frontiers | Frontiers in Robotics and AI

TYPE Original Research
PUBLISHED 25 September 2025
DOI 10.3389/frobt.2025.1649154

OPEN ACCESS

EDITED BY
Poramate Manoonpong,
University of Southern Denmark, Denmark

REVIEWED BY
Dai Owaki,
Tohoku University, Japan
Jing Wang,
University of Missouri, United States

*CORRESPONDENCE
Taisuke Kobayashi,
kobayashi@nii.ac.jp

RECEIVED 18 June 2025
ACCEPTED 25 August 2025
PUBLISHED 25 September 2025

CITATION
Takahashi K, Kobayashi T, Yamanokuchi T and Matsubara T (2025) Weber–Fechner law in temporal difference learning derived from control as inference.
Front. Robot. AI 12:1649154.
doi: 10.3389/frobt.2025.1649154

Weber–Fechner law in temporal difference learning derived from control as inference

Keiichiro Takahashi[1], Taisuke Kobayashi[2]*, Tomoya Yamanokuchi[1] and Takamitsu Matsubara[1]

[1]Division of Information Science, Nara Institute of Science and Technology, Ikoma, Japan, [2]National Institute of Informatics (NII) and The Graduate University for Advanced Studies (SOKENDAI), Tokyo, Japan

This study investigates a novel nonlinear update rule for value and policy functions based on temporal difference (TD) errors in reinforcement learning (RL). The update rule in standard RL states that the TD error is linearly proportional to the degree of updates, treating all rewards equally without any bias. On the other hand, recent biological studies have revealed that there are nonlinearities in the TD error and the degree of updates, biasing policies towards being either optimistic or pessimistic. Such biases in learning due to nonlinearities are expected to be useful and intentionally leftover features in biological learning. Therefore, this research explores a theoretical framework that can leverage the nonlinearity between the degree of the update and TD errors. To this end, we focus on a *control as inference* framework utilized in the previous work, in which the uncomputable nonlinear term needed to be approximately excluded from the derivation of the standard RL. By analyzing it, the Weber–Fechner law (WFL) is found, in which perception (i.e., the degree of updates) in response to a change in stimulus (i.e., TD error) is attenuated as the stimulus intensity (i.e., the value function) increases. To numerically demonstrate the utilities of WFL on RL, we propose a practical implementation using a reward–punishment framework and modify the definition of optimality. Further analysis of this implementation reveals that two utilities can be expected: i) to accelerate escaping from the situations with small rewards and ii) to pursue the minimum punishment as much as possible. We finally investigate and discuss the expected utilities through simulations and robot experiments. As a result, the proposed RL algorithm with WFL shows the expected utilities that accelerate the reward-maximizing startup and continue to suppress punishments during learning.

KEYWORDS

reinforcement learning, temporal difference learning, control as inference, reward–punishment framework, Weber–Fechner law, robot control

1 Introduction

Reinforcement learning (RL) (Sutton and Barto, 2018) provides robots with policies that allow them to interact in unknown and complex environments, replacing conventional model-based control with it. Temporal difference (TD) learning (Sutton, 1988) is a fundamental methodology in RL. For example, it has been introduced as the basis for

proximal policy optimization (PPO) (Schulman et al., 2017) and soft actor-critic (SAC) (Haarnoja et al., 2018), the most famous algorithms in recent years, both of which are implemented on popular RL libraries (Raffin et al., 2021; Huang et al., 2022) and applied to many real robots (Andrychowicz et al., 2020; Wahid et al., 2021; Nematollahi et al., 2022; Kaufmann et al., 2023; Radosavovic et al., 2024). In TD learning, the future value predicted from the current state is compared to that from the state after transition, which is the so-called TD error. The value function for that prediction can be learned by making this TD error 0, and its learning convergence is theoretically supported by the Bellman equation (although some residuals tend to remain in practice). In addition, actor-critic methods often utilize the TD error as the weight of the policy gradient (Sutton et al., 1999) since it indicates the direction of maximizing the future value.

Although TD learning plays an important role in RL theories and algorithms as above, TD learning can explain many biological behaviors. In particular, a strong correlation between TD errors and the amount of dopamine or the firing rate of dopamine neurons, which affects memory and learning in organisms, has been reported (Schultz et al., 1993; O'Doherty et al., 2003; Starkweather and Uchida, 2021), and behavioral learning in organisms is also hypothesized to be based on RL (Dayan and Balleine, 2002; Doya, 2021). Recently, a more detailed investigation of the relationship between TD errors and dopamine has revealed that it is not a simple linear relationship, as suggested by standard TD learning, but is biased and nonlinear (Dabney et al., 2020; Muller et al., 2024). It has also been reported that some of the nonlinearities may stabilize learning performance (Hoxha et al., 2025). In the context of RL theory, nonlinearly transformed TD learning has been proposed to obtain risk-sensitive behavior (Shen et al., 2014; Noorani et al., 2023) and robustness to outliers (Sugiyama et al., 2009; Cayci and Eryilmaz, 2024). The above studies suggest that the implicit biases introduced by nonlinearities would be effective both theoretically and biologically. In other words, discovering new nonlinearities theoretically or experimentally and understanding their utilities have both an engineering value, such as robot control, and a biological value, such as modeling the principles of behavioral learning in organisms. The aim of this study is to discover new nonlinearities theoretically and reveal their functions experimentally, standing on a constructivist approach using robots (Kuniyoshi et al., 2007).

Moreover, our previous study has found that conventional TD learning can be approximately derived using *control as inference* (Levine, 2018), given appropriate definitions of optimality and divergence (Kobayashi, 2022b). At the same time, it also revealed that updating the value and policy functions according to TD errors becomes optimistic by modifying the definition of the divergence. In a subsequent study, it was additionally found that modifying the definition of optimality leads to pessimistic updates (Kobayashi, 2024b). Thus, RL based on *control as inference* has the capacity to capture various nonlinearities due to the generality of the optimization problems it addresses. This study also follows the new derivation of TD learning in these previous studies to find/investigate the novel nonlinearity undiscovered so far.

In particular, we focus on the fact that an approximation was necessary to derive the conventional TD learning from control as inference with linearity between the TD errors and the degrees of updating. This approximation was generally unavoidable to eliminate an unknown variable and allow numerical computation. However, as the term excluded by the approximation is nonlinear, it should be worth analyzing its utilities as the first contribution of this study. To numerically evaluate the utilities, we propose a novel biologically plausible algorithm that combines a reward–punishment framework (Kobayashi et al., 2019; Wang et al., 2021) with a modified definition of optimality (Kobayashi, 2024b), making the nonlinear term computable in any task covered by RL. In this study, biological plausibility is defined as the presence of nonlinearities in organisms within contexts that are beyond learning.

As a result, we show analytically that the nonlinear term, which has been previously excluded, gives rise to the *Weber–Fechner law* (WFL), a well-known biologically plausible characteristic (Scheler, 2017; Portugal and Svaiter, 2011; Nutter and Esker, 2006; Binhi, 2023). In particular, the degree of update of the value and policy functions corresponding to the intensity of perception is logarithmically affected by the scale of the value function, which is the base stimulus: with the small scale, the update is sensitive to even a small TD error; with the large scale, only a large TD error allows the update enough. This WFL is dominant when the optimality is highly uncertain, while the conventional linear behavior is found when the optimality becomes deterministic. Although organisms have been reported to behave in ways that reduce the uncertainty of predictions (Parr et al., 2022), they are nevertheless forced to make decisions under conditions of uncertainty. Hence, we can anticipate that WFL under the uncertain optimality may also be found in the biological relationship between TD errors and dopamine in organisms.

Through simulations and real-robot experiments, we also confirm that the RL algorithm incorporating the derived WFL can effectively learn optimal policies properly and exert special effects on learning processes and outcomes. In particular, the proposed RL algorithm acquires tasks, and the WFL added in the right balance maximizes rewards eventually while suppressing punishments during learning. In addition, the capability to accelerate learning from a small reward phase allows the robot to efficiently learn a valve-turning task (Ahn et al., 2020) on a real robot, decreasing the error from the target stably. Thus, WFL is useful in RL, raising expectations that organisms have the same (or similar) utilities.

2 Preliminaries

2.1 Reinforcement learning

In RL, an agent aims to optimize a learnable policy so that the accumulation of future rewards from an unknown environment (so-called return) is maximized (Sutton and Barto, 2018) under a Markov decision process (MDP). In other words, an environment with a task to be solved is (implicitly) defined as the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{R}, p_0, p_e)$. Here, $\mathcal{S} \subset \mathbb{R}^{|\mathcal{S}|}$ and $\mathcal{A} \subset \mathbb{R}^{|\mathcal{A}|}$ denote the state and action spaces, respectively, with the $|\mathcal{S}|$-dimensional state s and the $|\mathcal{A}|$-dimensional action a. $\mathcal{R} \subseteq \mathbb{R}$ is the subset on which rewards exist, and the specific values (and even existences) of its upper

and lower boundaries $\mathcal{R} \subseteq (\underline{R}, \overline{R})$ are usually unknown. $p_0 : \mathcal{S} \mapsto \mathbb{R}_+$ denotes the probability for sampling the initial state of each trajectory, and $p_e : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto \mathbb{R}_+$ is known as the state transition probability (or dynamics).

With such a definition, the agent repeatedly interacts with the environment at the current state s according to the action a determined by its policy $\pi : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}_+$ with its learnable parameters ϕ, resulting in the next state s' and the corresponding reward r, which is computed using the reward function $r : \mathcal{S} \times \mathcal{A} \mapsto \mathcal{R}$. As a result, the agent obtains the return R_t from the time step t as presented in Equation 1:

$$R_t = (1-\gamma) \sum_{k=0}^{\infty} \gamma^k r_{t+k}, \tag{1}$$

where $\gamma \in [0, 1)$ denotes the discount factor. Note that $1-\gamma$ is multiplied for normalization to match the implementation used in this study, although the definition without it is common.

