Continuous-Time Reward Machines

Amin Falah[1] , **Shibashis Guha**[2] , and **Ashutosh Trivedi**[1]
[1]University of Colorado Boulder
[2]Tata Institute of Fundamental Research
amin.falah@colorado.edu, shibashis@tifr.res.in, ashutosh.trivedi@colorado.edu

Abstract

Reinforcement Learning (RL) is a sampling-based method for sequential decision-making, where a learning agent iteratively converges to an optimal policy by leveraging feedback from the environment in the form of scalar rewards. While timing information is often abstracted in discrete-time domains, time-critical applications—such as queuing systems, population processes, and manufacturing workflows—are naturally modeled as Continuous-Time Markov Decision Processes (CTMDPs). Since the seminal work of Bradtke and Duff, model-free RL for CTMDPs has become well-understood. In many practical settings, however, high-quality information about system rates—often derived from queuing theory—is available and could be exploited to accelerate learning. Yet classical RL algorithms for CTMDPs typically re-learn such parameters through sampling. We propose *Continuous-Time Reward Machines* (CTRMs), a framework that embeds reward functions and real-time state-action dynamics into a unified structure. CTRMs enable RL agents to navigate dense-time environments effectively, leveraging both reward shaping and counterfactual experiences to accelerate convergence. Empirical results demonstrate that CTRMs improve learning efficiency in time-critical settings.

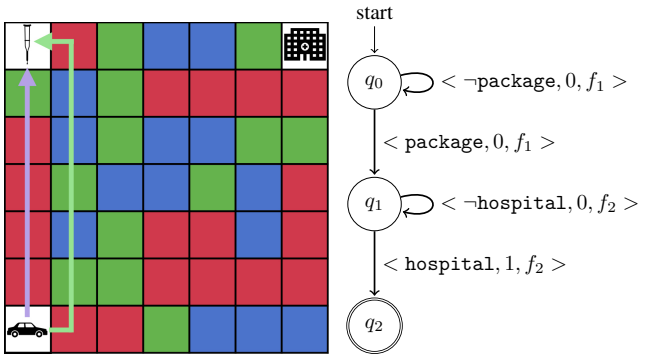
1 Introduction

Reinforcement Learning (RL) provides a versatile framework for sequential decision-making, enabling agents to learn optimal strategies in uncertain environments through feedback in the form of rewards. A central challenge in RL is designing reward functions that effectively encode task objectives, especially in safety-critical settings. Reward Machines [Icarte *et al.*, 2022] offer a programmatic way to specify (non-Markovian) rewards using finite-state machines. By employing them, designers can encode high-fidelity information about the environment, which agents can exploit during learning. As a result, reward machines support both reward shaping and reduced sample complexity. This work investigates their role in continuous-time domains.

RL for Continuous-Time Environments. Classic RL assumes an abstract notion of time in the interaction between an agent and its environment, typically modeled as discrete-time Markov decision processes (DTMDPs), where each interaction takes a fixed unit of time. However, in many real-world, time-critical applications—such as queuing systems, manufacturing processes, and population dynamics—actions incur variable delays. Continuous-time models, such as semi-MDPs [Baykal-Gürsoy, 2011] and continuous-time Markov decision processes (CTMDPs) [Guo and Hernández-Lerma, 2009], support decision-making over dense time, capturing the fluid and asynchronous nature of real-world interactions. CTMDPs, in particular, model the duration of state residence using exponential distributions governed by state-action-specific rates, thus providing a principled way to represent stochastic timing.

Separation of Concerns. Accurately estimating transition probabilities in real-world systems is challenging due to the complexity and variability of underlying processes, which often depend on latent factors and fluctuating conditions. In contrast, the timing of transitions—when governed by exponential rates—is typically more accessible and can often be derived from historical data or prior studies. These rates provide a compact yet expressive characterization of system dynamics and have been widely used in domains such as network traffic [Becchi, 2008], engineering reliability [O'Connor, 2011], and road traffic modeling [Oumaima *et al.*, 2020]. CTMDPs are well-suited for such systems, representing transitions through exponential sojourn times. A CTMDP can also be viewed as a DTMDP augmented with rate information for each state-action pair. This perspective motivates our approach: we decompose a CTMDP into a DTMDP and a finite-state automaton encoding rate information, enabling a clear *separation of concerns*. This decomposition isolates uncertainties associated with transition probabilities from those related to time delays. Consequently, different inference techniques can be applied: RL methods for estimating transition probabilities, and empirical or theoretical tools—such as those from queuing theory—for inferring rate parameters.

Continuous-Time Reward Machines. Reward machines [Icarte *et al.*, 2022] enable the encoding of non-Markovian reward mechanisms in discrete-time settings.



(a) (b)

Figure 1: (a) The grid world representing the autonomous driving environment and (b) CTRM specifying the rewards and rates.

Analogously, we *propose continuous-time reward machines* (CTRMs) to encode both non-Markovian rewards and rate mechanisms in continuous-time environments. CTRMs generalize systems with exponential sojourn times by allowing multiple rates for the same environment state, conditioned on regular properties over past event occurrences. For example, in an urban traffic system, the traversal speed of an ambulance may depend on whether its emergency lights are activated. This scenario can be modeled as a CTRM, with distinct states representing such conditions. A standard CTMDP is a special case of this framework, represented as a DTMDP paired with a single-state CTRM encoding the rates. In addition to rates, CTRMs encode rewards aligned with learning objectives, providing high-fidelity representations of environments with complex non-Markovian structure.

