

Experimental Research of Hybrid Reinforcement Learning Methods for QoS-Aware Web Service Composition: an Actor-Critic Approach

Irina Grishanova^{1,*†}, Julia Rogushina^{1,2†} and Pylyp Andon^{1,†}

¹*Institute of Software Systems, National Academy of Sciences of Ukraine, 44 Glushkov Pr., Kyiv, 03680, Ukraine*

²*Institute for Digitalisation of Education, National Academy of Educational Sciences of Ukraine, 9 M. Berlynskoho Str., Kyiv, 04060, Ukraine*

Abstract

We investigate Reinforcement Learning (RL) approaches for the problem of constrained web service composition with quality-of-service (QoS) optimization. The task is formulated as a sequential decision-making problem in an environment where services are described by input/output parameters and QoS attributes, and compositions must satisfy functional and structural constraints. We propose a hybrid framework that integrates Reinforcement Learning with rule-based constraints, including action masking, no-repetition policies, and SLA-based limitations. This approach enables the reduction of the search space and improves the validity of generated service compositions. Comparison of existing RL methods, such as Q-learning, Deep Q-Network (DQN), Hybrid QoS-aware DQN, Actor-Critic, and Proximal Policy Optimization (PPO), shows that the Actor-Critic approach achieves faster convergence and more stable performance compared to value-based methods. While PPO theoretically provides more stable policy updates, in the considered constrained environment it does not demonstrate significant improvement and requires careful tuning. An important finding of this study is that in structured environments with explicit constraints, the advantages of advanced policy optimization methods such as PPO are significantly reduced. Simpler Actor-Critic architectures combined with domain knowledge are aimed at more efficient and robust solutions.

Keywords

Machine Learning, Web service composition, Multigoal planning, Actor-Critic, Proximal Policy Optimization

1. Introduction

Web service composition [1] is a fundamental problem in modern distributed and service-oriented systems, involving the automated construction of service workflows that satisfy a given goal. With the rapid growth of available services, this task becomes increasingly complex due to the combinatorial explosion of possible compositions [2].

In practical scenarios, service composition is not limited to functional correctness but must also account for non-functional properties, commonly referred to as Quality of Service (QoS), such as execution cost, duration, and reliability or rating [3]. This transforms the problem into a multi-objective optimization task, requiring a balance between competing criteria.

Traditional approaches, including graph-based search and planning methods [4], often struggle with scalability and adaptability in dynamic environments. In contrast, RL provides a promising framework for sequential decision-making [5], allowing an agent to learn optimal composition strategies through interaction with the environment.

However, applying pure RL methods to service composition introduces significant challenges. The action space is highly constrained, as services can only be applied when their input requirements are satisfied. Additionally, strict constraints such as Service Level Agreements (SLA) have to be respected,

2nd Workshop "Intelligent information technologies" UkrProg-IIT'2026 co-located with 16th International Scientific and Practical Programming Conference UkrPROG'2026, May 5-6, 2026, Kyiv, Ukraine

*Corresponding author.

† These authors contributed equally.

✉ i26031966@gmail.com (I. Grishanova); ladamandraka2010@gmail.com (J. Rogushina); andon@isofts.kiev.ua (P. Andon)

ORCID 0000-0003-4999-6294 (I. Grishanova); 0000-0001-7958-2557 (J. Rogushina); 0009-0000-1737-5579 (P. Andon)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

and invalid actions have to be avoided entirely. These characteristics make unconstrained, purely data-driven approaches inefficient and potentially infeasible.

To address these limitations, this paper proposes a hybrid approach that combines RL with rule-based constraints. In particular, we employ Actor-Critic methods as the primary learning framework. In addition, Proximal Policy Optimization (PPO) is considered as an extension to investigate whether more advanced policy optimization techniques provide additional benefits in constrained environments.

The main contributions of this work deal with: (1) a hybrid RL framework integrating rule-based constraints into the decision process; (2) a formal formulation of constrained web service composition as an MDP with explicit feasibility enforcement; (3) an empirical comparison between Actor-Critic and PPO in constrained environments; and (4) an experimental demonstration that Actor-Critic can outperform PPO when structural constraints are explicitly enforced.

2. Related Work and Problem Evolution

The web service composition problem in the traditional approach uses graph-based and planning techniques, where services are modeled as transitions between states defined by input and output parameters: they reduce the problem to a path-finding task in a directed graph. However, they suffer from scalability issues as the number of services increases. Semantic approaches, based on ontologies and formal knowledge representation, enable reasoning about service compatibility and functionality. They provide a structured way to represent domain knowledge, but often lack efficient mechanisms for optimizing QoS or handling large search spaces.

RL is applied to service composition as a means of adaptive optimization. Classical value-based methods, such as Q-learning [6], allow agents to learn optimal policies by estimating action values. The concept of constrained decision-making in RL is formally captured by the Constrained Markov Decision Process (CMDP) framework, which extends standard MDPs with additional constraints. However, many CMDP-based methods rely on Lagrangian relaxation and do not guaranty strict constraint satisfaction during learning.