The optimal policy π^* is defined for this, as shown in Equation 2:

$$\pi^*(\cdot \mid s) = \arg\max_{\pi} \mathbb{E}_{p_\tau}[R_t \mid s_t = s], \tag{2}$$

where p_τ denotes the probability for the trajectory, defined as the joint probability of p_e and π from t to ∞. ϕ is optimized to represent π^* for any state.

As a remark, the maximization target is modeled as the (state) value function $V : \mathcal{S} \mapsto \mathcal{R}$ with its learnable parameters θ. When $a_t = a$ is also given as the additional condition for computing the above expectation as $\mathbb{E}_{p_\tau}[R_t \mid s_t = s, a_t = a]$, the action value function $Q : \mathcal{S} \times \mathcal{A} \mapsto \mathcal{R}$ is defined for the agent's policy. Here, $Q(s, a)$ can be approximated by $r + \gamma V(s')$ by following the recursive definition of return (i.e., Bellman equation), and the difference between it and $V(s)$ is defined as the TD error, $\delta \coloneqq r + \gamma V(s') - V(s)$, which should be minimized as much as possible by optimizing θ for any state. In addition, δ can be utilized for updating ϕ so that π is more likely to generate actions that make δ more positive (i.e., larger return than expected).

2.2 Update rule derived from control as inference

To interpret the above optimal control problem as a type of inference problem, *control as inference* introduces the stochastic variable for the trajectory's optimality $O = \{0, 1\}$ (Levine, 2018). As it is relevant to the return, its conditional probability is defined as shown in Equation 3:

$$\begin{aligned} p(O = 1 \mid s) &= e^{\beta\left(V(s) - \overline{R}\right)} =: p_V \\ p(O = 1 \mid s, a) &= e^{\beta\left(Q(s,a) - \overline{R}\right)} =: p_Q, \end{aligned} \tag{3}$$

where $\beta \in \mathbb{R}_+$ denotes the inverse temperature parameter. Note that $p(O = 0)$ can also be given as $1 - p(O = 1)$ since O is binary. From this definition, O can be explained so that it is more likely to be 1 if the value is higher. When β is small, optimality is ambiguous, and as β increases, optimality becomes deterministic.

With the probability of optimality, the optimal and non-optimal policies are inferred according to Bayes theorem. In particular, with the baseline policy $b(a \mid s)$ for sampling actions, $\pi(a \mid s, O)$ is obtained, as presented in Equation 4:

$$\begin{aligned} \pi(a \mid s, O) &= \frac{p(O \mid s, a)\, b(a \mid s)}{p(O \mid s)} \\ &= \begin{cases} \dfrac{e^{\beta\left(Q(s,a) - \overline{R}\right)}}{e^{\beta\left(V(s) - \overline{R}\right)}} b(a \mid s) & O = 1 \\ \dfrac{1 - e^{\beta\left(Q(s,a) - \overline{R}\right)}}{1 - e^{\beta\left(V(s) - \overline{R}\right)}} b(a \mid s) & O = 0. \end{cases} \end{aligned} \tag{4}$$

Based on this definition, a previous study (Kobayashi, 2022b) considered the minimization problem presented in Equation 5 for optimizing θ, the parameters of the value function V.

$$\min_{\theta} \mathbb{E}_{p_e, b}[\mathrm{KL}(p(O \mid s) \mid p(O \mid s, a))], \tag{5}$$

where $\mathrm{KL}(p_1 \mid p_2) = \mathbb{E}_{x \sim p_1}[\ln p_1(x) - \ln p_2(x)]$ is Kullback–Leibler (KL) divergence. More specifically, as the target probability is on the right side, this can be regarded as "reverse" KL divergence. Since $p(O \mid s, a)$ has more information than $p(O \mid s)$, this minimization problem makes $p(O \mid s)$ more informative to represent the optimality. To solve this problem, its gradient with respect to θ, g_θ, is derived as shown in Equation 6:

$$\begin{aligned} g_\theta &= \mathbb{E}_{p_e, b}\left[\nabla_\theta p_V \ln \frac{p_V}{p_Q} - \nabla_\theta p_V \ln \frac{1 - p_V}{1 - p_Q} + p_V \nabla_\theta \ln p_V + (1 - p_V) \nabla_\theta \ln (1 - p_V)\right] \\ &= \mathbb{E}_{p_e, b}\left[-\nabla_\theta V(s) \beta p_V \left\{\beta(Q(s,a) - V(s)) + \ln \frac{1 - p_V}{1 - p_Q}\right\}\right] \\ &\propto \mathbb{E}_{p_e, b}\left[-\nabla_\theta V(s) \left\{(1 - \lambda_\beta)(Q(s,a) - V(s)) + \lambda_\beta \ln \frac{1 - p_V}{1 - p_Q}\right\}\right], \end{aligned} \tag{6}$$

where $\lambda_\beta \coloneqq (1 + \beta)^{-1} \in (0, 1)$. The last proportion is obtained by dividing the raw gradient by $\beta(1 + \beta)p_V$. Since β is constant, it can be absorbed into the learning rate, but p_V appears to introduce a bias in the convergence point, as noted in the previous study (Kobayashi, 2022b). However, we found that the Fisher information for $p(O|V)$ is given as $\beta^2 p_V (1 - p_V)^{-1}$, and dividing the raw gradient by p_V can be interpreted as the sum of the raw and natural gradients (details are in Appendix 1), which is expected to converge to the same destination without bias (Landro et al., 2020).

As a practical problem, p_V and p_Q cannot be numerically computed since they include the unknown $\overline{R}$, the upper bound of the reward and return. The previous study approximates the above gradient by assuming $\lambda_\beta \to 0$ (i.e., $\beta \to \infty$), resulting in standard TD learning (by assuming $Q(s, a) \simeq r + \gamma V(s')$).

In addition to the value function, the policy π (more precisely, its parameter ϕ) is also optimized through the following minimization problem, as presented in Equation 7, with the reverse KL divergences.

$$\begin{aligned} &\min_{\phi} \mathbb{E}_{p_e}[\mathrm{KL}(\pi(a \mid s) \mid \pi(a \mid s, O = 1)) - \mathrm{KL}(\pi(a \mid s) \mid \pi(a \mid s, O = 0))] \\ &= \min_{\phi} \mathbb{E}_{p_e, \pi}\left[\ln \frac{\pi(a \mid s, O = 0)}{\pi(a \mid s, O = 1)}\right]. \end{aligned} \tag{7}$$

In particular, the policy tries to be close to the optimal policy, while being far away from the non-optimal policy. The gradient with respect to ϕ, g_ϕ, is also derived analytically, as shown in Equation 8:

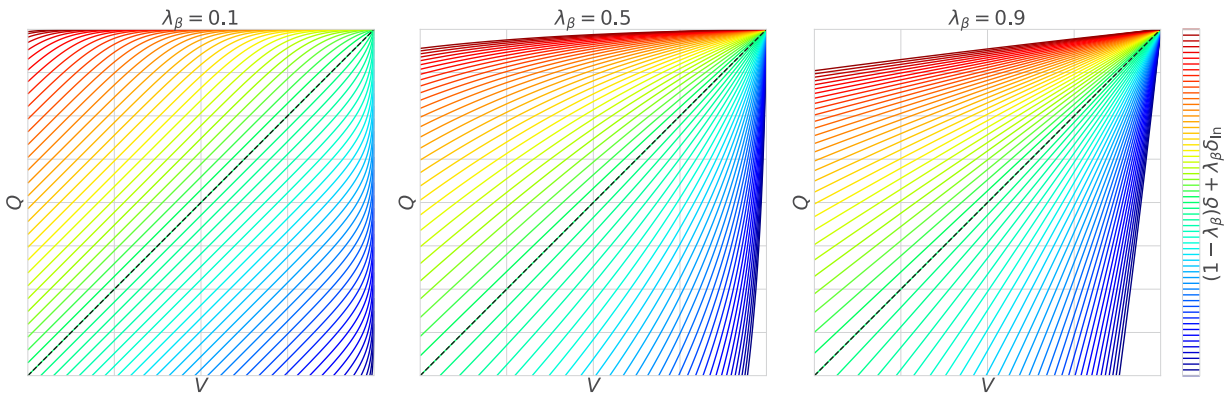

FIGURE 1
Effects of the nonlinear term $\delta_{\ln}$: when δ is dominant, the degrees of updates depicted by the contour lines are mostly equally spaced in parallel to the line of $V = Q$; when the influence of $\delta_{\ln}$ increases, the contour lines radiate out from the upper bound $\overline{R}$.

$$
\begin{aligned}
g_\phi &= \mathbb{E}_{p_e,\pi}\left[\frac{\nabla_\phi \pi(a \mid s)}{\pi(a \mid s)} \ln \frac{\pi(a \mid s, O=0)}{\pi(a \mid s, O=1)}\right] \\
&= \mathbb{E}_{p_e,\pi}\left[-\nabla_\phi \ln \pi(a \mid s)\left\{\beta(Q(s,a) - V(s)) + \ln \frac{1-p_V}{1-p_Q}\right\}\right] \\
&\propto \mathbb{E}_{p_e,b}\left[-\frac{\pi(a \mid s)}{b(a \mid s)} \nabla_\phi \ln \pi(a \mid s)\left\{(1-\lambda_\beta)(Q(s,a) - V(s)) + \lambda_\beta \ln \frac{1-p_V}{1-p_Q}\right\}\right],
\end{aligned} \tag{8}
$$

where the last proportion is given by dividing the raw gradient by $(1+\beta)$. In addition, at the final step, the importance sampling replaces π in the expectation operation with the baseline policy b. Along with the value function, the approximation of $\lambda_\beta \to 0$ makes this gradient computable, resulting in the standard policy gradient in actor-critic algorithms.