Example 1 (CTRMs)**.** *To illustrate the concept of CTRMs, consider an autonomous vehicle operating in an urban road network represented as a grid world (Figure 1a). Each grid cell corresponds to a traffic intersection, where the vehicle can move left, right, go straight, or make a U-turn. The stochastic dynamics of intersection traversal times are modeled using exponential distributions derived from historical traffic data (inspired by [Oumaima* et al.*, 2020]) and integrated into a reward machine (Figure 1b) to encode task objectives.*

The vehicle's task is to fetch medical supplies from a depot and deliver them to a hospital, starting from a distribution center. The reward machine encodes this task using three states: q_0 (navigating traffic without the package), q_1 (delivery phase with the package), and q_2 (accepting state upon task completion). Transitions in the CTRM are labeled with atomic propositions (e.g., `package` *and* `hospital`*), rewards, and rate functions. For instance, the transition to q_2 yields a positive reward, marking task completion. Rate functions f_1 and f_2 define average traversal times under varying traffic conditions, with f_2 prioritizing emergency deliveries. Grid cells are color-coded by average intersection time: red, blue, and green represent decreasing durations.*

Contributions. We consider three scenarios: (1) the environment and CTRM are entirely unknown (black-box RL), (2) the CTRM structure is known but rate information is unavailable, and (3) the complete CTRM is known. In all cases, the DTMDP environment is treated as unknown to the agent, allowing us to assess the impact of varying levels of prior information. Our key contributions are as follows:

1. We introduce *continuous-time reward machines* (CTRMs), modeled as finite-state automata whose transitions are triggered by environmental events. CTRMs enrich the modeling framework by encoding both state-specific sojourn rates and reward functions accessible to the learning agent.
2. Building on Q-learning for CTMDPs [Bradtke and Duff, 1994], we propose an algorithm for settings where the system is modeled as a DTMDP and objectives are specified via CTRMs. This method applies even when the agent lacks knowledge of the CTRM structure or rate information, relying solely on observed transitions and rewards. Modeling CTMDPs as DTMDPs paired with CTRMs yields a novel, modular abstraction for reasoning about continuous-time behavior.
3. Inspired by reward machines [Icarte *et al.*, 2022], we leverage the known structure of CTRMs to generate *counterfactual experiences*, allowing the agent to accelerate learning by simulating alternative outcomes. We examine two scenarios: (1) when both structure and rate information are available, and (2) when only structure and rewards are known. For the latter, we introduce a sampling-based approach to dynamically approximate rates—representing the first use of counterfactual reasoning and rate estimation in the CTRM framework.
4. The *reward shaping* technique of [Ng *et al.*, 1999] accelerates convergence in discrete-time settings by densifying rewards while preserving optimality. Extending reward shaping to continuous-time systems requires nontrivial adaptation. We propose a novel reward shaping scheme for CTRMs and formally prove its correctness under both known and unknown rate settings, addressing a gap not covered in prior work.
5. We empirically validate our approaches on a range of continuous-time benchmarks, demonstrating the benefits of each method across levels of prior knowledge.

2 Preliminaries

We use the symbols $\mathbb{N}, \mathbb{R}, \mathbb{R}_{\geq 0}, \mathbb{Q}$ and $\mathbb{Q}_{\geq 0}$ to denote the sets of natural numbers, real numbers, non-negative real numbers, rational numbers, and non-negative rational numbers, respectively. For $n \in \mathbb{N}$, we write $[n]$ for the set $\{1, \ldots, n\}$. The time domain is denoted by $\mathbb{T}$ and is defined as the set $\mathbb{R}_{\geq 0}$ of non-negative real numbers.

Given a finite set A, a *probability distribution* over A is a function $p\colon A \to [0,1] \cap \mathbb{Q}$ such that $\sum_{a \in A} p(a) = 1$ and $\mathsf{Supp}(p) = \{a \in A \mid p(a) > 0\}$ is the *support* of p. We denote the set of probability distributions on A by $\mathcal{D}(A)$.

Discrete-time MDP (DTMDP). A DTMDP is defined as a tuple $\mathcal{M} = (\mathsf{S}, s_0, \mathsf{Act}, P, \mathbf{AP}, L)$ where S is the finite set of states, $s_0 \in \mathsf{S}$ is the initial state, Act is the finite set of actions,

$P : \mathsf{S} \times \mathsf{Act} \to p(\mathsf{S})$ is the transition probability function, **AP** is the finite set of atomic proposition, and $L : \mathsf{S} \to 2^{\mathbf{AP}}$ is the labelling function that provides the set of atomic propositions that are true in a state. To denote the set of available actions in a state s, we use the notation $\mathsf{Act}(s)$.

Continuous-time MDP (CTMDP). A CTMDP [Puterman, 2014] is defined as $\mathcal{M} = (\mathsf{S}, s_0, \mathsf{Act}, P, \lambda, \mathbf{AP}, L)$, where the exit rate function $\lambda : \mathsf{S} \times \mathsf{Act} \to \mathbb{R}_{\geq 0}$ is introduced, while all other notations remain consistent with those of a DTMDP. Starting from the initial state s_0, an agent selects actions that lead to transitions between states according to the distribution P. The time spent in a state s is governed by an exponential distribution with exit rate $\lambda(s, a)$, which depends on the chosen action a. The probability of transitioning from state s to another state within time t under action a is given by the cumulative distribution function $F(t|s, a) = 1 - e^{-\lambda(s,a)t}$.

In a CTMDP, an infinite run is an ω-word: $(s_1, (\tau_1, a_1), s_2, (\tau_2, a_2), \ldots) \in \mathsf{S} \times ((\mathbb{R}_{\geq 0} \times \mathsf{Act}) \times \mathsf{S})^\omega$, where $s_i \in \mathsf{S}$, $a_i \in \mathsf{Act}(s_i)$, and τ_i is the time spent in s_i. Finite runs are sequences $(s_1, (\tau_1, a_1), \ldots, (\tau_{n-1}, a_{n-1}), s_n)$ for $n \in \mathbb{N}$. The sets of infinite and finite runs in $\mathcal{M}$ are denoted by $Runs^{\mathcal{M}}$ and $FR^{\mathcal{M}}$, and runs starting from a state s are $Runs^{\mathcal{M}}(s)$ and $FR^{\mathcal{M}}(s)$. For any $r \in FR^{\mathcal{M}}$, $last(r)$ represents its last state.

To handle non-determinism, a *scheduler* (or policy) $\sigma{:}FR^{\mathcal{M}} \to \mathcal{D}(\mathsf{Act})$ maps finite runs to probability distributions over actions available in $last(r)$. A scheduler is *deterministic* if it always chooses a single action (Dirac distribution), and *randomized* otherwise. It is *stationary* if decisions are always consistent for runs ending in the same state and *pure* if both deterministic and stationary. The set of all schedulers for $\mathcal{M}$ is $\Sigma_{\mathcal{M}}$. Under a scheduler σ, the CTMDP becomes a continuous-time Markov chain (CTMC), denoted $\mathcal{M}^\sigma$, where the non-determinism is resolved and transitions depend only on the current state, not on actions. The cumulative distribution function under σ is $F_\sigma(t|s) = F(t|s, \sigma(s))$. The sets of infinite and finite runs in $\mathcal{M}^\sigma$ are $Runs_\sigma^{\mathcal{M}}$ and $FR_\sigma^{\mathcal{M}}$, analogous to their CTMDP counterparts.