Deep RL methods [7], including Deep Q-Networks [8], extend Q-learning to large state spaces, but often suffer from instability and difficulty in handling constraints. Actor-Critic methods [9], particularly Proximal Policy Optimization [10], provide improved stability and are better suited for problems with complex action spaces. Several research directions emerge to address constraints in RL: safe RL focuses on minimizing risk during learning, often allowing temporary violations of constraints [11], shielded RL introduces a rule-based layer that prevents unsafe or invalid actions [12], and neuro-symbolic approaches combine neural learning with symbolic reasoning. These approaches highlight the importance of integrating domain knowledge into RL, particularly in environments where correctness is critical.

Previous research by the authors has explored the application of ML and RL methods to web service composition, demonstrating both the potential and limitations of purely data-driven approaches. In [13], we analyze dynamic programming and machine learning techniques, including Q-learning for adaptive service composition. The study showed that while these methods can effectively incorporate QoS parameters, they require more flexible formulations to handle variable-length service compositions and complex constraints.

This line of research is developed in [14], where a Q-learning-based approach was proposed for constructing service composition flows toward a target state. The method incorporated mechanisms for avoiding loops and redundant services, as well as a memory-based strategy for incremental knowledge accumulation. The results indicated that even in value-based RL, there is a need for explicit modeling of structural knowledge, including valid transitions and undesirable action sequences. In parallel, research on personalized learning trajectories, such as [15, 16], emphasized the importance of explicitly representing goals, prerequisites, and domain constraints. These findings suggest that domain knowledge should be integrated not only in the evaluation stage but also in the construction of the state space and the action set. The next step in this research direction was presented in [17], where a hybrid

DQN-based approach was introduced. This work demonstrated that combining deep RL with rule-based constraints, adaptive learning mechanisms, and QoS optimization improves stability and generalization. In particular, the integration of rule-based filtering was shown to reduce TD-error variance and guide policy formation.

The results of previous studies reveal a clear progression: from classical machine learning approaches to Q-learning-based adaptive composition, to deep RL methods, and finally to hybrid RL frameworks incorporating rule-based constraints. They indicate that a purely data-driven RL is insufficient for solving constrained service composition problems. Instead, effective solutions require integration of domain knowledge, explicit handling of constraints, and more stable policy optimization methods. The proposed method can be interpreted as a form of shielded reinforcement learning, where a rule-based layer enforces action validity and safety constraints. However, in contrast to standard shielded RL approaches, the proposed framework incorporates domain-specific constraints related to service composition, including functional compatibility, QoS-aware optimization, and trajectory-level restrictions. This results in a structured decision space and significantly improved learning stability.

3. Main Concepts of the WSC Task

The web service composition problem is modeled as a sequential decision-making process, where each service is defined by a set of input and output parameters and a vector of QoS attributes. Let $A = \{a_1, a_2, \dots, a_N\}$ be the set of available services. Each service $a_i \in A$ is defined as

$$a_i = \langle I_i, O_i, q_i \rangle, \quad (1)$$

where I_i is the set of input signals (preconditions), O_i is the set of output signals, and $q_i = (q_i^{(1)}, q_i^{(2)}, \dots, q_i^{(K)})$ is the QoS vector.

Let U be the universal set of all possible signals. The state at time t is defined as a set of available signals

$$s_t \subseteq U. \quad (2)$$

Thus, each state represents the currently available knowledge. -- The environment state S is defined as a set of currently available signals $s_t \subseteq U$, where U is the universal set of all possible signals. For computational purposes, the state is encoded as a binary vector $z_t \in \{0, 1\}^{|U|}$, where

$$z_t[j] = \begin{cases} 1, & \text{if } u_j \in s_t, \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

At each step, the agent selects an action $a_t \in A$, where each action corresponds to the execution of a service. However, the effective action space is constrained by feasibility conditions. A service a_i is applicable in state s_t if $I_i \subseteq s_t$. Additionally, to avoid redundant transitions, $O_i \not\subseteq s_t$, i.e., the service must introduce at least one new signal that extends the current state s_t .

The transition function is defined as

$$s_{t+1} = \begin{cases} s_t \cup O_{a_t}, & \text{accumulative mode,} \\ O_{a_t}, & \text{replacement mode.} \end{cases} \quad (4)$$

In the experimental setup, the accumulative mode is used; however, the transition (replacement) mode is configurable and can be adjusted depending on the problem formulation.

Let $G \subseteq U$ be the set of target signals. The goal is achieved when $G \subseteq s_t$. For a trajectory $\tau = (s_1, s_2, \dots, s_T)$, the composite QoS vector is defined as

$$Q^{(k)}(\tau) = \begin{cases} \sum_{t=1}^T q_{a_t}^{(k)}, & \text{if } k \in \mathcal{K}_{sum}, \\ \frac{1}{T} \sum_{t=1}^T q_{a_t}^{(k)}, & \text{if } k \in \mathcal{K}_{avg}. \end{cases} \quad (5)$$

In the current configuration, Cost and Duration are aggregated using summation, while Rating is aggregated using average.

The reward is defined on the basis of normalized QoS values. For each QoS dimension,

$$\hat{q}^{(k)} = \frac{q^{(k)} - q_{min}^{(k)}}{q_{max}^{(k)} - q_{min}^{(k)}}. \quad (6)$$

For minimized parameters, the normalized value is inverted: $\hat{q}^{(k)} \leftarrow 1 - \hat{q}^{(k)}$. The reward is defined as

$$R_t = \sum_{k=1}^K w_k \cdot \hat{q}^{(k)} - c, \quad (7)$$

where w_k are importance weights, $\hat{q}^{(k)}$ are normalized QoS values, and c is a step penalty.

