Generalized Gradient Emphasis Learning for Off-Policy Evaluation and Control with Function Approximation

Jiaqing Cao (jqcao@stu.suda.edu.cn)
Soochow University
Quan Liu
Soochow University
Lan Wu
Soochow University
Qiming Fu
Suzhou University of Science and Technology
Shan Zhong
Changshu Institute of Technology

Research Article

Keywords: reinforcement learning, off-policy learning, emphatic approach, gradient temporal-difference learning, gradient emphasis learning

Posted Date: October 4th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-2115364/v1

Generalized Gradient Emphasis Learning for Off-Policy Evaluation and Control with Function Approximation

Jiaqing Cao[1], Quan Liu[1*], Lan Wu[1], Qiming Fu[2] and Shan Zhong[3]

[1*]School of Computer Science and Technology, Soochow University, Suzhou, 215006, China.
[2]School of Electronics and Information Engineering, Suzhou University of Science and Technology, Suzhou, 215009, China.
[3]School of Computer Science and Engineering, Changshu Institute of Technology, Changshu, 215500, China.

*Corresponding author(s). E-mail(s): quanliu@suda.edu.cn;

Abstract

Emphatic temporal-difference (TD) learning (Sutton et al. 2016) is a pioneering off-policy reinforcement learning method involving the use of the *followon trace*. The recently proposed Gradient Emphasis Learning (GEM, Zhang et al. 2020) algorithm is used to fix the problems of unbounded variance and large emphasis approximation error introduced by the *followon trace* from the perspective of stochastic approximation. In this paper, we rethink GEM and introduce a novel generalized GEM($\boldsymbol{\beta}$) algorithm to learn the true emphasis. The key to the construction of the generalized GEM($\boldsymbol{\beta}$) algorithm is introducing a tunable hyperparameter $\boldsymbol{\beta}$ that is not necessarily the same as the discount factor $\boldsymbol{\gamma}$ to the *GEM operator*. We then apply the emphasis estimated by the proposed GEM($\boldsymbol{\beta}$) algorithm to the value estimation gradient and the policy gradient, respectively, yielding the corresponding emphatic TD variant for off-policy evaluation and actor-critic algorithm for off-policy control. Finally, we demonstrate empirically the advantage of the proposed algorithms across a range of problems, for both off-policy evaluation and off-policy control, and for both linear and nonlinear function approximation.

Keywords: reinforcement learning, off-policy learning, emphatic approach, gradient temporal-difference learning, gradient emphasis learning

1 Introduction

Off-policy learning, where an agent learns its current target policy while following a different behavior policy, provides an agent with opportunities to accumulate a wealth of knowledge by learning about the effects of different behavioral policies and underpins many practical implementations of reinforcement learning (RL, [3]) [4, 5, 6]. In many cases, we would prefer off-policy learning to some extent to learn about the greedy policy while exploring [7, 8], to enable data to be generated from one behavior policy to evaluate multiple target policies simultaneously [9, 10], to improve data efficiency via experience replay [11], or to correct data discrepancies introduced by the distributed computation [12]. Unfortunately, the combination of off-policy learning, function approximation, and bootstrapping via temporal-difference (TD) updates, known as the *deadly triad*, can destabilize learning resulting in "soft divergence" and slow convergence [3, 13, 14, 15].

In general, off-policy learning is challenging because the sampling distribution is different from the distribution under the desired evaluation policy, which is referred to as the *distribution mismatch* [16, 17, 18]. There are two kinds of tasks in off-policy RL, evaluation and control. The problem of off-policy evaluation (OPE, [19, 20]), where we want to predict the performance of a given target policy (averaged reward in the continuing setting or expected total discounted reward in the episode setting [21]) with samples collected by one or more different behavior policies, serves as a crucial step for developing efficient off-policy policy optimization algorithms [22, 23]. The work of off-policy policy optimization algorithms began with the Off-Policy Actor-Critic (Off-PAC) algorithm proposed by Degris et al. [24]. However, Off-PAC ignores the *distribution mismatch* between the behavior and target policies, and is convergent only in the tabular setting.

The Emphatic TD (ETD, [1]) algorithm resolves the instability due to the *distribution mismatch* by partially adjusting the updates of off-policy TD to be under the on-policy distribution through the introduced *followon trace* [1]. After this, Imani et al. [25] developed a new off-policy actor-critic algorithm called Actor-Critic with Emphatic weightings (ACE), which reweights Off-PAC updates with the use of *followon trace* as the emphasis. However, the *followon trace* tends to have unbounded variance and large emphasis approximation error, limiting the applicability of the methods. To remedy, Zhang et al. [2] proposed a gradient-based stochastic approximation algorithm, Gradient Emphasis Learning (GEM), to estimate the true emphasis, which poses a promising solution to the bias-variance trade-off. At the same time, a Convergent Off-Policy Actor-Critic (COF-PAC) algorithm is developed based on GEM.

In this paper, we focus on emphatic methods, and how to better learn the emphasis and derive more efficient OPE and off-policy control algorithms is the focus of our work. Thus, in this work, we draw inspiration from Hallak et al. [26] and develop a flexible and general algorithm for the bias-variance trade-off of the emphasis learning to alleviate the problem-dependent

dilemma. The contributions of our paper are as follows:

- We introduce a freely tunable hyperparameter β to the *GEM operator*, yielding the generalized GEM(β) algorithm. We then extend the GEM(β) algorithm to the ETD variant, resulting in the corresponding GEM-ETD(β) algorithm for OPE.
- We present the convergence analysis of the GEM(β) algorithm under standard off-policy and stochastic approximation conditions, and establish a theoretical characterization of the stability for the derivative GEM-ETD(β) algorithm.
- We empirically investigate the merits of GEM(β) in emphasis approximation on the diagnostic OPE benchmark under various function representations. For OPE, we compare the GEM-ETD(β) algorithm with the vanilla off-policy versions of ETD, GEM-ETD, and ETD(0, β) algorithms, and demonstrate the advantage of our GEM-ETD(β) algorithm by showcasing its improved performance.
- To investigate the practical benefits of our GEM(β) algorithm when used at scale, we further extend it to the actor-critic algorithm and name it COF-PAC(β). Finally, we evaluate COF-PAC(β) empirically on classic control tasks with neural network function approximators, and demonstrate the scalability and broad applicability of COF-PAC(β).

The structure of this paper is as follows. Sec. 2 describes the related work. Sec. 3 explains the notation and background. In Sec. 4, the generalized GEM(β) algorithm and its extensions are developed, and the corresponding theoretical analysis is presented in detail. In Sec. 5, the experimental results are presented and discussed. Sec. 6 is the conclusion and suggestions for future work.

2 Related Work

TD learning presented by Sutton [27] is perhaps the most powerful method for policy evaluation in RL. The divergence of off-policy linear TD was well documented by Tsitsiklis and Van Roy [28], as highlighted in the seminal counterexample by Baird [29]. The fundamental issue of divergence is the combination of function approximation, off-policy learning, and bootstrapping, which is known as the *deadly triad* [3, 13, 14, 15]. To address such an issue, GTD methods [30, 31, 32] were proposed. Unlike the semi-gradient TD algorithm, GTD methods are true stochastic gradient methods and enjoy convergence guarantees. However, they involve two sets of parameters and the need to tune the second stepsize in a problem-dependent way for good performance, making them hard to use in practice. We will include the most pertinent work to ours below due to the extensive research on emphatic methods.

The ETD approach, which was originally proposed in the pioneering work of Sutton et al. [1], involves a scalar *followon trace* to surmount the *distribution mismatch* issue with off-policy learning. On the other hand, Hallak et al. [26] demonstrated that the variance of the *followon trace* can be unbounded over a long or infinite time horizon. They further proposed the ETD(0, β) framework to bound the *followon trace* with a tunable hyperparameter but at the cost of a possibly large bias error, where β is a variable decay rate used for bias-variance trade-off. Jiang et al. [13] provided a new variant of ETD, where the *followon trace* is extended to cope with multi-step TD methods like V-trace [12]. Zhang and Whiteson [33] proposed to truncate the update of the *followon trace* to bound the variance of ETD. In the same period, Guan et al. [15] proposed a novel PER-ETD algorithm, which restarts and updates the *followon trace* only for a finite period for each update of the value function parameter, effectively reducing the variance of the *followon trace*.

The above variants belong to the Monte Carlo trace. In this work, we focus on estimating and using the expectation of trace from the perspective of function approximation instead of the instantaneous trace. The idea of learning expected trace as a function of state has been explored by van Hasselt et al. [34]. Jiang et al. [14] utilized the simple semi-gradient TD update for learning expected *followon trace* and studied ETD with deep neural loss function. However, since the asymptotic update matrix is not guaranteed to be positive definite, resulting in the update fails to satisfy the prerequisite stability for full convergence of the stochastic algorithm. Jiang et al. [14] then proposed two stabilization techniques to facilitate at-scale learning. Motivated by the guaranteed convergence of GTD methods under off-policy learning, the GEM algorithm proposed by Zhang et al. [2] to learn the expected *followon trace* through the GTD2-style update can somewhat alleviate the high-variance issue of the original *followon trace*, which is the inspiration for our work.

An extension to the emphatic method is the ACE algorithm [25], which applies the *followon trace* to policy gradient updates. Estimating the emphasis of a state using the *followon trace* is similar to estimating the value of a state using a single Monte Carlo return. As a result, the *followon trace* can have unbounded variance and large emphasis approximation error, complicating the convergence of ACE. Instead of using the *followon trace*, Zhang et al. [2] propose a novel stochastic approximation algorithm, GEM, to approximate the emphasis in COF-PAC. This reduces well-known variance issues with the *followon trace*.

In addition to emphatic methods, there are also other methods for addressing the *distribution mismatch* problem of off-policy learning, including density-ratio-based methods [16, 17, 18, 35, 36, 37, 38] and target-network-based methods [39]. Zhang et al. [2, 40] showed that some density ratios can be interpreted as special emphasis.

Traces provide temporal credit assignment in the backward view and thus rely on previous historical experience. Hence, the learning algorithms designed to estimate the expected trace require the time-indexed reversed. The idea of

bootstrapping in the reverse direction was explored by Gelada and Bellemare [37]; Wang et al. [41]; Hallak and Mannor [42]; Zhang et al. [43].

3 Background

We use time-indexed uppercase letters (e.g., S_t) to denote random variables and lowercase letters (e.g., $S_t = s$) to denote the values obtained. Multi-dimensional functions or vectors are bolded (e.g., $\boldsymbol{\theta}$), as are matrices (e.g., $\boldsymbol{\Phi}$). When it does not confuse, we use vectors and functions interchangeably. All vectors are column vectors. We use $\|\boldsymbol{x}\|_{\boldsymbol{\Xi}} \doteq \sqrt{\boldsymbol{x}^\top \boldsymbol{\Xi} \boldsymbol{x}}$ to denote the norm induced by a positive definite matrix $\boldsymbol{\Xi}$, which induces the matrix norm $\|\boldsymbol{A}\|_{\boldsymbol{\Xi}} \doteq \sup_{\|\boldsymbol{x}\|_{\boldsymbol{\Xi}}=1} \|\boldsymbol{A}\boldsymbol{x}\|_{\boldsymbol{\Xi}}$. We use $\mathbf{I}$ to denote the identity matrix and $\mathbf{1}$ to denote an all-one matrix. We indicate sets by calligraphic font (e.g., $\mathcal{S}$). For all state-dependent functions, we also allow time-dependent shorthands (e.g., $\boldsymbol{\phi}_t = \boldsymbol{\phi}(S_t)$).