The behavior of $\mathcal{M}$ under scheduler σ starting from state s is defined over a probability space $(Runs_\sigma^{\mathcal{M}}(s), \theta, \Pr_\sigma^{\mathcal{M}}(s))$. Here, the sample space $Runs_\sigma^{\mathcal{M}}(s)$ is the set of runs from state s, θ is the σ-algebra, and $\Pr_\sigma^{\mathcal{M}}(s)$ is the probability measure. The σ-algebra θ is defined such that for each $\pi \in FR_\sigma^{\mathcal{M}}(s)$ contains the set

$$Cyl_\sigma(\pi) = \{\eta \in Runs_\sigma^{\mathcal{M}}(s) | \exists \eta' \in Runs_\sigma^{\mathcal{M}} : \eta = \pi \cdot \eta'\},$$

called the cylinder set of the finite path π. The unique probability measure $\Pr_\sigma^{\mathcal{M}}(s)$ is defined such that for each $\pi = (s_1, \tau_1, \ldots, \tau_{n-1}, s_n) \in FR_\sigma^{\mathcal{M}}(s)$, it holds that

$$\Pr_\sigma^{\mathcal{M}}(s)(\pi) = \prod_{i=0}^{n-1} P_\sigma(s_i, s_{i+1}) \cdot F_\sigma(\tau_i | s_i).$$

Given a random variable $f : Runs_\sigma^{\mathcal{M}} \to \mathbb{R}$, we use $\mathbb{E}_\sigma^{\mathcal{M}}(s)\{f\}$ to denote the expectation of f over the runs of $\mathcal{M}^\sigma$. Lastly, for $n \geq 1$, we denote X_n, Y_n, D_n, and T_n as the random variables representing the n-th state, the action taken from the n-th state, time-delay in the n-th state, and cumulative time up to the n-th state, respectively, with $D_0 = T_0 = 0$.

Discounted Reward Objective. A rewardful CTMDP, $(\mathcal{M}, rew_D, rew_C)$, extends a CTMDP with reward functions $rew_D : \mathsf{S} \times \mathsf{Act} \times \mathsf{S} \to \mathbb{R}$ for discrete rewards and $rew_C : \mathsf{S} \times \mathsf{Act} \times \mathsf{S} \to \mathbb{R}$ for continuous rewards. Taking action a from state s to s' in t time units yields a reward $rew_D(s, a, s') + t \cdot rew_C(s, a, s')$, discounted by a factor $e^{-\alpha t}$, where $\alpha > 0$ is the discount parameter. The expected discounted reward $\mathsf{DR}^{\mathcal{M}^\sigma}(s, \alpha)$ from $s \in S$ under $\sigma \in \Sigma_{\mathcal{M}}$ is:

$$\mathbb{E}_\sigma^{\mathcal{M}}(s)\Big[\sum_{n=1}^{\infty} e^{-\alpha T_{n-1}} \big(rew_D(X_n, Y_n, X_{n+1}) + \int_{T_{n-1}}^{T_n} e^{-\alpha(t - T_{n-1})} rew_C(X_n, Y_n, X_{n+1}) dt\big)\Big].$$

Here, $rew_D(X_n, Y_n, X_{n+1})$ is the discrete reward for transitioning from X_n to X_{n+1} via action Y_n, while $rew_C(X_n, Y_n, X_{n+1})$ is the continuous reward accrued over $t - T_{n-1}$, both discounted by $e^{-\alpha T_{n-1}}$.

3 Continuous-Time Reward Machines

A reward machine (RM) [Icarte *et al.*, 2022] is a finite state machine defined as $\mathcal{R}_M = (\mathcal{U}, u_0, \mathcal{F}, \delta_u, \delta_r)$, where:

- $\mathcal{U}$ is the set of states, u_0 is the initial state, and $\mathcal{F} \subseteq \mathcal{U}$ is the set of final states.
- $\delta_u : \mathcal{U} \times 2^{\mathbf{AP}} \to \mathcal{U}$ is the transition function, and $\delta_r : \mathcal{U} \to [\mathsf{S} \times \mathsf{Act} \times \mathsf{S} \to \mathbb{R}]$ maps each state to a discrete reward function.

Each RM state maps transitions to rewards. Final states, though optional, can mark task completion. We extend RMs with timing rates per state-action pair, forming continuous-time reward machines.

A *continuous-time reward machine* (CTRM) is defined as $\mathcal{R}_C = (\mathcal{U}, u_0, \mathcal{F}, \delta_u, \delta_r, \delta_c, \beta_r)$, extending the RM with:

- $\delta_c : \mathcal{U} \to [\mathsf{S} \times \mathsf{Act} \times \mathsf{S} \to \mathbb{R}_{\geq 0}]$, a continuous reward function based on state and transitions in the MDP; and
- $\beta_r : \mathcal{U} \to [\mathsf{S} \times \mathsf{Act} \to \mathbb{R}_{\geq 0}]$, a rate function specifying the timing behavior of state-action pairs, varying with the state of the CTRM.

These extensions allow CTRMs to capture timing behavior and event-dependent changes in the environment, providing a richer framework for modeling continuous-time systems.

