1 + p_n^{m,\cdot} g_n^{m,\cdot} / \sigma^2), \tag{1}$$

where $p_n^{m,\cdot}$ and σ^2 are the transmit and noise power, respectively. The channel gain $g_n^{m,\cdot} = 10^{-PL_n^{m,\cdot}/10}$ follows the path loss $PL_n^{m,\cdot} = 128.1 + 37.6 \log_{10}(d_n^{m,\cdot})$ [46], with $d_n^{m,\cdot}$ being the distance to the selected server in km.

C. Computing Model

If $\alpha_n^m \neq 0$, the α_n^m portion of $\mathcal{T}_n^m$ is executed locally at UD m. The corresponding local computing latency is given by

$$t_n^{m,\mathrm{ud}} = \frac{\alpha_n^m h_n^m u^{\mathrm{com}}}{f^{\mathrm{ud}}}. \tag{2}$$

The corresponding energy consumption is represented as

$$e_n^{m,\mathrm{ud}} = \xi (f^{\mathrm{ud}})^2 \alpha_n^m h_n^m u^{\mathrm{com}}, \tag{3}$$

where ξ is the energy efficiency coefficient of UD's chip.

If $\beta_n^m \neq 0$, the β_n^m portion of $\mathcal{T}_n^m$ is offloaded to an ES for execution. The total edge computing latency is given as

$$t_n^{m,\mathrm{D2E}} = t_n^{m,\mathrm{tran}} + t_n^{m,\mathrm{wait}} + t_n^{m,\mathrm{exe}} + t_n^{m,\mathrm{back}}, \tag{4}$$

where $t_n^{m,\mathrm{tran}} = \beta_n^m h_n^m / r_n^{m,\mathrm{D2E}}$ is the D2E transmission time. The queuing delay is denoted by $t_n^{m,\mathrm{wait}} = (\sum_{m' \in \mathcal{M}'} \beta_n^{m'} h_n^{m'} + Q_n^{\mathrm{es}}) u^{\mathrm{com}} / f^{\mathrm{es}}$, which reflects the waiting time caused by tasks already queued at the ES and those arriving earlier from other UDs. Here, $\mathcal{M}' = \{m' \in \mathcal{M} \mid m' \neq$

Authorized licensed use limited to: TU Wien Bibliothek. Downloaded on November 03,2026 at 09:06:24 UTC from IEEE Xplore. Restrictions apply.

$m, \beta_n^{m'} \neq 0, \epsilon_n^{m',dest} = \epsilon_n^{m,dest}, t_n^{m',\text{tran}} < t_n^{m,\text{tran}}\}$ denotes the set of UDs whose tasks are transmitted to the same ES before UD m. The execution time on the ES is $t_n^{m,\text{exe}} = \beta_n^m h_n^m u^{\text{com}}/f^{\text{es}}$, and $t_n^{m,\text{back}} = h_n^{m,\text{back}}/r_n^{m,\text{D2E}}$ is the result feedback time. The energy consumption for edge offloading is given by

$$e_n^{m,\text{D2E}} = p^{\text{tran}}(\beta_n^m h_n^m)/r_n^{m,\text{D2E}}, \tag{5}$$

where p^{tran} is the transmission power.

If $\gamma_n^m \neq 0$, the γ_n^m portion of $\mathcal{T}_n^m$ is offloaded to a CS for execution. Since CS is highly similar to ES, the total cloud computing latency and energy consumption are given as

$$t_n^{m,\text{D2C}} = t_n^{m,\text{tran}} + t_n^{m,\text{wait}} + t_n^{m,\text{exe}} + t_n^{m,\text{back}}, \tag{6}$$

$$e_n^{m,\text{D2C}} = p^{\text{tran}}(\gamma_n^m h_n^m)/r_n^{m,\text{D2C}}. \tag{7}$$

For cross-domain offloading, where $\Pi_n^m = 1$, the cryptographic overhead relies on the offloaded data volume $(\beta_n^m + \gamma_n^m)h_n^m$. The local encryption latency at the UD and the remote decryption latency are defined as $t_n^{m,\text{enc}} = (\beta_n^m + \gamma_n^m)h_n^m u^{\text{enc}}/f^{\text{ud}}$ and $t_n^{m,\text{dec}} = (\beta_n^m + \gamma_n^m)h_n^m u^{\text{enc}}/f^{dest}$, respectively, where $f^{dest} \in \{f^{es}, f^{cs}\}$. The UD's encryption energy consumption is $e_n^{m,\text{enc}} = \xi(f^{\text{ud}})^2(\beta_n^m + \gamma_n^m)h_n^m u^{\text{enc}}$. Thus, the comprehensive end-to-end completion time is:

$$t_n^{m,\cdot} = t_n^{m,\text{tran}} + t_n^{m,\text{wait}} + t_n^{m,\text{exe}} + t_n^{m,\text{back}} + \Pi_n^m(t_n^{m,\text{enc}} + t_n^{m,\text{dec}}). \tag{8}$$

D. System Cost Model

Since $\mathcal{T}_n^m$ can be executed in parallel on the UD, ES, and CS, the completion latency is determined by the bottleneck among active paths. Let $\mathcal{K} = \{\text{ud, D2E, D2C}\}$ be the set of potential execution modes. The total latency $t_n^{m,\mathcal{T}}$ is formulated as:

$$t_n^{m,\mathcal{T}} = \max\left\{t_n^{m,k} \mid k \in \mathcal{K}, \phi_{n,k}^m > 0\right\}, \tag{9a}$$

$$\text{s.t.} \quad t_n^{m,\mathcal{T}} \leq t^{\max}. \tag{9b}$$

where $\phi_{n,k}^m \in \{\alpha_n^m, \beta_n^m, \gamma_n^m\}$ corresponds to the proportions assigned to each mode and $t^{\max}$ is the maximum tolerable latency specified in $\mathcal{T}_n^m$.

The total energy consumption for $\mathcal{T}_n^m$, encompassing all active execution paths, can be given by

$$e_n^m = e_n^{m,\text{ud}} + e_n^{m,\text{D2C}} + e_n^{m,\text{D2E}} + \Pi_n^m e_n^{m,\text{enc}}. \tag{10}$$

For each UD $m \in \mathcal{M}$, the available energy budget can be given by

$$v_n^m = v_{n-1}^m - e_n^m, \tag{11a}$$

$$\text{s.t.} \quad v_n^m \geq 0. \tag{11b}$$

Although ECDH and AES enhance the confidentiality of cross-domain transmissions, potential privacy disclosure risks may still exist. To further mitigate such risks, we encourage reducing cross-domain computation when the overall system cost is comparable. The security cost $\chi_n^{m,\mathcal{T}}$ is defined as $\chi_n^{m,\mathcal{T}} = \Lambda_s \Pi_n^m$, where Λ_s denotes the unit security risk coefficient associated with cross-domain interactions. Accordingly, the total cost $Z_n^{m,\mathcal{T}}$ for $\mathcal{T}_n^m$ is formulated as

$$Z_n^{m,\mathcal{T}} = \chi_n^{m,\mathcal{T}} + \Lambda_t t_n^{m,\mathcal{T}} + \Lambda_e e_n^m, \tag{12}$$

where Λ_t and Λ_e represent the weighting coefficients for time cost and energy consumption, respectively, satisfying $\Lambda_s + \Lambda_t + \Lambda_e = 1$. Hence, the overall system cost c over N time slots is given by

$$c = \frac{1}{NM}\sum_{n=1}^{N}\sum_{m=1}^{M} Z_n^{m,\mathcal{T}}. \tag{13}$$

E. Problem Formulation

Given the DS-MEC model in Section III, at each slot $n \in \mathcal{N}$ each UD $m \in \mathcal{M}$ chooses a partial-offloading action $a_n^m = \{\alpha_n^m, \beta_n^m, \gamma_n^m, \epsilon_n^{m,\text{dest}}, c_n^{m,\text{dest}}\}$. The feasible destination set Γ_n^m and the cross-domain encryption indicator Π_n^m are determined by the domain-security rule described in Section III: sensitive tasks ($\iota_n^m = 1$) must be processed within-domain, while normal tasks ($\iota_n^m = 0$) may be offloaded across domains with $\Pi_n^m = 1$ and incur cryptographic overhead and security cost.

