Diagnosing Constraint Awareness for Trustworthy Reinforcement Learning in MEC

Rim Sayegh[‡§], Ali Al Housseini[*†], Sahar Hoteit[§‖], Hela Marouane[‡], Silvia Santini[†], Omran Ayoub[*],
[‡]ESME Research Lab, Ivry sur Seine, France
[§]Université Paris-Saclay, CNRS, CentraleSupélec, Laboratoire des Signaux et Systèmes (L2S), Gif-sur-Yvette, France
[*]University of Applied Sciences and Arts of Southern Switzerland, Lugano, Switzerland
[†]University of Southern Switzerland, Lugano, Switzerland
[‖]Institut Universitaire de France (IUF), France
Corresponding authors: rim.sayegh@centralesupelec.fr, ali.alhousseini@supsi.ch

***Abstract*—Action masking is a well-known technique in hard-constrained RL for multi-access edge computing (MEC) orchestration that improves training efficiency by preventing infeasible decisions. However, it optimizes reward while externalizing feasibility logic from the policy, including resource capabilities and migration constraints. This paper studies this trade-off in an MEC-enabled Vehicle-to-Everything (V2X) setting by comparing two matched training regimes: one that removes infeasible actions before decision-making and one that learns feasibility through penalty-driven interaction. To analyze the basis of decision-making, we apply SHAP and introduce feasibility-aware metrics, including the Feasibility Attribution Ratio (FAR) and the Optimization Attribution Ratio (OAR). We then evaluate robustness to shifts in the feasibility set, such as increased demand or reduced MEC capacity. The results show that the unmasked model is more affected by feasibility-related constraints and more resilient when feasibility conditions shift, whereas the masked model improves convergence but is less stable under stress. These findings suggest that reward optimization alone is insufficient to assess trustworthy MEC-RL and motivate an evaluation protocol that combines performance metrics with explanation and robustness checks.**

I. Introduction

Reinforcement learning (RL) is increasingly studied as a framework for sequential decision-making in 5G/6G networks, with applications ranging from service placement and task offloading to routing and multi-access edge computing (MEC) orchestration [1], [2]. These applications share a common structure: at each step, an RL policy must select an action based on resource capacities, latency budgets, and the current network state, which may change between consecutive decisions. Training optimizes a scalar reward signal that captures objectives such as resource efficiency and request acceptance. However, high cumulative reward during training does not necessarily mean that the policy encodes the problem's feasibility structure; a policy may consistently select valid actions without attending to the state features that determine why those actions are valid. Determining whether a policy has internalized this structure, rather than relying on external mechanisms such as action masking that restrict its choices, is key to ensuring operator trust under deployment, especially in conditions unseen during training.

The most common mechanism for enforcing feasibility in constrained networking RL is action masking, which filters infeasible actions before the policy samples from the action space. This approach is effective and often necessary because it prevents infeasible decisions and improves training efficiency. However, despite its effectiveness, it may also distort the learning problem [3]. This is because when infeasible actions are systematically excluded from interaction, the policy receives less direct corrective feedback about why those actions are invalid and therefore has less incentive to internalize the feasibility constraints. As a result, part of the constraint logic is offloaded to the external mechanism rather than handled by the policy itself.

This can be problematic in operational settings, where the feasible action set may shift due to traffic surges or tighter service requirements. Under such perturbations, the key question is not only *whether* a policy achieved high training reward, but *whether* it learned to be aware of the state variables that govern valid decision-making. In most networking RL studies, policies are evaluated mainly through aggregate indicators such as reward, convergence, and acceptance ratio. Whether successful policies actually encode feasibility-relevant structure, or succeed mainly because infeasible behavior is filtered externally, remains insufficiently examined.

In this paper, we investigate feasibility-relevant awareness in a MEC-enabled V2X environment, a representative setting in which sequential application-offloading decisions must satisfy coupled compute, communication, and delay constraints under dynamic conditions [4], [5]. We develop two matched RL agents that solve the same task under the same reward formulation: one uses hard invalid-action masking, while the other acts over the full action space and learns feasibility through penalty-driven interaction. To quantify how these regimes shape the learned policy structure, we use SHAP-based feature attribution (SVERL) as a diagnostic probe of policy behavior rather than as a post-hoc visualization tool [6]. We group state features into feasibility-related and optimization-

This work has been partially supported by the EUREKA CELTIC-NEXT SUSTAINET-Advance project, funded by the Swiss Innovation Agency Innosuisse No. 119.588 INT-ICT, and by MUSMET project funded by the EIC Pathfinder Open scheme of the European Commission (grant agreement n. 101184379), and has received funding from the Swiss State Secretariat for Education, Research and Innovation (SERI).