3.1 Product Construction

Given an MDP $\mathcal{M}{=}(\mathsf{S}, s_0, \mathsf{Act}, P, \mathbf{AP}, L)$ and a CTRM $\mathcal{R}_C{=}(\mathcal{U}, u_0, \mathcal{F}, \delta_u, \delta_r, \delta_c, \beta_r)$, we define their product $\mathcal{M}_{\mathcal{R}_C}$ as a CTMDP with the form $((\mathsf{S}', s_0', \mathsf{Act}', P', \lambda, L', \mathbf{AP}'), rew_D, rew_C)$, where:

- $\mathsf{S}' = \mathsf{S} \times \mathcal{U}$; $s_0' = (s_0, u_0)$; $\mathsf{Act}' = \mathsf{Act}$;
- $P'((s, u), a, (s', u'))$ equals $P(s, a, s')$ if $u \in \mathcal{F}$ and $u' = u$, or $u \notin \mathcal{F}$ and $u' = \delta_u(u, L(s))$, and equals 0, otherwise;
- $L'(s, u) = L(s)$; $\mathbf{AP}' = \mathbf{AP}$;
- exit rate $\lambda((s, u), a)$ equals $\beta_r(u)(s, a)$ if $a \in \mathsf{Act}'(s)$, and is 0 otherwise; and

- the reward functions are defined as:

$$rew_D((s,u),a,(s',u')) = \delta_r(u)(s,a,s'), \text{ and}$$
$$rew_C((s,u),a) = \delta_c(u)(s,a).$$

3.2 Q-Learning

The extension of Q-learning [Watkins and Dayan, 1992] from DTMDP environments to CTMDPs was proposed in [Bradtke and Duff, 1994]. The algorithm estimates the Q-value, representing the expected total discounted reward for starting in state s, taking action a, and subsequently following policy σ. The Q-value for a discounted sum objective with discount factor α is given as:

$$\mathcal{Q}_\sigma(s,a)=r(s,a)+\frac{\lambda(s,a)}{\lambda(s,a)+\alpha}\sum_{s'\in\mathsf{S}}P(s,a,s')\mathcal{Q}_\sigma(s',\sigma(s')),$$

where $r(s,a)$ is defined as

$$\sum_{s'\in\mathsf{S}}P(s,a,s')\cdot rew_D(s,a,s')+\frac{rew_C(s,a,s')}{\alpha+\lambda(s,a)}.$$

The optimal Q-value, $\mathcal{Q}^*(s,a)$, can be characterized as

$$r(s,a)+\frac{\lambda(s,a)}{\lambda(s,a)+\alpha}\sum_{s'\in\mathsf{S}}P(s,a,s')\max_{a'\in\mathsf{Act}}\mathcal{Q}^*(s',a').$$

Q-learning uses stochastic approximation to estimate $\mathcal{Q}^*$ [Sutton and Barto, 2018]. For a transition (s,a,s') with delay τ, the Q-value update is:

$$\mathcal{Q}^{(k+1)}(s,a) := (1-\theta_k)\mathcal{Q}^{(k)}(s,a)+\theta_k\Big(\mathrm{rew}(s,a,s') + e^{-\alpha\tau}\max_{a'}\mathcal{Q}^{(k)}(s',a')\Big), \quad (1)$$

where $\mathrm{rew}(s,a,s')=\mathrm{rew}_D(s,a,s')+\frac{1-e^{-\alpha t}}{\alpha}\mathrm{rew}_C(s,a,s')$, and $\theta_k\in[0,1]$ is the learning rate. When the rate parameter λ is known, we use this information instead of the sampled delay τ, and the Q-update becomes:

$$\mathcal{Q}^{(k+1)}(s,a) := (1-\theta_k)\mathcal{Q}^{(k)}(s,a)+ \theta_k\Big(r(s,a)+\frac{\lambda(s,a)}{\lambda(s,a)+\alpha}\max_{a'}\mathcal{Q}^{(k)}(s',a')\Big). \quad (2)$$

Equation 2 differs from Equation 1 in both the reward function and the discounting approach. Equation 1 uses rew, which is based on sampled times, while Equation 2 uses r, reflecting the expected time spent. Similarly, the discounting in Equation 1 relies on sampled times, whereas in Equation 2, it is computed using the expected timing behavior of the model. As the objective is to estimate the expected discounted value of a state-action pair, Equation 2 converges faster by leveraging the known rate information for more precise updates.

4 CTRM Based Reinforcement Learning

In this section, we explore three reinforcement learning approaches for solving the product CTMDP, each utilizing different levels of information from the CTRM.

The first approach applies a standard RL algorithm for CTMDPs, treating the problem as a black box. It does not exploit the structure of the CTRM and instead relies solely on observed transitions and rewards to learn an optimal policy. While general-purpose, this method requires extensive exploration due to its lack of prior knowledge about the environment's dynamics.

The second approach leverages the known structure of the CTRM to improve learning efficiency. By generating counterfactual experiences based on CTRM transitions, the agent significantly expands its experience set, reducing reliance on exhaustive exploration. Additionally, embedded rate information is used to estimate expected state residence times, enabling more informed policy updates and faster convergence. We focus on pure schedulers, which suffice to achieve optimality under the discounted objective [Puterman, 2014].

The third approach introduces a novel reward shaping (RS) framework specifically designed for CTRMs. Unlike prior work on potential-based RS for DTMDPs with reward machines [Icarte *et al.*, 2022; Ng *et al.*, 1999], extending these techniques to CTRMs is non-trivial due to continuous-time dynamics and varying rate parameters. To address these challenges, we propose an automated RS scheme that is both practical and provably optimal. Our approach accommodates scenarios in which rate information is either known or unknown to the agent, marking a significant advancement in reward shaping for continuous-time systems.

Next, we provide a detailed explanation of these approaches, with an emphasis on the theoretical foundations and empirical performance of the proposed RS framework.

4.1 RL on Product CTMDP

In the previous section, we discussed RL algorithms for CTMDPs. When a CTRM is combined with an unknown DTMDP environment, their product yields a CTMDP, as described in Section 3.1. Our initial approach applies the RL method from [Bradtke and Duff, 1994] directly to this product CTMDP. This method uses a tabular RL algorithm that iteratively updates Q-values for state-action pairs in the product CTMDP. An episode—defined as a sequence of interactions between the agent and the environment starting from the initial state—consists of l steps in which the agent selects actions and observes outcomes. Over k episodes, the algorithm selects actions using an RL policy (e.g., ϵ-greedy), observes the next state and a sampled time τ, and updates Q-values according to Equation 1. At the end of training, the optimal policy is extracted by selecting, for each state, the action with the highest Q-value.