Original Optimization Problem: By (13), the long-term performance objective is to minimize the average system cost $c = \frac{1}{NM}\sum_{n=1}^{N}\sum_{m=1}^{M} Z_n^{m,\mathcal{T}}$, where $Z_n^{m,\mathcal{T}}$ already accounts for latency, energy, and cross-domain security risk. Therefore, the DS-MEC offloading problem can be formulated as

$$\mathbf{P1}: \min_{\{a_n^m\}} \; c = \frac{1}{NM}\sum_{n=1}^{N}\sum_{m=1}^{M} Z_n^{m,\mathcal{T}} \tag{14a}$$

$$\text{s.t. } \alpha_n^m + \beta_n^m + \gamma_n^m = 1, \quad \alpha_n^m, \beta_n^m, \gamma_n^m \in [0,1], \tag{14b}$$

$$\gamma_n^m \, \beta_n^m = 0, \tag{14c}$$

$$\epsilon_n^{m,\text{dest}} \in \Gamma_n^m \cap \mathcal{E} \text{ if } \beta_n^m \neq 0, \tag{14d}$$

$$c_n^{m,\text{dest}} \in \Gamma_n^m \cap \mathcal{C} \text{ if } \gamma_n^m \neq 0, \tag{14e}$$

$$t_n^{m,\mathcal{T}} \leq t^{\max}, \quad \forall n, m, \tag{14f}$$

$$v_n^m \geq 0, \quad \forall n, m, \tag{14g}$$

$$\Pi_n^m \in \{0,1\} \text{ follows domain constraints.} \tag{14h}$$

Problem $\mathbf{P1}$ is a mixed-integer nonlinear program (MINLP) due to the coupled continuous proportions $(\alpha_n^m, \beta_n^m, \gamma_n^m)$ and the discrete destination selection under queueing-and-rate dependent latency/energy expressions, which is consistent with the well-known MINLP nature of MEC offloading formulations in the literature. Moreover, even without the domain-security constraints, MEC offloading with assignment and capacity coupling is NP-hard in general; thus $\mathbf{P1}$ is also NP-hard, and obtaining optimal solutions via classical programming becomes prohibitive in dynamic, heterogeneous DS-MEC settings.

Sequential Decision Reformulation for MARL: Since $\mathbf{P1}$ optimizes the long-term average cost across slots, it naturally matches the objective of a discrete-time sequential decision process. Following the common MEC-to-MDP transformation adopted in recent offloading studies [47], we reformulate $\mathbf{P1}$ as a Markov decision process (MDP) (or equivalently, a decentralized partially observable MDP under decentralized execution) defined by the tuple $\langle s_n, a_n, \mathcal{P}, r, \pi\rangle$:

Authorized licensed use limited to: TU Wien Bibliothek. Downloaded on November 03,2026 at 09:06:24 UTC from IEEE Xplore. Restrictions apply.

State: The global state at slot n is given by $s_n = \{Q_n^{\text{cs}}, Q_n^{\text{es}}, \upsilon_n^m, \mathcal{T}_n^m\}$, which characterizes the queue states of CS and ES, the remaining energy levels, and task descriptors.

Observation: Under decentralized execution, each UD m observes a local view $s_n^m \in s_n$, including its own task $\mathcal{T}_n^m$, remaining energy υ_n^m, and locally accessible server/queue indicators (consistent with CTDE).

Action: Each UD selects $a_n^m \in a_n$ in the form of partial-offloading proportions and destination indices, and the joint action is $a_n = \{a_n^m\}_{m\in\mathcal{M}}$.

Transition: $\mathcal{P}(s_{n+1} \mid s_n, a_n)$ captures the DS-MEC dynamics, including queue evolution, channel-rate induced transmission time, and energy depletion.

Reward: Since MARL maximizes accumulated rewards, we define an instantaneous reward that is negatively correlated with the per-slot system cost, while penalizing constraint violations. A convenient choice is

$$r_n = -\frac{1}{M}\sum_{m\in\mathcal{M}} Z_n^{m,\mathcal{T}} - \omega_t \sum_m \left[t_n^{m,\mathcal{T}} - t^{\max}\right]_+ - \omega_e \sum_m \left[-\upsilon_n^m\right]_+, \quad (15)$$

where $[x]_+ = \max\{x, 0\}$ and $\omega_t, \omega_e \geq 0$ weight the penalties. This design follows the standard practice of converting constrained cost minimization into reward maximization with penalties in MEC offloading.

Finally, the learning objective becomes

$$\mathbf{P2}: \max_{\{\pi^m\}} \mathbb{E}\left[\sum_{n=1}^{N} \delta^{n-1} r_n\right], \quad (16)$$

where $\delta \in (0, 1]$ is the discount factor. In Section IV, we solve **P2** using CTDE-based MARL and further augment r_n with the proposed CMI auxiliary reward to encourage secure and effective cooperation.

F. Threat Model

We consider multi-tiered threats in the DS-MEC system:

First, cross-domain links may be exposed to network-level adversaries that attempt to eavesdrop on or tamper with offloaded task data. To mitigate this risk, sensitive tasks are restricted to local or within-domain edge execution, while normal cross-domain tasks are protected by ECDH-based key establishment and AES encryption.

Second, remote edge/cloud servers are assumed to be semi-trusted: they correctly execute assigned tasks but may attempt to infer private user context or sensitive semantic information from legitimate task/state representations. The DIB module compresses the raw state into a decision-sufficient representation, reducing representation-level information leakage.

Third, stochastic network fluctuations and adversarial perturbations may disturb policy learning and induce spurious correlations among agents. DIB further improves robustness by filtering decision-irrelevant perturbations.

Finally, hardware-level attacks (e.g., side-channel attacks) are excluded from our threat model, as they exploit physical implementation flaws that require fundamentally different countermeasures. Addressing such threats is orthogonal to our work and is left for future extensions.

G. Extension to Multi-Level Security Policies

In practical industrial IoT scenarios, security policies often exhibit a hierarchical structure (e.g., top-secret, confidential, secret, and unclassified), rather than a strict binary classification. We emphasize that the binary flag employed in our DS-MEC model serves as an operational abstraction representing domain-crossing eligibility, which does not limit the framework to simplistic binary systems. Any multi-level security policy can be effectively mapped onto this abstraction via a threshold function without altering the fundamental structure of the optimization problem. For instance, given a network with four security tiers (Levels 1–4), highly sensitive tasks (Level 4) can be strictly mapped to $\iota_n^m = 1$ to enforce intra-domain execution. Conversely, low-sensitivity tasks (Levels 1, 2, 3) are mapped to $\iota_n^m = 0$, permitting cross-domain offloading under cryptographic protection. Under this extension, the unit security risk coefficient Λ_s can be dynamically scaled based on the security tiers to penalize resource access across different administrative boundaries accordingly, thereby preserving the scalability of MITRA-DIB.

IV. Algorithm Design

As illustrated in Fig. 2, MITRA-DIB is developed under the CTDE paradigm for secure cooperative offloading in DS-MEC. The framework mainly consists of a DIB-based representation module, a CMI-based cooperation mechanism, a quality-aware experience buffering strategy, and a multi-agent policy learning process. To clear the mathematical density and provide a structural roadmap of the proposed framework, the key characteristics of each module within MITRA-DIB—including their respective inputs, outputs, training objectives, and core functionalities—are systematically summarized in Table II before delving into technical details.

A. DIB-Based Task Representation Bottleneck

The raw state contains perturbations unrelated to action decisions, such as transient fast-fading channel states, random load fluctuations, and redundant components weakly correlated with the value of action. These amplify the variance of mutual-information estimates and disturb multi-agent coordination. We propose a decision-oriented information filter: first, it learns a task representation z, ensuring that each agent can select offloading actions based on a more stable and decision-relevant representation, thereby reducing instability caused by noise and spurious correlations; then, it uses this representation to guide policy learning and advantage shaping. DIBEncoder optimizes the variational approximation of the classic information bottleneck trade-off, which is achieved through the joint constraint of the actor/critic learning objective acting on the encoder parameters ψ and the KL bottleneck.

Authorized licensed use limited to: TU Wien Bibliothek. Downloaded on November 03,2026 at 09:06:24 UTC from IEEE Xplore. Restrictions apply.