ISBN 978-3-903176-82-9 © 2026 IFIP

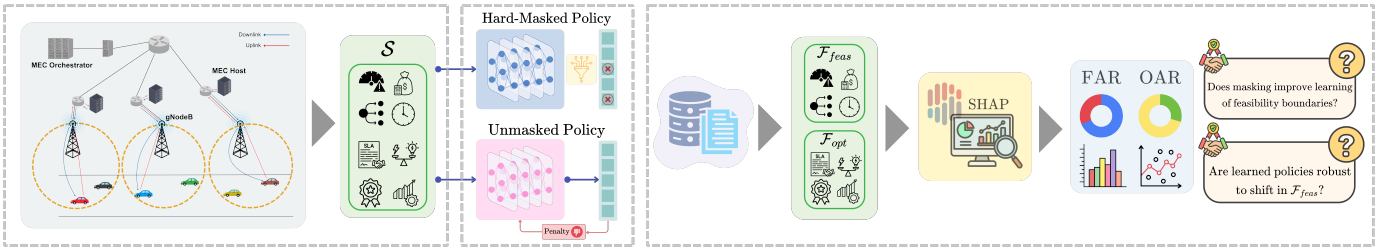

Fig. 1: Proposed workflow for the masked-vs.-unmasked MEC-RL study. A VEC environment provides the same decision task to two matched agents: a hard-masked policy and an unmasked penalty-based policy. After training, both agents are analyzed on a shared evaluation-state set using SHAP. Feature attributions are grouped into feasibility-related and optimization-related components to compute FAR, OAR, which are then connected to robustness under feasibility-set perturbations for trustworthiness assessment.

related sets, measure the policy's dependence on these sets, and test whether these attribution patterns are reflected in robustness under shifts in the feasible action set, including host-capacity reduction and increased demand. Our results show that hard masking improves convergence and nominal reward but reduces the policy's reliance on feasibility-defining features, whereas the unmasked regime develops stronger constraint awareness and incurs smaller performance loss under feasibility-set shifts unseen during training, suggesting that reward alone is insufficient to assess trustworthiness in deployed MEC-RL policies.

II. Related Work

RL for constrained network control: RL has become a standard approach for sequential decision problems in next-generation networks, with recent surveys documenting its widespread use in MEC task offloading, resource management, and edge orchestration [2]. Representative studies include deep RL (DRL) for delay-sensitive task offloading in MEC [7], multi-agent RL for hybrid-decision-space task offloading in MEC [8], and DRL-based latency-aware slicing in O-RAN [9]. This literature demonstrates the maturity of RL for network control, evaluation while remains predominantly on performance, focusing on reward, latency, and acceptance ratio.

Within RL, action masking is a common strategy for preventing agents from selecting infeasible actions in constrained environments. Prior work shows that this mechanism is particularly important when the number of invalid actions is large [3] and when performance gains are required [10]. However, its effect on what the policy learns about environment feasibility remains underexplored, particularly in networking RL, where masking is often treated as an implementation detail rather than as a factor that may shape how the agent learns feasibility-relevant structure.

Explainability for networking RL: DRL models remain largely opaque, which weakens operators' trust in their deployment for safety- and performance-critical network-control tasks. This has increased interest in explainable reinforcement learning (XRL), where the objective is not only to improve human understanding of learned policies, but also to support reliable use in operational settings. Surveys such as [11] organize this literature into several main families, including policy summarization and saliency- or attribution-based explanation methods. In [12], SHAP is applied to multi-agent DRL for V2X resource allocation, enabling feature selection and model reduction, whereas [5] improves interpretability in vehicular network slicing by combining attention with Shapley-value supervision. While [13] focuses on explaining the behavior of an RL agent for network slice admission control, existing approaches are used mainly to describe learned behavior after training rather than to analyze how specific training choices affect what the agent internalizes.

Position of this work: Our work lies at the intersection of MEC-oriented RL, constrained RL, and XRL. As in prior MEC-RL studies, we consider a sequential network-control problem; as in constrained RL, feasibility must be respected during decision making; and as in XRL, we use post-hoc feature attribution to inspect learned policies. The main contribution of this paper is to investigate how feasibility handling mechanism in RL for constrained network control problems influences performance under stringent requirements not captured during training.