3 Weber–Fechner law in TD learning

3.1 Numerical analysis with an explicit upper bound

The gradients to optimize the value and policy functions are derived in Equations 6, 8, respectively. However, as the upper bound of the reward function $\overline{R}$ is unknown and p_V and p_Q cannot be calculated numerically, it was necessary to exclude the uncomputable term by setting $\lambda_\beta \to 0$. As a result, the previous study (Kobayashi, 2022b) found the conventional update rule, where the gradients are weighted by $Q(s,a) - V(s) \simeq r + \gamma V(s') - V(s) = \delta$ (i.e., the TD error). On the other hand, if the nonlinear term excluded (i.e., $\delta_{\ln} \coloneqq \ln(1-p_V) - \ln(1-p_Q)$) is computable, it is interesting how it affects the gradients, and this analysis is the main focus of this study.

Therefore, we assume that $\overline{R}$ is known at once in this section. With this assumption, the gradient including $\delta_{\ln}$ is analyzed. First, we numerically visualize the gradient according to $\lambda_\beta \in (0,1)$ and estimate the role of $\delta_{\ln}$, which has a stronger influence when λ_β increases (i.e., β decreases). For this purpose, $(1-\lambda_\beta)\delta + \lambda_\beta\delta_{\ln}$ (i.e., the degree of updates) for $\lambda_\beta = \{0.1, 0.5, 0.9\}$ in the case $\mathcal{R} = (-1, 1)$ is illustrated in Figure 1.

First, at $\lambda_\beta = 0.1$, the contour lines representing the degree of updates are spaced equally and parallel to the line of $V = Q$. This is mainly because δ is dominant; that is, the degree of updates is linearly proportional to the TD error. Note that the behavior is slightly different for $V, Q \simeq \overline{R}$ because $\delta_{\ln}$ remains. The remaining $\delta_{\ln}$, however, easily converges to 0 since the large β (= 9 in this case) causes p_V and p_Q converge to 0, even with only a small difference between V, Q and $\overline{R}$ (i.e., optimality is deterministic).

On the other hand, when $\lambda_\beta = 0.9$, $\delta_{\ln}$ is dominant, causing the contour lines to extend radially from $\overline{R}$. In other words, when the value is close to $\overline{R}$, the update is significantly activated even with a small TD error, whereas when the value is far from $\overline{R}$, only a large TD error can sufficiently drive the update. Unlike the case with $\lambda_\beta = 0.1$, $\delta_{\ln}$ has a strong effect even when the value is far from $\overline{R}$ because p_V and p_Q with the small β (= 1/9 in this case) are changed at approximately 1/2 without converging to 0 (i.e., optimality is uncertain).

Finally, $\lambda_\beta = 0.5$ yields an intermediate behavior between the above two characteristics. In particular, when the value is somewhat close to $\overline{R}$, the radial spread from $\overline{R}$ is observed due to the influence of $\delta_{\ln}$, and when it falls below a certain level, δ dominates, and it switches to parallel contour lines. However, it should be noted that this trend depends on the range of $\mathcal{R}$, so it is not always true for $\lambda_\beta = \{0.1, 0.5, 0.9\}$.

3.2 Mathematical analysis using the Taylor expansion

We further analyze the characteristics of $\delta_{\ln}$ found in the above numerical results. Since these characteristics become apparent when V and Q are close to $\overline{R}$, we apply a Taylor expansion to p_V and p_Q around $\overline{R}$, as presented in Equations 9, 10:

$$
p_V = \sum_{n=0}^{\infty} \frac{\beta^n \left(V - \overline{R}\right)^n}{n!} \simeq 1 + \beta\left(V - \overline{R}\right), \tag{9}
$$

$$
p_Q = \sum_{n=0}^{\infty} \frac{\beta^n \left(Q - \overline{R}\right)^n}{n!} \simeq 1 + \beta\left(Q - \overline{R}\right). \tag{10}
$$

Accordingly, $\delta_{\ln}$ is presented in Equation 11, :

$$\delta_{\ln} = \ln \frac{1-p_V}{1-p_Q} \simeq -\ln \frac{\overline{R}-Q}{\overline{R}-V}. \tag{11}$$

At this point, let us interpret $\overline{R}-V$ as the baseline stimulus strength, $\overline{R}-Q$ as the stimulus strength after the change (or $-\delta$ in $\overline{R}-Q \simeq \overline{R}-V-\delta$ as the change in stimulus strength), and $-\delta_{\ln}$ as the intensity of perception. If this is the case, these are subject to the WFL. In other words, how strongly the stimulus change $-\delta$ is perceived (i.e., $-\delta_{\ln}$) depends on the baseline of the stimulus strength $\overline{R}-V$: the smaller $\overline{R}-V$ is, the more acute the sensation becomes, and *vice versa*. This is exactly the characteristic found in the right side of Figure 1, indicating that the approximation by the Taylor expansion is valid.

We then conclude the analysis that WFL is hidden even in the update rule of RL derived from control as inference. WFL has also been found in areas closely related to brain functions such as neuron firing patterns (Scheler, 2017) and cognition (Portugal and Svaiter, 2011). Recently, it has been shown that the time steps in RL can theoretically be a nonlinear log scale (i.e., WFL), leading to adaptive temporal discounting (Maini and Tiganj, 2025). Therefore, it is not implausible to find it in RL, which is also attracting attention as a biological decision-making model (Dayan and Balleine, 2002; Doya, 2021). This hypothesis would be supported by the fact that WFL is activated when optimality is uncertain, which is consistent with the conditions faced by organisms (Parr et al., 2022).

Furthermore, its applicability to learning and practical engineering value should be verified through numerical experiments, with the exception that WFL represents a useful characteristic in RL and has been evolutionary preserved in organisms. However, the above analysis was performed under the assumption that $\mathcal{R}$ is known, which is contrary to the general problem statement for RL. In the next section, therefore, we propose a practical implementation that enables the computation of $\delta_{\ln}$ even when $\mathcal{R}$ is unknown in a biologically plausible manner, followed by an experimental verification of the benefits of WFL in RL.

4 Practical implementation

4.1 Introduction of the reward–punishment framework

First, we address the unknown $\mathcal{R}$ without giving prior knowledge of the problem to be solved. The requirements are i) the boundary of $\mathcal{R}$ to define the optimality and ii) the range of $\mathcal{R}$ to determine β (or λ_β) for which WFL is valid. A naive solution would be to empirically estimate the boundary $(\underline{R}, \overline{R})$. The estimation of the range $(\overline{R}-\underline{R})$ does not need to be rigorous, so learning can proceed stably if it is updated slowly. However, for p_V and p_Q to consistently satisfy the definition of probability, $\overline{R}$ may need to change frequently, which could cause instability in learning. Moreover, the empirically estimated $\overline{R}$ is likely to be underestimated relative to its true value, thereby preventing the full utility of WFL from being realized.

Therefore, in this study, we introduce a more reliable solution, the *reward–punishment framework* (Kobayashi et al., 2019; Wang et al., 2021). Although rewards are generally defined as scalar values that can be either positive or negative, this is a way to separate positive rewards $r^+ \in \mathcal{R}_+ \subseteq \mathbb{R}_+$ and negative rewards, i.e., punishments, $r^- \in \mathcal{R}_- \subseteq \mathbb{R}_-$. This can be applied to any RL problem, either i) by having the environment output $r^{+,-}$ as in the experiments of this study (see the top of Figure 2) or ii) by distributing $r \in \mathcal{R}$ from the environment as $r^+ = \max(r, 0)$ and $r^- = \min(r, 0)$ (see the bottom of Figure 2).

The reward–punishment framework learns the value and policy functions for $r^{+,-}$, respectively. In particular, the returns and the value functions for $r^{+,-}$ are first defined as shown in Equation 12, :

$$\begin{aligned} R_t^{+,-} &= (1-\gamma)\sum_{k=0}^{\infty}\gamma^k r_{t+k}^{+,-} \\ V^{+,-}(s) &= \mathbb{E}_{p_\tau}\left[R_t^{+,-} \mid s_t = s\right] \\ Q^{+,-}(s,a) &= \mathbb{E}_{p_\tau}\left[R_t^{+,-} \mid s_t = s, a_t = a\right]. \end{aligned} \tag{12}$$

The policies $\pi^{+,-}$, which attempt to maximize them separately, are also introduced.

Here, since only one action can be passed to the environment, even if the agent has two policies, it is necessary to synthesize them. Following the previous study (Wang et al., 2021), a mixture distribution with a mixing ratio based on the value function is designed, as shown in Equations 13, 14:

$$b(a \mid s) = w\pi^+(a \mid s) + (1-w)\pi^-(a \mid s), \tag{13}$$

$$w = \frac{e^{\beta_w V^+(s)}}{e^{\beta_w V^+(s)} + e^{-\beta_w V^-(s)}}. \tag{14}$$

With this design, however, only one of the policies might be activated and the other might be ignored if the difference in the scales of $r^{+,-}$ is large. To alleviate this issue, a policy regularization method, PPO-RPE (Kobayashi, 2023), for the density ratio $\pi^{+,-}/b$ with importance sampling [see Equation 8], is introduced in this study. As it yields $\pi^+ \simeq \pi^-$ *via* b, the past mixture distributions transfer and share the acquired skills with each other. In addition, as PPO-RPE is a type of actor-critic algorithm, it can be applied not only to continuous but also to discrete action spaces.