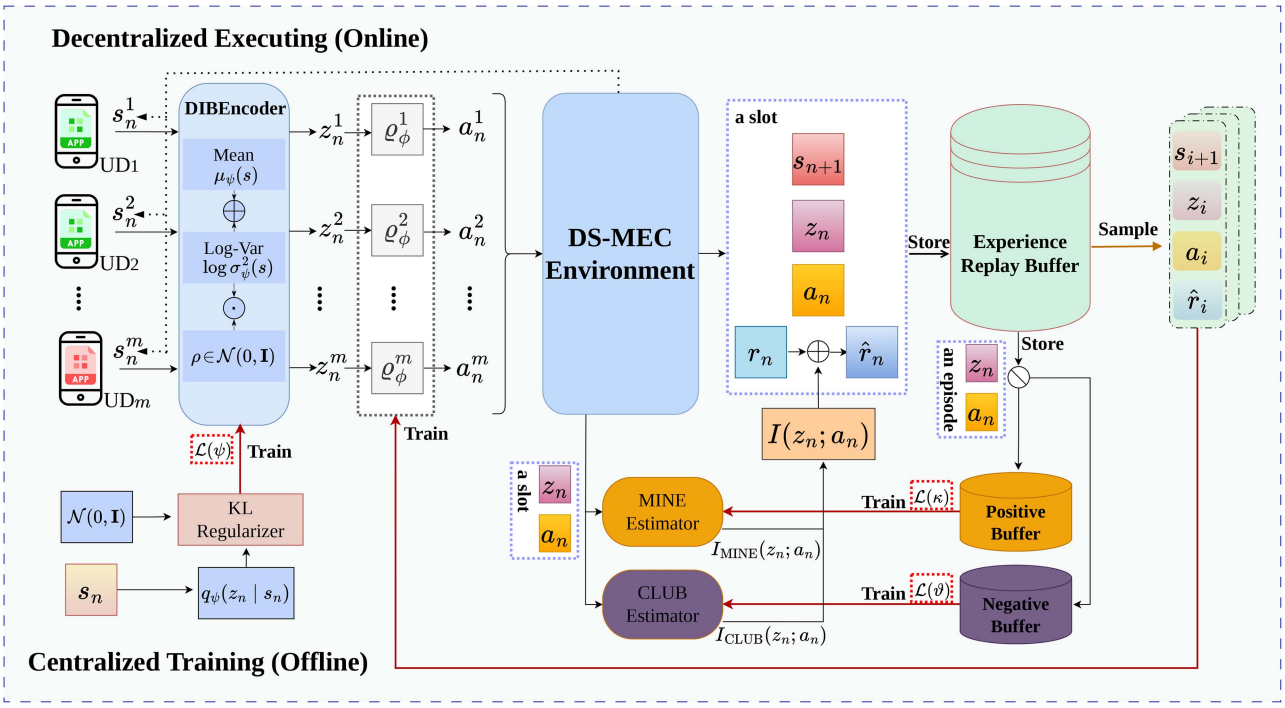


Fig. 2. The algorithm overview of MITRA-DIB.

TABLE II
OPERATIONAL SUMMARY OF MODULES IN MITRA-DIB

Module Name	Input Variables	Output Signals	Training Objective	Primary Functionality
DIB Encoder	Raw local state s_n^m	Task-adaptive representation z_n^m	Minimize KL divergence bottleneck loss $\mathcal{L}(\psi)$	Suppresses high-variance, decision-irrelevant environmental perturbations.
CMI Estimators (MINE & CLUB)	Samples $(z_i,a_i),(z_i,a_j)$ from $\mathcal{D}^+$ and $\mathcal{D}^-$	MI lower/upper bounds $(I_{\mathrm{MINE}}, I_{\mathrm{CLUB}})$	Jointly optimize $\mathcal{L}(\kappa)$ (maximize MINE) and $\mathcal{L}(\vartheta)$ (minimize CLUB)	Computes the composite Credence Mutual Information to provide coordination signals.
Positive/Negative Buffers ($\mathcal{D}^+$&$\mathcal{D}^-$)	Episode trajectories ζ' and cumulative rewards r^ζ	Segregated experience mini-batches	–	Separates cooperative behaviors that should be reinforced from low-quality behaviors that should be attenuated.
Decentralized Actor	Local compressed representation z_n^m	Offloading action a_n^m	Maximize the expected augmented return under CTDE	Generates local offloading proportions and destination decisions for each UD during decentralized execution.
Centralized Critic	Joint representations z_n, joint actions a_n, and augmented reward $\hat{r}_n$	Centralized critic value V_θ	Minimize the Bellman error under the augmented reward $\hat{r}_n$	Evaluates joint actions during centralized training and guides decentralized actors.
Security Mechanism	Cross-domain normal-task data and encryption indicator Π_n^m	decrypted task data	–	Protects cross-domain transmissions using ECDH-based key establishment and AES encryption.

Specifically, we design a stochastic encoder that maps s to a stochastic representation and constrain it via a controllable KL regularizer to realize an information bottleneck:

$$z = \mathrm{DIBEncoder}(s) = \mu_\psi(s) + \exp\left(\frac{1}{2}\log\sigma^2_\psi(s)\right)\odot\rho,$$
$$\log\sigma^2_\psi(s) = 4\tanh(W_{\log}g_\psi(s)+b_{\log}), \qquad (17)$$

where $\odot$ denotes element-wise multiplication and $\rho\in\mathcal{N}(0,\mathbf{I})$. The encoder is parameterized by the mean $\mu_\psi(s)$ and the log-variance $\log\sigma^2_\psi(s)$. Here, $g_\psi(s)$ denotes a neural feature extractor of state s, and $W_{\log}$ and $b_{\log}$ are the learnable parameters of the log-variance head. The scaling $4\tanh(\cdot)$ bounds $\log\sigma^2_\psi(s)$ within $(-4,4)$ to avoid extreme variances and stabilize stochastic encoding (and the subsequent MI estimation). The loss function:

$$\mathcal{L}(\psi) = \mathrm{KL}\big(\mathcal{N}\left(\mu_\psi(s),\,\mathrm{diag}(\exp(\log\sigma^2_\psi(s)))\right)\,\big\|\,\mathcal{N}(0,\mathbf{I})\big). \qquad (18)$$

Authorized licensed use limited to: TU Wien Bibliothek. Downloaded on November 03,2026 at 09:06:24 UTC from IEEE Xplore. Restrictions apply.

DIBEncoder constrains the difference between the encoding distribution and the prior distribution through a KL bottleneck regularization term, which can serve as a controllable proxy for the information capacity of the representation. The degree of compression is determined by the trade-off between decision performance and compression regularization: a stronger KL constraint encourages the encoder to discard more redundant or decision-irrelevant information, whereas a weaker constraint allows more state features to be retained.

During the CMI regularization and advantage-shaping stages, we treat z as a constant to avoid representation collapse and non-stationarity driven by the estimators or target values.

B. Credence Mutual Information

As a quality-aware collaborative signal, CMI guides agents to learn which interaction relationships should be repeated or avoided, thereby improving the quality of collaboration under decentralized execution. We utilize the mutual information between the joint action a and the task representation z, which is formally expressed as follows:

$$I(z;a) = H(a) - H(a^m \mid z) - H(a^{-m} \mid a^m, z), \tag{19}$$

where $H(\cdot)$ denotes entropy, $H(X \mid Y)$ denotes the conditional entropy of X given Y, a^m denotes the action of an arbitrary UD m, and a^{-m} denotes the joint action of all UDs except m. As shown in (19), $I(z;a)$ can be decomposed into three distinct terms: (1) $H(a)$ measures the exploration capacity over joint actions. Maximizing $H(a)$ promotes diverse offloading policy and prevents policy collapse; (2) $-H(a^m \mid z)$ describes each UD's confidence in its own decision given the task representation z; (3) $-H(a^{-m} \mid a^m, z)$ captures a UD's belief about the offloading behaviors of other UDs, given the task representation and its own action, thereby implicitly characterizing the correlation of actions among UDs.

C. Positive–Negative Sample Buffers

In MEC scenarios, due to the stochasticity of system dynamics and the fluctuations arising from the learning process of UD offloading policies, the offloading experiences (s_n, a_n, r_n, s_{n+1}) consisting of offloading actions with diverse r_n are gathered during MARL. Essentially, CMI characterizes the dependency strength between the task representation and the multi-agent joint offloading actions. Therefore, high mutual information can only indicate that there is a strong correlation and high-confidence coordination between agents based on task features, which is to say a certain 'stable coordination', but this coordination is not necessarily good. If all offloading experiences were stored directly in a replay buffer and sampled for the sole purpose of maximizing $I(z;a)$, experiences originating from both optimal and suboptimal policies could be uniformly amplified. Motivated by prior work on mutual-information-regularized MARL [48], we further partition offloading experiences into high-quality and low-quality experiences, which are stored in the positive sample buffer $\mathcal{D}^+$ and the negative sample buffer $\mathcal{D}^-$, respectively. Positive samples correspond to trajectories with higher cumulative rewards; maximizing CMI on these samples can reinforce beneficial coordination. Conversely, negative samples correspond to low-quality trajectories; minimizing CMI on these negative samples can weaken such undesirable coordination and prevent the policy from imitating sub-optimal coordination patterns. Concretely, we determine the quality of an episode ζ by its cumulative reward r^ζ. Let r_{low} denote the minimum reward currently stored in $\mathcal{D}^+$. When a segment's $r^\zeta > r_{low}$, its experience is stored in $\mathcal{D}^+$ until $\mathcal{D}^+$ is full; if $\mathcal{D}^+$ is full, the experience with the lowest reward is dequeued. Conversely, when a segment's $r^\zeta < r_{low}$, its experience is inserted into $\mathcal{D}^-$ using a first-in-first-out (FIFO) policy, since recent low-quality behaviors should be avoided as a priority.

Considering the intractability of directly computing $I(z;a)$, especially in the case of continuous variables with unknown distributions, we adopt MINE/CLUB estimators to compute the CMI auxiliary reward. MINE estimates a lower bound of $I(z;a)$ used for maximization, and CLUB estimates an upper bound of $I(z;a)$ used for minimization, based on $\mathcal{D}^+$ and $\mathcal{D}^-$, respectively.