We show that hard masking improves convergence and reward, but reduces the policy's reliance on feasibility-relevant features, while unmasked training leads to stronger dependence on constraint-relevant features and more robust behavior under feasibility-set shifts. These results provide direct evidence that trustworthiness in MEC-oriented RL cannot be assessed from reward alone, and instead requires evaluating both what the policy attends to and how it behaves when feasibility conditions change.

III. System Model and RL Formulation

This section introduces the vehicular MEC environment (III-A), formulates the RL decision problem (III-B), defines the two training regimes (III-C), and presents the feature partition and trust-oriented evaluation metrics (III-D, III-E).

A. Vehicular MEC Environment

Fig. 1 shows a schematic representation of the framework. We consider a vehicular edge computing (VEC) environment (left panel of Fig. 1) in which a set of vehicles $\mathcal{V} = \{v_1, \ldots, v_n\}$ travel along a highway segment served by roadside base stations (gNodeBs), each co-located with a MEC host from the set $\mathcal{M} = \{m_1, \ldots, m_h\}$. The deployment follows the ETSI MEC reference architecture [14]. Each vehicle runs a safety-critical sensing application [4] that must be offloaded to a

MEC host for real-time processing. A centralized orchestrator assigns each vehicle's task to a suitable host based on resource availability, proximity, and load conditions.

Decisions are made sequentially following a consecutive set of events: at each event e, a vehicle v_i *either* initiates a new offloading request *or* requests migration due to mobility, and the orchestrator selects a target host m_j via the binary assignment $x_{ij}^e \in \{0,1\}$. Each MEC host $m_j \in \mathcal{M}$ is characterized by a set of resources $k \in \mathcal{R} = \{\text{CPU}, \text{RAM}, \text{Disk}\}$. We denote by $C_{j,k}^{\max}$ and $\rho_{j,k}^e$ the maximum capacity and the remaining capacity of resource k for MEC m_j at event e, respectively. Similarly, each vehicle $v_i \in \mathcal{V}$ requires $Q_{i,k}^e$ units of resource k. The overall objective is to *maximize* the number of successfully served vehicles while *balancing* load and *minimizing* latency. A complete specification of the environment, including the safety application, traffic model, and full optimization formulation, is provided in [4].

B. MDP Formulation

We model the environment as a Markov decision process with state-dependent action constraints: MDP = $(\mathcal{S},\ \mathcal{A},\ \mathcal{T},\ \mathcal{R}_w,\ \mathcal{C})$ where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the full discrete action space, $\mathcal{T}$ is the transition function, $\mathcal{R}_w$ is the reward function, and $\mathcal{C} : \mathcal{S} \to 2^{\mathcal{A}}$ is the *feasibility constraint function* that maps each state to the subset of valid actions. At each decision event e, the agent observes $s^e \in \mathcal{S}$, selects $a^e \in \mathcal{A}$, and receives reward $r^e \in \mathcal{R}_w$.

State space: The state matrix at event e aggregates three information groups:

$$s^e = \big[\underbrace{V_i^{\text{info}}, V_i^{\text{req}}}_{\text{requesting vehicle}}, \underbrace{\mathcal{M}^{\text{info}}}_{\text{all hosts}}\big] \tag{1}$$

Where, $V_i^{\text{info}} = (v_i^{id}, v_i^p, v_i^s, v_i^h, v_i^m, v_i^d)$ captures the requesting vehicles' ID, position, speed, heading, current host assignment, and distances to all hosts, respectively. $V_i^{\text{req}} = (Q_{i,1}^e, Q_{i,2}^e, Q_{i,3}^e)$ encodes its resource requirements. In addition, $\mathcal{M}^{\text{info}}$ contains information about MECs. A MEC $m_j \in \mathcal{M}$ is described by the tuple (m_j^{id}, m_j^c, m_j^l), where m_j^{id} is the MEC ID, $m_j^c = (\rho_{j,1}^e, \rho_{j,2}^e, \rho_{j,3}^e)$ are the available resources and m_j^l is the load level of m_j in the current event e. The total dimension of the flattened state vector is $6|\mathcal{M}|+10$.

Action space: The agent selects $a^e \in \mathcal{A} = \{0, 1, \ldots, M\}$, where the first $M-1$ actions represent MEC host indexes, while last action corresponds to *drop* the request.