We use the following constraints: (1) no repetition constraint, where each service can be used at most once, $a_t \notin \{a_1, \dots, a_{t-1}\}$; (2) deadlock conditions, where the episode is terminated if no valid actions exist or the state does not change for M steps; and (3) SLA constraints. Let the composite QoS vector be $Q(\tau)$. SLA constraints are defined as

$$Q^{(k)}(\tau) \leq \theta_k^{max}, \quad k \in \mathcal{K}_{max}, \quad (8)$$

$$Q^{(k)}(\tau) \geq \theta_k^{min}, \quad k \in \mathcal{K}_{min}. \quad (9)$$

where θ_k^{max} and θ_k^{min} are the upper and lower bounds for the k -th QoS dimension, and $\mathcal{K}_{max}$ and $\mathcal{K}_{min}$ denote the sets of QoS indices corresponding to maximization and minimization objectives, respectively. Two types of SLA constraints are considered: hard constraints that terminate a trajectory and soft constraints that apply a penalty.

4. Problem Formulation

The problem is formulated as an MDP: $M = (\mathcal{S}, \mathcal{A}, P, R, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, P is the transition function, R is the reward function and γ is the discount factor. The goal of the work is to find a policy π that maximizes the expected cumulative reward:

$$\pi^* = \arg \max_{\pi} \mathbb{E} \left[\sum_{t=0}^T \gamma^t R_t \right], \quad (10)$$

subject to functional feasibility constraints, rule-based restrictions, and SLA-based QoS constraints.

The problem can also be formulated as a constrained optimization problem, where the objective is to maximize the expected cumulative reward while satisfying the constraint conditions:

$$\max_{\pi} \mathbb{E}_{\pi}[R(\tau)] \quad \text{subject to} \quad C_j(\tau) \leq d_j, \quad j = 1, \dots, J. \quad (11)$$

Here, π denotes the policy, τ is a trajectory generated by the policy, and $R(\tau)$ is the cumulative reward of the trajectory. The functions $C_j(\tau)$ represent constraint metrics (e.g., QoS-related criteria), and d_j are the corresponding threshold values. However, instead of relaxing constraints as in CMDP, this work enforces them directly through action masking, rule-based filtering, and early termination.

5. Proposed Method

We propose a hybrid RL framework that integrates Actor-Critic methods with rule-based constraints. The system combines a learning-based policy with explicit domain knowledge, ensuring both optimality and feasibility of the resulting service compositions.

The architecture consists of two main components: an RL agent (Actor-Critic) and a rule-based layer enforcing constraints such as action validity, no repetition, and SLA compliance. This hybrid design reduces the effective search space and improves training stability.

The interaction loop between the agent and the environment can be formalized as follows:

1. Observe current state s_t .
2. Apply rule-based filtering to obtain valid action mask M_t .
3. Compute policy distribution $\pi_\theta(a \mid s_t, M_t)$.
4. Sample action a_t .
5. Execute action (service) in the environment.
6. Receive next state s_{t+1} and reward r_t .
7. Store transition.
8. Update policy and value function.

The rule-based module enforces domain-specific constraints directly in the environment. Action validity rules allow an action a_t only if $I_{a_t} \subseteq s_t$ and $O_{a_t} \not\subseteq s_t$. This means that a service can be applied only if all its required input signals are already present in the current state s_t , and it produces at least one new signal that is not yet included in the state. The no-repetition constraint is defined as $a_t \notin \{a_1, \dots, a_{t-1}\}$. Deadlock detection terminates an episode when no valid actions exist or no progress is observed for a predefined number of steps.

SLA enforcement uses the composite QoS vector $Q(\tau)$. Hard constraints, $Q^{(k)}(\tau) > \theta_k^{max}$, terminate a trajectory. Soft constraints, $Q^{(k)}(\tau) < \theta_k^{min}$, apply a reward penalty $r_t \leftarrow r_t - \lambda$. Hard constraints define unacceptable regions of the solution space; if any QoS dimension exceeds the specified maximum threshold, the trajectory is immediately terminated. In contrast, soft constraints allow temporary deviations below the desired minimum levels but introduce a penalty to the reward, discouraging such actions without fully excluding them. This mechanism ensures that the agent prioritizes feasible and high-quality compositions while maintaining flexibility during exploration. As a result, the learning process is guided toward solutions that satisfy QoS requirements, balancing strict compliance with adaptability in constrained environments.

Action masking is implemented by modifying the output of the policy network before sampling. The network produces unnormalized action scores (logits) z_i for each action $a_i \in S$ in the action space. A binary mask $M_i \in \{0, 1\}$ indicates whether action a_i is valid in the current state s_t . Invalid actions are masked by assigning them a value of $-\infty$ prior to applying the softmax function:

$$\tilde{y}_i = \begin{cases} y_i, & \text{if } M_i = 1, \\ -\infty, & \text{if } M_i = 0, \end{cases} \quad \pi(a_i \mid s_t) = \text{softmax}(\tilde{y})_i. \quad (12)$$

Here, y_i denotes the unnormalized preference (logit) for action a_i , $\tilde{y}_i$ is the masked score, and M_i is the validity indicator derived from rule-based constraints. This ensures that invalid actions receive zero probability and cannot be selected by the policy.