D. MI Bounds

Based on jointly sampled pairs (z_i, a_i) from $\mathcal{D}^+$ and independently sampled actions a_j from $\mathcal{D}^+$, the lower bound of $I(z;a)$ is approximated by MINE as follows:

$$I(z;a) \geq I_{\text{MINE}}(z;a) = \sup_\kappa \underbrace{\mathbb{E}_{\mathbb{P}_{ZA}}[\log \sigma(\Upsilon_\kappa(z_i, a_i))] + \mathbb{E}_{\mathbb{P}_Z \otimes \mathbb{P}_A}[\log \sigma(-\Upsilon_\kappa(z_i, a_j))]}_{-\mathcal{L}(\kappa)}, \tag{20}$$

where $\mathbb{P}_{ZA}$ is the joint distribution from which the matched pairs (z_i, a_i) are sampled, while (z_i, a_j) is a randomly paired sample, approximately drawn from the product of marginals $\mathbb{P}_Z \otimes \mathbb{P}_A$. Υ_κ is a neural network with parameters κ and $\sigma(x) = \frac{1}{1+e^{-x}}$ denotes the sigmoid function.

Similarly, CLUB approximates an upper bound of $I(z;a)$:

$$I(z;a) \leq I_{\text{CLUB}}(z;a) = \underbrace{\mathbb{E}_{\mathbb{P}_{ZA}}[\log \Xi_\vartheta(z_i \mid a_i)]}_{-\mathcal{L}(\vartheta)} - \mathbb{E}_{\mathbb{P}_Z \otimes \mathbb{P}_A}[\log \Xi_\vartheta(z_i \mid a_j)], \tag{21}$$

where Ξ_ϑ is a neural network with parameters ϑ, which provides a variational approximation to the true conditional distribution. The joint distribution and marginals distribution are consistently sampled as in MINE, but are based on $\mathcal{D}^-$.

To enable MINE and CLUB to progressively guide agent behavior (toward or away from superior and inferior behaviors) and to provide effective quantitative signals, we define

$$I(z;a) = \lambda_1 I_{\text{MINE}}(z;a) - \lambda_2 I_{\text{CLUB}}(z;a), \tag{22}$$

where λ_1 and λ_2 are hyperparameters that balance the composite mutual-information reward and the original reward.

The stability of the MINE/CLUB estimator is improved through prior noise filtering by the DIB module, warm-up after sufficient samples, and periodic training. Furthermore, when calculating the CMI reward, the task representation is treated as a fixed quantity, which helps reduce the non-stationarity caused by the estimator and prevent representation collapse.

Authorized licensed use limited to: TU Wien Bibliothek. Downloaded on November 03,2026 at 09:06:24 UTC from IEEE Xplore. Restrictions apply.

E. Centralized Training With MI-Auxiliary Reward

We primarily adopt the CTDE paradigm to learn collaborative offloaders. During the centralized training phase, the learning of each UD's offloading policy ϱ_ϕ^m (parameterized by ϕ) is guided by a central critic V_θ (parameterized by θ), which can access the states and actions of the other UDs. The central critic is required exclusively during offline training, plays no role in online offloading decisions, and can be deployed within the cloud domain. The backbone of our model is MADDPG and we incorporate the $I(z;a)$ computed by (22) as an auxiliary reward term to the original reward in order to suppress redundant coupling and encourage necessary correlations at the value-learning level. The objective function becomes

$$J(V) = \sum_{n=1}^{N} \mathbb{E}\delta^{n-1} \left[r(s_n, a_n) + I(z_n; a_n) \right], \tag{23}$$

where δ is the discount factor for rewards. The raw reward $r(s_n, a_n)$ produced at time slot n is the positively/negatively correlated result of the system cost c and the constraints (9b) and (11b) after executing action a_n. The training losses of the MINE and CLUB neural estimators are $\mathcal{L}(\kappa)$ and $\mathcal{L}(\vartheta)$, respectively. Therefore, the total loss function is

$$\mathcal{L}_{\text{total}}(\kappa, \vartheta, \psi) \;=\; \mathcal{L}(\kappa) + \mathcal{L}(\vartheta) + \lambda_3 \mathcal{L}(\psi), \tag{24}$$

where λ_3 is a tunable hyperparameter that controls the strength of DIB compression. A larger λ_3 imposes stronger information compression, reducing redundant information in the representation, but may lead to the loss of decision-relevant features; a smaller λ_3, by contrast, preserves more original state information, but may introduce high-variance perturbations.

F. Security Mechanism

To enhance the confidentiality of data transmitted cross-domain, we novelly introduce a lightweight key exchange protocol ECDH, to share a fresh key between the task offload party UD and the potential task execution party that is outside the domain of UD. Concretely, when $\Pi_n^m = 1$, UD m and the potential selected server ES/CS run the ECDH protocol to obtain a shared secret key K. Then, the task data TD will be encrypted by the UD with (25), and the corresponding ciphertext $\text{Cipher}_{\text{TD}}$ will be transmitted to the potential selected server ES/CS, which will execute (26) to obtain the plaintext.

$$\text{Cipher}_{\text{TD}} \leftarrow \text{AES.Enc}(K, \text{TD}), \tag{25}$$

$$\text{TD} \leftarrow \text{AES.Dec}(K, \text{Cipher}_{\text{TD}}). \tag{26}$$

Based on the above security mechanism, our MITRA-DIB demonstrates favorable scalability. The joining or departure of any entity does not impact the rest of the system, because cryptographic keys are independent and can be renegotiated using ECDH protocol.

G. Overall Algorithm

Algorithm 1 outlines the CTDE-based **MITRA-DIB** training. It initializes centralized critics, decentralized actors, the DIBEncoder, and MI estimators (MINE/CLUB), supported by a standard buffer $\mathcal{B}$ and quality-aware $\mathcal{D}^+/\mathcal{D}^-$ buffers.

During each slot n, each UD m maps its local state s_n^m to a representation z_n^m to sample an action a_n^m. The environment enforces domain-security rules: sensitive tasks are restricted to within-domain execution, while normal tasks undergo encrypted cross-domain offloading. A CMI-augmented reward $\hat{r}_n = r_n + \lambda_1 I_{\text{MINE}} - \lambda_2 I_{\text{CLUB}}$ is computed to provide coordination signals, with transitions stored in $\mathcal{B}$. Post-episode, *experience shaping* categorizes trajectories into $\mathcal{D}^+$ or $\mathcal{D}^-$ based on their cumulative rewards. The update process adopts an alternating pipeline, MINE maximizes MI on $\mathcal{D}^+$, CLUB minimizes

Algorithm 1: The training algorithm of MITRA-DIB.

Initialize critic parameters θ^m of V_θ and actor parameters ϕ^m of offloaders ϱ_ϕ, and corresponding target networks θ'^m and ϕ'^m, $m \in \mathcal{M}$
Initialize DIBEncoder parameters ψ, MINE estimator parameters κ, and CLUB estimator parameters ϑ
Initialize replay buffer $\mathcal{B}$, positive buffer $\mathcal{D}^+$, and negative buffer $\mathcal{D}^-$
repeat
Initialize the DS-MEC environment Env (with the domain-security rule and encryption indicator Π_n^m)
Initialize an empty episode trajectory $\zeta' \leftarrow \emptyset$
for each slot n in episode ζ **do**
Each UD m observes its local state s_n^m (consistent with CTDE)
Compute task representation $z_n^m \leftarrow \text{DIBEncoder}_\psi(s_n^m)$
Each UD samples an offloading action: $a_n^m \sim \varrho_\phi^m(\cdot \mid z_n^m)$
(i) if $\iota_n^m = 1$, offloading is restricted to the domain that m belongs to
(ii) if $\iota_n^m = 0$, cross-domain offloading is allowed, but it must be encrypted and will incur additional overhead
Env executes the joint action $a_n = \{a_n^m\}_{m \in \mathcal{M}}$ and returns (r_n, s_{n+1})
Compute MI-auxiliary reward $I(z_n; a_n) \leftarrow \lambda_1 I_{\text{MINE},\kappa}(z_n; a_n) - \lambda_2 I_{\text{CLUB},\vartheta}(z_n; a_n)$
Set augmented reward $\hat{r}_n \leftarrow r_n + I(z_n; a_n)$
Store transition $(z_n, a_n, \hat{r}_n, s_{n+1})$ into $\mathcal{B}$
end for
Compute the episode cumulative reward r^ζ
Let r_{low} be the minimum episode reward currently stored in $\mathcal{D}^+$ (or $-\infty$ if $\mathcal{D}^+$ is empty)
if $r^\zeta > r_{\text{low}}$, store $\zeta' = \{z_n, a_n\}_{n=1}^N$ into $\mathcal{D}^+$ and evict the lowest-score episode if full;
else store $\zeta' = \{z_n, a_n\}_{n=1}^N$ into $\mathcal{D}^-$ using FIFO.
Periodically update κ by maximizing $I_{\text{MINE}}(z; a)$ using samples from $\mathcal{D}^+$ (i.e., minimizing $\mathcal{L}(\kappa)$)
Periodically update ϑ by minimizing $I_{\text{CLUB}}(z; a)$ using samples from $\mathcal{D}^-$ (i.e., minimizing $\mathcal{L}(\vartheta)$)
Periodically update ψ by minimizing the KL bottleneck loss $\mathcal{L}(\psi)$ (Decision Information Bottleneck)
For each $m \in \mathcal{M}$, update critic θ^m and actors ϕ^m using samples from $\mathcal{B}$ (CTDE with augmented reward $\hat{r}_n$)
For each $m \in \mathcal{M}$, periodically update target parameters θ'^m and ϕ'^m
until convergence
return $\{\varrho_\phi^m\}_{m \in \mathcal{M}}$

Authorized licensed use limited to: TU Wien Bibliothek. Downloaded on November 03,2026 at 09:06:24 UTC from IEEE Xplore. Restrictions apply.