Action Feasibility constraints: The feasible set $\mathcal{C}(s^e) \subseteq \mathcal{A}$ is defined by two conditions:

$$x_{ij}^{e-1} \le 1 - x_{ij}^e, \tag{2}$$

$$\textstyle\sum_{i \in \mathcal{V}} x_{ij}^e\, Q_{i,k}^e \le \rho_{j,k}^e, \quad \forall k \in \mathcal{R}. \tag{3}$$

Constraint (2) enforces the migration decision rule of the environment, ensuring that an already assigned request is not kept on the same host when a migration decision is triggered. Constraint (3) ensures that aggregate resource demand does not exceed the available capacity of the selected host either when a vehicle requests a new offloading request or when a migration is triggered. An action a^e belongs to $\mathcal{C}(s^e)$ if and only if both constraints are satisfied.

Reward: The reward function is designed to guide the agent toward selecting MEC hosts that are simultaneously resource-feasible, load-balanced, and geographically close to the requesting vehicle, mirroring the joint optimization objectives of MEC orchestration in vehicular settings [4]. For feasible actions, the reward aggregates a base bonus R_s, a clipped load-variance reduction $r_l^e = \phi(\Delta\text{Var}^e(\mathbf{L}))$, where $\mathbf{L} = (m_1^l, \ldots, m_h^l)$ encouraging balanced load among MEC hosts, and a proximity-mobility term $r_d^e = -\hat{d}^e + \cos(\theta^e)$ favoring nearby hosts aligned with the vehicle's heading; where $\hat{d}^e$ corresponds to the distance/proximity to MEC hosts and θ^e to the angle between the vehicle velocity (heading) and the vector pointing to the selected MEC.

$$r^e = \begin{cases} R_s + r_l^e + r_d^e, & \text{if } a^e \in \mathcal{C}(s^e), \\ r_{pen}, & \text{if } a^e \notin \mathcal{C}(s^e), \\ r_{drop}, & \text{if } \mathcal{C}(s^e) = \emptyset \end{cases} \tag{4}$$

where r_{pen} is a penalty (negative value) in case of selecting an infeasible action (unmasked agent) and r_{drop} is a negative reward in case there is no feasible action. Please note that this reward becomes null if there is no feasible actions and the agent decides to drop the request. The treatment of infeasible actions depends on the training regime described next. This distinction is central to our study, since the same MEC decision problem can be solved either by *externally enforcing* feasibility through $\mathcal{C}(s^e)$ or by requiring the policy to *learn* feasibility-relevant regularities from interaction.

C. Training Regimes

The objective of this study is to examine how the way feasibility is enforced during training influences how much an RL policy learns about the environment's constraints. For a rigorous evaluation, feasibility handling must be the only factor that differs between the compared agents. To this end, we instantiate two matched training regimes over the same MDP (Fig. 1). The policy architecture, optimizer, and all training parameters are fixed; the regimes differ *only* in how the agent is exposed to $\mathcal{C}(s^e)$ during learning.

Regime M (Hard Mask): The agent is trained using MaskablePPO [3]. At each step, the environment computes $\mathcal{C}(s^e)$ and constructs a binary mask $\mathbf{m}(s^e) \in \{0,1\}^M$, where $\mathbf{m} = 1$ iff $j \in \mathcal{C}(s^e)$. Infeasible actions receive logit $-\infty$ before the softmax, so π_M (policy) assigns them zero probability. As a result, the policy cannot select an infeasible action and therefore receives no learning signal from violating the feasibility constraints. This regime may improve learning efficiency and nominal performance (reward), but it also externalizes part of the decision logic through an oracle-like constraint filter.

Regime U (Unmasked + Penalty): The agent is trained using standard PPO [15] over the full action space $\mathcal{A}$, without any masking. When the agent selects $a^e \notin \mathcal{C}(s^e)$, the environment returns a fixed penalty $r_{\text{pen}} < 0$, the network state remains unchanged, and the episode continues. The agent must therefore learn, through trial and error, which regions of the state

TABLE I: Partition of state features into feasibility ($\mathcal{F}_{\text{feas}}$) and optimization ($\mathcal{F}_{\text{opt}}$) groups.