We consider the mathematically idealized form of the RL problem, modeled as an infinite-horizon Markov Decision Process (MDP, [21]) $(\mathcal{S}, \mathcal{A}, p, r, \gamma)$ consisting of a finite state space $\mathcal{S}$ of $|\mathcal{S}|$ states, a finite action space $\mathcal{A}$ of $|\mathcal{A}|$ actions, a state-transition distribution $p : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$, a reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, and a discount factor $\gamma \in [0, 1)$. The agent and environment interact continually. Given a target policy π mapping states to distributions over actions, the agent takes an action $A_t \in \mathcal{A}$ at state $S_t \in \mathcal{S}$ according to $\pi(\cdot|S_t) \in [0, 1]$ at time step t, where $\pi(A_t|S_t)$ denotes the probability of taking action A_t at state S_t. In response, the environment then transitions to the next state $S_{t+1} \in \mathcal{S}$ according to $p(\cdot|S_t, A_t)$ and emits a reward $R_{t+1} \doteq r(S_t, A_t, S_{t+1})$. We use $\mathbf{P}_\pi \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$ to denote the state transition probability matrix induced by π, i.e., $\mathbf{P}_\pi[s, s'] \doteq \sum_a \pi(s, a)p(s'|s, a)$. The goal of policy evaluation is to estimate the value function $\boldsymbol{v}_\pi$ (Assumption 1 below ensures $\boldsymbol{v}_\pi$ is well-defined), defined as the expectation of the discounted total rewards under the target policy π:

$$v_\pi(s) \doteq \mathbb{E}_\pi \left[\sum_{t=0}^{T} \gamma^t R_{t+1} | S_t = s \right], \text{ for all } s \in \mathcal{S}, \tag{1}$$

where $\mathbb{E}_\pi$ denotes the expectation w.r.t. the probability distribution of states, actions, and rewards, generated under the target policy π, and T is for instance the time the current episode terminates or $T = \infty$. It is well-known that $\boldsymbol{v}_\pi$ satisfies the so-called *Bellman equation*:

$$\begin{aligned} \boldsymbol{v}_\pi &= \boldsymbol{R} + \gamma \mathbf{P}_\pi \boldsymbol{v}_\pi \\ &\doteq B\boldsymbol{v}_\pi, \end{aligned} \tag{2}$$

where $\boldsymbol{R} \in \mathbb{R}^{|\mathcal{S}|}$ denotes the reward vector induced by policy π, with $R(s) = \mathbb{E}_\pi[R_{t+1}|S_t = s]$, B is known as the *Bellman operator*.

We consider function approximation and seek to learn the parameter vector $\boldsymbol{\theta} \in \mathbb{R}^n$, with $n \ll |\mathcal{S}|$, such that $v_{\boldsymbol{\theta}}(s) \approx v_\pi(s)$, for all $s \in \mathcal{S}$, under an arbitrary fixed target policy π. In the case of off-policy learning, we expect to select actions with the action selection probability $\mu(a|s) \in [0, 1]$ following the behavior policy μ and then generate a series of state transitions and rewards as the observation data. Let $\boldsymbol{d}_\mu \in \mathbb{R}^{|\mathcal{S}|}$ and $\boldsymbol{d}_\pi \in \mathbb{R}^{|\mathcal{S}|}$ be the stationary distributions of μ and π, respectively. We define $\mathbf{D} \doteq diag(\boldsymbol{d}_\mu) \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$ and $\mathbf{D}_\pi \doteq diag(\boldsymbol{d}_\pi) \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$. Assumption 3 below ensures $\boldsymbol{d}_\mu$ exists and $\mathbf{D}$ is invertible.

3.1 Off-Policy Evaluation

We use the linear function approximation to demonstrate the RL algorithms for OPE. In the linear function approximation setting, the value function $\boldsymbol{v}_\pi$ is approximated as a linear combination of some features representing the state: $v_\pi(s) \approx \boldsymbol{\theta}^\top \boldsymbol{\phi}(s)$, where $\boldsymbol{\phi}(s) \in \mathbb{R}^n$ is the feature vector characterizing state s. We use $\boldsymbol{\Phi} \in \mathbb{R}^{|\mathcal{S}|\times n}$ to denote the feature matrix, where each row of $\boldsymbol{\Phi}$ is $\boldsymbol{\phi}(s)^\top$.

Emphatic TD(λ, β): Prior work by Precup et al. [44] attempted to completely correct the *distribution mismatch* using the product of all importance sampling (IS) ratios from time 0, and thereby reweighting the updates of off-policy TD(λ). It is theoretically possible to convert the state distribution from $\boldsymbol{d}_\mu$ to $\boldsymbol{d}_\pi$. Unfortunately, the variance of this method is extremely large, thus limiting its practicality. Lately, Sutton et al. [1] proposed the ETD(λ) algorithm with much less variance than the IS-TD(λ) method. The variance is tamped down by incorporating the discount factor γ over the product of IS ratios. More generally, the ETD(λ, β) algorithm, which encompasses the vanilla ETD(λ), using a free hyperparameter β in place of the discount factor γ to control variance. For $\lambda = 0$, ETD$(0, \beta)$ updates the parameter vector $\boldsymbol{\theta}$ recursively as

$$M_t = \beta\rho_{t-1}M_{t-1} + i_t,\ M_0 = i(S_0), \tag{3}$$

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha_t\rho_t M_t\delta_t\boldsymbol{\phi}_t, \tag{4}$$

where $\alpha_t > 0$ is the stepsize parameter, $\beta \in (0, 1]$ is a variable decay rate that is not necessarily the same as γ, in particular, for $\beta = \gamma$, M_t is the *followon trace* [1]. $\rho_t \doteq \frac{\pi(A_t|S_t)}{\mu(A_t|S_t)}$ is the IS ratio, which is used to compensate for the data value difference caused by sampling according to the behavior policy μ (Assumption 3 below ensures ρ is well-defined), $\delta_t = R_{t+1} + \gamma\boldsymbol{\theta}_t^\top\boldsymbol{\phi}_{t+1} - \boldsymbol{\theta}_t^\top\boldsymbol{\phi}_t$, is the conventional TD error, and $\boldsymbol{i} : \mathcal{S} \to [0, \infty)$ is the interest function used to represent the user's preference for different states. In practice, the interest is usually set to 1 for all states, meaning that they are all equally important.

Gradient Emphasis Learning: Unlike the previous Monte Carlo style trace M_t, GEM seeks to learn the limiting expected emphatic trace $\boldsymbol{m}_\pi$ using function approximation. We refer to $\boldsymbol{m}_\pi$ as the emphasis in the rest of this paper. To elaborate, GEM introduces a linear parametric function $\boldsymbol{m}_{\boldsymbol{w}} \doteq \boldsymbol{\Phi}\boldsymbol{w}$

such that $m_{\boldsymbol{w}}(s)$ approximates $m_\pi(s) = \lim_{t\to\infty} \mathbb{E}_\mu[M_t|S_t = s]$ when $\beta = \gamma$.[1] Inspired by GTD methods, GEM seeks to find an approximate solution that satisfies $\boldsymbol{m}_{\boldsymbol{w}} = \Pi\hat{\mathcal{T}}\boldsymbol{m}_{\boldsymbol{w}}$ via minimizing the projection objective $\left\|\Pi\bar{\delta}_{\boldsymbol{w}}\right\|^2_{\mathbf{D}}$, where $\Pi = \boldsymbol{\Phi}(\boldsymbol{\Phi}^\top\mathbf{D}\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^\top\mathbf{D}$ is the projection operator (Assumption 4 below ensures the existence of $(\boldsymbol{\Phi}^\top\mathbf{D}\boldsymbol{\Phi})^{-1}$), $\hat{\mathcal{T}}$ is the *GEM operator* defined as $\hat{\mathcal{T}}\boldsymbol{m}_{\boldsymbol{w}} \doteq \boldsymbol{i} + \gamma\mathbf{D}^{-1}\mathbf{P}_\pi^\top\mathbf{D}\boldsymbol{m}_{\boldsymbol{w}}$, and $\bar{\delta}_{\boldsymbol{w}} \doteq \hat{\mathcal{T}}\boldsymbol{m}_{\boldsymbol{w}} - \boldsymbol{m}_{\boldsymbol{w}}$. The GEM algorithm updates $\boldsymbol{\kappa}$ and $\boldsymbol{w}$ recursively as

$$\boldsymbol{\kappa}_{t+1} = \boldsymbol{\kappa}_t + \zeta_t(\bar{\delta}_t - \boldsymbol{\kappa}_t^\top\boldsymbol{\phi}_{t+1})\boldsymbol{\phi}_{t+1}, \tag{5}$$

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \zeta_t[(\boldsymbol{\phi}_{t+1} - \gamma\rho_t\boldsymbol{\phi}_t)\boldsymbol{\kappa}_t^\top\boldsymbol{\phi}_{t+1}], \tag{6}$$

where $\{\zeta_t\}$ is a sequence of deterministic nonnegative non-increasing learning rates satisfying Assumption 6, and $\bar{\delta}_t \doteq i_{t+1} + \gamma\rho_t\boldsymbol{w}_t^\top\boldsymbol{\phi}_t - \boldsymbol{w}_t^\top\boldsymbol{\phi}_{t+1}$ is the *GEM error* in reversed time with a reward signal of i_{t+1} for each time step t.

3.2 Off-Policy Control

In this paper, we focus on policy-based control. In off-policy control, the target policy π is a differentiable function parameterized by $\boldsymbol{\varpi} \in \mathbb{R}^n$, with $n \ll |\mathcal{S}|$. The goal of off-policy actor-critic methods [2, 24, 25] is to maximize the excursion objective $J(\pi) \doteq \sum_{s\in\mathcal{S}} d_\mu(s)i(s)v_\pi(s)$ by adapting the target policy π. For $J(\pi)$, the policy gradient is

$$\nabla J(\pi) = \mathbb{E}_{s\sim d_\mu, a\sim\mu}[m_\pi(s)\rho(s,a)q_\pi(s,a)\nabla\log\pi(a|s)], \tag{7}$$

where

$$q_\pi(s,a) \doteq \mathbb{E}_\pi\left[\sum_{t=0}^{T}\gamma^t R_{t+1}|S_t = s, A_t = a\right], \text{ for all } s\in\mathcal{S}, a\in\mathcal{A},$$

is the state-action value function of π. To compute $\nabla J(\pi)$, we need $\boldsymbol{q}_\pi$ and $\boldsymbol{m}_\pi$, to which we typically do not have access. For $\boldsymbol{q}_\pi$, Imani et al. [25] and Zhang et al. [2] commonly use the conventional TD error δ_t as the alternative to $\boldsymbol{q}_\pi$. The key difficulty is then in estimating $\boldsymbol{m}_\pi$. Degris et al. [24] ignore the emphasis $\boldsymbol{m}_\pi$ and update $\boldsymbol{\varpi}$ as $\boldsymbol{\varpi}_{t+1} = \boldsymbol{\varpi}_t + \alpha\rho_t q_\pi(S_t, A_t)\nabla\log\pi(A_t|S_t)$ in Off-PAC, which is theoretically justified only in the tabular setting. Imani et al. [25] approximate $m_\pi(S_t)$ with the *followon trace* M_t, resulting in the ACE update

$$\boldsymbol{\varpi}_{t+1} = \boldsymbol{\varpi}_t + \alpha\rho_t M_t q_\pi(S_t, A_t)\nabla\log\pi(A_t|S_t). \tag{8}$$

[1] Sutton et al. [1] show that $\lim_{t\to\infty}\mathbb{E}_\mu[M_t|S_t = s] = m_\pi(s)$. The existence of this limit is established in Lemma 1 in Zhang et al. [45].

COF-PAC proposed by Zhang et al. [2] is based on the gradient expression in the ACE algorithm. The difference is that COF-PAC uses the emphasis learned by GEM to reweight the update. COF-PAC updates $\boldsymbol{\varpi}$ as

$$\boldsymbol{\varpi}_{t+1} = \boldsymbol{\varpi}_t + \alpha\rho_t m_{\boldsymbol{w}_t} q_\pi(S_t, A_t)\nabla \log \pi(A_t|S_t), \tag{9}$$

where $m_{\boldsymbol{w}_t} \doteq \boldsymbol{w}_t^\top \boldsymbol{\phi}_t$ and $\boldsymbol{w}_t$ is updated according to GEM (Eqs. (5) and (6)).

3.3 Assumptions

In order to analyze the off-policy parametric TD-style and GTD-style algorithms in the literature, we now make the following standard assumptions.

Assumption 1 (Condition on the target policy) The target policy π is such that $(\mathbf{I} - \gamma \mathbf{P}_\pi)^{-1}$ exists.

Assumption 2 (Condition on the features) The feature matrix $\boldsymbol{\Phi}$ has full column rank.

Assumption 3 (Convergence of behavior policy) The behavior policy μ induces an ergodic Markov chain on $\mathcal{S}$, and, moreover, for all $(s, a) \in \mathcal{S} \times \mathcal{A}$, $\mu(a|s) > 0$ if $\pi(a|s) > 0$.

Assumption 4 (Problem solvability) The matrices $\mathbf{C} \doteq \boldsymbol{\Phi}^\top \mathbf{D} \boldsymbol{\Phi}$ and $\bar{\mathbf{A}} \doteq \boldsymbol{\Phi}^\top \mathbf{D}(\mathbf{I} - \beta \mathbf{P}_\pi^\top)\boldsymbol{\Phi}$ are nonsingular.