TABLE III
PRIMARY MEC PARAMETERS

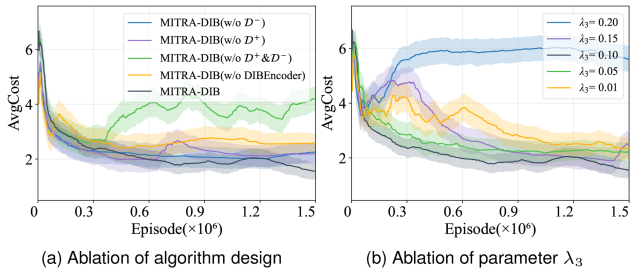
Parameter	Description	Value
f^{ud}	UD computing capacity	1 GHz
u^{com}	CPU cycles/bit	900
ξ	Energy efficiency coefficient	10^{-27}
σ^2	White Gaussian noise	-100 dBm
N	Number of slots	10
$\lvert\mathcal{B}\rvert$	Training batch size	32
B	Bandwidth	10 MHz
Δt	Slot length	1 s
$t^{\max}$	maximum task delay	0.9 s
p^{tran}	UD transmitting power	0.1 W

MI on $\mathcal{D}^-$, and the DIBEncoder is optimized via KL bottleneck loss to stabilize representations. Finally, critics and actors are updated using mini-batches from $\mathcal{B}$ with soft target updates until convergence, yielding optimized decentralized policies $\{\varrho_\phi^m\}_{m\in\mathcal{M}}$. It is worth noting that MINE/CLUB does not directly participate in the actor's gradient backpropagation, but instead estimates CMI first and feeds it as a detached auxiliary reward into the subsequent critic/actor update process.

V. Simulation Results

A. Experimental Settings

The MEC system was implemented in Python and PyTorch. All experiments were trained and executed on a single server equipped with an Intel Xeon(R) Gold 6430 CPU and an NVIDIA RTX 4090 GPU. All experiments are repeated with five different random seeds, and the results are reported in terms of the mean and standard deviation. Our simulation experiments consider four different MEC configurations: (1) 1CS-5ES-12UD (CS_1: 32 GHz, ES_1: 6 GHz, ES_2: 8 GHz, ES_3: 11 GHz, ES_4: 14 GHz, ES_5: 19 GHz). It is divided into six domains (CS_1; ES_1 and two UDs; ES_2 and two UDs; ES_3 and two UDs; ES_4 and three UDs; ES_5 and three UDs). (2) 1CS-3ES-7UD (CS_1: 28 GHz, ES_1 : 6 GHz, ES_2: 11 GHz, ES_3: 19 GHz). It is divided into four domains (CS_1; ES_1 and two UDs; ES_2 and two UDs; ES_3 and three UDs). (3) 1CS-2ES-5UD (CS_1: 26 GHz, ES_1: 10 GHz, ES_2: 19 GHz). It is divided into three domains (CS_1; ES_1 and two UDs; ES_2 and three UDs). (4) 1CS-2ES-3UD (CS_1: 22 GHz, ES_1: 8 GHz, ES_2: 14 GHz). It is divided into two domains (CS_1; ES_1, ES_2 and three UDs). For the 1CS-5ES-12UD scenario, the task sizes of UDs 1–12 are modeled as Gaussian random variables with means of 7, 6.5, 6, 5.5, 5.5, 5, 5, 4.5, 4.5, 4, 3.5, and 3 kb, respectively, and a variance of 1 kb^2. Each UD has a certain probability of generating sensitive tasks. Unless otherwise specified, the main DS-MEC parameters are listed in Table III. The discount factor δ is 0.99, the sizes of the positive and negative sample buffers are 1000, and the hyperparameters are $\lambda_1 = 3$, $\lambda_2 = 1$, and $\lambda_3 = 0.1$. The optimizer used is Adam, where the learning rates for the actor, MINE, and CLUB are 1e-4, and the learning rates for the critic and DIB encoder are 1e-3. For baseline methods, except for algorithm-specific settings, all baselines adopt the same training environment and evaluation protocols as MITRA-DIB.



(a) Ablation of algorithm design (b) Ablation of parameter λ_3

Fig. 3. Ablation experimental results.

The baselines comprise four representative state-of-the-art MARL algorithms commonly used in recent MEC studies: MADDPG [37], FACMAC [38], HAPPO [39], and MATD3 [49]. All algorithms follow the CTDE paradigm and performance evaluation metrics fall into two categories: (1) The first is the average system cost (AvgCost) c defined in (13) of all successful episodes that represents the quality of service of all tasks in MEC. (2) The second is the task success rate, calculated by $\sum_n \mathbb{I}(t_n^{m,\zeta} < t^{\max} \mid \forall m \in \mathcal{M})/N$, that indicates the degree to which all tasks are completed during one episode ζ.

B. Ablation Study

To validate the contribution of each module in MITRA-DIB, we conducted systematic ablation experiments on the 1CS-3ES-7UD scenario. As shown in Fig. 3(a), performance degradation is most severe when both positive and negative sample buffers are removed, highlighting the critical role of the CMI-guided coordination mechanism in enabling effective collaboration. Removing the DIBEncoder significantly raises the learning curve and increased its volatility, indicating that DIB plays a crucial role in filtering redundancy and stabilizing policy learning. Finally, removing either the positive or negative sample buffers alone harms performance, proving that encouraging high-quality actions and suppressing low-quality actions are both essential for optimizing policy learning. Fig. 3(b) illustrates results regarding hyperparameter λ_3 in the 1CS-3ES-7UD scenario. When λ_3 is large, AvgCost increases significantly. This indicates that excessive information compression leads to the loss of features critical for decision-making, making policy learning difficult. When λ_3 is small, excessive redundancy is retained in the state representation, disrupting effective collaboration among agents.

C. Main Experiment Analysis

We evaluate MITRA-DIB against four representative CTDE-based MARL baselines under four DS-MEC configurations with increasing degrees of heterogeneity and multi-domain coupling: 1CS-5ES-12UD, 1CS-3ES-7UD, 1CS-2ES-5UD, and 1CS-2ES-3UD. The evaluation focuses on three complementary metrics: (i) the average system cost (AvgCost) defined in (13), (ii) the training reward defined in (15) (negatively correlated with the per-slot cost while penalizing constraint violations), and (iii) the task success rate summarized in Table IV. These results jointly characterize solution quality, learning stability,

Authorized licensed use limited to: TU Wien Bibliothek. Downloaded on November 03,2026 at 09:06:24 UTC from IEEE Xplore. Restrictions apply.

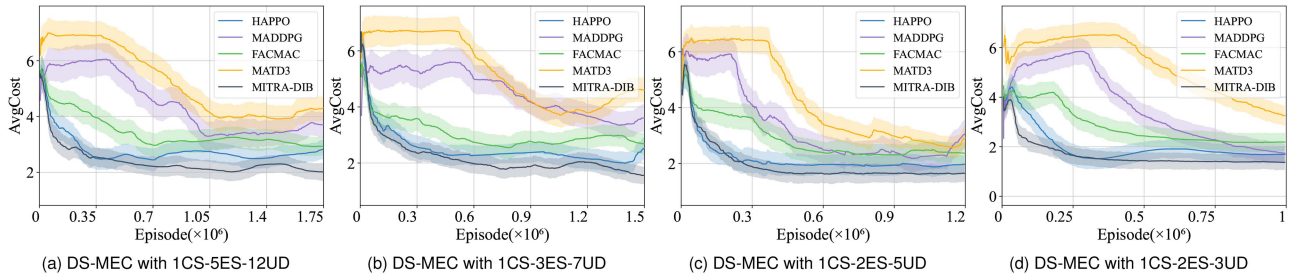


Fig. 4. AvgCost during training in four DS-MEC scenarios.

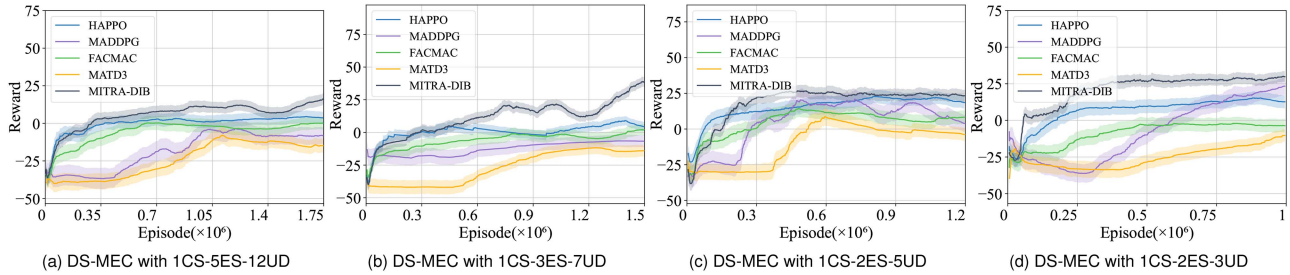


Fig. 5. Reward during training in four DS-MEC scenarios.