In any case, within the reward–punishment framework, the upper bound of r^- is given to be 0, making $\delta_{\ln}$ computable. On the other hand, r^+ has only a lower bound of 0, so $\delta_{\ln}$ for it remains uncomputable. In the next section, therefore, we derive $\delta_{\ln}$ utilizing this lower bound.

The range of rewards, $\sigma^{+,-}$, which is necessary for designing β where WFL is effectively manifested, can be estimated from the empirical $r^{+,-}$. However, the assumption when deriving Equations 6, 8 (i.e., β is constant) is violated if $\sigma^{+,-}$ fluctuates too much. In addition, since the experienced scale of r^- is likely to gradually decrease, the approach to record the maximum scale is not suitable for this case. From the above, $\sigma^{+,-}$ is estimated and used for the design of β as shown in Equation 15, :

$$\begin{aligned} \sigma_{\max}^{+,-} &\leftarrow \max\left(\zeta\sigma_{\max}^{+,-}, |r^{+,-}|\right) \\ \sigma^{+,-} &\leftarrow \zeta\sigma^{+,-} + (1-\zeta)\sigma_{\max}^{+,-} \\ \beta^{+,-} &= \frac{\beta_0}{\sigma^{+,-}}, \end{aligned} \tag{15}$$

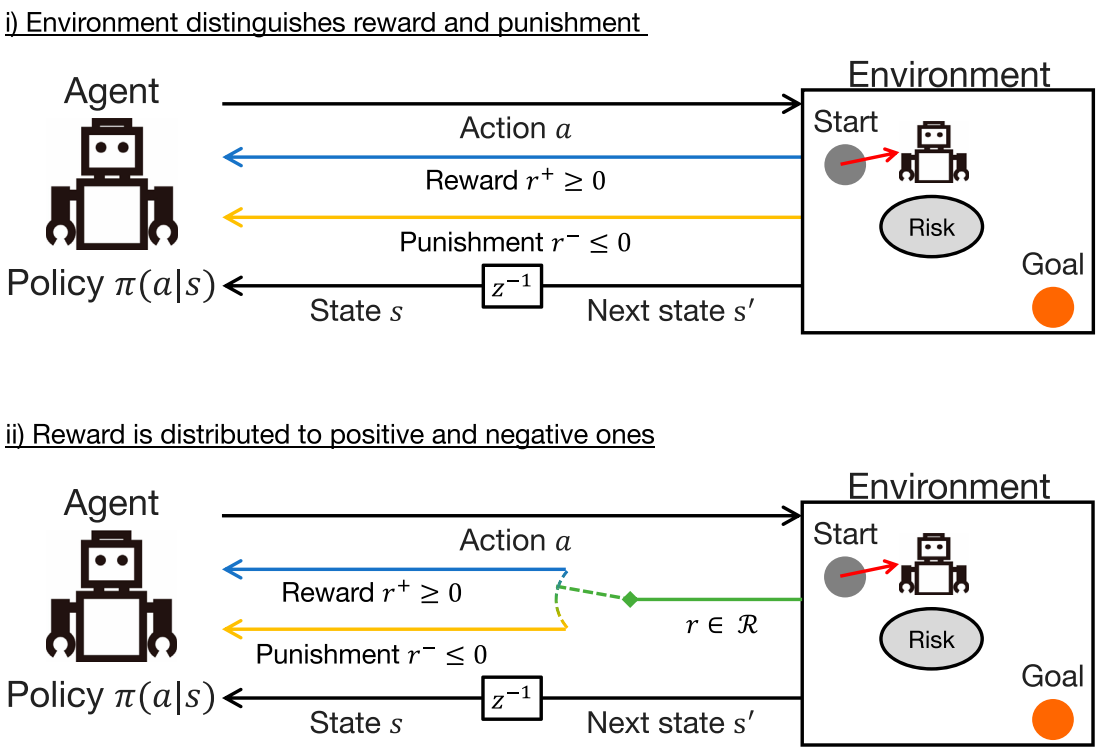

FIGURE 2
Reward–punishment framework: in the upper case, both rewards and punishments are directly given from the environment; in the bottom case, scalar rewards in a subset of real space are treated by distinguishing between positive and negative ones as rewards and punishments, respectively.

where $\zeta \in (0,1)$ denotes the gradualness of adaptation (generally, ζ is close to 1) and $\beta_0 \in \mathbb{R}_+$ denotes the baseline. This design allows β to reflect the scale while limiting frequent fluctuations of β by using the recent maximum scale and updating to that value gradually. In addition, as this update rule does not use $V^{+,-}$, it can avoid adverse effects due to the estimation uncertainty of $V^{+,-}$.

4.2 Inversion of the definition of optimality using the lower bound

As mentioned above, although $r^- \leq 0$ can numerically compute Equations 6, 8 without any approximation, $r^+ \geq 0$ cannot do so since it only has the lower bound. To solve this issue, we introduce a new method for deriving gradients with WFL, inspired by a previous study (Kobayashi, 2024b). In particular, the inversion of the definition of optimality in Equation 3 is considered the starting point.

$$\begin{aligned} p(O=0 \mid s) &= e^{-\beta(V(s)-\underline{R})} =: p_V \\ p(O=0 \mid s,a) &= e^{-\beta(Q(s,a)-\underline{R})} =: p_Q. \end{aligned} \tag{16}$$

In Equation 16, the lower bound $\underline{R}$ of $\mathcal{R}$ is utilized for satisfying the definition of probability. Note that the aliases p_V and p_Q are given for $p(O=0)$, unlike Equation 3.

As the previous study did not derive the gradients of Equations 5, 7 using Equation 16, their derivations are described below. First, g_θ for Equation 5 is derived as shown in Equation 17:

$$\begin{aligned} g_\theta &= \mathbb{E}_{p_e,b}\left[\nabla_\theta p_V \ln\frac{p_V}{p_Q} - \nabla_\theta p_V \ln\frac{1-p_V}{1-p_Q} + p_V \nabla_\theta \ln p_V + (1-p_V)\nabla_\theta \ln(1-p_V)\right] \\ &\propto \mathbb{E}_{p_e,b}\left[-\nabla_\theta V(s)\left\{(1-\lambda_\beta)(Q(s,a)-V(s)) - \lambda_\beta \ln\frac{1-p_V}{1-p_Q}\right\}\right]. \end{aligned} \tag{17}$$

Except for the different definitions of p_V and p_Q and the sign reversal of the second term, it is symmetric to Equation 6. Similarly, g_ϕ for Equation 7 is shown in Equation 18:

$$\begin{aligned} g_\phi &= \mathbb{E}_{p_e,\pi}\left[\frac{\nabla_\phi \pi(a \mid s)}{\pi(a \mid s)} \ln\frac{\pi(a \mid s, O=0)}{\pi(a \mid s, O=1)}\right] \\ &\propto \mathbb{E}_{p_e,b}\left[-\frac{\pi(a \mid s)}{b(a \mid s)}\nabla_\phi \ln \pi(a \mid s)\left\{(1-\lambda_\beta)(Q(s,a)-V(s)) - \lambda_\beta \ln\frac{1-p_V}{1-p_Q}\right\}\right], \end{aligned} \tag{18}$$

where $\pi(a \mid s, O=0) = p_Q p_V^{-1} b(a \mid s)$ and $\pi(a \mid s, O=1) = (1-p_Q)(1-p_V)^{-1} b(a \mid s)$ are also redefined in this study. This gradient is also in the same format as Equation 17, except that the sign of the second term is reversed, with different definitions for p_V and p_Q. Note that the impact of redefining optimality can be confirmed when the minimization problem is designed with "forward" KL divergence, as analyzed in the previous study (Kobayashi, 2024b).

Both have the same degree of updates multiplied by the gradients, and the first term coincides with the TD error δ as in the original. The crucial second term appears to be different, but as depicted in Figure 3, the contour lines extend radially from $\underline{R}$, as in the original. If the Taylor expansion around $\underline{R}$ is applied to p_V and p_Q, which are substituted into $\delta_{\ln} := -\ln(1-p_V) + \ln(1-p_Q)$, the same

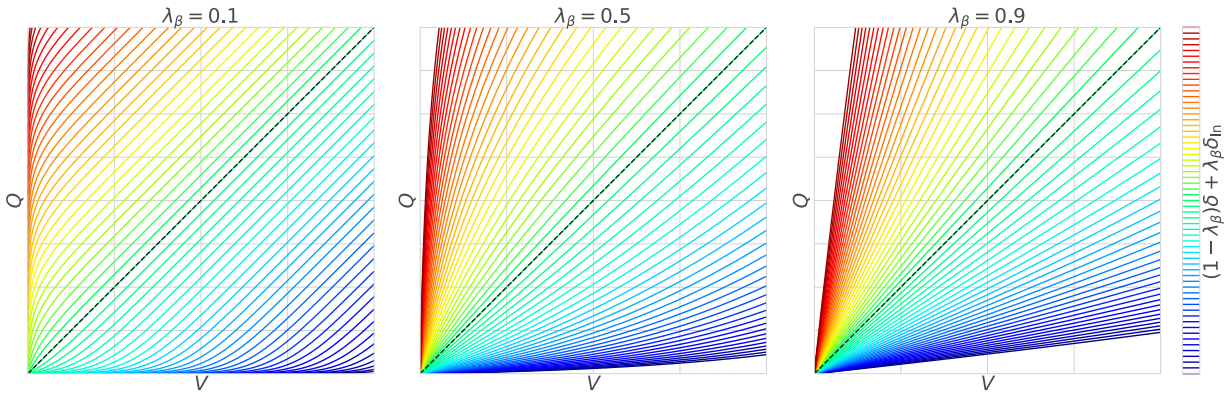

FIGURE 3
Weber–Fechner law using the lower bound: with the large λ_β, the contour lines become narrower with V and Q closer to their lower bound $\underline{R}$ and wider with V and Q farther from $\underline{R}$.