4.2 Counterfactual Experience RL for CTRMs

While the previous method provides an effective RL algorithm for CTMDPs, it does not fully utilize the known structure of the CTRM. Inspired by [Icarte *et al.*, 2022], we propose a counterfactual experience-based RL algorithm that leverages the transition dynamics of the CTRM to generate multiple experiences from a single transition in the unknown environment. Consider an unknown DTMDP environment $\mathcal{M}$ paired with a CTRM $\mathcal{R}_C$. Each step taken by the agent provides the information (s,u,a,τ,r,s',u'), where:

- s' is the next state observed when taking action a from state s in $\mathcal{M}$,
- u' is the next state in the CTRM, determined by the transition $\delta_u(u, L(s))$,
- τ is the sampled time, based on the rate $\beta_r(u)(s,a)$, and
- r is the reward, computed as

$$r = \delta_r(u)(s,a,s') + \tau \cdot \delta_c(u)(s,a).$$

As noted in [Icarte *et al.*, 2022], only the transition (s,a,s') is unknown; the remaining information (u,r,τ,u') is determined by the known CTRM structure. By leveraging this, the agent generates counterfactual experiences—alternative outcomes that simulate multiple scenarios. For each observed transition (s,a,s'), the agent generates the corresponding (u,r,τ,u') for each state $u \in \mathcal{U}$, where $\tau = \frac{1}{\beta_r(u)(s,a)}$, representing the expected time spent based on the rate. We define the set of counterfactual experiences as:

$$CE(s,a,s') = \bigcup_{u \in \mathcal{U}} \{(s,u,a,\tau,rew(s,a,s'),s',\delta_u(u,L(s)))\},$$

where $rew(s,u,a,s') = \delta_r(u)(s,a,s') + \frac{\delta_c(u)(s,a)}{\beta_r(u)(s,a)+\alpha}$.

Counterfactuals via Sampling. When the rates $\beta_r(u)(s,a)$ are unknown but the CTRM structure is available, the agent approximates the expected time spent in a state by sampling transition times over n trials and averaging the results, $\frac{1}{\hat{\beta}_r(u)(s,a)} = \frac{1}{n}\sum_{i=0}^{n-1}\tau_i$. Using this approximation, the agent generates counterfactual experiences (u,r,τ,u'), where

$$r = \delta_r(u)(s,a,s') + \frac{\delta_c(u)(s,a)}{\hat{\beta}_r(u)(s,a) + \alpha}$$

and $u' = \delta_u(u, L(s))$. This approach enables counterfactual updates based on the CTRM structure, even without direct access to rates. These counterfactual updates involve using Equation 2 for Q-value updates, as the approximated rate is used for each experience. The Q-values are updated for all counterfactual experiences generated, ensuring efficient learning. Details of the algorithm and its empirical evaluation are provided in Section 5.

4.3 Reward Shaping for CTRMs

Reward shaping (RS), introduced in [Ng *et al.*, 1999], is a technique for DTMDPs that assigns additional rewards to ensure the same optimal solutions for the discounted objective as in the original system while giving giving faster convergence. By introducing intermediate rewards, this method reduces the sporadic nature of rewards, creating a more continuous signal to guide the agent effectively toward its goals. This smoother feedback accelerates learning and mitigates sequences of uninformative actions.

The approach in [Ng *et al.*, 1999] defines an RS function $F(s,a,s') = \gamma\phi(s') - \phi(s)$, where γ is the discount factor and ϕ is a potential function over states. Adding F to the original rewards preserves the optimal policy's value. This idea was extended in [Icarte *et al.*, 2022] to reward machines for discrete systems, where potential functions were computed via value iteration over the reward machine.

Adapting this approach to CTMDPs, however, is more challenging. Unlike DTMDPs, CTMDPs involve continuous reward rates, as described in Section 2, where rewards and discounting depend not only on transitions but also on the time spent in each state. To address this complexity, we propose a novel RS function specifically for CTRMs and provide theoretical guarantees of its correctness, ensuring it preserves the optimal policy under the discounted reward objective.

As outlined in Section 3.2, our RL algorithm uses two Q-learning update equations (Equations 1 and 2) depending on the availability of rate information. The reward shaping (RS) function is accordingly defined in two forms. Below, we describe the general structure of the RS function and its variations based on the specific RL implementation.

Formally, the RS function comprises of two components—one for scalar rewards and one for continuous rewards—defined as $F_S : \mathsf{S} \times \mathsf{Act} \times \mathbb{T} \times \mathsf{S} \to \mathbb{R}$ and $F_C : \mathsf{S} \times \mathsf{Act} \times \mathbb{T} \times \mathsf{S} \to \mathbb{R}$. For a transition (s,a,τ,s'), where τ is the sampled time, the RS functions are given by:

$$F_S(s,a,\tau,s') = \gamma(s,a,\tau)\cdot\phi_S(s') - \phi_S(s),$$

$$F_C(s,a,\tau,s') = \frac{1}{\omega(s,a,\tau)}\big(\gamma(s,a,\tau)\cdot\phi_C(s') - \phi_C(s)\big),$$

where ϕ_S and ϕ_C are potential functions over states. The functions $\gamma : \mathsf{S} \times \mathsf{Act} \times \mathbb{T} \to \mathbb{R}$ and $\omega : \mathsf{S} \times \mathsf{Act} \times \mathbb{T} \to \mathbb{R}$ vary based on the Q-learning update equation.

RS when the rates are unknown. When the rate $\lambda(s,a)$ is unknown to the agent, we use the Q-update equation 1. In this case, the functions γ and ω are defined based on the sampled time τ as $\gamma(\tau) = e^{-\alpha\tau}$ and $\omega(\tau) = \frac{1-e^{-\alpha\tau}}{\alpha}$, here α is the discount parameter. These definitions rely on observed transitions, ensuring that the RS functions integrate seamlessly with the RL algorithm.

RS when the rates are known. If the rate $\lambda(s,a)$ is known, we use the Q-update equation 2. Here, γ and ω are based on the expected behavior of the model rather than the sampled time. Specifically, $\gamma(s,a) = \frac{\lambda(s,a)}{\alpha+\lambda(s,a)}$ and $\omega(s,a) = \frac{1}{\alpha+\lambda(s,a)}$. This approach leverages the additional rate information to improve accuracy.