TABLE IV
TASK SUCCESS RATE OF ALL EVALUATED ALGORITHMS. THE BEST RESULTS ARE HIGHLIGHTED IN **BOLD**. VALUES ARE IN %.

DS-MEC	MADDPG	FACMAC	HAPPO	MATD3	MITRA-DIB
1CS-5ES-12UD	21.37 ± 1.13	37.32 ± 1.48	55.40 ± 2.14	19.78 ± 1.12	79.53 ± 2.14
1CS-3ES-7UD	31.87 ± 1.53	42.41 ± 1.23	68.14 ± 1.85	27.88 ± 0.92	94.25 ± 1.91
1CS-2ES-5UD	52.78 ± 1.27	39.51 ± 1.31	61.39 ± 1.69	25.16 ± 1.10	88.38 ± 2.12
1CS-2ES-3UD	62.23 ± 1.53	47.28 ± 1.56	67.57 ± 1.33	41.14 ± 1.55	97.91 ± 2.09

and feasibility satisfaction under the domain-security rule and cryptographic overhead.

Cost optimality and convergence: Fig. 4 reports AvgCost trajectories during training. Across all four scenarios, MITRA-DIB converges to the lowest cost region and exhibits a faster reduction in AvgCost compared with baselines. In the most challenging setting (1CS-5ES-12UD), the baselines present pronounced oscillations and long transient plateaus, indicating sensitivity to non-stationary multi-agent dynamics and stochastic perturbations in queues/channels. By contrast, MITRA-DIB reduces the cost steadily and maintains a comparatively smooth learning curve. This behavior is consistent with the intended effects of the DIBEncoder: by compressing the raw state into a decision-sufficient representation, DIB suppresses high-variance, decision-irrelevant perturbations that otherwise induce spurious correlations and unstable gradients in MARL optimization. Meanwhile, the CMI-guided auxiliary reward provides an explicit coordination signal that complements the global cost-based reward and improves credit alignment among decentralized offloaders, which is especially beneficial when cross-domain choices incur additional security costs and cryptographic latency/energy overheads.

Reward dynamics and training stability: Fig. 5 further corroborates the above findings from the reward perspective. MITRA-DIB achieves consistently higher rewards during training and shows improved stability, whereas the baselines often suffer from high-variance reward fluctuations and slow recovery from suboptimal coordination regimes. Notably, HAPPO demonstrates relatively competitive reward behavior in some cases, but it remains inferior in convergence quality and robustness under larger-scale coupling (e.g., 1CS-5ES-12UD). The superior reward profiles of MITRA-DIB reflect its ability to (i) reduce overall system cost while respecting latency/energy constraints through the penalty design in (15), and (ii) avoid amplifying detrimental inter-agent correlations via the positive/negative experience buffers that respectively encourage high-quality joint behaviors and suppress low-quality ones.

Feasibility and service reliability: Table IV summarizes task success rates in four DS-MEC scenarios. MITRA-DIB yields the best success rate in all settings, achieving **79.53%**, **94.25%**, **88.38%**, and **97.91%**, respectively. This indicates that MITRA-DIB not only reduces AvgCost but also more reliably satisfies the end-to-end latency constraint (9b) (and implicitly the energy feasibility (11b)) under the domain-security rule. The gap is most pronounced in the largest scenario, where the higher degree of resource heterogeneity and multi-domain interactions makes coordination more difficult. Overall, these results validate that MITRA-DIB can learn secure and cooperative offloading strategies that generalize across DS-MEC scales, balancing efficiency (low AvgCost), stability (smooth learning curves), and reliability (high success rate) under cross-domain security constraints.

D. Visual Analytics

As shown in Fig. 6, to better illustrate how the domain-security rule affects practical offloading behaviors, we visualize the execution-location distribution of *Normal* and *Sensitive* tasks across four destinations: (i) Local execution, (ii) Within-domain

Authorized licensed use limited to: TU Wien Bibliothek. Downloaded on November 03,2026 at 09:06:24 UTC from IEEE Xplore. Restrictions apply.

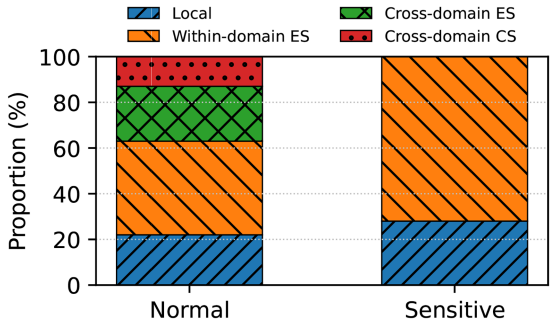


Fig. 6. Execution-location distribution for Normal and Sensitive tasks.

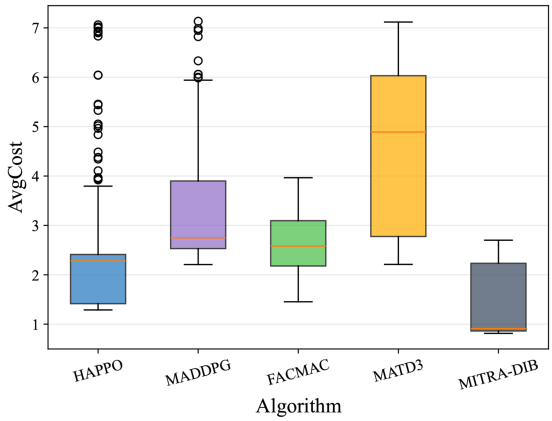


Fig. 7. Distribution of AvgCost over the last 200 training episodes in the 1CS-3ES-7UD DS-MEC scenario.

ES, (iii) Cross-domain ES, and (iv) Cross-domain CS. As shown, Normal tasks are distributed across all destinations, with a substantial portion executed at within-domain ESs and non-negligible cross-domain offloading (ES/CS). In contrast, Sensitive tasks are fully constrained to within-domain processing, with all offloaded workload handled by within-domain ESs and the remainder executed locally, consistent with the enforced security constraint.

To further examine the steady-state performance after convergence, Fig. 7 visualizes the AvgCost distribution of five algorithms over the last 200 training episodes in the 1CS-3ES-7UD scenario. It can be observed that **MITRA-DIB** achieves the lowest median AvgCost and a relatively compact interquartile range, indicating superior cost optimality. In contrast, the median and interquartile range of MATD3 are both significantly higher. Although HAPPO has lower costs for some samples, it contains many high-cost outliers. MADDPG has many high-cost outliers, while the fluctuation of FACMAC is relatively smaller, but its cost level is still higher than that of MITRA-DIB. Overall, this visualization complements the learning-curve evidence in Fig. 4 by confirming that, in the converged regime, MITRA-DIB sustains a lower and more reliable system-cost level under the domain-security rule and cryptographic overhead.

E. Privacy, Robustness, and Decision Sufficiency of DIB

To verify whether the task representation z still retains the information required for making offloading decisions, we conducted a decision sufficiency probe experiment in Table V. After training the DIB-enhanced policy, we froze the learned encoder and trained lightweight probe networks on held-out trajectories. These probes used different inputs to predict the Q-values estimated by the critic. The results show that the compressed representation retains the decision-relevant information required by the value function.

TABLE V
Q-VALUE PREDICTION PERFORMANCE WITH DIFFERENT INPUTS

Input	Q-value prediction R^2
Raw state s	0.9995 ± 0.0015
Task representation z	0.9982 ± 0.0018
Random projection	0.9331 ± 0.0026

TABLE VI
ATTRIBUTE INFERENCE ATTACKS AND BOUNDED-PERTURBATION EXPERIMENTS UNDER THE 1CS-2ES-3UD SCENARIO

Metric	MITRA-DIB	MITRA-DIB(w/o DIB)
Attribute inference accuracy	$68.15\% \pm 5.71\%$	$96.18\% \pm 4.27\%$
Task success rate	$90.89\% \pm 4.63\%$	$73.16\% \pm 4.40\%$

TABLE VII
PARAMETER COUNTS AND TRAINING-PHASE LATENCY MEASUREMENTS

Incorporating module	Relative Increase	Parameter Count
DIBEncoder	$14.39\% \pm 0.12\%$	57,344
MINE	$27.54\% \pm 0.09\%$	2,560
CLUB	$14.58\% \pm 0.07\%$	5,084
DIBEncoder + MINE + CLUB	$60.13\% \pm 0.23\%$	64,988

To validate the ability of the DIB module to prevent representation-level privacy leakage, we trained a neural network as an eavesdropping attacker to predict a class-balanced binary sensitive attribute from the representations available during agent interactions. To avoid trajectory leakage, we partition the trajectories at the round level and split them into training, validation, and test sets at a ratio of 6:2:2. Meanwhile, during the testing phase, we impose bounded perturbations with a magnitude of 0.1 on the channel gains, task sizes, and server queue states to evaluate the stability of the system in the presence of malicious interference or channel fluctuations. As shown in Table VI, DIB can reduce information exposure and improve the model's performance under observation perturbations.