Group	Features
$\mathcal{F}_{\text{feas}}$	***Vehicle info:*** Current MEC assignment (v_i^h) ***Vehicle req.:*** CPU/RAM/Disk demand ($Q_{i,k}$) ***MEC host:*** Index (m_j^{id}), Avail. CPU/RAM/Disk ($\rho_{j,k}$)
$\mathcal{F}_{\text{opt}}$	***Vehicle info:*** Position (v_i^p), Speed (v_i^s), Heading (v_i^m), Distances to hosts (v_i^d) ***MEC host:*** Load level (m_j^l)

space are associated with infeasible decisions and how they affect returns.

The central hypothesis is that these two regimes, despite solving the same task under the same reward, can produce policies with different internal structures. When feasibility is enforced externally, the policy may achieve high reward without learning to attend to the state features that govern action validity, making it more brittle when the feasible set shifts due to capacity reduction, host unavailability, or demand surges that exceed training conditions.

D. Feature Partition

We partition the state variables into roles in decision-making before computing feature attribution.[1]The objective of this partition is to distinguish features that govern *action feasibility* from those that mainly affect *action quality*. Tab. I details each group. This distinction is essential for our analysis: it allows us to interpret attribution scores not only in terms of which inputs influence a decision, but also in terms of *what kind of reasoning* the policy relies on. In particular, a policy that assigns more attribution mass to feasibility-related features can be interpreted as relying more strongly on constraint-relevant information, whereas a policy dominated by optimization-related features may be relying more on reward refinement. Accordingly, we define two feature groups:

- **Feasibility features** ($\mathcal{F}_{\text{feas}}$): variables that determine whether an action satisfies constraints (2)–(3). These include the MEC host index m_j^{id}, the available host resources $\rho_{j,k}^e$, the vehicle's current MEC assignment, and the vehicle's resource requirements $Q^e s_{i,k}$. These variables directly govern whether a candidate assignment is admissible.
- **Optimization features** ($\mathcal{F}_{\text{opt}}$): variables that influence the reward of a feasible decision without determining validity itself. These include vehicle motion and location descriptors, vehicle-to-host distances, and host load indicators. These variables shape which feasible action is preferable, rather than which actions are allowed.

E. Evaluation Metrics

For a collected evaluation dataset $\mathcal{D}$, we compute the SHAP values $\phi_p(s^e)$ for all state features p in $6|\mathcal{M}|+10$. We define the following metrics:

[1]This partition is functional rather than strictly disjoint: it assigns each feature to the role it primarily plays in the decision structure (constraints described above).

TABLE II: Simulation and training parameters.

Parameter	Value
Environment	
MEC hosts (h) / Vehicles (n)	[2, 5] / [5, 30]
Vehicle speed / Request lifetime	$\mathcal{U}(40, 140)$ km/h / $\mathcal{U}(30, 180)$ s
Road length / Simuation time	20 km / 200 s
Training	
Regime M / Regime U	MaskablePPO / PPO
Policy network / Learning rate	MLP $[128, 128, 128]$ / 3×10^{-4}
Timesteps / Seeds per regime	$\geq$500K / 5

Feasibility Attribution Ratio (FAR): Measures the fraction of total attribution mass on feasibility-relevant features $\mathcal{F}_{\text{feas}}$. While $\overline{\text{FAR}}$ is the average over samples in the evaluation dataset.

$$\text{FAR}(s^e) = \frac{\sum_{f\in\mathcal{F}_{\text{feas}}}|\phi_f(s^e)|}{\sum_p|\phi_p(s^e)|}, \quad \overline{\text{FAR}} = \frac{1}{|\mathcal{D}|}\sum_{s=1}^{|\mathcal{D}|}\text{FAR}(s^e) \quad (5)$$

Optimization Attribution Ratio (OAR): Measures the fraction of total attribution mass on optimization-relevant features $\mathcal{F}_{\text{opt}}$. While $\overline{\text{OAR}}$ is the average over samples in the states set. By construction, OAR is the complement of FAR.

$$\text{OAR}(s^e) = \frac{\sum_{o\in\mathcal{F}_{\text{opt}}}|\phi_o(s^e)|}{\sum_p|\phi_p(s^e)|}, \quad \overline{\text{OAR}} = \frac{1}{|\mathcal{D}|}\sum_{s=1}^{|\mathcal{D}|}\text{OAR}(s^e) \quad (6)$$

IV. Results and Analysis

This section evaluates the two regimes from three complementary perspectives: (i) learning efficiency under nominal conditions, (ii) attribution analysis of policies under different regimes, and (iii) robustness to shifts in the feasible action set. We compare convergence and invalid-action rates to assess training efficiency, analyze feature attributions to quantify reliance on $\mathcal{F}_{\text{feas}}$ versus $\mathcal{F}_{\text{opt}}$, and examine whether these differences translate into robustness under deployment-time deviations. The main simulation and training parameters are reported in Tab. II. We denote by U and M, the RL agents trained under regimes U and M, respectively.