TABLE 1 Correspondence between WFL and the proposed update rule.

	WFL	Update rule for r^+	Update rule for r^-
Formula	$p \propto \frac{S}{S_0}$	$\delta_{\ln} \propto \ln \frac{Q}{V}$	$-\delta_{\ln} \propto \ln \frac{-Q}{-V}$
Baseline of stimulus strength	S_0	V	$-V$
Stimulus strength after change	S	Q	$-Q$
Change in stimulus strength	$S - S_0$	$Q - V \simeq \delta$	$-(Q-V) \simeq -\delta$
Intensity of perception	p	$\delta_{\ln}$	$-\delta_{\ln}$

WFL is confirmed. Thus, it is possible to compute the gradients with WFL even for r^+, where only $\underline{R} = 0$ is known.

4.3 Expected utilities

As described above, we proposed a novel algorithm including the terms with WFL, which had been excluded in the previous study (Kobayashi, 2022b) (and the standard RL algorithms) because they are computationally infeasible. Table 1 summarizes the correspondence between WFL and the update rule in the proposed algorithm. Note that since WFL is a law about the signal strength, the terms in punishments are converted for the punishment strength by reversing their signs.

In this study, we summarize the basic utilities of WFL in this algorithm. First, for rewards r^+, the updates of the value and policy functions are actively promoted at $V^+ \simeq 0$, whereas the updates are relatively suppressed over a certain level, $V^+ \gg 0$. Conversely, for punishments r^-, the updates are slow under a certain level $V^- \ll 0$, but $V^- \simeq 0$ is pursued eventually. It is known in gradient-based optimization that large gradients per update lead to a solution robust to small perturbations, while small gradients lead to one of the local solutions (Smith, 2018; Foret et al., 2021). Therefore, WFL in the proposed algorithm can also be interpreted as seeking an early local solution for r^+ and a stable global solution for r^-. Note that, as shown in Figures 1, 3, the utilities of WFL can be suppressed by adjusting λ_β (or β) with the activation of the standard TD learning, which is not affected by the baseline stimulus strength (i.e., $|V|$).

5 Numerical verification

5.1 Toy problem

First, we investigate the feasibility of learning the optimal policy under the proposed algorithm with WFL and the effects of WFL on the learning process and results. As a toy problem, *Pendulum-v0* implemented in OpenAI Gym is applied, while its reward function is redefined to fit the reward–punishment framework.

$$\begin{aligned} r^+ &= 1 + \cos q \\ r^- &= -(0.1|\dot{q}| + 0.001|\tau|)(1 + \cos q), \end{aligned} \tag{19}$$

In Equation 19, q denotes the pendulum angle, $\dot{q}$ denotes its angular velocity, and τ denotes the torque applied to the pendulum (i.e., action). In particular, the agent gets high rewards if the pendulum is close to upright, while it is punished when the pendulum is not stopped. In addition, this punishment is stronger when the pendulum is close to upright. Since we know that *Pendulum-v0* is a standard and classic benchmark that eventually yields an optimal policy, it is useful for analyzing the changes in the learning process.

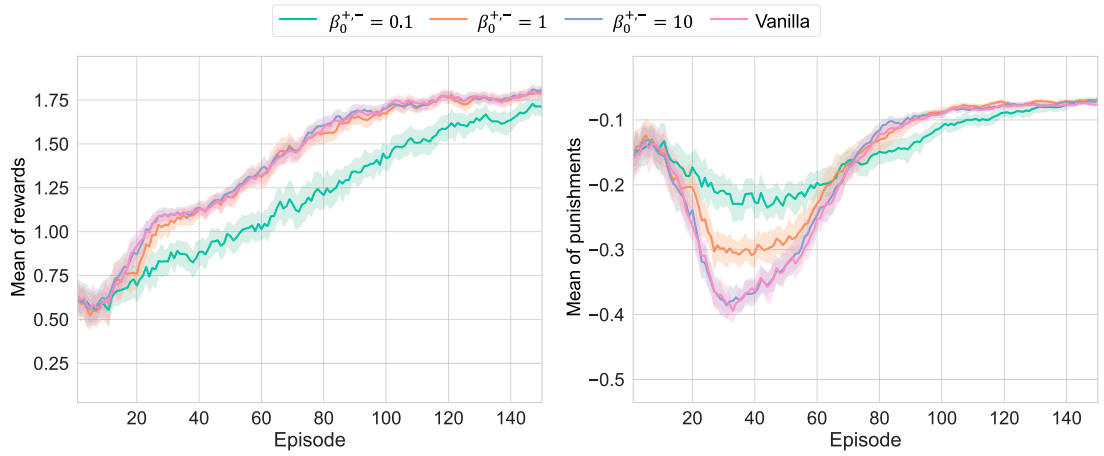

FIGURE 4
Learning results of *Pendulum-v0* with different $\beta_0^{+,-}$: the upper and bottom curves depict the learning curves for episodic averages of r^+ and r^-, respectively; when $\beta_0^{+,-}$ was small, the pursuit of r^+ became slower, and r^- was preferentially suppressed to 0.

With $\beta_0^{+,-} = \{0.1, 1, 10, \infty\}$ (∞ indicates conventional TD learning), four respective models are trained 50 times with different random seeds in order to achieve the statistical learning results shown in Figure 4. Note that the learning conditions, including network architectures, are summarized in Appendix 2. As shown in the results, learning can proceed for any $\beta_0^{+,-}$ without collapse, indicating that RL is valid even if the term $\delta_{\ln}$ with WFL is used. The increase in $\beta_0^{+,-}$ made the learning curves approach the conventional curves, as expected. On the other hand, when $\beta_0^{+,-}$ becomes smaller, unique behaviors were observed. In other words, for $\beta_0^{+,-} = 0.1$, optimization with respect to r^+ was delayed, while the temporary deterioration of r^- was restricted. This can be attributed to the fact that optimization worked to reduce r^- to 0 as much as possible while simultaneously maximizing r^+ to some extent. However, in the former case, it might be possible that the strong effort to reduce r^- to 0 suppressed the exploration, causing a delay in the discovery of the optimal solution for r^+. Note that $\beta_0^{+,-} = 1$ achieved the similar learning curve of reward to the case of Vanilla due to the nonlinear effect of $\beta_0^{+,-}$, but their confidence intervals were hardly overlapped at the early stages of learning.

Then, to take the advantages of both, the results of setting $\beta_0^+ = 10$ and $\beta_0^- = 0.1$ asymmetrically are depicted in Figure 5. Under this setting, WFL's efforts to reduce r^- to 0 remained and suppressed the temporary deterioration of r^-, while r^+ was successfully optimized without much delay. In other words, the delay in learning about r^+ at $\beta_0^{+,-} = 0.1$ can be attributed to the characteristics of WFL. Note that the delay in maximizing r^+ at approximately 20 episodes is considered an effect of the suppression of exploration. On the other hand, r^+ was maximized more efficiently, and r^- was smaller in conjunction with it.

In any case, WFL's utilities analyzed in this study were confirmed as expected in the numerical verification. In addition, as suggested in Figure 5, the optimization behaviors for $r^{+,-}$ can be adjusted by setting $\beta_0^{+,-}$ separately. However, we need to remark that the separation of $\beta_0^{+,-}$ does not mean that they function independently since $r^{+,-}$ necessarily depend on each other in the learning process, as in the exploration suppression described above.

5.2 Robotic task

The above numerical verification showed that the proposed method with WFL can optimize the policies with its expected learning characteristics. Based on this finding, we additionally demonstrate that the proposed method can be useful in more practical robotic tasks. This study focuses on the D'Claw task in ROBEL (Ahn et al., 2020), in which three 3-DOF robotic fingers manipulate a valve (see Figure 6). This benchmark is unique in that it includes not only a robotic simulation but also a real-robot version, which can automatically be initialized to restart episodes. Hence, this task is useful to verify that the proposed algorithm works in a real-world setting. Note that the code for this system is not the original code but a modified version in the literature (Yamanokuchi et al., 2022). Its state space consists of the angles and angular velocities of the finger joints and of the valve (in total, 22 dimensions), while the action space consists of nine dimensions of angular changes in the finger joints.

5.2.1 Simulations

First, the simulations confirm that the behaviors when WFL is activated for $r^{+,-}$ individually are consistent with the toy problem. Since $\beta_0^{+,-} = 0.1$ was too extreme and $\beta_0^{+,-} = 1$ was insufficient for WFL effects, $\beta_0^{+,-} = 0.5$ is adopted to activate WFL from here as *WFL-R/P*.

The reward function is defined as shown in Equation 20:

$$\begin{aligned} r^+ &= q_v \mathbb{I}_{q_v > 0} \\ r^- &= -0.01 \| q_j + \Delta q_j \|_2^2, \end{aligned} \tag{20}$$

where q_v and q_j denote the valve angle and the joint angles, respectively, and Δq_j denotes the angular changes in joints (i.e., action). In particular, the goal is to turn the valve as much as possible while keeping the fingers in the initial posture to some extent. Note that the sign of q_v reverses after one turn, so the actual goal is to stop just before one turn.