Assumption 5 (Boundedness and i.i.d. conditions) $(\boldsymbol{\phi}_t, R_t, \boldsymbol{\phi}_{t+1}, \rho_t)_{t\geq 0}$ is an independent, identically distributed (i.i.d.) sequence with uniformly bounded second moments for states and rewards.

Assumption 6 (Stepsize condition) The learning rates $\{\zeta_t\}$ are deterministic, nonnegative, non-increasing, and satisfies the Robbins-Monro condition [46], i.e., $\sum_{t=0}^{\infty} \zeta_t = \infty$, $\sum_{t=0}^{\infty} \zeta_t^2 < +\infty$.

Remark 1 Assumptions 2 and 3 are standard in the off-policy RL literature [3, 47, 48]. In Assumption 3, ergodicity holds if the Markov chain induced by the behavior policy μ is irreducible and aperiodic [49]. Such an assumption ensures a unique stationary distribution $\boldsymbol{d}_\mu$. Assumptions 4 and 5 is commonly used in parametric GTD methods [2, 16, 30, 31, 32, 50]. The non-singularity of matrices in Assumption 4 can be satisfied by using linearly independent features.

4 Proposed Emphatic Algorithms

In this paper, we develop the generalized GEM(β) algorithm by introducing a freely tunable hyperparameter $\beta \in (0, 1]$ to the *GEM operator* $\hat{\mathcal{T}}$. In this

case, $\hat{\mathcal{T}}_\beta$ is defined as $\hat{\mathcal{T}}_\beta \boldsymbol{y} \doteq \boldsymbol{i} + \beta \mathbf{D}^{-1}\mathbf{P}_\pi^\top \mathbf{D}\boldsymbol{y}$, for any vector $\boldsymbol{y} \in \mathbb{R}^{|\mathcal{S}|}$, and the true emphasis $\boldsymbol{m}_\pi \doteq \mathbf{D}^{-1}(\mathbf{I} - \beta \mathbf{P}_\pi^\top)^{-1}\mathbf{D}\boldsymbol{i}$. Given Prop. 1 below, the GEM(β) algorithm can be established following the same derivation routine as Zhang et al. [2]. We now present the GEM(β) algorithm, which updates $\boldsymbol{\kappa}$ and $\boldsymbol{w}$ recursively as

$$\boldsymbol{\kappa}_{t+1} = \boldsymbol{\kappa}_t + \zeta_t(\bar{\delta}_t(\beta) - \boldsymbol{\kappa}_t^\top \boldsymbol{\phi}_{t+1})\boldsymbol{\phi}_{t+1}, \tag{10}$$

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \zeta_t[(\boldsymbol{\phi}_{t+1} - \beta\rho_t\boldsymbol{\phi}_t)\boldsymbol{\kappa}_t^\top \boldsymbol{\phi}_{t+1}], \tag{11}$$

where $\bar{\delta}_t(\beta) \doteq i_{t+1} + \beta\rho_t \boldsymbol{w}_t^\top \boldsymbol{\phi}_t - \boldsymbol{w}_t^\top \boldsymbol{\phi}_{t+1}$. Then the derivative GEM-ETD$(0, \beta)$ updates $\boldsymbol{\theta}$ iteratively as

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha_t \boldsymbol{m}_{\boldsymbol{w}_t} \rho_t \delta_t \boldsymbol{\phi}_t, \tag{12}$$

where $\boldsymbol{m}_{\boldsymbol{w}_t} \doteq \boldsymbol{w}_t^\top \boldsymbol{\phi}_t$ and $\boldsymbol{w}_t$ is updated according to GEM(β) (Eqs. (10) and (11)). Algorithm 1 provides the pseudocode of GEM-ETD$(0, \beta)$ for OPE.

Motivation: The inspiration for the algorithm comes from the work of Hallak et al. [26]. Hallak et al. [26] demonstrate that by controlling β in the ETD(λ, β) algorithm, the variance of the algorithm can be reduced while still maintaining a reasonable bias, thus allowing for improved performance. The vanilla GEM algorithm for emphasis estimation and its variant GEM-ETD for OPE truly trade off bias and variance. However, they are problem-dependent. For example, as pointed out in Zhang et al. [2], "if the states are heavily aliased, the GEM emphasis estimation may be heavily biased, as well GEM-ETD" in the more challenging modified Baird's counterexample [29]. Zhang et al. [2] demonstrate that this is mainly due to the fact that the magnitude of the *GEM error* $\bar{\delta}_t$ varies dramatically across different states. Thus, in order to obtain the optimal choice for different problems, we investigate whether the bias and variance of the algorithm can be managed flexibly by providing more freedom in the choice of β.

Proposition 1 $\hat{\mathcal{T}}_\beta$ is a contraction mapping w.r.t some weighted maximum norm and $\boldsymbol{m}_\pi$ is its unique fixed point.

Proof The proof involves Corollary 6.1 in Bertsekas and Tsitsiklis [51]. Details are provided in Appendix A.1. □

Remark 2 In this paper, for the sake of brevity, we consider the simplest setting of $\lambda = 0$, as does Zhang et al. [2]. The focus of our work is to observe qualitative properties such as convergence, bias, and variance that arise from the generalized GEM algorithm, rather than eligibility traces. From here on we refer to GEM-ETD$(0, \beta)$ as GEM-ETD(β) and ETD$(0, \beta)$ as ETD(β).

Algorithm 1 GEM-ETD$(0, \beta)$ for OPE

Input: $\{\alpha_t\}$, $\{\zeta_t\}$: stepsize sequence; $\beta \in (0, 1]$: decay rate; $\gamma \in [0, 1]$: discount factor; π: target policy; μ: behavior policy; $\boldsymbol{i} : \mathcal{S} \to [0, \infty)$: interest function.

for each episode **do**
  Initialize $\boldsymbol{w}_0$, $\boldsymbol{\kappa}_0$, $\boldsymbol{\theta}_0$.
  Get an initial random state S_0.
  Set $\rho_i = \pi(A_i|S_i)/\mu(A_i|S_i)$, for $i = 0, 1, 2, ..., T$.
  for $t = 0, 1, 2, ..., T$ **do**
    Take A_t from S_t according to μ, and arrive at S_{t+1}.
    Observe sample $(\boldsymbol{\phi}_t, R_{t+1}, \boldsymbol{\phi}_{t+1})$ at time step t.
    $\bar{\delta}_t \leftarrow i_{t+1} + \beta\rho_t \boldsymbol{w}_t^\top \boldsymbol{\phi}_t - \boldsymbol{w}_t^\top \boldsymbol{\phi}_{t+1}$.
    $\boldsymbol{\kappa}_{t+1} \leftarrow \boldsymbol{\kappa}_t + \zeta_t(\bar{\delta}_t - \boldsymbol{\kappa}_t^\top \boldsymbol{\phi}_{t+1})\boldsymbol{\phi}_{t+1}$.
    $\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t + \zeta_t[(\boldsymbol{\phi}_{t+1} - \beta\rho_t\boldsymbol{\phi}_t)\boldsymbol{\kappa}_t^\top \boldsymbol{\phi}_{t+1}]$.
    $\delta_t \leftarrow R_{t+1} + \gamma\rho_{t+1}\boldsymbol{\theta}_t^\top \boldsymbol{\phi}_{t+1} - \boldsymbol{\theta}_t^\top \boldsymbol{\phi}_t$.
    $\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t + \alpha_t(\boldsymbol{w}_t^\top \boldsymbol{\phi}_t)\rho_t\delta_t\boldsymbol{\phi}_t$.
  end for
end for

4.1 Convergence Analysis

We have discussed the motivations and ideas that led to the development of the generalized GEM(β) algorithm, which is used for the bias-variance trade-off in estimating $\boldsymbol{m}_\pi$. We now analyze the resulting GEM(β) algorithm theoretically.

Let $\boldsymbol{y}_t^\top \doteq [\boldsymbol{\kappa}_t^\top, \boldsymbol{w}_t^\top]$, we rewrite the GEM$(\beta)$ updates (Eqs. (10) and (11)) as

$$\boldsymbol{y}_{t+1} = \boldsymbol{y}_t + \zeta_t(\boldsymbol{g}_{t+1} - \mathbf{G}_{t+1}\boldsymbol{y}_t), \tag{13}$$

where

$$\mathbf{G}_{t+1} \doteq \begin{bmatrix} \boldsymbol{\phi}_{t+1}\boldsymbol{\phi}_{t+1}^\top & \boldsymbol{\phi}_{t+1}(\boldsymbol{\phi}_{t+1} - \beta\rho_t\boldsymbol{\phi}_t)^\top \\ -(\boldsymbol{\phi}_{t+1} - \beta\rho_t\boldsymbol{\phi}_t)\boldsymbol{\phi}_{t+1}^\top & 0 \end{bmatrix},$$

$$\boldsymbol{g}_{t+1} \doteq \begin{bmatrix} i_{t+1}\boldsymbol{\phi}_{t+1} \\ 0 \end{bmatrix}.$$

Then the limiting behavior of GEM(β) is governed by

$$\mathbf{G} \doteq \mathbb{E}[\mathbf{G}_t] = \begin{bmatrix} \mathbf{C} & \bar{\mathbf{A}} \\ -\bar{\mathbf{A}}^\top & 0 \end{bmatrix}, \boldsymbol{g} \doteq \mathbb{E}[\boldsymbol{g}_t] = \begin{bmatrix} \boldsymbol{\Phi}^\top \mathbf{D}\boldsymbol{i} \\ 0 \end{bmatrix}.$$

Theorem 1 (Convergence of GEM(β)) Under Assumptions (3-6), we have $\lim_{t\to\infty} \boldsymbol{y}_t = \mathbf{G}^{-1}\boldsymbol{g}$ almost surely.

Proof The main step of the proof is to show that $\mathbf{G}$ is strictly positive definite. We provide a detailed proof of Theorem 1 in Appendix A.2, which is inspired by Sutton et al. [30, 31] and Maei [32]. □

4.2 Asymptotic Stability Analysis

We now show that GEM-ETD(β) achieves asymptotic stability as defined in Sutton et al. [1]. Let $\mathbf{D}_{\boldsymbol{m}_{\boldsymbol{w}}} \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$ be the diagonal matrix with diagonal entires $[\mathbf{D}_{\boldsymbol{m}_{\boldsymbol{w}}}]_{ss} \doteq d_\mu(s) m_{\boldsymbol{w}}(s)$ for any state s. In GEM(β), we approximate $\lim_{t\to\infty} \mathbb{E}_\mu[M_t|S_t = s]$ with $m_{\boldsymbol{w}}(s)$. In this, we also define $\mathbf{D}_{\bar{\boldsymbol{m}}} \doteq diag(d_\mu(s) \lim_{t\to\infty} \mathbb{E}_\mu[M_t|S_t = s]) \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$, and their difference, $\mathbf{D}_\epsilon \doteq \mathbf{D}_{\boldsymbol{m}_{\boldsymbol{w}}} - \mathbf{D}_{\bar{\boldsymbol{m}}}$. The GEM-ETD($\beta$) update (12) can be rewritten to make the stability issues more transparent:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha_t(\mathbf{b}_t - \mathbf{A}_t\boldsymbol{\theta}_t), \tag{14}$$

where $\mathbf{A}_t \doteq \boldsymbol{m}_{\boldsymbol{w}_t}\rho_t\boldsymbol{\phi}_t(\boldsymbol{\phi}_t - \gamma\boldsymbol{\phi}_{t+1})^\top$, and $\mathbf{b}_t \doteq \boldsymbol{m}_{\boldsymbol{w}_t}\rho_t R_{t+1}\boldsymbol{\phi}_t$. It can be computed that

$$\mathbf{A} \doteq \lim_{t\to\infty} \mathbb{E}_\mu[\mathbf{A}_t] = \boldsymbol{\Phi}^\top \mathbf{D}_{\boldsymbol{m}_{\boldsymbol{w}}}(\mathbf{I} - \gamma\mathbf{P}_\pi)\boldsymbol{\Phi}.$$

According to Sutton et al. [1], we now establish stability by proving that $\mathbf{A}$ is positive definite.

Lemma 1 (Stability) Under Assumptions 2 & 3, there exists a positive constant $\vartheta > 0$ such that if $\|\mathbf{D}_\epsilon\| < \vartheta$, then $\mathbf{A}$ is positive definite.