A. Training Convergence

Fig. 2 compares the learning dynamics of the RL agents trained under both regimes over 1.5×10^6 training timesteps. The top panel reports the mean episodic reward, while the bottom panel shows the invalid-action rate. A clear separation emerges between the two regimes. As expected, M converges substantially faster, reaching a stable reward plateau after approximately 5×10^5 timesteps, whereas U requires roughly 9×10^5 timesteps to stabilize. In addition to faster convergence, M attains a higher final reward level, fluctuating around 280, while U stabilizes closer to 240–250. This confirms that hard masking improves both training efficiency and nominal optimization performance. The lower panel clarifies the source of this difference. By construction, M maintains an invalid-action rate of 0 throughout training, since infeasible actions are removed from the policy's choice set before sampling. In contrast, U initially selects infeasible actions very frequently, with an invalid-action rate peaking near 65%, because it must discover feasibility constraints through interaction alone. As

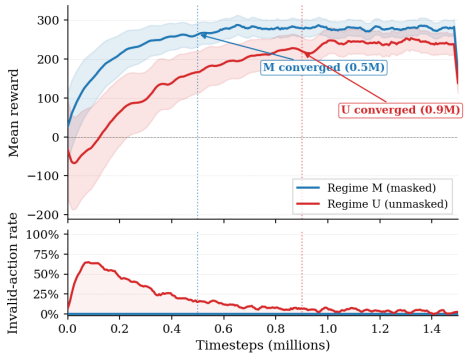

Fig. 2: Mean episodic reward and invalid-action rates of RL agents under both training regimes.

training proceeds, this rate decreases steadily and approaches a low residual level of about 3%, showing that the unmasked policy gradually learns to avoid infeasible decisions without external filtering. The slower reward improvement of U is therefore not incidental, but reflects the additional exploration cost of learning feasibility structure directly from experience.

Takeaway 1: Masking accelerates convergence by eliminating infeasible exploration, whereas unmasked training converges more slowly because the agent must learn feasibility through interaction.

B. Attribution Patterns and Feasibility Awareness

We next examine how RL agents trained under the two different regimes differ in the type of information they rely on. Tab. III reports $\overline{\text{FAR}}$ and $\overline{\text{OAR}}$ across both regimes. The main result is immediate: Agent U exhibits a higher $\overline{\text{FAR}}$ (0.597 ± 0.027 vs. 0.519 ± 0.056) than agent M. In other words, when infeasible actions are not removed a priori, the learned policy allocates a larger share of its attribution to features that govern feasibility. This is consistent with a policy that must internalize constraint-relevant structure rather than relying on an external mask to enforce it. Fig. 3 provides a feature-group view of this result by plotting mean absolute SHAP values of the various features when the RL agent is trained under each regime. The features are partitioned into $\mathcal{F}_{\text{feas}}$ and $\mathcal{F}_{\text{opt}}$ categories. The largest contrast appears in the $\mathcal{F}_{\text{feas}}$ features, especially *MEC identity & capacity*, which is the most influential group under both regimes but receives larger attribution under regime U. Smaller attribution is given to *current MEC* and *N_MEC*, with U still having higher attribution than M for the same sets of features. These variables directly affect host admissibility; thus, their stronger influence for U indicates greater sensitivity to the structure of the feasible action set. By contrast, the regime M places relatively more emphasis on $\mathcal{F}_{\text{opt}}$ components compared with U, especially *vehicle kinematics*, while *MEC distance & load* remains important in both regimes. This pattern is consistent with the role of masking: once infeasible actions are filtered out externally, the policy can focus more on ranking the remaining valid actions according to performance-related criteria.