Theorem 1. *Given a CTMDP $\mathcal{M}$ with scalar and continuous reward functions rew_D and rew_C, and reward shaping functions F_S and F_C, the optimal policies for the discounted reward objective remain unchanged. Specifically, for the rewardful CTMDPs $\mathcal{M}_R = (\mathcal{M}, rew_D, rew_C)$ and $\mathcal{M}'_R = (\mathcal{M}, rew_D + F_S, rew_C + F_C)$, the optimal policies are identical.*

Proof Sketch. In summary, the RS functions F_S and F_C are designed to ensure that the cumulative reward difference between any two policies remains invariant. This invariance guarantees that the ranking of policies under the discounted reward objective is unaffected, preserving the optimal policy. This result is established through algebraic manipulation of the Q-value equations for both cases, demonstrating that the additional rewards introduced by F_S and F_C do not alter the value of the optimal policy. □

Theorem 1 establishes that adding rewards of the form F_S and F_C does not alter the optimal policies. We now discuss how to construct such a reward shaping function within a reinforcement learning (RL) framework.

As defined, F_S and F_C for a transition (s, a, t, s') depend on the exit rate $\lambda(s, a)$ or the sampled time t, as well as potential functions ϕ_S and ϕ_C. The CTRM framework provides the agent with either the rate associated with each state-action pair or with the sampled transition time, enabling computation of $\gamma(s, a, t)$ and $\omega(s, a, t)$ dynamically during execution.

To compute the potential functions ϕ_S and ϕ_C, we adopt the approach from [Icarte *et al.*, 2022], applying value iteration over the reward machine with a fixed discount factor. This process assigns a value $V(q)$ to each state q in the reward machine, which is then used to define $\phi_S(s, q)$ and $\phi_C(s, q)$ for each product CTMDP state (s, q). Specifically, we set $\phi_S(s, q) = \phi_C(s, q) = V(q)$, where $V(q)$ is obtained through value iteration on the CTRM.

For simplicity, we assume here that the rewards δ_c and δ_r are functions of the form $\mathcal{U} \times 2^{\mathbf{AP}} \to \mathbb{R}$. This simplification is necessary in the absence of prior knowledge about the environment's states for value iteration. However, if specific reward values for each state in the reward machine are known in advance, this restriction is not needed.

Reward shaping aims to enhance the guidance provided to the agent by adjusting the rewards assigned by the reward machine. Instead of relying solely on the original rewards, δ_c and δ_r, shaping introduces adjusted rewards: $\delta_c(s, a, s') + F_C(s, a, s')$ for continuous rewards and $\delta_r(s, a, s') + F_S(s, a, s')$ for discrete rewards. Since reward shaping only modifies rewards, it integrates seamlessly with both RL approaches discussed earlier.

5 Experimental Results

In this section, we evaluate the performance of our proposed approaches across benchmark environments to assess their efficiency and effectiveness[1]. Each benchmark consists of an unknown DTMDP environment and a CTRM specifying the objective, with different levels of information available to the learning agent. We compare three approaches: (1) *Baseline RL*, which applies tabular Q-learning to the product CTMDP without leveraging CTRM information, (2) *Counterfactual RL*, which utilizes the CTRM structure to generate counterfactual experiences, and (3) *Counterfactual RL with Sampling*, which estimates rates dynamically when only the CTRM structure is known. Each method is evaluated both with and without reward shaping (RS). While we use tabular Q-learning to isolate the effects of CTRMs, counterfactual updates, and shaping, the framework is modular and compatible with deep RL and model-based CTMDP solvers, as it operates purely on sampled transition data.

For all experiments, we set the discount parameter $\alpha = 0.001$ and learning rate $\theta = 0.1$. The RL algorithms follow an ϵ-greedy exploration strategy, with ϵ initially set to 0.7 and decaying exponentially at a rate of 0.01 to transition from exploration to exploitation. Figure 2 presents the performance of each approach across four benchmarks. The Y-axis represents the average reward per step as a ratio of the optimal value (normalized to 1 using value iteration), while the X-axis shows the number of episodes. We conduct 10 independent runs and report the median performance with shaded regions indicating the 25th and 75th percentiles. Each graph includes results for six approaches: the three methods with and without reward shaping. Next, we provide descriptions and performance analysis of our benchmarks.

1) Autonomous Vehicle in Urban Environment. This benchmark, introduced in Section 1, models an autonomous vehicle navigating a stochastic gridworld (121 states) with varying traffic rates. The CTRM (3 states) encodes the objective of collecting a package and delivering it to a hospital. As shown in Figure 2(a), the baseline RL approach struggles, even with reward shaping. Counterfactual RL methods perform significantly better, with reward shaping further accelerating convergence. Notably, counterfactual RL with sampling performs comparably to the full-information variant, despite lacking rate information.

2) Firefighter and Traffic Coordination. This benchmark models an urban scenario where a firetruck and a car must navigate interdependent routes to reach their respective destinations. The firetruck's movement initially slows traffic, impacting the car's progress until the path is cleared. The CTRM encodes this objective by rewarding successful sequential task completion and adjusting movement rates based on traffic dynamics. The environment comprises 64 states with a 4-state CTRM. As shown in Figure 2(b), all methods perform poorly without reward shaping. However, with reward shaping, performance improves significantly, with counterfactual approaches achieving superior results.

3) Synchronized Firefighter and Traffic System. This benchmark extends the Firefighter and Traffic Coordination scenario by introducing synchronous decision-making, where both the firetruck and the car select their actions simultaneously rather than sequentially. The CTRM encodes the objective of reaching their respective destinations while accounting for the dynamic traffic conditions influenced by their joint movement. As shown in Figure 2(c), counterfactual methods outperform the baseline RL approach by effectively utilizing structural information. However, unlike other benchmarks, reward shaping does not provide a significant acceleration.

4) Treasure Hunt. Inspired by [Rens and Raskin, 2020], this benchmark models an amateur treasure hunter exploring a 121-cell gridworld to locate a hidden treasure and sell it for profit. The agent must first obtain a map to reveal the treasure's location. After acquiring the map, the agent can choose between two options: purchasing basic digging tools at a lower cost or investing in both tools and a horse, which allows for faster travel but at a higher expense. The CTRM (7 states) encodes these sequential objectives and associated rewards, with different travel rates depending on the agent's choices. As shown in Figure 2(d), counterfactual approaches perform well with or without reward shaping. However, the baseline RL method improves significantly with reward shaping but struggles without it.