Proof Details are provided in Appendix A.3. □

4.3 Fixed Points Analysis

In addition to the stability analysis, there is the question of the quality of the approximation at the fixed point. We now analyze the fixed point of GEM-ETD(β).

We denote by $\Pi_{\mathbf{D}}$ a projection operator weighted by the diagonal matrix $\mathbf{D}$. First we consider the fixed point of GEM(β). The fixed point of GEM(β) is a solution of the following projected equation, i.e., $\boldsymbol{\Phi}\boldsymbol{w}^* = \Pi_{\mathbf{D}}(\boldsymbol{i} + \beta\mathbf{D}^{-1}\mathbf{P}_\pi^\top\mathbf{D}\boldsymbol{\Phi}\boldsymbol{w}^*)$. Rearranging the terms yields the analytical solution of GEM(β):

$$\boldsymbol{w}^* = \left(\boldsymbol{\Phi}^\top\mathbf{D}(\boldsymbol{\Phi} - \beta\mathbf{D}^{-1}\mathbf{P}_\pi^\top\mathbf{D}\boldsymbol{\Phi})\right)^{-1}\boldsymbol{\Phi}^\top\mathbf{D}\boldsymbol{i}.$$

We define $\boldsymbol{m}_{\boldsymbol{w}}^* \doteq \boldsymbol{\Phi}\boldsymbol{w}^*$. Let $\mathbf{D}_{\boldsymbol{m}_{\boldsymbol{w}}^*} \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$ be the diagonal matrix with diagonal entires $\left[\mathbf{D}_{\boldsymbol{m}_{\boldsymbol{w}}^*}\right]_{ss} \doteq d_\mu(s) m_{\boldsymbol{w}}^*(s)$ for any state s. Then the fixed point of GEM-ETD(β) is a solution of the *Bellman operator* followed by a projection, i.e., $\boldsymbol{\Phi}\boldsymbol{\theta}^* = \Pi_{\mathbf{D}_{\boldsymbol{m}_{\boldsymbol{w}}^*}}(\boldsymbol{R} + \gamma\mathbf{P}_\pi\boldsymbol{\Phi}\boldsymbol{\theta}^*)$. Rearranging the terms yields the analytical solution of GEM-ETD(β):

$$\boldsymbol{\theta}^* = \left(\boldsymbol{\Phi}^\top\mathbf{D}_{\boldsymbol{m}_{\boldsymbol{w}}^*}(\mathbf{I} - \gamma\mathbf{P}_\pi)\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^\top\mathbf{D}_{\boldsymbol{m}_{\boldsymbol{w}}^*}\boldsymbol{R}. \tag{15}$$

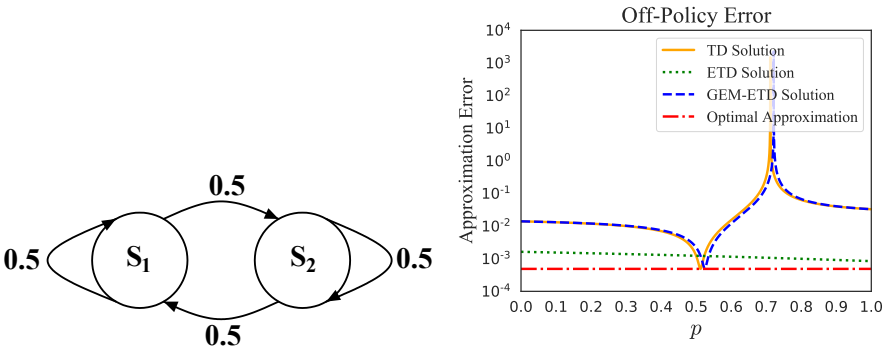


Fig. 1 Panel a): Two-state Markov chain presented by Kolter [52] for the off-policy counterexample. Panel b): The mean squared error for different off-policy distributions, along with the error of the optimal approximation.

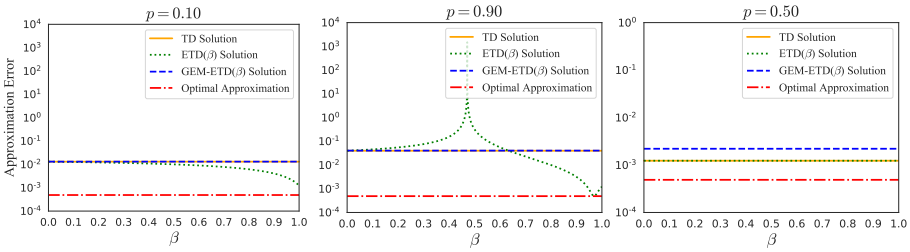


Fig. 2 The mean squared error by varying the decay rate β under different off-policy distributions.

Numerical Illustration: Here we introduce a concrete numerical example to illustrate the fixed point of GEM-ETD(β). Consider the two-state Markov chain shown in Fig. 1(a), which is an off-policy counterexample presented by Kolter [52] to show the bias bound of the off-policy TD, and used by Hallak et al. [26] to demonstrate the bias bound of the ETD. In this MDP model, the transition probability matrix $\mathbf{P}_\pi = (1/2)\mathbf{1}$, the discount factor $\gamma = 0.99$, and the true value function $\boldsymbol{v}_\pi = [1 \;\; 1.05]^\top$. The features are $\boldsymbol{\Phi} = [1 \;\; 1.05 + \varepsilon]^\top$, where $\varepsilon \in \mathbb{R}$. Clearly, in this example, $\boldsymbol{d}_\pi = [0.5 \;\; 0.5]^\top$. We use $\boldsymbol{d}_\mu = [p \;\; 1-p]^\top$ for off-policy learning, where $p \in [0, 1]$. The interest function is always 1.

We benchmark the vanilla TD, the vanilla ETD(β), and GEM-ETD(β) in this off-policy counterexample. Fig. 1(b) shows a plot of the mean squared error $\|\boldsymbol{\Phi}\boldsymbol{\theta}^* - \boldsymbol{v}_\pi\|_{\mathbf{D}_\pi}$ for the example above with $\beta = \gamma = 0.99$ and $\varepsilon = 0.001$, varying p from 0 to 1, where $\boldsymbol{\theta}^*$ is obtained from different equations for various methods. This becomes most clear when juxtaposing the equations

$$\boldsymbol{\theta}^* = \left(\boldsymbol{\Phi}^\top \mathbf{D}(\boldsymbol{\Phi} - \gamma \mathbf{P}_\pi \boldsymbol{\Phi})\right)^{-1} \boldsymbol{\Phi}^\top \mathbf{D} \boldsymbol{R}, \tag{TD}$$

$$\boldsymbol{\theta}^* = \left(\boldsymbol{\Phi}^\top \mathbf{D}_{\bar{\boldsymbol{m}}}(\boldsymbol{\Phi} - \gamma \mathbf{P}_\pi \boldsymbol{\Phi})\right)^{-1} \boldsymbol{\Phi}^\top \mathbf{D}_{\bar{\boldsymbol{m}}} \boldsymbol{R}, \tag{ETD(β))}$$

$$\boldsymbol{\theta}^* = \left(\boldsymbol{\Phi}^\top \mathbf{D}_{\boldsymbol{m}^*_{\boldsymbol{w}}}(\boldsymbol{\Phi} - \gamma \mathbf{P}_\pi \boldsymbol{\Phi})\right)^{-1} \boldsymbol{\Phi}^\top \mathbf{D}_{\boldsymbol{m}^*_{\boldsymbol{w}}} \boldsymbol{R}. \tag{GEM-ETD(β))}$$

Fig. 1(b) also shows the optimal error $\|\Pi_{\mathbf{D}_\pi} \boldsymbol{v}_\pi - \boldsymbol{v}_\pi\|_{\mathbf{D}_\pi}$. We can find that for certain behavior distributions ($p \approx 0.715$), the bias of TD approaches infinity (see Sutton et al. [1] for an extensive discussion and analysis of this essential problem), which matches previous results on off-policy TD in Kolter [52]. Again, for all behavior distributions, the bias of ETD is bounded and the bias bound decreases with the growth of p, which is consistent with the results on ETD in Hallak et al. [26]. Finally, GEM-ETD performs comparably to TD in this case ($\beta = \gamma$), and the bias would be unbounded for $p \approx 0.721$.

We now present the mean squared error results by varying β from 0 to 1, and consider three different behavior distributions: $p = 0.1$, $p = 0.5$, and $p = 0.9$. ε is still set to 0.001. As shown in Fig. 2, the main points to note are: (1) In the case of off-policy ($p = 0.1$ or $p = 0.9$), the bias of ETD(β) approaches infinity at $p = 0.9$, $\beta \approx 0.472$. The bias bound of GEM-ETD(β) does not change across all β values for each p, indicating that it is stably independent of β under extreme off-policy learning. (2) In particular, under on-policy learning ($p = 0.5$), the bounds of vanilla TD and ETD(β) coincide, which are higher than the optimal error and lower than the bias bound of GEM-ETD(β).

Interpretation 1 Due to stochastic function approximation, the objective of GEM-ETD(β) replaces the true emphasis $\boldsymbol{m}_\pi$ with an estimate $\boldsymbol{m}_{\boldsymbol{w}}$. Consequently, the optimal solution under this objective depends on the features used to approximate the emphasis, as confirmed by the behavior observed in the following experiments (see Sec. 5.1.2). We remark that the bias bound of GEM-ETD(β) depends on the features constructed.

4.4 Extension to Actor-Critic

The fact that actor-critic agents are more susceptible to off-policy learning than value-based agents is one of the main reasons we choose to concentrate on emphatic actor-critics in this paper. We can extend GEM(β) to a new actor-critic algorithm by simply applying the learned emphasis to the policy gradient, following existing work on ACE and COF-PAC. We name it COF-PAC(β) because it builds on COF-PAC. For COF-PAC(β), we use neural networks to parameterize $\boldsymbol{v}_{\boldsymbol{\theta}}$, $\boldsymbol{m}_{\boldsymbol{w}}$, and $\boldsymbol{\pi}_{\boldsymbol{\varpi}}$. The joint loss function of critic and emphasis for COF-PAC(β) is

$$\left(R_{t+1} + \gamma v(S_{t+1};\ \bar{\boldsymbol{\theta}}_t) - v(S_t;\ \boldsymbol{\theta}_t)\right)^2 + \left(i_{t+1} + \beta \rho_t m(S_t;\ \bar{\boldsymbol{w}}_t) - m(S_{t+1};\ \boldsymbol{w}_t)\right)^2, \tag{16}$$

where $\bar{\boldsymbol{w}}$ and $\bar{\boldsymbol{\theta}}$ indicate the parameters of the target network for $\boldsymbol{v}_{\boldsymbol{\theta}}$ and $\boldsymbol{m}_{\boldsymbol{w}}$, respectively. Then following the derivation of policy gradient in Zhang et al. [2], the actor of COF-PAC(β) takes the following update rules:

$$\boldsymbol{\varpi}_{t+1} = \boldsymbol{\varpi}_t + \alpha \rho_t m_{\boldsymbol{w}_t} q_\pi(S_t, A_t) \nabla \log \pi(A_t|S_t). \tag{17}$$

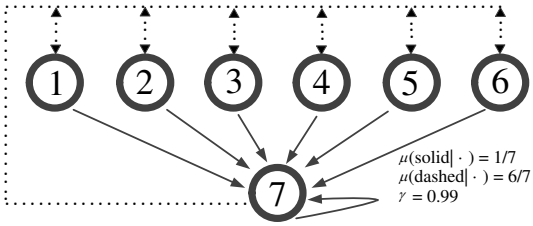


Fig. 3 Baird's Counterexample from Chapter 11.2 of Sutton and Barto [3]. There are two actions available at each state, dashed and solid. The solid action always results in the state 7 and a reward 0, and the dashed action results in one of states 1-6 with equal probability and a reward 1. The initial state is sampled randomly from all seven states.

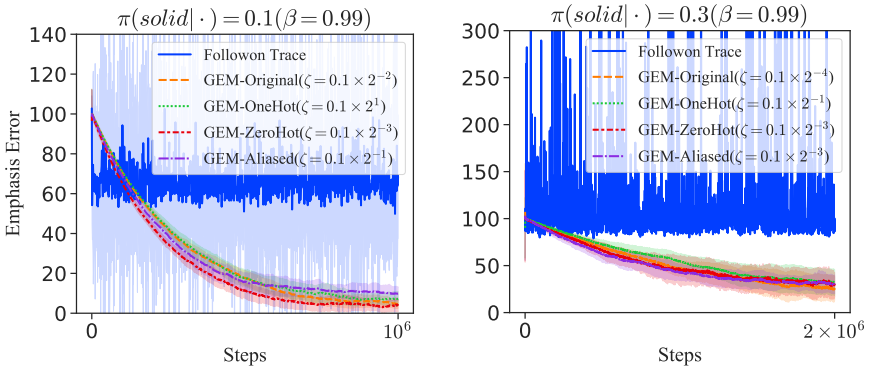


Fig. 4 Averaged emphasis approximation error for the vanilla GEM and the *followon trace* with different features. Learning rates used are bracketed.