The Actor-Critic method is used as the baseline learning approach. It consists of two components: an actor that represents the policy $\pi(a_i \mid s_t)$ and a critic that estimates the value function $V(s)$. The advantage function is defined as

$$A_t = G_t - V(s_t). \quad (13)$$

where A_t is the advantage estimate at time step t , G_t is the return (cumulative discounted reward) from time step t , and $V(s_t)$ is the value function estimating the expected return from state s_t .

The actor is updated using

$$L_{actor} = -\mathbb{E} [\log \pi(a_t \mid s_t) \cdot A_t], \quad (14)$$

where L_{actor} is the policy loss used to update the actor, encouraging the selection of actions with positive advantage and discouraging suboptimal actions, $\pi(a_t \mid s_t)$ is the policy, i.e., the probability of selecting action a_t in state s_t , and A_t is the advantage function estimating how much better the selected action is compared to the expected value of the state.

The critic is trained to minimize the value loss

$$L_{value} = \mathbb{E} [(V(s_t) - G_t)^2], \quad (15)$$

where $V(s_t)$ is the value function estimating the expected return from state s_t , and G_t is the observed return (cumulative discounted reward).

The final loss is defined as a weighted combination:

$$L = L_{actor} + \lambda_v L_{value} - \lambda_e H(\pi), \quad (16)$$

where λ_v and λ_e are weighting coefficients and $H(\pi)$ is the entropy of the policy, encouraging exploration by preventing premature convergence to deterministic policies.

This formulation allows for simultaneous policy improvement and value estimation, leading to efficient learning in structured environments.

As an extension, we also consider Proximal Policy Optimization (PPO), which introduces a clipped surrogate objective to stabilize policy updates by limiting large changes in the policy.

The probability ratio is defined as

$$r_t(\theta) = \frac{\pi_\theta(a_t | s_t)}{\pi_{\theta_{old}}(a_t | s_t)}, \quad (17)$$

where π_θ is the current policy, $\pi_{\theta_{old}}$ is the policy used to generate the trajectories, and $r_t(\theta)$ measures how much the probability of selecting action a_t in state s_t has changed.

The clipped surrogate objective L^{CLIP} (policy loss) is used to update the policy, ensuring that policy updates remain within a bounded range and is defined as

$$L^{CLIP} = \mathbb{E} [\min(r_t(\theta)A_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon)A_t)], \quad (18)$$

where A_t is the advantage function and ϵ is a hyperparameter controlling the allowed deviation from the previous policy. The clipping operation $\text{clip}(x, a, b)$ restricts the ratio $r_t(\theta)$ to the interval $[1 - \epsilon, 1 + \epsilon]$, preventing excessively large policy updates and improving training stability.

The final PPO objective (final loss) becomes

$$L = L^{CLIP} + \lambda_v L_{value} - \lambda_e H(\pi), \quad (19)$$

where L_{value} is the value loss (critic loss), $H(\pi)$ is the entropy term encouraging exploration, and λ_v , λ_e are weighting coefficients.

The proposed approach follows standard Actor-Critic and PPO formulations for policy and value updates. The key distinction is the incorporation of domain-specific constraints, implemented through rule-based action masking, trajectory restrictions, and SLA enforcement. These modifications alter the effective action space and guide the learning process toward feasible and high-quality solutions.

Unlike standard shielded RL approaches, the proposed method not only prevents invalid actions but also improves learning efficiency by reducing the effective search space and guiding the agent toward feasible trajectories.

As shown in Equation (7), the reward function integrates QoS optimization and trajectory efficiency. Optionally, a composite QoS-based reward is used, $R_t = f(Q(\tau_t)) - f(Q(\tau_{t-1}))$, which aligns learning with the final objective.

The basic Actor-Critic algorithm operates by simultaneously learning a policy and a value function. At each step, the agent interacts with the environment, collects transitions, computes the return, and updates both the actor and the critic using the advantage function.

The PPO-based algorithm follows a similar interaction pipeline with the environment, as illustrated in Fig. 1(b), but differs in the policy update mechanism. Instead of performing direct policy gradient updates using the advantage function, PPO introduces a clipped surrogate objective that constrains the magnitude of policy updates. This stabilizes training by preventing large deviations from the previous policy.

<pre> /* (a) Actor-Critic */ Initialize actor parameters θ Initialize critic parameters φ for each episode do: reset environment observe initial state s_0 for $t = 0 \dots T$ do: apply action mask M_t sample action $a_t \sim \pi_\theta(a s_t, M_t)$ execute action in environment observe reward r_t, next state s_{t+1}, terminal flag store transition (s_t, a_t, r_t, s_{t+1}), terminal if terminal: break compute returns G_t for all steps for each step t do: compute advantage: $A_t = G_t - V\varphi(s_t)$ update actor: $\theta \leftarrow \theta + \nabla \log \pi_\theta(a_t s_t) * A_t$ update critic: $\varphi \leftarrow \varphi - \nabla (V\varphi(s_t) - G_t)^2$ end for </pre>	<pre> /* (b) PPO */ Initialize policy parameters θ Initialize value function parameters φ for each iteration do: collect trajectories using current policy π_θ for each episode do: reset environment observe initial state s_0 for $t = 0 \dots T$ do: apply action mask M_t sample action $a_t \sim \pi_\theta(a s_t, M_t)$ execute action observe reward r_t and next state s_{t+1} store transition (s_t, a_t, r_t) if terminal: break compute returns G_t compute advantages A_t for K epochs do: update θ using PPO clipped objective update φ using value loss end for </pre>
---	--

Figure 1: Hybrid (a) Actor–Critic and (b) PPO algorithm for Service Composition.