[1]Our implementation can be accessed at: https://github.com/falahamin1/Continuous-Time-Reward-Machines.git

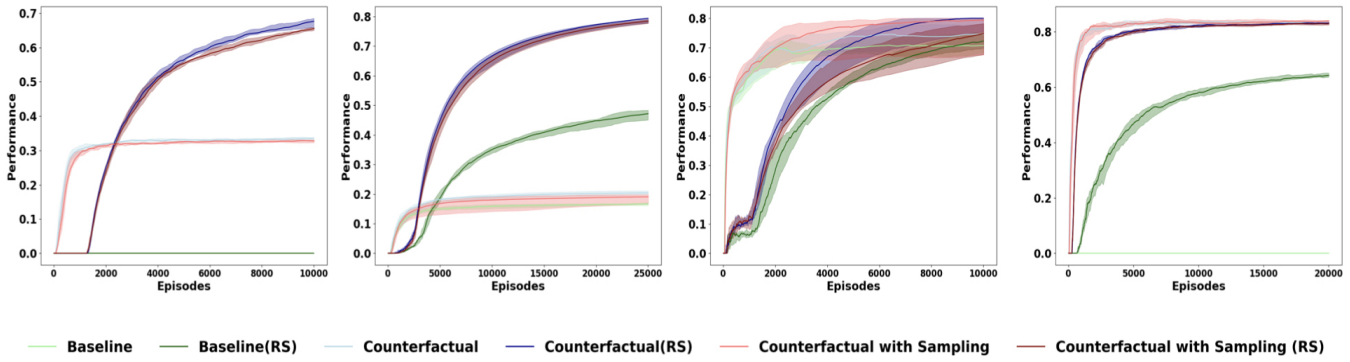


Figure 2: Comparison of the three reinforcement learning approaches—Classic, Counterfactual, and Counterfactual with Sampling—across the following benchmarks: (a) autonomous vehicle in an urban environment, (b) firefighter and traffic coordination, (c) synchronized firefighter and traffic system, and (d) treasure hunt environment.

Key Takeaways. Our experimental evaluation highlights several key trends. Counterfactual approaches consistently outperform the baseline RL method across all benchmark environments. Reward shaping is shown to play a pivotal role in accelerating learning and enhancing convergence in most scenarios, but does not show much improvement for benchmarks with small CTRMs. Finally, counterfactual RL with sampling, despite lacking rate information, achieves performance comparable to the full-information counterfactual approach, underscoring its effectiveness even in situations with partial information.

6 Related Work

The use of structured rewards in reinforcement learning has been extensively studied, particularly in the context of discrete-time Markov decision processes (DTMDPs). Reward machines, introduced in [Icarte *et al.*, 2022], provide a modular and formal framework for specifying learning objectives using deterministic finite automata, where reward functions are associated with automaton states. Building on [Ng *et al.*, 1999], reward shaping techniques tailored for reward machines were also proposed. Further improvements in shaping methods appear in [Gupta *et al.*, 2023].

Several works have explored formal structures for specifying rewards in RL. Logic-based frameworks have been used to define objectives [Sadigh *et al.*, 2014; Camacho *et al.*, 2019; Li *et al.*, 2017; Jothimurugan *et al.*, 2019; Jiang *et al.*, 2021; Vaezipoor *et al.*, 2021], while automata-based approaches have been applied in [Hahn *et al.*, 2019; Icarte *et al.*, 2022; Hahn *et al.*, 2023; Hahn *et al.*, 2024], including domains such as autonomous driving [Huang *et al.*, 2023]. Non-Markovian rewards have also been studied through temporal logic [Brafman *et al.*, 2018; Giacomo *et al.*, 2020] and active learning [Rens and Raskin, 2020]. Notably, all of these approaches assume DTMDPs as the environment model, leveraging their discrete-time structure.

Reinforcement learning algorithms for CTMDPs were first introduced in [Bradtke and Duff, 1994]. Although CTMDPs have been extensively studied in the formal methods community for various objectives [Baier *et al.*, 2005; Rabe and Schewe, 2011], these works typically assume full knowledge of the model, rather than learning directly from interaction with the environment. More recently, RL algorithms targeting specific objectives in continuous-time domains have emerged. For example, RL for *omega-regular* objectives in unknown CTMDPs was studied in [Falah *et al.*, 2023], while similar objectives were explored in semi-MDPs in [Oura and Ushio, 2022]. To the best of our knowledge, no prior work has adapted reward machines specifically for continuous-time settings. Our work addresses this gap by extending the reward machine framework to continuous-time domains, providing a structured mechanism for specifying learning objectives in environments with continuous dynamics.

7 Conclusion

This paper introduces *continuous-time reward machines* (CTRMs) as a novel framework for modeling rich temporal and reward dynamics in continuous-time systems. We propose a Q-learning algorithm for scenarios where both the CTRM structure and rates are unknown, providing a robust baseline for black-box reinforcement learning. By leveraging CTRM structure, we accelerate policy learning through counterfactual experiences, introducing methods for both full and partial information settings, including a sampling-based approach for approximating unknown rates. We further extend reward shaping to continuous-time domains, offering formal correctness guarantees and an implementation compatible with CTRMs. Empirical evaluations across diverse benchmarks demonstrate the effectiveness and versatility of our methods, highlighting their potential to advance the state of the art in continuous-time reinforcement learning. These contributions lay the groundwork for structured, modular, and explainable RL in real-time and safety-critical systems.

Acknowledgements

This research was supported in part by the National Science Foundation under CAREER Award CCF-2146563. Ashutosh Trivedi holds the position of Royal Society Wolfson Visiting Fellow and gratefully acknowledges the support of the Wolfson Foundation and the Royal Society for this fellowship.

References

[Baier *et al.*, 2005] Christel Baier, Holger Hermanns, Joost-Pieter Katoen, and Boudewijn Haverkort. Efficient computation of time-bounded reachability probabilities in uniform continuous-time Markov decision processes. *Theoretical Computer Science*, 345(1):2–26, 2005.

[Baykal-Gürsoy, 2011] Melike Baykal-Gürsoy. *Semi-Markov Decision Processes*. John Wiley & Sons, Inc., Hoboken, NJ, USA, 2011.