The detailed algorithm implementation and experimental settings are described in Section 5.2.

5 Experiments

We design experiments aiming to answer the following questions: 1) Can GEM(β) approximate the true emphasis as promised? 2) Can we flexibly manage the bias and variance of GEM(β) and its variant GEM-ETD(β) by providing more freedom in the choice of β? 3) Can GEM(β) scale up to classic control tasks from OpenAI Gym [53] with neural network function approximators?

5.1 Diagnostic Experiments

In this section, we first establish the performance of GEM(β) and its variant GEM-ETD(β) on a diagnostic MDP modified from Baird's counterexample [29]. This MDP model is a well-known star counterexample for evaluating the performance of off-policy convergent algorithms and has been used to demonstrate the soundness of emphasis-based TD learning methods under off-policy

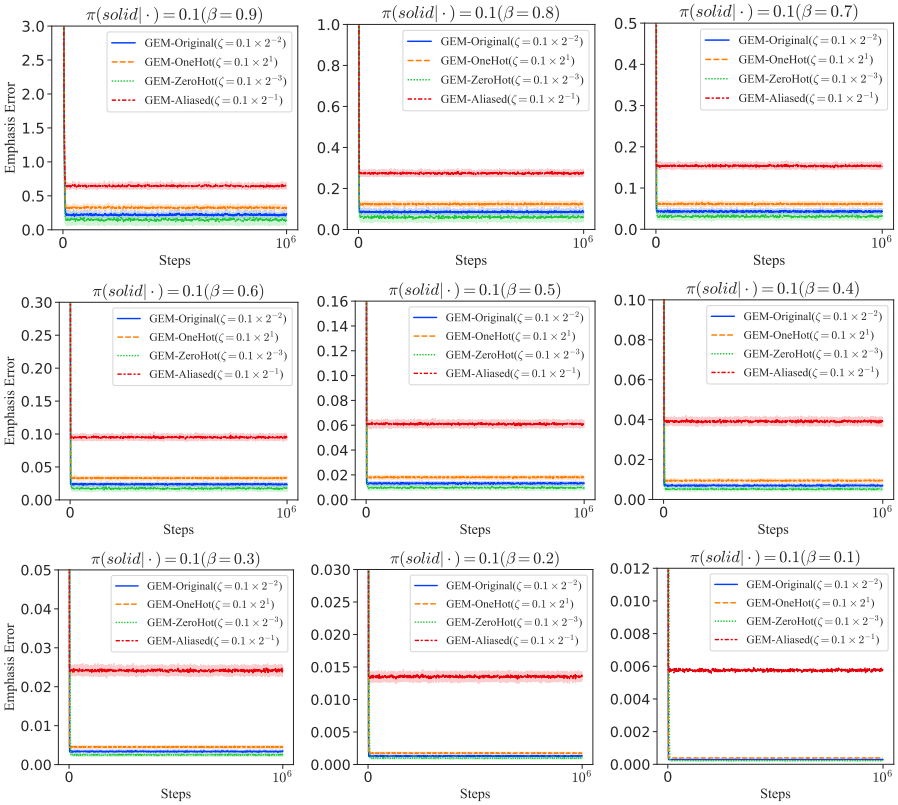


Fig. 5 Averaged emphasis approximation error for GEM(β) by varying β with different features when $\pi(solid|\cdot) = 0.1$. Learning rates used are bracketed.

learning [2, 3, 13, 14, 15], so provides a convincing testbed that can verify the performance of our algorithms. As shown in Fig. 3, there are seven states with linear features and two actions. The behavior policy μ always takes the solid action with probability $\frac{1}{7}$. Our goal is to observe how qualitative properties such as convergence, approximation error, learning speed, and variance manifest in practice, and to illustrate the theoretical results previously obtained. For all states, the interest is set to 1, and the discount factor $\gamma \doteq 0.99$.

In this problem, four different types of features are considered: original features, one-hot features, zero-hot features, and aliased features. Original features are the features used by Sutton and Barto [3], where each state's features lie in $\mathbb{R}^8$. One-hot features use one-hot encoding, where each feature lies in $\mathbb{R}^7$, and zero-hot features are the complements of one-hot features. In particular, aliased features are the features lying in $\mathbb{R}^6$, where the quantities of interest may not lie in the feature space, resulting in state aliasing (details are described in Appendix A.4).

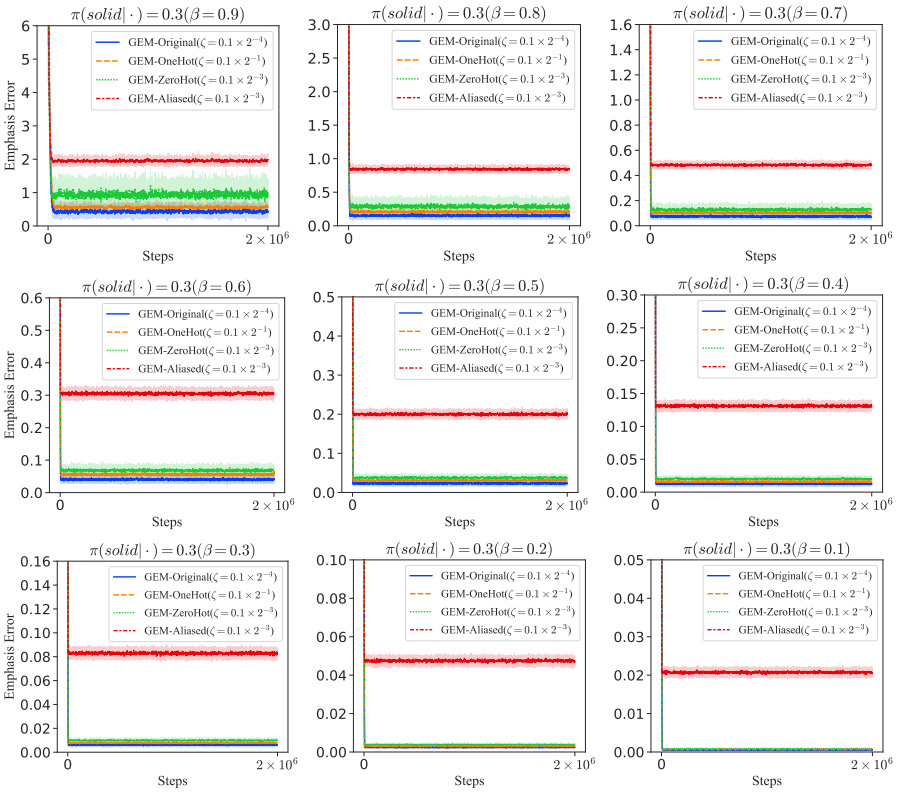


Fig. 6 Averaged emphasis approximation error for GEM(β) by varying β with different features when $\pi(solid|\cdot) = 0.3$. Learning rates used are bracketed.

In addition to reporting the performance of the best learning curves, we extensively investigate the sensitivity to the hyperparameter β. So we extensively sweep the values of β by varying it in $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.99\}$. To create the best learning curve, a large number of combinations of different hyperparameters are used to run each algorithm, and the results are plotted using the best hyperparameters that minimized the Area Under the learning Curve (AUC) for all algorithms in the solid lines. The average is taken over 30 independent runs for all curves, with the shaded regions indicating standard errors.

5.1.1 Emphasis Approximation

We first compare the performance of approximating the true emphasis $\boldsymbol{m}_\pi$ with GEM(β) (Eqs. (10) and (11)) and the vanilla GEM (Eqs. (5) and (6)). Additionally, we also benchmark the *followon trace*. The emphasis approximation error for the *followon trace* and GEM(β) at time step t is computed as $|M_t - m_\pi(S_t)|$ and $|m_{\boldsymbol{w}}(S_t) - m_\pi(S_t)|$ respectively, where $\boldsymbol{m}_\pi$ is computed

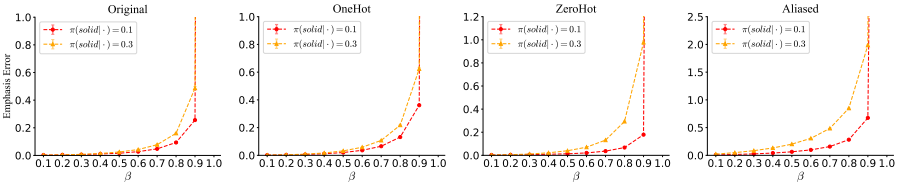


Fig. 7 Sensitivity to the decay rate β for emphasis approximation. For each specific value of β, All methods choose the fixed value of ζ under different features. Values swept are $\beta \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.99\}$. Note that for $\beta = \gamma = 0.99$, we have the vanilla GEM.

Table 1 Average area values obtained under the emphasis error learning curve for each case corresponding to Fig. 7.

Cases	Original		OneHot		ZeroHot		Aliased	
	$\pi(solid\|\cdot) = 0.1$	$\pi(solid\|\cdot) = 0.3$	$\pi(solid\|\cdot) = 0.1$	$\pi(solid\|\cdot) = 0.3$	$\pi(solid\|\cdot) = 0.1$	$\pi(solid\|\cdot) = 0.3$	$\pi(solid\|\cdot) = 0.1$	$\pi(solid\|\cdot) = 0.3$
$\beta = 0.99$	28.106±1.460	52.774±5.696	29.165±1.784	57.418±5.943	23.507±1.689	52.416±6.205	28.490±2.167	50.617±3.308
$\beta = 0.90$	0.256±0.004	0.487±0.012	0.361±0.002	0.623±0.008	0.179±0.007	0.978±0.013	0.675±0.004	1.985±0.008
$\beta = 0.80$	0.094±0.003	0.160±0.003	0.132±0.001	0.218±0.002	0.067±0.003	0.293±0.004	0.282±0.002	0.849±0.002
$\beta = 0.70$	0.048±0.002	0.078±0.002	0.066±0.001	0.108±0.001	0.036±0.002	0.132±0.001	0.158±0.002	0.486±0.002
$\beta = 0.60$	0.027±0.002	0.042±0.001	0.036±0.001	0.058±0.001	0.021±0.002	0.071±0.001	0.098±0.002	0.307±0.001
$\beta = 0.50$	0.016±0.001	0.025±0.001	0.020±0.001	0.032±0.001	0.013±0.001	0.040±0.001	0.064±0.002	0.202±0.001
$\beta = 0.40$	0.011±0.001	0.014±0.001	0.011±0.001	0.017±0.001	0.009±0.001	0.022±0.001	0.041±0.001	0.133±0.001
$\beta = 0.30$	0.006±0.001	0.008±0.001	0.006±0.001	0.009±0.001	0.006±0.001	0.011±0.001	0.026±0.001	0.084±0.001
$\beta = 0.20$	0.004±0.001	0.004±0.001	0.003±0.001	0.004±0.001	0.004±0.001	0.006±0.001	0.016±0.001	0.049±0.001
$\beta = 0.10$	0.002±0.001	0.002±0.001	0.002±0.001	0.001±0.001	0.003±0.001	0.002±0.001	0.007±0.001	0.022±0.001

analytically. For GEM(β), we consider a fixed learning rate ζ and tune it from $\{0.1 \times 2^{-6}, ..., 0.1 \times 2^{2}\}$. Note that for $\beta = \gamma = 0.99$, we have the vanilla GEM.

Here we consider two target policies: $\pi(solid|\cdot) = 0.1$ and $\pi(solid|\cdot) = 0.3$. For the tuning of hyperparameters, we first tune the hyperparameters of the vanilla GEM, and GEM(β) inherits the common hyperparameters of GEM to investigate whether the tunable decay rate β plays a role in the performance improvement of the algorithm. Fig. 4 presents the results for the *followon trace* and the vanilla GEM. The *followon trace* exhibits large approximation error and variance, and the vanilla GEM by comparison shows to be an effective way to mitigate these two problems from the *followon trace* to some extent, which matches previous experiments reported by Zhang et al. [2]. When we reduce the value of β, as shown in Figs. 5 and 6, GEM(β) with a lower value of β achieves lower variance and approximation error.