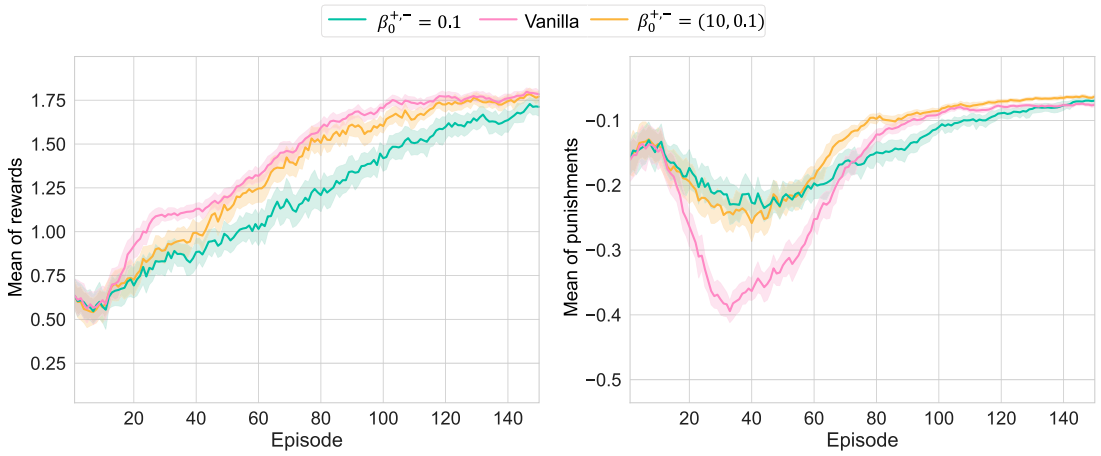

FIGURE 5
Learning results of *Pendulum-v0* with the asymmetric $\beta_0^{+,-}$: because of small $\beta_0^- = 0.1$, the deterioration of r^- was restricted; although the exploration was more or less limited, the large $\beta_0^+ = 10$ value enabled to maximize r^+ to the same level as the conventional method.

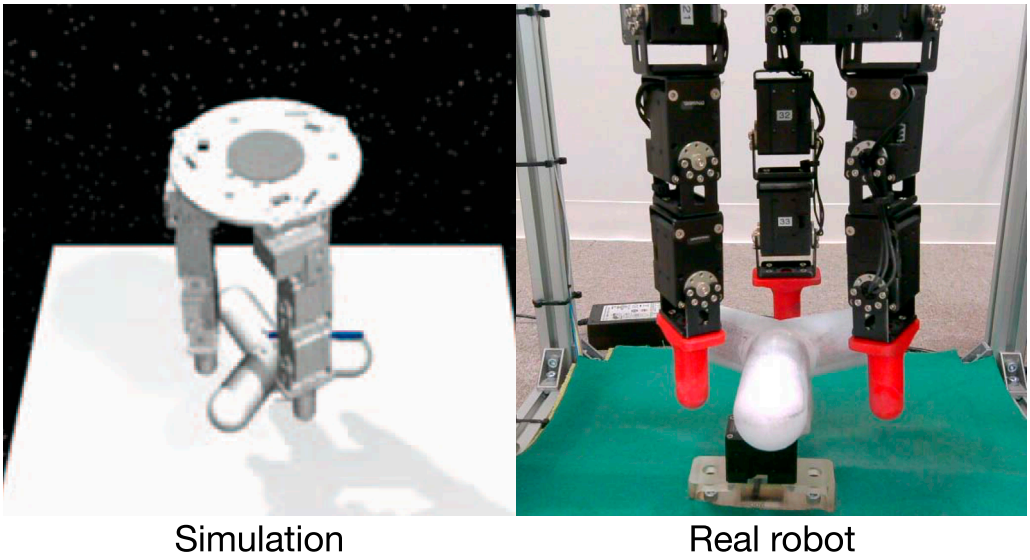

FIGURE 6
D'Claw task (Ahn et al., 2020): it is simulated on MuJoCo (Todorov et al., 2012).

The learning results of each condition with 20 different random seeds are shown in Figure 7. Here, *WFL-R* and *WFL-P* denote the asymmetric models with $\beta_0^{+,-} = (0.5, \infty), (\infty, 0.5)$, respectively. Note that because the punishments were very small, unlike in the toy problem, we plotted the sum of rewards/punishments per episode rather than their mean. First, the pursuit of r^+ was slowed down in *WFL-R* with WFL for r^+. As its side effect, r^- always outperformed the conventional method due to the reduced robot motion. On the other hand, *WFL-P* with WFL for r^- showed improvement after approximately 4000 episodes, reflecting the pursuit of reducing punishments and finally achieving the best performance. In addition, probably because the range of motion of each finger joint was maintained as its side effect, the speed of improvement in r^+ was significantly increased compared to others. Thus, it was suggested that the appropriate addition of WFL, including its side effects, can improve learning performance in practical robotic tasks.

5.2.2 Real-robot experiments

Next, we demonstrate how WFL works in learning on the real world. For simplicity, WFLs for $r^{+,-}$ are both activated simultaneously and compared with the conventional method.

Since the real-robot valve angle has a different domain from the simulation angle and the angle jumps to $\pi \to -\pi$ in a half turn, the reward function is modified accordingly, as shown in Equation 21:

$$\begin{aligned} r^+ &= |q_v| \mathbb{I}_{q_v > 0 \vee q_v < -\frac{3}{4}\pi} \\ r^- &= -0.01 \|q_j + \Delta q_j\|_2^2. \end{aligned} \tag{21}$$

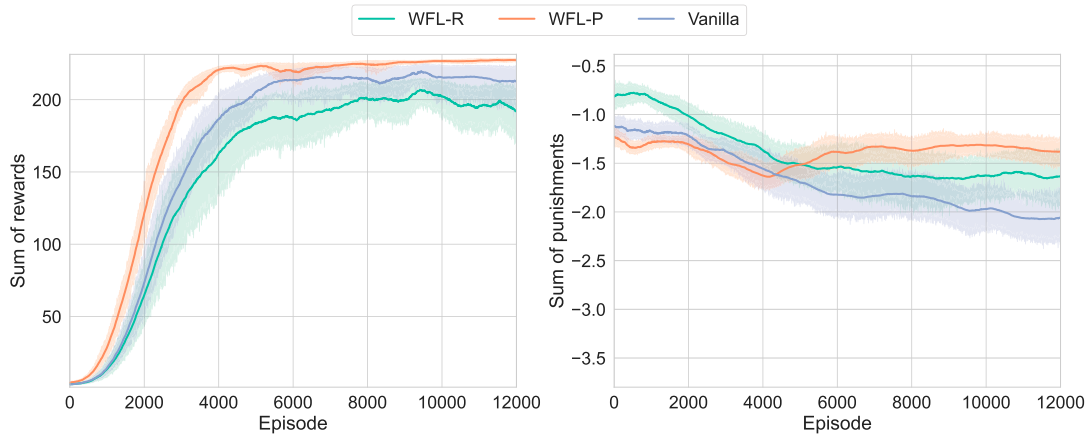

FIGURE 7
Learning results of D'Claw simulation: $\beta_0^{+,-} = (0.5, \infty), (\infty, 0.5)$ are labeled as *WFL-R* and *WFL-P*, respectively; the proposed method with WFL (especially, WFL-P) improved the maximization of rewards while reducing punishments.

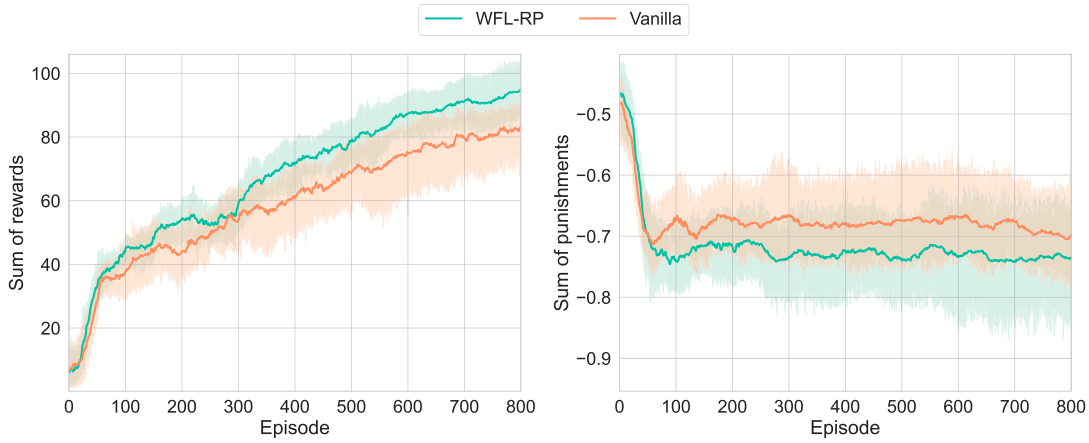

FIGURE 8
Learning results of real-world D'Claw: $\beta_0^{+,-} = (0.5, 0.5)$ is labeled as *WFL-RP*; accelerating the increase in a small reward was confirmed, although the increase in punishment by the side effect of the behavior for rotating the valve was not suppressed.

In other words, the goal is to stop the valve half a turn while allowing some overshoot. Note that, as the other differences from the above simulations, the ranges of motion and actions (i.e., the exploration capability) are restricted to avoid hardware malfunction.

First, the learning results with five trials are shown in Figure 8. Note that it was difficult to verify the asymmetric $\beta_0^{+,-}$ due to the cost of the real-robot experiments, even with automatic episode initialization. Therefore, the robustness of the proposed method to hyperparameters is demonstrated by adopting the symmetric *WFL-RP* as the proposed method under the assumption that there is limited prior knowledge (i.e., *WFL-P* was better in the simulations).