F. Overhead Analysis

We analyzed the overhead of the DIBEncoder and MINE/CLUB estimators. Estimators are only used during the training phase for mutual information reward shaping. During inference, none of them participate in online decision-making, and therefore introduce no additional overhead. In Table VII, we provide the parameter counts and measure the latency of a single training update with a batch size of 1024 in the 1CS-3ES-7UD scenario.

It should be noted that the 60.13% increase in additional latency is a single instantaneous training overhead, rather than the total overhead of the entire training process. MINE/CLUB updates are executed periodically, triggered once every 100 timesteps by default. In addition, when the number of samples in the good buffer is less than 500, the algorithm degrades to

Authorized licensed use limited to: TU Wien Bibliothek. Downloaded on November 03,2026 at 09:06:24 UTC from IEEE Xplore. Restrictions apply.

TABLE VIII
CRYPTOGRAPHIC OVERHEAD COMPARISON

Scheme	Computation	Communication
MITRA-DIB	1.374 ms	1,000,224 bits
MADDPG with Paillier encryption	5.379 s	2,003,000 bits

a normal actor-critic update. Therefore, after amortizing it over the entire training process, the total training overhead is far lower than 60.13%.

In Table VIII, we compare MITRA-DIB with a reinforcement learning baseline (MADDPG) based on partially homomorphic encryption [50]. We consider Paillier additive homomorphic encryption, which allows aggregation within the encrypted domain but introduces modular exponentiation computation overhead and ciphertext expansion.

The overhead is estimated under the same 1,000,000 bits plaintext data volume. The computation overhead includes encryption, decryption, homomorphic aggregation, and amortized ECDH key exchange. The communication overhead denotes encrypted traffic volume. The results indicate that compared with the lightweight AES/ECDH path used in MITRA-DIB, the partially homomorphic encryption baseline incurs significantly higher computation overhead and Communication overhead, which is costly for delay-sensitive MEC offloading.

VI. CONCLUSION

MITRA-DIB is a multi-agent reinforcement learning framework for secure and cooperative task offloading in Multi-access Edge Computing. It tackles both data-security risks in cross-domain transmission and coordination inefficiencies under decentralized execution. To improve robustness and stability, we introduce DIB that suppresses decision-irrelevant perturbations and redundant semantics, thereby mitigating spurious correlations and avoiding unnecessary inter-agent coupling. The proposed CMI auxiliary reward provides informative coordination signals during centralized training with decentralized execution. In addition, a domain-security rule enforces within-domain execution for sensitive tasks and secures cross-domain transmission for normal tasks through lightweight cryptography primitives. Future work will explore extending MITRA-DIB to highly dynamic and mobile settings, such as scenarios with time-varying domain membership and fluctuating numbers of user devices.

REFERENCES

[1] Z. Xiao, J. Shu, H. Jiang, G. Min, H. Chen, and Z. Han, "Perception task offloading with collaborative computation for autonomous driving," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 2, pp. 457–473, Feb. 2023.

[2] D. C. Nguyen et al., "6G Internet of Things: A comprehensive survey," *IEEE Internet Things J.*, vol. 9, no. 1, pp. 359–383, Jan. 2022.

[3] L. Kong et al., "Edge-computing-driven Internet of Things: A survey," *ACM Comput. Surv.*, vol. 55, no. 8, 2023, Art. no. 174.

[4] R. Deb and S. Roy, "A comprehensive survey of vulnerability and information security in SDN," *Comput. Netw.*, vol. 206, 2022, Art. no. 108802.

[5] X. He, R. Jin, and H. Dai, "Peace: Privacy-preserving and cost-efficient task offloading for mobile-edge computing," *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 1814–1824, Mar. 2020.

[6] Y. Zhao et al., "Joint content caching, service placement, and task offloading in UAV-enabled mobile edge computing networks," *IEEE J. Sel. Areas Commun.*, vol. 43, no. 1, pp. 51–63, Jan. 2025.

[7] A. Daghayeghi and M. Nickray, "Decentralized computation offloading via multi-agent deep reinforcement learning for noma-assisted mobile edge computing with energy harvesting devices," *J. Syst. Architecture*, vol. 151, 2024, Art. no. 103139.

[8] Y. Qu, S. Yu, L. Gao, K. Sood, and Y. Xiang, "Blockchained dual-asynchronous federated learning services for digital twin empowered edge-cloud continuum," *IEEE Trans. Serv. Comput.*, vol. 17, no. 3, pp. 836–849, May/Jun. 2024.

[9] S. Ding, L. Zhang, M. Pan, and X. Yuan, "Patrol: Privacy-oriented pruning for collaborative inference against model inversion attacks," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2024, pp. 4716–4725.

[10] X. Xu et al., "A stealthy wrongdoer: Feature-oriented reconstruction attack against split learning," in *Proc. IEEE/CVF Conf. Comput. Vis. pattern Recognit.*, 2024, pp. 12130–12139.

[11] C. Li, B. BAO, and Y. Gao, "In-context fully decentralized cooperative multi-agent reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2025, pp. 1–6.

[12] W. Ding, L. Shi, Y. Chi, and D. Zhao, "Seeing is not believing: Robust reinforcement learning against spurious correlation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 36, 2023, pp. 66328–66363.

[13] G. Wang, P. Cheng, Z. Chen, B. Vucetic, and Y. Li, "Inverse reinforcement learning with graph neural networks for full-dimensional task offloading in edge computing," *IEEE Trans. Mobile Comput.*, vol. 23, no. 6, pp. 6490–6507, Jun. 2024.

[14] Y. Chen, J. Xu, Y. Wu, J. Gao, and L. Zhao, "Dynamic task offloading and resource allocation for NOMA-aided mobile edge computing: An energy efficient design," *IEEE Trans. Serv. Comput.*, vol. 17, no. 4, pp. 1492–1503, Jul./Aug. 2024.

[15] L. Tan, Z. Kuang, L. Zhao, and A. Liu, "Energy-efficient joint task offloading and resource allocation in OFDMA-based collaborative edge computing," *IEEE Trans. Wireless Commun.*, vol. 21, no. 3, pp. 1960–1972, Mar. 2022.

[16] H. Teng, Z. Li, K. Cao, S. Long, S. Guo, and A. Liu, "Game theoretical task offloading for profit maximization in mobile edge computing," *IEEE Trans. Mobile Comput.*, vol. 22, no. 9, pp. 5313–5329, Sep. 2023.

[17] Y. Liang, H. Tang, H. Wu, Y. Wang, and P. Jiao, "Lyapunov-guided offloading optimization based on soft actor-critic for ISAC-aided internet of vehicles," *IEEE Trans. Mobile Comput.*, vol. 23, no. 12, pp. 14708–14721, Dec. 2024.

[18] L. He et al., "An online joint optimization approach for QoE maximization in UAV-enabled mobile edge computing," in *Proc. IEEE Conf. Comput. Commun.*, 2024, pp. 101–110.

[19] Z. Hu, J. Niu, T. Ren, and M. Guizani, "Achieving fast environment adaptation of DRL-based computation offloading in mobile edge computing," *IEEE Trans. Mobile Comput.*, vol. 23, no. 5, pp. 6347–6362, May 2024.

[20] L. Huang, S. Bi, and Y.-J. A. Zhang, "Deep reinforcement learning for online computation offloading in wireless powered mobile-edge computing networks," *IEEE Trans. Mobile Comput.*, vol. 19, no. 11, pp. 2581–2593, Nov. 2020.

[21] S. Zhang, S. Bao, K. Chi, and Z. Shi, "A mixed DRL framework for computing offloading and resource allocation in digital twin assisted wireless powered MEC network," *IEEE Trans. Mobile Comput.*, vol. 25, no. 8, pp. 12472–12487, Aug. 2026.

[22] Y. He, B. Sheng, H. Yin, D. Yan, and Y. Zhang, "Multi-objective deep reinforcement learning based time-frequency resource allocation for multi-beam satellite communications," *China Commun.*, vol. 19, no. 1, pp. 77–91, 2022.

[23] Z. Lin, Z. Feng, K. Guo, A. Nauman, D. Niyato, and J. Wang, "AI-driven seamless and massive access in space-air-ground integrated networks," *IEEE Wireless Commun.*, vol. 32, no. 3, pp. 72–79, Jun. 2025.

[24] A. Alwarafy, K. A. Al-Thelaya, M. Abdallah, J. Schneider, and M. Hamdi, "A survey on security and privacy issues in edge computing-assisted Internet of Things," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4004–4022, Mar. 2021.

[25] Y. Xiao, Y. Jia, C. Liu, X. Cheng, J. Yu, and W. Lv, "Edge computing security: State of the art and challenges," *Proc. IEEE*, vol. 107, no. 8, pp. 1608–1631, Aug. 2019.