To better facilitate comparison and analysis, we also present a line graph in Fig. 7 and a table of values in Table 1 that includes all the AUC values shown in Fig. 7. The standard error of each entry in the table are statistically reported as well, although some of them are invisible due to being small. As we can see in Fig. 7 and the corresponding Table 1, we in fact find that compared with the vanilla GEM, GEM(β) is indeed a better way to improve performance in approximating the true emphasis.

5.1.2 Overall Performance

Our ultimate goal is to apply these approximated emphasis derived to the value estimate gradient, and obtain the corresponding ETD variants. We now investigate and compare the performance of these ETD variants for OPE.

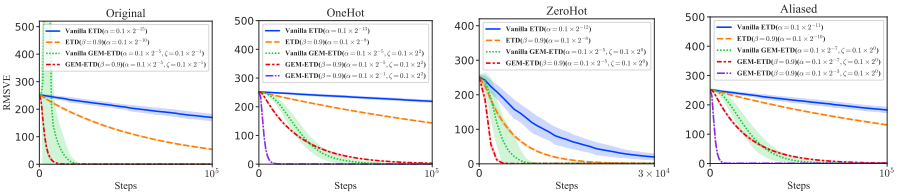


Fig. 8 The comparison of the best learning curve under different features using the AUC criterion. Learning rates used are bracketed.

We extreme the difference between the behavior policy and the target policy and consider a target policy $\pi(solid|\cdot) = 0.05$. We train each algorithm with every different hyperparameters combination for 10^5 steps and evaluate the Root Mean Square Value Error (RMSVE) at each time step during training, computed as RMSVE $= \|\boldsymbol{v_\theta} - \boldsymbol{v}_\pi\|_{\mathbf{D}}$, where $\boldsymbol{v_\theta}$ is the estimated value function and $\boldsymbol{v}_\pi$ is computed analytically. We use the same architecture and sweep range of stepsize parameter α as Zhang et al. [2] and tune α from $\{0.1\times2^{-19}, ..., 0.1\times 2^0\}$. Note that for $\beta = \gamma = 0.99$, we have the vanilla versions of ETD and GEM-ETD.

We first tune hyperparameters for the vanilla versions of ETD, ETD(β), and GEM-ETD. GEM-ETD(β) inherits the common hyperparameters from the vanilla GEM-ETD, and then we tune the hyperparameter β. The results with hyperparameters combinations that minimized the AUC in the solid lines are reported in Fig. 8. The main points to note are: (1) Under all four sets of features, ETD($\beta = 0.9$) enjoys a lower variance than the vanilla ETD—consistent with previous results on ETD(β) in Hallak et al. [26]. (2) The stable convergence of GEM-ETD(β) occurs, illustrating Lemma 1. Further, GEM-ETD($\beta = 0.9$) has a clear win over the vanilla GEM-ETD. GEM-ETD($\beta = 0.9$) outperforms the vanilla GEM-ETD in terms of learning speed and variance under original and zero-hot features. (3) Under one-hot and aliased features, although GEM-ETD($\beta = 0.9$) learns slower than the vanilla GEM-ETD, it exhibits lower variance, thus, a higher value for α can be set to accelerate convergence. We also show the results for the higher value of α ($\alpha = 0.1 \times 2^{-1}$ for the one-hot features, and $\alpha = 0.1 \times 2^{-3}$ for aliased features), the results show that when we increase the stepsize parameter α, GEM-ETD($\beta = 0.9$) converges faster, and the variance is still smaller than that of the vanilla GEM-ETD. Overall, we conclude by highlighting that compared to vanilla GEM-ETD, GEM-ETD(β) does provide a reasonable strategy to obtain substantial improvements by controlling the decay rate β.

5.1.3 Sensitivity to β for OPE:

Until now, we have only reported the results using the best performing stepsizes. In this section, we extensively investigate the sensitivity of GEM-ETD(β) to the decay rate β for OPE. For the stepsize parameter α and the emphasis learning rate ζ, GEM-ETD(β) inherits the best performing values of the

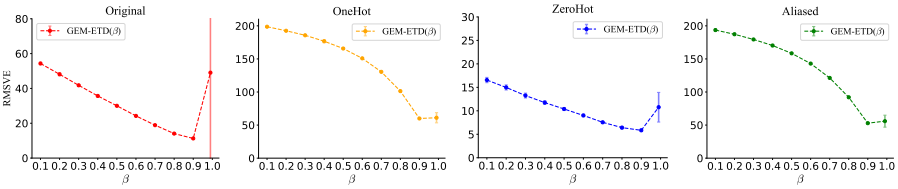


Fig. 9 Sensitivity to the decay rate β for OPE. For each specific value of β, All methods choose the fixed values of α and ζ under different features. Values swept are $\beta \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.99\}$. Note that for $\beta = \gamma = 0.99$, we have the vanilla versions of ETD and GEM-ETD.

Table 2 The lowest AUC values for all algorithms corresponding to Fig. 9. The lowest averaged area values obtained after comparison under the RMSVE learning curve are highlighted in bold.

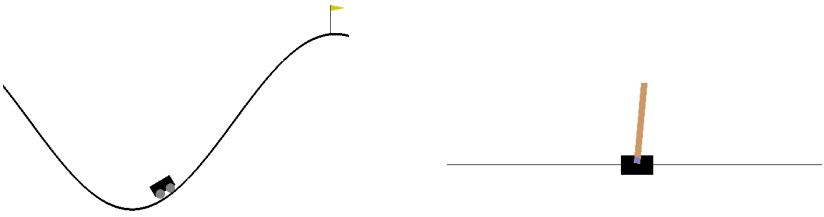
Cases	Original	OneHot	ZeroHot	Aliased
$\beta = 0.99$	49.054±99.641	61.102±7.716	10.767±3.157	55.916±9.050
$\beta = 0.90$	**8.273±0.471**	**59.976±1.209**	**5.045±0.258**	**52.873±1.307**
$\beta = 0.80$	14.071±0.368	101.342±0.601	6.417±0.270	92.019±0.939
$\beta = 0.70$	18.894±0.312	130.466±0.613	7.550±0.216	121.275±0.920
$\beta = 0.60$	24.222±0.373	150.876±0.687	9.012±0.263	142.966±1.105
$\beta = 0.50$	30.023±0.426	165.725±0.598	10.385±0.322	158.434±1.216
$\beta = 0.40$	35.686±0.561	176.723±2.292	11.750±0.380	170.273±1.699
$\beta = 0.30$	41.791±0.506	185.734±0.567	13.249±0.518	179.423±1.390
$\beta = 0.20$	48.120±0.827	192.558±0.631	14.977±0.540	187.263±1.742
$\beta = 0.10$	54.329±0.699	198.672±0.868	16.555±0.564	193.603±1.594

vanilla GEM-ETD obtained above for a fair comparison. For the sake of completeness, Table 2 contains the lowest values of all cases shown in Fig. 9 as a table of values. The standard error of each entry in the table is also statistically reported. The bold entries highlight the lowest RMSVE for the given case. The results in Fig. 9 and the corresponding Table 2 show that GEM-ETD(β) with $\beta = 0.9$ performs the best under all four sets of features. When it comes to the bias-variance trade-off, the optimal choice is usually problem-dependent. In conclusion, our experiments results demonstrate that our GEM-ETD(β) is indeed a promising approach with a broader applicability to different problem situations.

5.2 Classic Control Tasks

To answer the posed question 3) at the beginning of Sec. 5, we benchmark ACE, COF-PAC, and COF-PAC(β) on classic control tasks from OpenAI Gym [53], which are illustrated in Fig. 10. Additionally, we also benchmark ACE(β), where we introduce the tunable decay rate β to the *followon trace* in ACE. The detailed descriptions of the classic control tasks are shown in Table 3.

All curves are averaged over 10 independent runs and shaded regions indicate standard errors. For all non-termination transitions, we set the discount



(a) MountainCarContinuous-v0 (b) CartPole-v1

Fig. 10 Screenshots for the classic control tasks from OpenAI Gym used in our environments

Table 3 Descriptions of OpenAI Gym classic control tasks used in our experiments

Task	Number of states	Number of actions	Control type	Training goal
MountainCarContinuous-v0	2	1	Continuous	Drive up a big hill
CartPole-v1	4	2	Discrete	Balance a pole on a cart

Table 4 Hyperparameters table

Parameter	Value
Gradient norm clip	0.5
Discount factor	0.99
Replay buffer size	10^6
Batch size of the replay buffer	10
Importance sampling ratio clip	[0, 2]
Decay rate β	(0, 1]
Warm-up steps before learning	100 environment steps
Target network update frequency	200 optimization steps
Optimizer	RMSProp with an initial learning rate 10^{-3}

factor γ to 0.99, and for all termination transitions, we set it to 0. Upon termination, the agent was transferred back to the initial state. The interest function was always 1. The behavior policy μ is a uniformly random policy, and the target policy π is a Gaussian policy. We evaluated $J(\pi)$ every 10^3 steps. To evaluate $J(\pi)$, we first sample a state from $\boldsymbol{d}_\mu$ and then follow π until episode termination, which we call an excursion. We use the averaged return of 10 excursions as an estimate of $J(\pi)$. For $\boldsymbol{q}_\pi$, we follow the existing work of Imani et al. [25] and Zhang et al. [2], and replace $\boldsymbol{q}_\pi$ with the TD error value δ_t. Further, inspired by the success of semi-gradient methods in large-scale RL [11], a semi-gradient version of GEM is utilized to train $\boldsymbol{m}_\pi$.

We conducted our experiments on a server with 56 Intel® Xeon® E5-2680 v4 CPUs and two Nvidia Tesla P40 GPUs. Our implementation is based on PyTorch. All algorithms share the same architecture and common parameterization. For ACE, ACE(β), COF-PAC, and COF-PAC(β), we use separate two-hidden-layer neural networks to parameterize $\boldsymbol{v}_{\boldsymbol{\theta}}$, $\boldsymbol{m}_{\boldsymbol{w}}$, and $\boldsymbol{\pi}_{\boldsymbol{\varpi}}$. Each hidden layer has 64 hidden units and a ReLU activation function. Particularly, we parameterized $\boldsymbol{\pi}_{\boldsymbol{\varpi}}$ as a diagonal Gaussian distribution with the mean being

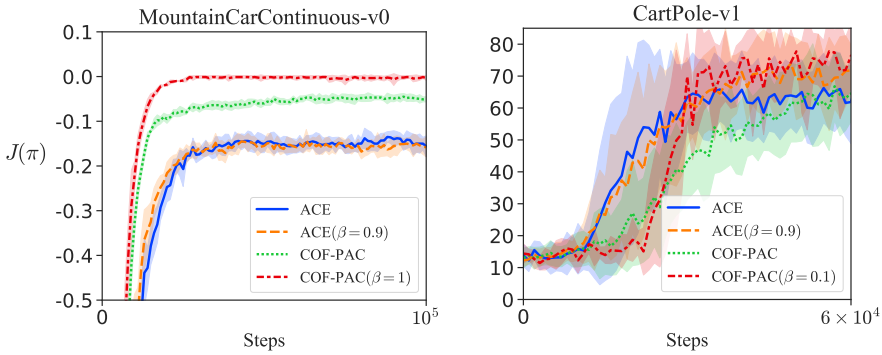


Fig. 11 Comparison among ACE, ACE(β), COF-PAC, and COF-PAC(β) on MountainCarContinuous and CartPole classic control tasks.

the output of the network. The COF-PAC(β) implementation is based on the COF-PAC implementation of Zhang et al. [2]. In order to reduce instability, we also clip the IS ratios [12]. Table 4 lists all hyperparameters used by COF-PAC(β), most of which follow the hyperparameters reported in Zhang et al. [2].

The results are reported in Fig. 11. In MountainCarContinuous, the performance of ACE and ACE(β) is similar. COF-PAC outperforms ACE and ACE(β), and COF-PAC(β) performs the best for the task with $\beta = 1$. In CartPole, ACE performs better than COF-PAC. By adjusting β, the performance of both ACE(β) and COF-PAC(β) is improved. In particular, although ACE(β) reaches the similar final performance as COF-PAC(β), COF-PAC(β) achieves more stable and fast convergence. To summarize, these experimental results support our claim that our GEM(β)-learned emphasis can indeed lead to performance improvements on classic control tasks at scale.