It was confirmed in r^+ that the proposed method always outperformed the conventional method. This is probably because the proposed method preferentially learned a small number of motion samples that rotated the valve forward, which were rarely obtained by chance with the limited exploration capability. Instead, r^- of the proposed method was slightly lower than that of the conventional method, probably because the side effect of the behavior to rotate the valve was larger than the behavior to maintain r^- at 0. Another possibility that should be noted is that the regularization of $\pi^+ \simeq \pi^-$ was added, but the large difference in scale between r^+ and r^- may have prevented it from functioning satisfactorily, and the policy to pursue r^+ may have been prioritized.

Next, task accomplishment, which cannot be evaluated from $r^{+,-}$ alone, is evaluated using the terminal valve angle with the post-learning policies. The five post-learning policies for each condition tested 10 episodes, as shown in Figure 9. As expected from the learning curve of r^+, the proposed method produced more results closer to the target angle, $\theta = \pi$. Moreover, when the range of $\pm\pi/4$ from the target angle is considered the success, the proposed method showed 41/50 (i.e., 82%), whereas the conventional method showed 35/50 (i.e., 70%).

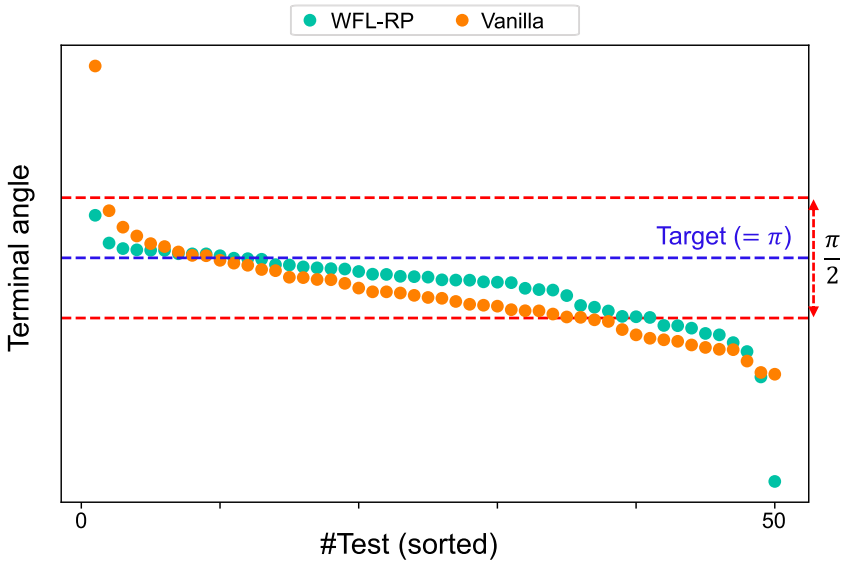

FIGURE 9
Terminal valve angles with the post-learning policies: 10 episodes were performed in each of the 5 models, and the (sorted) valve angles at the end of the episodes were plotted; *WFL-RP* tended to be closer to the target angle than that in the conventional method, and it also outperformed in the success rate where the error was within $\pi/4$.

6 Discussion and conclusion

6.1 Discussion

As shown above, although WFL in TD learning was confirmed in this study can be shown to produce more desirable learning processes and outcomes, whether WFL is more sensitive to reward design than conventional TD learning is remains an open question. For example, an inappropriate design may cause a conflict between the reward and punishment policies, potentially preventing them from achieving their respective objectives. Basically, objectives given as punishments r^- should have a high priority for achievement, and those with rewards r^+ should be regarded as value-added. However, prioritization among multiple objectives is often given as weights, which may lead to overlapping roles among multiple parameters and make it difficult to understand. The complexity is further increased by the fact that the impact of WFL can be adjusted by λ_β (or β).

Therefore, the need to assign such priorities to RL users and/or task designers may pose an obstacle to real-world applications. To alleviate this issue, further research on the design theory of reward functions suitable for this algorithm and/or the automation of assignment to $r^{+,-}$ (and tuning of hyperparameters) based on user preferences would increase the practical value of this algorithm. Recently, it might be a good idea to specify the context in LLM-based reward design (Ma et al., 2024) so that the necessary factors are set as r^- and those to be desired are r^+, while avoiding the conflict between them.

On the other hand, WFL, found in TD learning, originally explains the relationship between stimuli and perception in organisms, but it has not been discovered in brain activities related to TD learning. Considering that RL is also used as decision-making models for organisms and that the relationship between TD errors and brain activities has actually been verified, it is possible that WFL in TD learning may be latent in our brain activities.

Therefore, it would be important to verify the existence or absence of WFL using this algorithm for the analysis of brain-activity data. Moreover, the feedback from the findings may be able to elaborate on our algorithm: for example, the hyperparameters in the algorithm might be tuned by representing the brain-activity data. When conducting this investigation, it may be possible to derive a more sophisticated model if the WFL in the time direction derived by Maini and Tiganj (2025) can also be considered in addition to the proposed algorithm with the WFL in TD errors. Alternatively, brain-activity data may provide suggestions for new modeling of the heuristic update rule of $\beta^{+,-}$.

6.2 Conclusion

In this study, we revealed a novel nonlinearity in TD learning, WFL, which explains the relationship between stimuli and perception of organisms in the update rule of the value and policy functions in RL. Without loss of generality, it was implemented as a novel biologically plausible RL algorithm on the reward–punishment framework. We showed that the proposed method can be expected to explore a local solution to maximize rewards as early as possible while gradually aiming for a global solution to minimize punishments. Numerical verification indicated that the proposed method does not cause RL to collapse and retains the characteristics of WFL. The proposed method was also useful for robot control, and it outperformed the conventional method in the valve-turning task using D'Claw.

After addressing the limitations identified in this study, it would be valuable to test its generalizability (e.g., its capability to learn tasks with sparse rewards and/or discrete action spaces).

Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

Author contributions

KT: Investigation, Software, Validation, Visualization, Writing – original draft, Data curation, Formal analysis. TK: Conceptualization, Formal analysis, Funding acquisition, Methodology, Project administration, Software, Supervision, Writing – original draft, Writing – review and editing, Data curation. TY: Investigation, Resources, Software, Writing – review and editing. TM: Methodology, Project administration, Supervision, Writing – review and editing.

Funding

The author(s) declare that financial support was received for the research and/or publication of this article. This work was supported by JSPS KAKENHI, Development and validation of a unified theory of prediction and action, grant number JP24H02176.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The author(s) declare that no Generative AI was used in the creation of this manuscript.

Any alternative text (alt text) provided alongside figures in this article has been generated by Frontiers with the support of artificial intelligence and reasonable efforts have been made to ensure accuracy, including review by the authors wherever possible. If you identify any issues, please contact us.

Publisher's note

References

Ahn, M., Zhu, H., Hartikainen, K., Ponte, H., Gupta, A., Levine, S., et al. (2020). "Robel: robotics benchmarks for learning with low-cost robots," in *Conference on robot learning* (PMLR), 1300–1313.

Amari, S.-I. (1998). Natural gradient works efficiently in learning. *Neural Comput.* 10, 251–276. doi:10.1162/089976698300017746

Andrychowicz, O. M., Baker, B., Chociej, M., Jozefowicz, R., McGrew, B., Pachocki, J., et al. (2020). Learning dexterous in-hand manipulation. *Int. J. Robotics Res.* 39, 3–20. doi:10.1177/0278364919887447

Binhi, V. (2023). Magnetic navigation in animals, visual contrast sensitivity and the weber–fechner law. *Neurosci. Behav. Physiology* 53, 1036–1046. doi:10.1007/s11055-023-01497-3

Cayci, S., and Eryilmaz, A. (2024). Provably robust temporal difference learning for heavy-tailed rewards. *Adv. Neural Inf. Process. Syst.* 36.

Dabney, W., Kurth-Nelson, Z., Uchida, N., Starkweather, C. K., Hassabis, D., Munos, R., et al. (2020). A distributional code for value in dopamine-based reinforcement learning. *Nature* 577, 671–675. doi:10.1038/s41586-019-1924-6

Dayan, P., and Balleine, B. W. (2002). Reward, motivation, and reinforcement learning. *Neuron* 36, 285–298. doi:10.1016/s0896-6273(02)00963-7

Doya, K. (2021). Canonical cortical circuits and the duality of bayesian inference and optimal control. *Curr. Opin. Behav. Sci.* 41, 160–167. doi:10.1016/j.cobeha.2021.07.003

Foret, P., Kleiner, A., Mobahi, H., and Neyshabur, B. (2021). "Sharpness-aware minimization for efficiently improving generalization," in *International conference on learning Representations*.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. (2018). "Soft actor-critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning* (PMLR), 1861–1870.

Hoxha, I., Sperber, L., and Palminteri, S. (2025). Evolving choice hysteresis in reinforcement learning: comparing the adaptive value of positivity bias and gradual perseveration. *Proc. Natl. Acad. Sci. USA* 122, e2422144122. doi:10.1073/pnas.2422144122

Huang, S., Dossa, R. F. J., Ye, C., Braga, J., Chakraborty, D., Mehta, K., et al. (2022). Cleanrl: high-quality single-file implementations of deep reinforcement learning algorithms. *J. Mach. Learn. Res.* 23, 12585–12602.