In contrast to the Actor-Critic approach shown in Fig. 1(a), where the policy is updated directly based on the estimated advantage, PPO performs multiple optimization epochs over collected trajectories and uses a probability ratio between the current and previous policies. This results in more conservative updates but increases computational complexity.

In both approaches, the proposed hybrid modification incorporates rule-based constraints into the decision process. While the core difference between the methods lies in the policy optimization strategy, both are extended in this work through the integration of domain-specific constraints, forming a hybrid RL framework for service composition.

Figure 2 illustrates the differences in the learning process between the proposed Actor–Critic and PPO-based methods. Both approaches share the same interaction pipeline with the environment, which includes rule-based action masking and QoS-aware reward formulation.

In the classical setting, PPO introduces an additional mechanism based on a reference (previous) policy and a clipped surrogate objective, which constrains policy updates and improves training stability. In contrast, the Actor–Critic method performs direct policy updates based on the advantage function, resulting in faster but potentially less stable learning.

In the proposed framework, however, the service composition environment incorporates structural constraints through rule-based action masking and QoS-aware rewards. This reduces the effective action space and the variance of policy updates, partially mitigating instability issues typically associated with Actor–Critic methods.

6. Experimental Results

The proposed approach was evaluated on a dataset of 1000 web services (learning objects; see our previous works [13, 14, 15, 16, 17]), where each service is defined by a set of input signals, a set of output signals, and a QoS vector $q = (Cost, Duration, Rating)$. Cost and Duration are minimized, and Rating is maximized. The task consists of constructing a valid service composition trajectory from

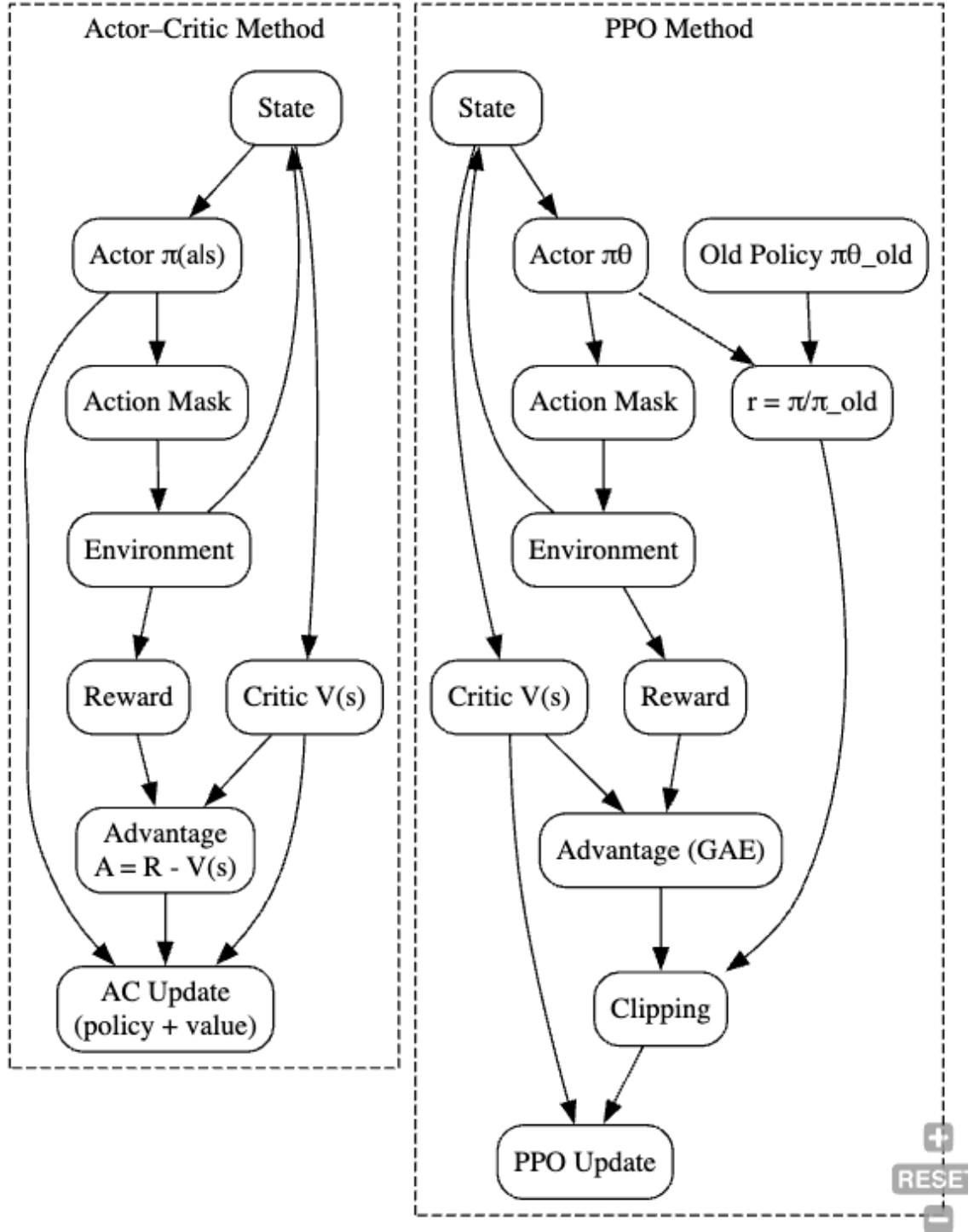


Figure 2: Algorithmic flow diagrams of the learning process for the proposed hybrid Actor-Critic and PPO methods with action masking and rule-based constraints.