6 Conclusion

In this paper, inspired by Hallak et al. [26], we developed a novel generalized GEM(β) algorithm for OPE and off-policy control. We have empirically demonstrated that a reasonable tuning of the decay rate β can indeed lead to significantly better performance. We also showcased the merits of the extended off-policy control algorithm COF-PAC(β) over existing emphatic actor-critic algorithms on classic control tasks from OpenAI Gym [53]. We analyzed our algorithms theoretically and empirically to increase the understanding of the concept. We conducted our empirical study on small-scale diagnostic benchmarks as a proof of concept. Further, we investigated the COF-PAC(β) algorithm in large-scale experiments to highlight both the scalability and broad applicability of GEM(β). As we know, GEM(β) uses a GTD2-style update, which relies so heavily on $\boldsymbol{\kappa}$ for learning $\boldsymbol{w}$: $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \alpha_t[(\boldsymbol{\phi}_{t+1} - \beta\rho_t\boldsymbol{\phi}_t)\boldsymbol{\kappa}_t^\top\boldsymbol{\phi}_{t+1}]$. In the beginning, when $\boldsymbol{\kappa}$ is inaccurate, the updates for $\boldsymbol{w}$ are poor. The TD with gradient correction (TDC) algorithm [31]

has been verified to demonstrate superior performance in the family of gradient TD algorithms, as reaffirmed by extensive experiments on GTD methods in Ghiassian et al. [50]. Thus, a possibility for further work is to apply the update rules of TDC to learn the emphasis for better performance. Finally, we anticipate that our GEM(β) algorithm can be extended to multi-agent RL optimal control algorithms to solve the Nash equilibrium problem.

Acknowledgments. We would like to thank Fei Zhu, Leilei Yan, and Xiaohan Zheng for their technical support. We would also like to thank the computer resources and other support provided by the Machine Learning and Image Processing Research Center of Soochow University.

Declarations

- Funding: This work is supported by the National Natural Science Foundation of China (Nos. 61772355, 61702055, 61876217, 62176175), Jiangsu Province Natural Science Research University major projects (18KJA520011, 17KJA520004), Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University (93K172014K04, 93K172017K18, 93K172021K08), Suzhou Industrial application of basic research program part (SYG201422), the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD).
- Competing interests: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.
- Ethics approval: Not applicable.
- Consent to participate: Not applicable.
- Consent for publication: Not applicable.
- Availability of data and materials: The datasets generated during and analysed during the current study are available from the corresponding author on reasonable request.
- Code availability: The code is publicly available at https://github.com/Caojiaqing0526/DeepRL.
- Authors' contributions: **Jiaqing Cao:** Conceptualization, Methodology, Software, Validation, Writing-original draft, Writing-review&editing. **Quan Liu:** Conceptualization, Resources, Writing-review&editing, Validation, Project administration, Funding acquisition, Supervision. **Lan Wu:** Investigation, Software, Visualization. **Qiming Fu:** Investigation, Software, Visualization. **Shan Zhong:** Investigation, Software, Visualization. All authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Appendix A

A.1 Proof of Proposition 1

Proof From the definitions of $\hat{\mathcal{T}}_\beta$ and $\boldsymbol{m}_\pi$, we have

$$\begin{aligned}\hat{\mathcal{T}}_\beta \boldsymbol{m}_\pi &= \boldsymbol{i} + \beta \mathbf{D}^{-1}\mathbf{P}_\pi^\top \mathbf{D}\boldsymbol{m}_\pi \\ &= \boldsymbol{i} + \beta \mathbf{D}^{-1}\mathbf{P}_\pi^\top (\mathbf{I} - \beta \mathbf{P}_\pi^\top)^{-1}\mathbf{D}\boldsymbol{i} \\ &= \left(\mathbf{D}^{-1}(\mathbf{I} - \beta \mathbf{P}_\pi^\top) + \beta \mathbf{D}^{-1}\mathbf{P}_\pi^\top\right)(\mathbf{I} - \beta \mathbf{P}_\pi^\top)^{-1}\mathbf{D}\boldsymbol{i} \\ &= \mathbf{D}^{-1}(\mathbf{I} - \beta \mathbf{P}_\pi^\top)^{-1}\mathbf{D}\boldsymbol{i} = \boldsymbol{m}_\pi.\end{aligned}$$

According to Theorem 1.3.22 in Horn and Johnson [54], given any two square matrices $\mathbf{A}$ and $\mathbf{B}$, the products $\mathbf{AB}$ and $\mathbf{BA}$ have the same eigenvalues. Thus, we have $\rho(\beta \mathbf{D}^{-1}\mathbf{P}_\pi^\top \mathbf{D}) = \rho\left((\beta \mathbf{P}_\pi^\top \mathbf{D})\mathbf{D}^{-1}\right) = \rho(\beta \mathbf{P}_\pi^\top) = \rho(\beta \mathbf{P}_\pi) < 1$, where $\rho(\cdot)$ is the spectral radius. Clearly, the matrix $\beta \mathbf{D}^{-1}\mathbf{P}_\pi^\top \mathbf{D}$ is nonnegative. Consequently, according to Corollary 6.1 in Bertsekas and Tsitsiklis [51], $\hat{\mathcal{T}}_\beta$ is a contraction mapping w.r.t some weighted maximum norm, which completes the proof.

□

A.2 Proof of Theorem 1

Proof This proof is inspired by Sutton et al. [30, 31] and Maei [32]. We provide the full proof here for completeness.

We first rewrite the iteration equation Eq. (13) in the following form:

$$\boldsymbol{y}_{t+1} = \boldsymbol{y}_t + \zeta_t(h(\boldsymbol{y}_t) + L_{t+1}),$$

where $h(\boldsymbol{y}) \doteq \mathbf{G}\boldsymbol{y} + \boldsymbol{g}$ and $L_{t+1} \doteq (\mathbf{G}_{t+1} - \mathbf{G})\boldsymbol{y}_t + (\boldsymbol{g}_{t+1} - \boldsymbol{g})$ is the noise sequence. Let $\Omega_t \doteq (\boldsymbol{y}_1, L_1, ..., \boldsymbol{y}_{t-1}, L_t)$ be σ-fields generated by the quantities $\boldsymbol{y}_i, L_i, i \leq k, k \geq 1$.

Now we apply the conclusions from Theorem 2.2 provided in Borkar and Meyn [55], i.e., the following preconditions must be satisfied: (i) The function $h(\boldsymbol{y})$ is Lipschitz, and there exists $h_\infty(\boldsymbol{y}) \doteq \lim_{c\to\infty} h(c\boldsymbol{y})/c$ for all $\boldsymbol{y} \in \mathbb{R}^{2n}$; (ii) The sequence (L_t, Ω_t) is a martingale difference sequence, and $\mathbb{E}[\|M_{t+1}\|^2|\Omega_t] \leq K(1+\|\boldsymbol{y}\|^2)$ holds for some constant $K > 0$ and any initial parameter vector $\boldsymbol{y}_1$; (iii) The nonnegative stepsize sequence a_t satisfies $\sum_t a_t = \infty$ and $\sum_t a_t^2 < +\infty$; (iv) The origin is a globally asymptotically stable equilibrium for the ordinary differential equation (ODE) $\dot{\boldsymbol{y}} = h_\infty(\boldsymbol{y})$; and (v) The ODE $\dot{\boldsymbol{y}} = h(\boldsymbol{y})$ has a unique globally asymptotically stable equilibrium.

First for condition (i), because $\left\|h(\boldsymbol{y}_i) - h(\boldsymbol{y}_j)\right\|^2 = \left\|\mathbf{G}(\boldsymbol{y}_i - \boldsymbol{y}_j)\right\|^2 \leq \mathbf{G}\left\|(\boldsymbol{y}_i - \boldsymbol{y}_j)\right\|^2$ for $\forall \boldsymbol{y}_i, \boldsymbol{y}_j$, therefore $h(\cdot)$ is Lipschitz. Meanwhile, $\lim_{c\to\infty} h(c\boldsymbol{y})/c = \lim_{c\to\infty}(c\mathbf{G}\boldsymbol{y} + \boldsymbol{g})/c = \lim_{c\to\infty} \boldsymbol{g}/c + \lim_{c\to\infty} \mathbf{G}\boldsymbol{y}$. Assumption 5 ensures that $\boldsymbol{g}$ is bounded. Thus, when $c \to \infty$, $\lim_{c\to\infty} \boldsymbol{g}/c = 0$, $\lim_{c\to\infty} h(c\boldsymbol{y})/c = \lim_{c\to\infty} \mathbf{G}\boldsymbol{y}$. Next, we establish that condition (ii) is true: because

$$\begin{aligned}\|L_{t+1}\|^2 &= \|(\mathbf{G}_t - \mathbf{G})\boldsymbol{y}_t + (\boldsymbol{g}_t - \boldsymbol{g})\|^2 \\ &\leq \|(\mathbf{G}_t - \mathbf{G})\|^2\|\boldsymbol{y}_t\|^2 + \|(\boldsymbol{g}_t - \boldsymbol{g})\|^2,\end{aligned}$$

let $K = \max\{\|(\mathbf{G}_t - \mathbf{G})\|^2, \|(\boldsymbol{g}_t - \boldsymbol{g})\|^2\}$, we have $\|L_{t+1}\|^2 \leq K(1 + \|\boldsymbol{y}_t\|^2)$. As a result, we see that condition (ii) is met. Further, condition (iii) is satisfied by

Assumption 6 in Theorem 1. Finally, for conditions (iv) and (v), we need to prove that the real parts of all the eigenvalues of $\mathbf{G}$ are negative.

We first show that $\mathbf{G}$ is nonsingular. Using the determinant rule for partitioned matrices, we have $\det(\mathbf{G}) = \det(\mathbf{C})\det(\bar{\mathbf{A}}^\top \mathbf{C}^{-1}\bar{\mathbf{A}}) = \det(\bar{\mathbf{A}}^\top \bar{\mathbf{A}}) = (\det \bar{\mathbf{A}})^2 \neq 0$. This indicates that all the eigenvalues of $\mathbf{G}$ are nonzero.

Now, let $\chi \in \mathbb{C}$, $\chi \neq 0$ be a nonzero eigenvalue of matrix $\mathbf{G}$ with normalized eigenvector $\mathbf{x} \in \mathbb{C}^{2n}$, i.e., $\mathbf{x}^*\mathbf{x} = 1$, where $\mathbf{x}^*$ is the complex conjugate of vector $\mathbf{x}$. Hence $\mathbf{x}^*\mathbf{G}\mathbf{x} = \chi$. Let $\mathbf{x}^\top \doteq [\mathbf{x}_1^\top, \mathbf{x}_2^\top]$, where $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{C}^n$. Clearly, we can obtain

$$\chi = \mathbf{x}^*\mathbf{G}\mathbf{x} = \mathbf{x}_1^*\mathbf{C}\mathbf{x}_1 + \mathbf{x}_1^*\bar{\mathbf{A}}\mathbf{x}_2 - \mathbf{x}_2^*\bar{\mathbf{A}}^\top \mathbf{x}_1,$$

because $\bar{\mathbf{A}}$ is real, $\bar{\mathbf{A}}^* = \bar{\mathbf{A}}^\top$. Consequently, $(\mathbf{x}_1^*\bar{\mathbf{A}}\mathbf{x}_2)^* = \mathbf{x}_2^*\bar{\mathbf{A}}^\top\mathbf{x}_1$, yielding $\mathrm{Re}(\mathbf{x}_1^*\bar{\mathbf{A}}\mathbf{x}_2 - \mathbf{x}_2^*\bar{\mathbf{A}}^\top\mathbf{x}_1) = 0$, where $\mathrm{Re}(\cdot)$ denotes the real-part of the eigenvalue. Finally, we have $\mathrm{Re}(\chi) = \mathrm{Re}(\mathbf{x}_1^*\mathbf{C}\mathbf{x}_1) = \|\mathbf{x}_1\|_{\mathbf{C}}^2 > 0$, which completes the proof.