Ilboudo, W. E. L., Kobayashi, T., and Matsubara, T. (2023). Adaterm: adaptive t-distribution estimated robust moments for noise-robust stochastic gradient optimization. *Neurocomputing* 557, 126692. doi:10.1016/j.neucom.2023.126692

Kaufmann, E., Bauersfeld, L., Loquercio, A., Müller, M., Koltun, V., and Scaramuzza, D. (2023). Champion-level drone racing using deep reinforcement learning. *Nature* 620, 982–987. doi:10.1038/s41586-023-06419-4

Kobayashi, T. (2022a). "L2c2: Locally lipschitz continuous constraint towards stable and smooth reinforcement learning," in *IEEE/RSJ international Conference on intelligent Robots and systems (IEEE)*, 4032–4039.

Kobayashi, T. (2022b). Optimistic reinforcement learning by forward kullback–leibler divergence optimization. *Neural Netw.* 152, 169–180. doi:10.1016/j.neunet.2022.04.021

Kobayashi, T. (2023). Proximal policy optimization with adaptive threshold for symmetric relative density ratio. *Results Control Optim.* 10, 100192. doi:10.1016/j.rico.2022.100192

Kobayashi, T. (2024a). "Consolidated adaptive t-soft update for deep reinforcement learning," in *International joint Conference on neural networks (IEEE)*, 1–8.

Kobayashi, T. (2024b). Drop: distributional and regular optimism and pessimism for reinforcement learning

Kobayashi, T., Aotani, T., Guadarrama-Olvera, J. R., Dean-Leon, E., and Cheng, G. (2019). "Reward-punishment actor-critic algorithm applying to robotic non-grasping manipulation," in *Joint IEEE international Conference on Development and Learning and Epigenetic robotics (IEEE)*, 37–42.

Kuniyoshi, Y., Yorozu, Y., Suzuki, S., Sangawa, S., Ohmura, Y., Terada, K., et al. (2007). Emergence and development of embodied cognition: a constructivist approach using robots. *Prog. brain Res.* 164, 425–445. doi:10.1016/s0079-6123(07)64023-0

Landro, N., Gallo, I., and La Grassa, R. (2020). Mixing adam and sgd: a combined optimization method

Levine, S. (2018). Reinforcement learning and control as probabilistic inference: Tutorial and review

Ma, Y. J., Liang, W., Wang, G., Huang, D.-A., Bastani, O., Jayaraman, D., et al. (2024). "Eureka: human-level reward design via coding large language models," in *International conference on learning Representations*.

Maini, S. S., and Tiganj, Z. (2025). Reinforcement learning with adaptive temporal discounting. *Reinf. Learn. J.*

Muller, T. H., Butler, J. L., Veselic, S., Miranda, B., Wallis, J. D., Dayan, P., et al. (2024). Distributional reinforcement learning in prefrontal cortex. *Nat. Neurosci.* 27, 403–408. doi:10.1038/s41593-023-01535-w

Nematollahi, I., Rosete-Beas, E., Röfer, A., Welschehold, T., Valada, A., and Burgard, W. (2022). "Robot skill adaptation via soft actor-critic Gaussian mixture models," in *International Conference on Robotics and automation (IEEE)*, 8651–8657.

Noorani, E., Mavridis, C. N., and Baras, J. S. (2023). "Exponential td learning: a risk-sensitive actor-critic reinforcement learning algorithm," in *American control conference (IEEE)*, 4104–4109.

Nutter, F. W., and Esker, P. D. (2006). The role of psychophysics in phytopathology: the weber–fechner law revisited. *Eur. J. Plant Pathology* 114, 199–213. doi:10.1007/s10658-005-4732-9

O'Doherty, J. P., Dayan, P., Friston, K., Critchley, H., and Dolan, R. J. (2003). Temporal difference models and reward-related learning in the human brain. *Neuron* 38, 329–337. doi:10.1016/s0896-6273(03)00169-7

Parr, T., Pezzulo, G., and Friston, K. J. (2022). *Active inference: the free energy principle in mind, brain, and behavior*. MIT Press.

Portugal, R. D., and Svaiter, B. F. (2011). Weber-fechner law and the optimality of the logarithmic scale. *Minds Mach.* 21, 73–81. doi:10.1007/s11023-010-9221-z

Radosavovic, I., Xiao, T., Zhang, B., Darrell, T., Malik, J., and Sreenath, K. (2024). Real-world humanoid locomotion with reinforcement learning. *Sci. Robotics* 9, eadi9579. doi:10.1126/scirobotics.adi9579

Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., and Dormann, N. (2021). Stable-baselines3: reliable reinforcement learning implementations. *J. Mach. Learn. Res.* 22, 1–8.

Scheler, G. (2017). Logarithmic distributions prove that intrinsic learning is Hebbian, *F1000Research* 6, 1222, doi:10.12688/f1000research.12130.1

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms

Schultz, W., Apicella, P., and Ljungberg, T. (1993). Responses of monkey dopamine neurons to reward and conditioned stimuli during successive steps of learning a delayed response task. *J. Neurosci.* 13, 900–913. doi:10.1523/jneurosci.13-03-00900.1993

Shen, Y., Tobia, M. J., Sommer, T., and Obermayer, K. (2014). Risk-sensitive reinforcement learning. *Neural Comput.* 26, 1298–1328. doi:10.1162/neco_a_00600

Smith, S. (2018). "Don't decay the learning rate, increase the batch size," in *International conference on learning Representations*.

Starkweather, C. K., and Uchida, N. (2021). Dopamine signals as temporal difference errors: recent advances. *Curr. Opin. Neurobiol.* 67, 95–105. doi:10.1016/j.conb.2020.08.014

Sugiyama, M., Hachiya, H., Kashima, H., and Morimura, T. (2009). "Least absolute policy iteration for robust value function approximation," in *IEEE international conference on robotics and automation* (IEEE), 2904–2909.

Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Mach. Learn.* 3, 9–44. doi:10.1007/bf00115009

Sutton, R. S., and Barto, A. G. (2018). *Reinforcement learning: an introduction*. MIT press.

Sutton, R. S., McAllester, D., Singh, S., and Mansour, Y. (1999). Policy gradient methods for reinforcement learning with function approximation. *Adv. neural Inf. Process. Syst.* 12.

Todorov, E., Erez, T., and Tassa, Y. (2012). "Mujoco: a physics engine for model-based control," in *IEEE/RSJ international conference on intelligent robots and systems (IEEE)*, 5026–5033.

Wahid, A., Stone, A., Chen, K., Ichter, B., and Toshev, A. (2021). "Learning object-conditioned exploration using distributed soft actor critic," in *Conference on robot learning* (PMLR), 1684–1695.

Wang, J., Elfwing, S., and Uchibe, E. (2021). Modular deep reinforcement learning from reward and punishment for robot navigation. *Neural Netw.* 135, 115–126. doi:10.1016/j.neunet.2020.12.001

Yamanokuchi, T., Kwon, Y., Tsurumine, Y., Uchibe, E., Morimoto, J., and Matsubara, T. (2022). Randomized-to-canonical model predictive control for real-world visual robotic manipulation. *IEEE Robotics Automation Lett.* 7, 8964–8971. doi:10.1109/lra.2022.3189156

Appendix A

1 Natural gradient for θ

The Fisher information for $p(O|V)$, $\mathcal{I}(V)$, is derived as follows:

$$\begin{aligned}\mathcal{I}(V) &= \mathbb{E}_{p(O|V)}\left[\left(\frac{\partial}{\partial V}\ln p(O|V)\right)^2\right] = p(O=1|V)\left(\frac{\partial}{\partial V}\ln p(O=1|V)\right)^2 \\ &+ p(O=0|V)\left(\frac{\partial}{\partial V}\ln p(O=0|V)\right)^2 = \beta^2 p_V + (1-p_V)\left(\frac{-\beta p_V}{1-p_V}\right)^2 \\ &= \beta^2 p_V\left(1+\frac{p_V}{1-p_V}\right) = \beta^2\frac{p_V}{1-p_V} \end{aligned} \tag{22}$$

where $p_V = p(O=1|V) = e^{\beta(V-\bar{R})}$.

The natural gradient is obtained by dividing the raw gradient by the Fisher information (Amari, 1998). As the raw gradient g^{raw} is represented as $-\{(1-\lambda_\beta)\delta + \lambda_\beta\delta_{\ln}\}\beta(1+\beta)p_V\nabla_\theta V$, its natural gradient g^{nat} can be given as $-\{(1-\lambda_\beta)\delta + \lambda_\beta\delta_{\ln}\}\beta^{-1}(1+\beta)(1-p_V)\nabla_\theta V$. By removing the constant coefficients related to β, their summation can be derived as $-\{(1-\lambda_\beta)\delta + \lambda_\beta\delta_{\ln}\}\nabla_\theta V$.

2 Learning conditions

The value and policy functions for $r^{+,-}$ are independently approximated by a common network structure. The structure has two fully connected layers as hidden layers, with 100 neurons for each, and ReLU function is employed as its activation function. AdaTerm (Ilboudo et al., 2023) is employed to optimize the network parameters, and its learning rate is set to 1×10^{-3} for the toy problem and 5×10^{-4} for the robotic task. The sample efficiency is improved with experience replay, although its buffer size is small (i.e. 1×10^{4}) to emphasize on-policyness. Half of the stored experiences are randomly replayed at the end of each episode with 32 the batch size. As tricks to stabilize learning, we introduce PPO-RPE (Kobayashi, 2023) introduced in the main text, target networks with CAT-soft update (Kobayashi, 2024a), and regularization to make the functions smooth by L2C2 (Kobayashi, 2022a). All of these remain the default hyperparameters. Other hyperparameters are $\gamma = 0.99$ for the discount rate and $\zeta = 0.999$ for the empirical TD error scale estimation in Equation 15.