an initial state to a predefined goal state.

The reward function is defined as Equation (7) with weights $w = [40, 10, 50]$ and step penalty $c = 1$. The following methods were evaluated: Hybrid Q-learning (tabular), Hybrid Rule-based DQN, Hybrid Rule-based Actor-Critic (A2C), and Hybrid Rule-based PPO.

Training results for proposed hybrid Actor-Critic and PPO methods are shown in Figures 3 and 4. The cumulative reward increases over training, indicating effective learning of the composition policy. The monotonic growth of the return curve demonstrates that the agent successfully improves its

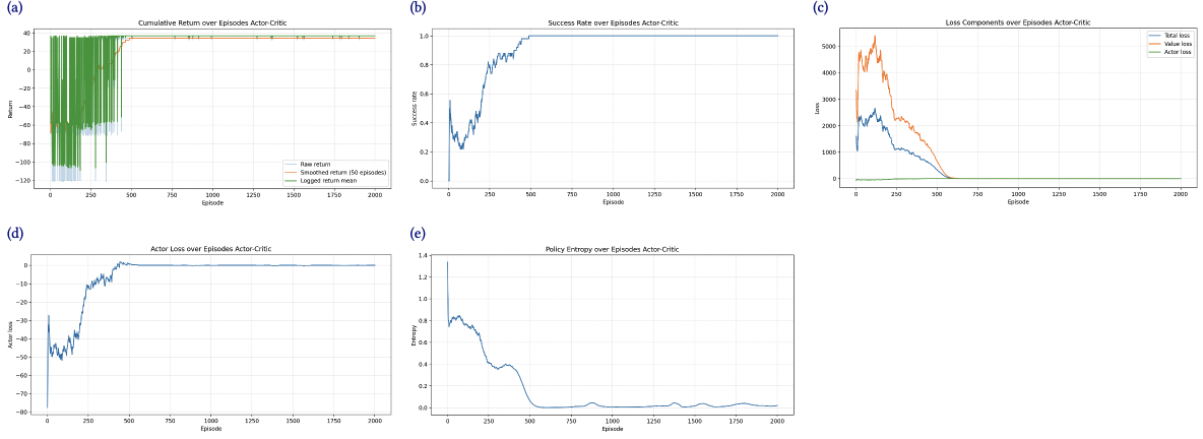


Figure 3: Training dynamics of the proposed method based on Actor-Critic: (a) cumulative reward, (b) success rate, (c) loss components, (d) actor loss, and (e) policy entropy.

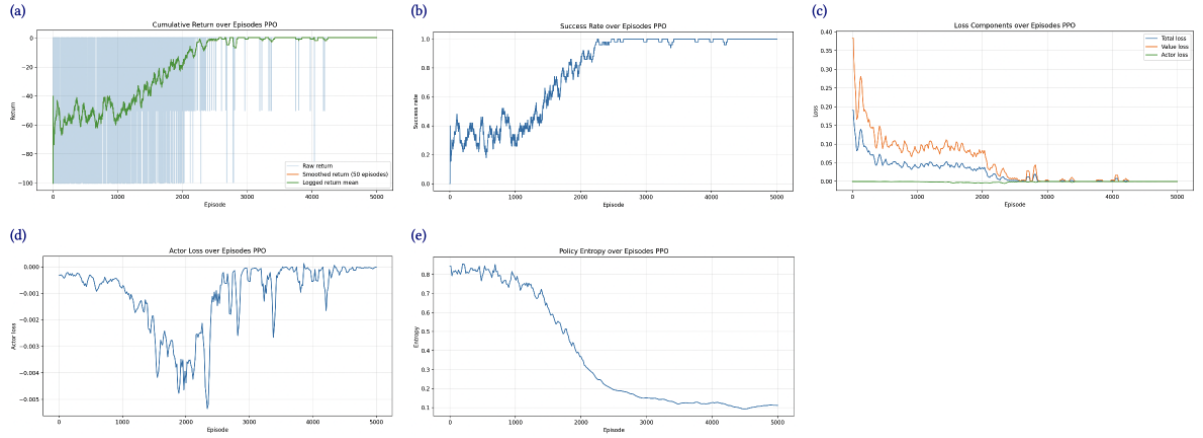


Figure 4: Training dynamics of the proposed hybrid PPO method: (a) cumulative reward, (b) success rate, (c) loss components, (d) actor loss, and (e) policy entropy.

decision-making strategy over time.