[26] Y. Sun et al., "Toward trusted 6G mobile edge computing: A secure batch large language models deployment framework," *IEEE Trans. Mobile Comput.*, vol. 25, no. 3, pp. 3328–3346, Mar. 2026.

[27] V. N. Ti et al., "Reliable intelligent reflecting surface-assisted mobile edge computing systems: A physical layer security and encryption design," *IEEE Trans. Mobile Comput.*, vol. 25, no. 2, pp. 1950–1966, Feb. 2026.

Authorized licensed use limited to: TU Wien Bibliothek. Downloaded on November 03,2026 at 09:06:24 UTC from IEEE Xplore. Restrictions apply.

[28] L. Zhi et al., "Self-powered absorptive reconfigurable intelligent surfaces for securing satellite-terrestrial integrated networks," *China Commun.*, vol. 21, no. 9, pp. 276–291, Sep. 2024.
[29] Z. Lin et al., "Wireless endogenous security for SAGINs: Achieving ubiquitous access and secure communication in symbiosis," *IEEE Netw.*, vol. 39, no. 6, pp. 155–163, Nov. 2025.
[30] Y. He, L. Hu, H. Li, and X. Wang, "A dynamic frequency modulation algorithm based on deep reinforcement learning for FDA-assisted satellite covert communication," *IEEE J. Sel. Areas Commun.*, vol. 44, pp. 1529–1544, 2026.
[31] C. Xu, P. Zhang, X. Xia, L. Kong, P. Zeng, and H. Yu, "Digital twin-assisted intelligent secure task offloading and caching in blockchain-based vehicular edge computing networks," *IEEE Internet Things J.*, vol. 12, no. 4, pp. 4128–4143, Feb. 2025.
[32] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy, "Deep variational information bottleneck," in *Proc. Int. Conf. Learn. Representations*, 2017, pp. 67612–67640.
[33] Z. Gao, L. Yang, and Y. Dai, "Large-scale computation offloading using a multi-agent reinforcement learning in heterogeneous multi-access edge computing," *IEEE Trans. Mobile Comput.*, vol. 22, no. 6, pp. 3425–3443, 2023.
[34] S. Hwang, H. Lee, J. Park, and I. Lee, "Decentralized computation offloading with cooperative UAVs: Multi-agent deep reinforcement learning perspective," *IEEE Wireless Commun.*, vol. 29, no. 4, pp. 24–31, Aug. 2022.
[35] F. Hu, Y. Deng, and A. H. Aghvami, "Scalable multi-agent reinforcement learning for dynamic coordinated multipoint clustering," *IEEE Trans. Commun.*, vol. 71, no. 1, pp. 101–114, Jan. 2023.
[36] Z. Lin, H. Cao, Z. Feng, T. M. Abdellatif, and S. Moussa, "RIS-SCMA Co-design for endogenous security and spectral efficiency: A multi-agent DRL approach in cognitive satellite-terrestrial networks," in *Proc. IEEE Glob. Commun. Conf.*, 2025, pp. 6472–6477.
[37] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6382–6393.
[38] B. Peng et al., "FACMAC: Factored multi-agent centralised policy gradients," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 12208–12221.
[39] J. G. Kuba et al., "Trust region policy optimisation in multi-agent reinforcement learning," in *Proc. Int. Conf. Learn. Representations*, 2021, pp. 6569–6595.
[40] J. Wang, Y. Li, Y. Li, S. Mao, Z. Dong, and Y. Tang, "GoMIC: Enhancing efficient collaboration in multi-agent reinforcement learning through group-specific mutual information," *IEEE Trans. Cogn. Devel. Syst.*, vol. 17, no. 6, pp. 1536–1547, Dec. 2025.
[41] X. Du, Y. Ye, P. Zhang, Y. Yang, M. Chen, and T. Wang, "Situation-dependent causal influence-based cooperative multi-agent reinforcement learning," in *Proc. AAAI Conf. Artif. Intell.*, 2024, pp. 17362–17370.
[42] S. Li et al., "Robust multi-agent reinforcement learning by mutual information regularization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 10, pp. 18118–18132, Oct. 2025.
[43] S. Ding, W. Du, L. Ding, L. Guo, and J. Zhang, "Learning efficient and robust multi-agent communication via graph information bottleneck," in *Proc. AAAI Conf. Artif. Intell.*, 2024, pp. 17346–17353.
[44] R. Yang, J. Wang, Q. Peng, R. Guo, G. Wu, and B. Li, "Learning robust representations with long-term information for generalization in visual reinforcement learning," in *Proc. Int. Conf. Learn. Representations*, 2025, pp. 67612–67640.
[45] Z. Xiao et al., "Multi-objective parallel task offloading and content caching in D2D-aided MEC networks," *IEEE Trans. Mobile Comput.*, vol. 22, no. 11, pp. 6599–6615, Nov. 2023.
[46] J. Peng, H. Qiu, J. Cai, W. Xu, and J. Wang, "D2D-assisted multi-user cooperative partial offloading, transmission scheduling and computation allocating for MEC," *IEEE Trans. Wireless Commun.*, vol. 20, no. 8, pp. 4858–4873, Aug. 2021.
[47] Z. Hu, J. Niu, T. Ren, X. Liu, and M. Guizani, "SITOff: Enabling size-insensitive task offloading in D2D-assisted mobile edge computing," *IEEE Trans. Mobile Comput.*, vol. 24, no. 3, pp. 1567–1584, Mar. 2025.
[48] D. Ye and Z. Lu, "Mutual-information regularized multi-agent policy iteration," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 36, 2023, pp. 2617–2635.
[49] J. Ackermann, V. Gabler, T. Osa, and M. Sugiyama, "Reducing overestimation bias in multi-agent domains using double centralized critics," in *NeurIPS Deep Reinforcement Learn. Workshop*, 2019.
[50] T. Yuan, H.-M. Chung, and X. Fu, "PP-MARL: Efficient privacy-preserving multi-agent reinforcement learning for cooperative intelligence in communications," *IEEE Netw.*, vol. 38, no. 5, pp. 196–203, Sep. 2024.

Kexin Li received the PhD degree from Northeastern University, Shenyang, China, in 2023. She is currently working as a lecturer with the University of Electronic Science and Technology of China. Her research interests include software-defined networking, edge computing, and machine learning.

Changjian Xu received the BS degree from the China University of Mining and Technology, Xuzhou, China, in 2024. He is currently working toward the MS degree with the University of Electronic Science and Technology of China. His research interests include edge computing, reinforcement learning, and multimodal large language models.

Hongwei Zhu received the BS degree from Southwest Jiaotong University, Chengdu, China, in 2024. He is currently working toward the MS degree with the University of Electronic Science and Technology of China. His research interests include task scheduling and edge computing.

Wen Yin received the MS degree in 2024 from University of Electronic Science and Technology of China, Chengdu, China, where he is currently working toward the PhD degree. His research interests include multimodal perception, multimodal large language models, and affective computing.

Yu Liu received the MS degree from The University of Western Australia, in 2023. He is currently working toward the PhD degree with the Institute of Information Engineering, Chinese Academy of Sciences. His research interests include retrieval-augmented generation and multi-agent collaborative reasoning for large language models.

Yu Peng (Member, IEEE) received the PhD degree in computer science and technology from Hunan University, Changsha, China, in 2024. He is currently an assistant researcher with the University of Electronic Science and Technology of China. His research interests include security and privacy issues in intelligent computing, cloud computing, networked applications, and blockchain.

Yong Wang received the PhD degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2008. He is currently a research professor with the University of Electronic Science and Technology of China. His research interests include Big Data privacy computing in public security, trusted computing, collaborative computing of intelligent models, and data protection technology.

Xingwei Wang received the BS, MS, and PhD degrees in computer science from Northeastern University, Shenyang, China, in 1989, 1992, and 1998, respectively. He is currently a professor with the College of Computer Science and Engineering, Northeastern University. He has authored or coauthored more than 100 journal articles, books and book chapters, and refereed conference papers. His research interests include cloud computing and future Internet. He was the recipient of the several best paper awards.

Schahram Dustdar (Fellow, IEEE) is currently a full professor of computer science with a focus on Internet Technologies heading the Distributed Systems Group with the TU Wien, Austria, also with ICREA, Barcelona, Spain. Since 2016, he has been the chairman of the Informatics Section of the Academia Europaea . He is a member of the IEEE Conference Activities Committee, Section Committee of Informatics of the Academia Europaea, and Academia Europaea: The Academy of Europe. He was the recipient of the ACM Distinguished Scientist Award (2009) and IBM Faculty Award (2012). He is an associate editor of *IEEE Transactions on Services Computing*, *ACM Transactions on the Web*, and *ACM Transactions on Internet Technology*, and on the editorial board of *IEEE Internet Computing*. He is the editor-in-chief of Computing (Springer).

Authorized licensed use limited to: TU Wien Bibliothek. Downloaded on November 03,2026 at 09:06:24 UTC from IEEE Xplore. Restrictions apply.