[Becchi, 2008] Michela Becchi. From Poisson processes to self-similarity: a survey of network traffic models. *Washington University in St. Louis, Tech. Rep*, 2008.

[Bradtke and Duff, 1994] Steven Bradtke and Michael Duff. Reinforcement learning methods for continuous-time Markov decision problems. In *NIPS Conference*, pages 393–400. MIT Press, 1994.

[Brafman *et al.*, 2018] Ronen Brafman, Giuseppe De Giacomo, and Fabio Patrizi. LTLf/LDLf non-markovian rewards. In *AAAI*, pages 1771–1778. AAAI Press, 2018.

[Camacho *et al.*, 2019] Alberto Camacho, Rodrigo Toro Icarte, Toryn Qwyllyn Klassen, Richard Anthony Valenzano, and Sheila Ann McIlraith. LTL and beyond: Formal languages for reward function specification in reinforcement learning. In *IJCAI*, volume 19, pages 6065–6073, 2019.

[Falah *et al.*, 2023] Amin Falah, Shibashis Guha, and Ashutosh Trivedi. Reinforcement learning for omega-regular specifications on continuous-time MDP. In *ICAPS*, volume 33, pages 578–586, 2023.

[Giacomo *et al.*, 2020] Giuseppe De Giacomo, Marco Favorito, Luca Iocchi, Fabio Patrizi, and Alessandro Ronca. Temporal logic monitoring rewards via transducers. In *KR*, pages 860–870, 2020.

[Guo and Hernández-Lerma, 2009] Xianping Guo and Onésimo Hernández-Lerma. Continuous-time markov decision processes. In *Continuous-Time Markov Decision Processes*, pages 9–18. Springer, 2009.

[Gupta *et al.*, 2023] Dhawal Gupta, Yash Chandak, Scott M. Jordan, Philip S. Thomas, and Bruno Castro da Silva. Behavior alignment via reward function optimization. In *NIPS*, pages 52759–52791, 2023.

[Hahn *et al.*, 2019] Ernst Moritz Hahn, Mateo Perez, Sven Schewe, Fabio Somenzi, Ashutosh Trivedi, and Dominik Wojtczak. Omega-regular objectives in model-free reinforcement learning. In *TACAS*, volume 11427 of *LNCS*, pages 395–412. Springer, 2019.

[Hahn *et al.*, 2023] Ernst Moritz Hahn, Mateo Perez, Sven Schewe, Fabio Somenzi, Ashutosh Trivedi, and Dominik Wojtczak. Omega-regular reward machines. In *ECAI*, pages 972–979. IOS Press, 2023.

[Hahn *et al.*, 2024] Ernst Moritz Hahn, Mateo Perez, Sven Schewe, Fabio Somenzi, Ashutosh Trivedi, and Dominik Wojtczak. Omega-regular decision processes. In *AAAI Press*, volume 38, pages 21125–21133, 2024.

[Huang *et al.*, 2023] Zhenbo Huang, Shiliang Sun, Jing Zhao, and Liang Mao. Multi-modal policy fusion for end-to-end autonomous driving. *Information Fusion*, 98:101834, 2023.

[Icarte *et al.*, 2022] Rodrigo Toro Icarte, Toryn Q Klassen, Richard Valenzano, and Sheila A McIlraith. Reward machines: Exploiting reward function structure in reinforcement learning. *Journal of Artificial Intelligence Research*, 73:173–208, 2022.

[Jiang *et al.*, 2021] Yuqian Jiang, Suda Bharadwaj, Bo Wu, Rishi Shah, Ufuk Topcu, and Peter Stone. Temporal-logic-based reward shaping for continuing reinforcement learning tasks. In *AAAI*, pages 7995–8003. AAAI Press, 2021.

[Jothimurugan *et al.*, 2019] Kishor Jothimurugan, Rajeev Alur, and Osbert Bastani. A composable specification language for reinforcement learning tasks. In *NeurIPS 2019*, pages 13021–13030, 2019.

[Li *et al.*, 2017] Xiao Li, Cristian-Ioan Vasile, and Calin Belta. Reinforcement learning with temporal logic rewards. In *IROS*, pages 3834–3839. IEEE, 2017.

[Ng *et al.*, 1999] Andrew Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *ICML*, volume 99, pages 278–287, 1999.

[O'Connor, 2011] Andrew N O'Connor. *Probability distributions used in reliability engineering*. RiAC, 2011.

[Oumaima *et al.*, 2020] El Joubari Oumaima, Ben Othman Jalel, and Vèque Véronique. Continuous time Markov chain traffic model for urban environments. In *GLOBECOM*, pages 1–6. IEEE, 2020.

[Oura and Ushio, 2022] Ryohei Oura and Toshimitsu Ushio. Learning-based bounded synthesis for semi-MDPs with LTL specifications. *IEEE Control Systems Letters*, 6:2557–2562, 2022.

[Puterman, 2014] Martin Lee Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[Rabe and Schewe, 2011] Markus Norman Rabe and Sven Schewe. Finite optimal control for time-bounded reachability in CTMDPs and continuous-time Markov games. *Acta Informatica*, 48:291–315, 2011.

[Rens and Raskin, 2020] Gavin Rens and Jean-François Raskin. Learning non-Markovian reward models in MDPs. *arXiv preprint arXiv:2001.09293*, 2020.

[Sadigh *et al.*, 2014] Dorsa Sadigh, Eui Seok Kim, Samuel Coogan, Sosale Shankar Sastry, and Sanjit Arunkumar Seshia. A learning based approach to control synthesis of Markov decision processes for linear temporal logic specifications. In *CDC*, pages 1091–1096, 2014.

[Sutton and Barto, 2018] Richard Stuart Sutton and Andrew Gehret Barto. *Reinforcement Learning: An Introduction*. MIT Press, second edition, 2018.

[Vaezipoor *et al.*, 2021] Pashootan Vaezipoor, Andrew C. Li, Rodrigo Andrés Toro Icarte, and Sheila Ann McIlraith.

LTL2Action: Generalizing LTL instructions for multi-task RL. In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pages 10497–10508. PMLR, 2021.

[Watkins and Dayan, 1992] Christopher John Cornish Hellaby Watkins and Peter Samuel Dayan. Technical note Q-learning. *Machine Learning*, 8:279–292, 1992.