Loss stabilizes after initial fluctuations and the value loss decreases steadily. This indicates convergence of both the policy and value function. At the early stage, exploration dominates; in the middle stage, rapid policy improvement occurs; and at the late stage, convergence and stabilization are observed.

While PPO demonstrates more stable learning dynamics in theory, in the considered constrained environment it does not consistently outperform the Actor-Critic approach. In particular, PPO requires significantly more training iterations and careful hyperparameter tuning, and its performance remains less stable in the presence of strict constraints and sparse rewards. Importantly, the presence of rule-based constraints significantly stabilizes the learning process, reducing the variance typically observed in Actor-Critic methods. As a result, the advantage of PPO over Actor-Critic becomes less pronounced in constrained environments.

The similarity between total loss and value loss indicates that the actor contribution becomes negligible during training. Such behavior suggests that the policy updates are either saturated or suppressed due to clipping and low advantage variance, leading to limited policy improvement. In constrained environments, the critic may dominate the learning process, while the policy gradient becomes ineffective, reducing the practical benefit of PPO.

Experimental comparison shows that the Actor-Critic approach achieves faster convergence and comparable or higher QoS scores in early training stages. PPO demonstrates more stable learning

Table 1

Qualitative comparison of modified RL methods based on experimental results for QoS-aware web service composition.

Method	Avg Return	Convergence Speed	QoS Stability	Scalability	Notes
Q-learning modified	Low	Fast (small state)	Low	Poor	Tabular
HR-QoS-DQN	Medium	Medium	Medium	Medium	Improved
Actor-Critic modified	High	Fast	High	Good	Best result
PPO modified	Medium	Slow	Medium	Good	Requires tuning

dynamics but requires more training iterations and does not significantly outperform Actor-Critic in constrained environments. These findings suggest that simpler Actor-Critic architectures may be preferable for practical deployment in service composition systems.

In addition to the results presented in this paper, we conducted comparative experiments with our previous approaches, including hybrid Q-learning and hybrid DQN-based methods, which also incorporate rule-based constraints and QoS-aware mechanisms. The detailed experimental results are reported in our earlier works [13, 14, 15, 16, 17]. These results indicate that, in the considered setting, the Actor-Critic method tends to achieve faster convergence compared to value-based approaches such as DQN and HR-QoS-DQN. A qualitative summary of these findings is presented in Table 1, which compares the modified RL methods considered in this study and in our previous experiments.

Based on the experiments conducted, several key observations can be highlighted. First, the rule-based layer significantly improves convergence speed, trajectory validity, and training stability. Second, PPO does not outperform the Actor-Critic method in the considered constrained environment. Although PPO provides a theoretically more stable policy update through clipping, it typically requires more training iterations and careful hyperparameter tuning, which may increase computational cost and make its application less convenient in practice. In contrast, the Actor-Critic method demonstrates faster convergence and more stable practical performance when combined with rule-based constraints. Third, using QoS in the reward function enables direct optimization of service quality and aligns the learning objective with the final composition goal. Fourth, the combination of RL and rule-based constraints provides a significant advantage over purely data-driven approaches, as it integrates adaptive learning with explicit domain knowledge.

7. Conclusion

This paper presents an RL-based approach to the problem of constrained web service composition with QoS optimization. The study focuses on integrating machine learning techniques with domain-specific knowledge in the form of rule-based constraints.

The main contributions of this work are as follows. A hybrid environment model combines RL with explicit rule-based constraints. These constraints include functional compatibility of services, prevention of service repetition, and SLA-based QoS restrictions. This integration allows the agent to operate in a structured decision space and reduces the number of invalid actions. The effectiveness of Actor-Critic methods in constrained environments was demonstrated. This approach achieved fast convergence and high-quality solutions, consistently generating valid service composition trajectories with optimal QoS characteristics. The results show that while PPO provides theoretically more stable updates, its advantages are significantly reduced in environments where constraints are explicitly enforced. In such settings, the additional complexity of PPO does not lead to substantial performance gains.

Thus, the proposed hybrid approach offers a practical and efficient solution for QoS-aware service composition, balancing learning capability and domain knowledge.

In contrast to standard shielded RL approaches, which primarily focus on preventing unsafe actions, the proposed method integrates domain-specific constraints related to service composition, including functional compatibility, QoS-aware optimization, and trajectory-level restrictions.

Despite the obtained results, the study has revealed several limitations that should be considered in further research.

The experimental environment is simulated and assumes static service descriptions. However, the proposed framework allows for a certain degree of dynamic variability. In particular, changes in the action space (e.g., addition or removal of services), the state space (e.g., introduction of new signals or features) and QoS parameters can be accommodated without fundamental modifications to the model.

This is achieved due to the rule-based action masking mechanism and the state representation based on sets of signals, which enable flexible adaptation to structural changes in the environment. Preliminary observations from previous work indicate that moderate variations in service parameters do not significantly affect the performance of the approach. Nevertheless, fully dynamic environments require further investigation.

The reward function is based on a configurable aggregation of QoS attributes, which, in the experimental setup, follows a simplified formulation and may not fully capture complex multi-objective trade-offs.

The action space is discrete and structured due to rule-based filtering, reflecting the nature of the service composition problem. While this structure improves learning efficiency, it may affect the direct applicability of the results to environments with continuous or less constrained decision spaces.