Takeaway 2: External masking shifts part of the feasibility reasoning out of the policy. By contrast, unmasked training

TABLE III: Feasibility and Optimization Attribution Ratios

Metric	Agent M	Agent U
$\overline{\text{FAR}}$	0.519 ± 0.056	$\mathbf{0.597 \pm 0.027}$
$\overline{\text{OAR}}$	$\mathbf{0.481 \pm 0.056}$	0.403 ± 0.027

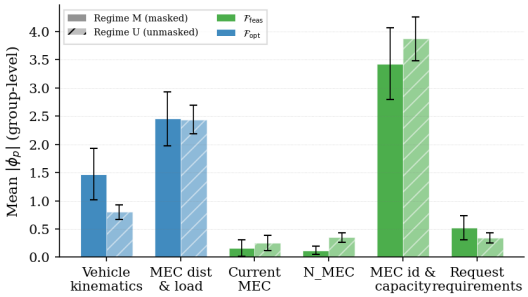

Fig. 3: SHAP Attribution Profile

places more attribution on constraint-defining features, thus internalizing the structure of feasible decision making. This, in turn, gives greater confidence in its deployment robustness, since operational changes in capacity, availability, or delay are more likely to be reflected in the policy's own decision process rather than handled only by an external mask.

C. Robustness Under Scaling Shift

We now evaluate whether agents trained under both regimes remain reliable under scaling shifts (unseen perturbations during training). To this end, we introduce two scaling changes at test time and measure the resulting acceptance ratio, which is the ratio of the assigned tasks to the total requested tasks in an episode. In the first, the number of MEC hosts is increased beyond the training range while keeping the total system capacity fixed; thus, capacity becomes distributed across a larger number of hosts, making individual host feasibility more restrictive. In the second, the number of vehicles is increased, which raises the request load and therefore the demand rate on the system. In both cases, remaining parameters are kept fixed. The gray-highlighted region in Fig. 4 indicates the training range, while the remaining points correspond to unseen operating conditions. Fig. 4 shows a clear pattern. Within or near training range (2 to 6 MECs), M achieves slightly higher acceptance than agent U, consistent with the advantage of masking when the test conditions closely match the training regime. However, once the environment moves into harder and unseen settings (7 MECs or higher), this advantage reverses. Under MEC scaling, the acceptance ratio of M drops sharply beyond the training range, whereas U degrades more gradually. A similar transition appears under vehicle scaling: as the number of requests increases, both regimes deteriorate, but the decline is steeper for M, while U maintains a consistently higher acceptance ratio in the high-load region. These results indicate that the masked policy is more brittle under scaling shift, even though it performs better under nominal conditions. Fig. 5 shows $\overline{\text{FAR}}$ and $\overline{\text{OAR}}$ for increasing number of MECs. Results show that U remains remarkably stable: $\overline{\text{FAR}}$ stays close to 0.58–0.60. In contrast, M exhibits substantially larger fluctuations, with $\overline{\text{OAR}}$ becoming more erratic as the test

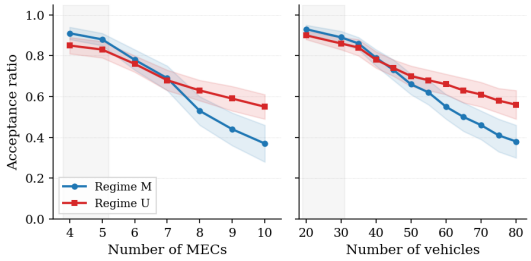

Fig. 4: Acceptance Ratio vs. Number of MECs and Vehicles

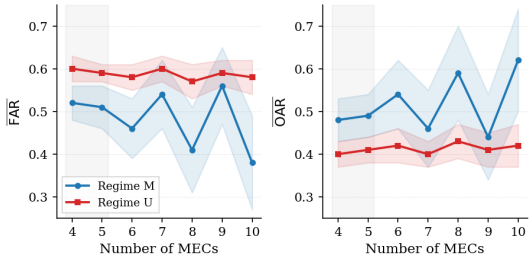

Fig. 5: FAR/OAR vs. Number of MECs

scenario becomes more demanding. This indicates that the unmasked policy maintains a consistent reliance on feasibility-relevant features even under unseen scaling, whereas the masked policy changes its attribution profile more noticeably when the operating conditions shift.

Moreover, U shows a higher robustness, because its decisions remain anchored to the same class of constraint-defining cues under stress, which increases confidence in its suitability for deployment. By contrast, M benefits from external feasibility enforcement under nominal conditions, yet its internal decision basis appears less stable once the environment becomes more demanding, making it less reliable despite its stronger in-range performance. The results reveal a consistent trade-off: Hard masking improves learning speed and reward by preventing infeasible exploration, but it also reduces the incentive for the policy to rely on $\mathcal{F}_{\text{feas}}$. U incurs a higher training cost, yet develops a more feasibility-aware attribution profile and preserves that profile more consistently under unseen scaling shifts. As a result, it degrades more gracefully when feasibility conditions become more demanding. *Takeaway 3:* Hard masking improves nominal performance, but it does not necessarily produce the more deployment-trustworthy policy. Under unseen scaling shifts, unmasked training allows to retain a more stable feasibility-oriented attribution profile and degrades more gracefully in acceptance ratio, which provides stronger evidence of robustness in operational settings.