Future research directions include the integration of ontology-based knowledge representation for more expressive modeling of service semantics and improved reasoning about compatibility and composition; the study of dynamic environments, where service availability and QoS parameters change over time, requiring adaptive and online learning mechanisms; and the development of explainable and knowledge-infused reinforcement learning methods.

More broadly, this work can be extended toward knowledge-infused learning approaches, where domain knowledge is systematically incorporated into the learning process. In particular, the integration of ontological models and rule-based reasoning with reinforcement learning opens opportunities for developing explainable and interpretable decision-making systems. This direction aligns with recent advances in explainable reinforcement learning, enabling not only improved performance but also transparency and reasoning capabilities in complex service composition tasks.

Overall, the proposed framework contributes to the development of machine learning methods based on knowledge-oriented models, combining data-driven learning with structured domain knowledge.

Declaration on Generative AI

During the preparation of this work, the authors used AI programs ChatGPT 4o and Copilot for grammar and spelling checks. After using these services, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

- [1] B. Benatallah, M. Dumas, Q. Z. Sheng, A. H. H. Ngu, Declarative composition and peer-to-peer provisioning of dynamic web services, in: Proc. 18th International Conference on Data Engineering, 2002, DOI: 10.1109/ICDE.2002.994738.
- [2] V. Uc-Cetina, F. Moo-Mena, R. Hernandez-Ucan, Composition of web services using Markov decision processes and dynamic programming, The Scientific World Journal, 2015, Article ID 545308. DOI: 10.1155/2015/545308
- [3] E. Zeng, B. Benatallah, A. H. H. Ngu, M. Dumas, J. Kalagnanam, H. Chang, QoS-aware middleware for Web services composition, IEEE Transactions on Software Engineering 30(5) (2004) 311–327, DOI: 10.1109/TSE.2004.11.
- [4] S. Dustdar, W. Schreiner, A survey on web services composition, International Journal of Web and Grid Services 1(1) (2005) 1–30, DOI: 10.1504/IJWGS.2005.007545.

- [5] R. S. Sutton, A. G. Barto, Reinforcement Learning: An Introduction, 2nd ed., MIT Press, Cambridge, MA, 2018. URL: <https://web.stanford.edu/class/psych209/Readings/SuttonBartoIPRLBook2ndEd.pdf>
- [6] C. J. C. H. Watkins, P. Dayan, Q-learning, Machine Learning 8(3–4) (1992) 279–292, DOI: 10.1007/BF00992698.
- [7] J. Fan, Z. Wang, Y. Xie, Z. Yang, A theoretical analysis of deep Q-learning, Proceedings of Machine Learning Research 120 (2020) 1–4, DOI: 10.48550/arXiv.1901.00137.
- [8] B. Jang, M. Kim, G. Harerimana, J. W. Kim, Q-learning algorithms: A comprehensive classification and applications, IEEE Access (2019), DOI: 10.1109/ACCESS.2019.2941229.
- [9] R. S. Sutton, D. A. McAllester, S. P. Singh, Y. Mansour, Policy gradient methods for reinforcement learning with function approximation, NeurIPS, 2000. URL: <https://papers.neurips.cc/paper/1713-policy-gradient-methods-for-reinforcement-learning-with-function-approximation.pdf>
- [10] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal Policy Optimization Algorithms, arXiv:1707.06347, 2017, DOI: 10.48550/arXiv.1707.06347.
- [11] J. Garcia, F. Fernandez, A comprehensive survey on safe reinforcement learning, Journal of Machine Learning Research 16 (2015) 1437–1480. URL: <https://www.jmlr.org/papers/volume16/garcia15a/garcia15a.pdf>
- [12] M. Alshiekh, R. Bloem, R. Ehlers, B. Konighofer, S. Niekum, U. Topcu, Safe reinforcement learning via shielding, AAAI Conference on Artificial Intelligence, 2018, DOI: 10.1609/aaai.v32i1.11797.
- [13] I. Grishanova, J. Rogushina, The technology of machine learning for a composite web service development, Problems in Programming 4 (2024) 3–13, DOI: 10.15407/pp2024.04.003.
- [14] I. Grishanova, J. Rogushina, Flow constructing and optimizing method for composite web service based on Q-learning, Problems in Programming 1 (2025) 82–93, DOI: 10.15407/pp2025.01.082.
- [15] I. Grishanova, J. Rogushina, AI planning for construction of personal learning pathways, ITS-2024: Information Technologies and Security, December 19, 2024, Kyiv, Ukraine, CEUR Workshop Proceedings 4068 (2024). URL: <https://ceur-ws.org/Vol-4068/paper15.pdf>
- [16] J. Rogushina, I. Grishanova, Use of semantic technologies and reinforcement learning in construction of personal learning pathways, in: Proceedings of the Artificial Intelligence in Science and Education AISE2025 International Conference, Chapter XIV, 2025, pp. 182–197, DOI: 10.33407/lib.NAES.id/748281.
- [17] I. Grishanova, J. Rogushina, P. Andon, Hybrid rule-based QoS-optimized modification of Deep Q-Learning Network: Properties and specifics, CEUR Workshop Proceedings, Vol-4049: Proceedings of the Workshop on Intelligent Information Technologies (UkrProg-IIT 2025) (2025) 176–188. DOI: 10.55056/ceur-ws.org/vol-4049/s_23_grishanova_rogushina_andon.pdf