Together, these three takeaways form a coherent picture: masking buys efficiency at the cost of internalizing constraint structure, and this cost only becomes visible under deployment-time stress. Trustworthy MEC-RL, therefore, requires evaluating not only how fast and how well a policy learns, but what it learns to rely on. These results reveal a trade-off between nominal efficiency and deployment robustness: masking improves training safety and efficiency, whereas unmasked training yields stronger internal feasibility modeling and greater robustness under shift. While shown here only in a vehicular MEC setting, similar trade-offs may arise in other constrained RL problems with externally enforced feasibility.

V. Conclusion

In this work, we examined the trade-off between nominal performance and internal constraint reasoning in MEC-RL, through a controlled comparison between hard-masked and unmasked training regimes in a V2X environment. Using SHAP-based attribution alongside the proposed FAR and OAR metrics, we find that masking accelerates convergence and eliminates infeasible exploration, but externalizes feasibility logic rather than encoding it within the policy. The unmasked regime, despite lower nominal reward, exhibits stronger attribution to feasibility-defining features and maintains more stable behavior under shifts in the feasible action set. These results indicate that reward and convergence metrics alone are insufficient to characterize trustworthy MEC-RL policies, and support an evaluation protocol that integrates performance measures with attribution analysis and robustness assessment. A useful extension would be to examine alternative feature partitions to assess how robust the attribution-based conclusions are to alternative group definitions.

References

[1] Z. Zabihi *et al.*, "Reinforcement learning methods for computation offloading: A systematic review," *ACM Comput. Surv.*, Aug. 2023.

[2] N. Yang *et al.*, "Beyond the edge: An advanced exploration of reinforcement learning for mobile edge computing, its applications, and future research trajectories," *IEEE Commun. Surv. Tutorials*, 2025.

[3] S. Huang *et al.*, "A closer look at invalid action masking in policy gradient algorithms," *The International FLAIRS Conference Proceedings*, 2022.

[4] R. Sayegh *et al.*, "Mobility aware task migration in vehicular edge computing networks," in *36th International Teletraffic Congress (ITC-36)*, 2025.

[5] H. Sun *et al.*, "An explainable ai framework for dynamic resource management in vehicular network slicing," in *IEEE 36th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. IEEE, 2025.

[6] D. Beechey *et al.*, "Explaining reinforcement learning with shapley values," in *Proceedings of the 40th International Conference on Machine Learning*. PMLR, 2023.

[7] M. Tang *et al.*, "Deep reinforcement learning for task offloading in mobile edge computing systems," *IEEE TMC*, 2020.

[8] J. Wang *et al.*, "Multi-agent reinforcement learning for task offloading with hybrid decision space in multi-access edge computing," *Ad Hoc Networks*, 2025.

[9] R. Raftopoulos *et al.*, "Drl-based latency-aware network slicing in o-ran with time-varying slas," *arXiv preprint arXiv:2401.05042*, 2024.

[10] O. Ayoub *et al.*, "Towards explainable reinforcement learning in optical networks: The rmsa use case," in *Optical Fiber Communications Conference and Exhibition (OFC)*, 2024.

[11] L. Wells *et al.*, "Explainable ai and reinforcement learning—a systematic review of current approaches and trends," *Frontiers in Artificial Intelligence*, 2021.

[12] N. Khan *et al.*, "Explainable ai-aided feature selection and model reduction for drl-based v2x resource allocation," *IEEE Transactions on Communications*, vol. 73, no. 9, pp. 7633–7649, 2025.

[13] J. P. Asdikian *et al.*, "Verifying behavior of reinforcement learning agents for network slice admission control," in *21st International Conference on Network and Service Management (CNSM)*, 2025, pp. 1–6.

[14] ETSI, "Multi-access Edge Computing (MEC); Framework and Reference Architecture," European Telecommunications Standards Institute, Tech. Rep. ETSI GS MEC 003, Oct. 2022.

[15] J. Schulman *et al.*, "Proximal policy optimization algorithms," *CoRR*, 2017.