□

A.3 Proof of Lemma 1

Proof As shown by Sutton et al. [1] and Hallak et al. [26], $\mathbf{D}_{\bar{\boldsymbol{m}}}(\mathbf{I} - \gamma\mathbf{P}_\pi)$ is positive definite, i.e., for any real vector $\mathbf{y}$, we have $g(\mathbf{y}) \doteq \mathbf{y}^\top \mathbf{D}_{\bar{\boldsymbol{m}}}(\mathbf{I} - \gamma\mathbf{P}_\pi)\mathbf{y} > 0$. Since $g(\mathbf{y})$ is a continuous function, it obtains its minimum value in the compact set $\mathcal{Y} \doteq \{\mathbf{y} : \|\mathbf{y}\| = 1\}$, i.e., there exists a positive constant $\vartheta_0 > 0$ such that $g(\mathbf{y}) \geq \vartheta_0 > 0$ holds for any $\mathbf{y} \in \mathcal{Y}$. In particular, for any $\mathbf{y} \in \mathbb{R}^{|\mathcal{S}|}$, we have $g(\frac{\mathbf{y}}{\|\mathbf{y}\|}) \geq \vartheta_0$, i.e., $\mathbf{y}^\top \mathbf{D}_{\bar{\boldsymbol{m}}}(\mathbf{I} - \gamma\mathbf{P}_\pi)\mathbf{y} \geq \vartheta_0\|\mathbf{y}\|^2$. Hence, we have

$$\begin{aligned}
&\mathbf{y}^\top \mathbf{D}_{\boldsymbol{m}_w}(\mathbf{I} - \gamma\mathbf{P}_\pi)\mathbf{y} \\
&= \mathbf{y}^\top \mathbf{D}_{\bar{\boldsymbol{m}}}(\mathbf{I} - \gamma\mathbf{P}_\pi)\mathbf{y} + \mathbf{y}^\top \mathbf{D}_\epsilon(\mathbf{I} - \gamma\mathbf{P}_\pi)\mathbf{y} \\
&\geq \vartheta_0\|\mathbf{y}\|^2 + \mathbf{y}^\top \mathbf{D}_\epsilon(\mathbf{I} - \gamma\mathbf{P}_\pi)\mathbf{y} \\
&\geq \vartheta_0\|\mathbf{y}\|^2 - |\mathbf{y}^\top \mathbf{D}_\epsilon(\mathbf{I} - \gamma\mathbf{P}_\pi)\mathbf{y}| \\
&\geq \vartheta_0\|\mathbf{y}\|^2 - \|\mathbf{y}\|^2\|\mathbf{D}_\epsilon\|\|\mathbf{I} - \gamma\mathbf{P}_\pi\| \\
&= (\frac{\vartheta_0}{\|\mathbf{I} - \gamma\mathbf{P}_\pi\|} - \|\mathbf{D}_\epsilon\|)\|\mathbf{I} - \gamma\mathbf{P}_\pi\|\|\mathbf{y}\|^2,
\end{aligned}$$

for any $\mathbf{y}$.

Let $\vartheta \doteq \frac{\vartheta_0}{\|\mathbf{I} - \gamma\mathbf{P}_\pi\|}$. Clearly, we can obtain that when $\|\mathbf{D}_\epsilon\| < \vartheta$ holds, $\mathbf{\Phi}^\top \mathbf{D}_{\boldsymbol{m}_w}(\mathbf{I} - \gamma\mathbf{P}_\pi)\mathbf{\Phi}$ is positive definite, which, together with Assumption 2, finally implies that $\mathbf{A}$ is positive definite and completes the proof.

□

A.4 Features of Baird's Counterexample

Original Features:

According to Sutton and Barto [3], we have $\boldsymbol{\phi}(s_1) \doteq [2,0,0,0,0,0,0,1]^\top$, $\boldsymbol{\phi}(s_2) \doteq [0,2,0,0,0,0,0,1]^\top$, $\boldsymbol{\phi}(s_3) \doteq [0,0,2,0,0,0,0,1]^\top$, $\boldsymbol{\phi}(s_4) \doteq [0,0,0,2,0,0,0,1]^\top$, $\boldsymbol{\phi}(s_5) \doteq [0,0,0,0,2,0,0,1]^\top$, $\boldsymbol{\phi}(s_6) \doteq [0,0,0,0,0,2,0,1]^\top$, $\boldsymbol{\phi}(s_7) \doteq [0,0,0,0,0,0,1,2]^\top$.

One-Hot Features:

$\boldsymbol{\phi}(s_1) \doteq [1,0,0,0,0,0,0,0]^\top$, $\boldsymbol{\phi}(s_2) \doteq [0,1,0,0,0,0,0,0]^\top$, $\boldsymbol{\phi}(s_3) \doteq [0,0,1,0,0,0,0,0]^\top$, $\boldsymbol{\phi}(s_4) \doteq [0,0,0,1,0,0,0,0]^\top$, $\boldsymbol{\phi}(s_5) \doteq [0,0,0,0,1,0,0,0]^\top$, $\boldsymbol{\phi}(s_6) \doteq [0,0,0,0,0,1,0,0]^\top$, $\boldsymbol{\phi}(s_7) \doteq [0,0,0,0,0,0,0,1]^\top$.

Zero-Hot Features:
$\boldsymbol{\phi}(s_1) \doteq [0,1,1,1,1,1,1]^\top$, $\boldsymbol{\phi}(s_2) \doteq [1,0,1,1,1,1,1]^\top$, $\boldsymbol{\phi}(s_3) \doteq [1,1,0,1,1,1,1]^\top$, $\boldsymbol{\phi}(s_4) \doteq [1,1,1,0,1,1,1]^\top$, $\boldsymbol{\phi}(s_5) \doteq [1,1,1,1,0,1,1]^\top$, $\boldsymbol{\phi}(s_6) \doteq [1,1,1,1,1,0,1]^\top$, $\boldsymbol{\phi}(s_7) \doteq [1,1,1,1,1,1,0]^\top$.
Aliased Features:
$\boldsymbol{\phi}(s_1) \doteq [2,0,0,0,0,0]^\top$, $\boldsymbol{\phi}(s_2) \doteq [0,2,0,0,0,0]^\top$, $\boldsymbol{\phi}(s_3) \doteq [0,0,2,0,0,0]^\top$, $\boldsymbol{\phi}(s_4) \doteq [0,0,0,2,0,0]^\top$, $\boldsymbol{\phi}(s_5) \doteq [0,0,0,0,2,0]^\top$, $\boldsymbol{\phi}(s_6) \doteq [0,0,0,0,0,2]^\top$, $\boldsymbol{\phi}(s_7) \doteq [0,0,0,0,0,2]^\top$.

References

[1] R. S. Sutton, A. R. Mahmood, M. White, An emphatic approach to the problem of off-policy temporal-difference learning, The Journal of Machine Learning Research 17 (2016) 2603–2631.

[2] S. Zhang, B. Liu, H. Yao, S. Whiteson, Provably convergent two-timescale off-policy actor-critic with function approximation, in: Proceedings of the 37th International Conference on Machine Learning, 2020, pp. 11204–11213.

[3] R. S. Sutton, A. G. Barto, Reinforcement learning: An introduction, 2nd ed., MIT press, 2018.

[4] J. Wang, Y. Wang, Z. Ji, Off-policy: Model-free optimal synchronization control for complex dynamical networks, Neural Processing Letters (2022) 1–18.

[5] V. Narayanan, H. Modares, S. Jagannathan, F. L. Lewis, Event-driven off-policy reinforcement learning for control of interconnected systems, IEEE Transactions on Cybernetics 52 (2022) 1936–1946.

[6] W. Meng, Q. Zheng, Y. Shi, G. Pan, An off-policy trust region policy optimization method with monotonic improvement guarantee for deep reinforcement learning, IEEE Transactions on Neural Networks and Learning Systems 33 (2022) 2223–2235.

[7] W. Li, F. Huang, X. Li, G. Pan, F. Wu, State distribution-aware sampling for deep q-learning, Neural Processing Letters 50 (2019) 1649–1660.

[8] J. Li, Z. Xiao, J. Fan, T. Chai, F. L. Lewis, Off-policy q-learning: Solving nash equilibrium of multi-player games with network-induced delay and unmeasured state, Automatica 136 (2022) 110076.

[9] M. Jaderberg, V. Mnih, W. M. Czarnecki, T. Schaul, J. Z. Leibo, D. Silver, K. Kavukcuoglu, Reinforcement learning with unsupervised auxiliary tasks, in: Proceedings of the 5th International Conference on Learning Representations, 2017.

[10] T. Zahavy, Z. Xu, V. Veeriah, M. Hessel, J. Oh, H. van Hasselt, D. Silver, S. Singh, A self-tuning actor-critic algorithm, in: Advances in Neural Information Processing Systems, 2020, pp. 20913–20924.

[11] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al., Human-level control through deep reinforcement learning, Nature 518

(2015) 529–533.

[12] L. Espeholt, H. Soyer, R. Munos, K. Simonyan, V. Mnih, T. Ward, Y. Doron, V. Firoiu, T. Harley, I. Dunning, S. Legg, K. Kavukcuoglu, IMPALA: scalable distributed deep-rl with importance weighted actor-learner architectures, in: Proceedings of the 35th International Conference on Machine Learning, 2018, pp. 1406–1415.

[13] R. Jiang, T. Zahavy, Z. Xu, A. White, M. Hessel, C. Blundell, H. van Hasselt, Emphatic algorithms for deep reinforcement learning, in: Proceedings of the 38th International Conference on Machine Learning, 2021, pp. 5023–5033.

[14] R. Jiang, S. Zhang, V. Chelu, A. White, H. van Hasselt, Learning expected emphatic traces for deep RL, in: Proceedings of the 36th AAAI Conference on Artificial Intelligence, 2022, pp. 12882–12890.

[15] Z. Guan, T. Xu, Y. Liang, PER-ETD: A polynomially efficient emphatic temporal difference learning method, in: 10th International Conference on Learning Representations, 2022.

[16] S. Zhang, B. Liu, S. Whiteson, Gradientdice: Rethinking generalized offline estimation of stationary values, in: Proceedings of the 37th International Conference on Machine Learning, 2020, pp. 11194–11203.

[17] Y. Liu, A. Swaminathan, A. Agarwal, E. Brunskill, Off-policy policy gradient with stationary distribution correction, in: Uncertainty in Artificial Intelligence, 2020, pp. 1180–1190.

[18] R. Zhang, B. Dai, L. Li, D. Schuurmans, Gendice: Generalized offline estimation of stationary values, in: Proceedings of the 8th International Conference on Learning Representations, 2020.

[19] A. M. Metelli, A. Russo, M. Restelli, Subgaussian and differentiable importance sampling for off-policy evaluation and learning, in: Advances in Neural Information Processing Systems, 2021, pp. 8119–8132.

[20] N. Kallus, M. Uehara, Double reinforcement learning for efficient off-policy evaluation in markov decision processes, The Journal of Machine Learning Research 21 (2020) 1–63.

[21] M. L. Puterman, Markov decision processes: discrete stochastic dynamic programming, John Wiley & Sons, 2014.

[22] H. Wai, M. Hong, Z. Yang, Z. Wang, K. Tang, Variance reduced policy evaluation with smooth function approximation, in: Advances in Neural Information Processing Systems, 2019, pp. 5776–5787.

[23] S. P. Shen, Y. J. Ma, O. Gottesman, F. Doshi-Velez, State relevance for off-policy evaluation, in: Proceedings of the 38th International Conference on Machine Learning, 2021, pp. 9537–9546.

[24] T. Degris, M. White, R. S. Sutton, Off-policy actor-critic, arXiv preprint arXiv:1205.4839 (2012).

[25] E. Imani, E. Graves, M. White, An off-policy policy gradient theorem using emphatic weightings, in: Advances in Neural Information Processing Systems, 2018, pp. 96–106.

[26] A. Hallak, A. Tamar, R. Munos, S. Mannor, Generalized emphatic temporal difference learning: Bias-variance analysis, in: Proceedings of the 30th AAAI Conference on Artificial Intelligence, 2016, pp. 1631–1637.

[27] R. S. Sutton, Learning to predict by the methods of temporal differences, Machine learning 3 (1988) 9–44.

[28] J. N. Tsitsiklis, B. Van Roy, An analysis of temporal-difference learning with function approximation, IEEE transactions on automatic control 42 (1997) 674–690.

[29] L. Baird, Residual algorithms: Reinforcement learning with function approximation, in: Proceedings of the 12th International Conference on Machine Learning, 1995, pp. 30–37.

[30] R. S. Sutton, C. Szepesvári, H. R. Maei, A convergent o(n) temporal-difference algorithm for off-policy learning with linear function approximation, in: Advances in Neural Information Processing Systems, 2008, pp. 1609–1616.

[31] R. S. Sutton, H. R. Maei, D. Precup, S. Bhatnagar, D. Silver, C. Szepesvári, E. Wiewiora, Fast gradient-descent methods for temporal-difference learning with linear function approximation, in: Proceedings of the 26th International Conference on Machine Learning, 2009, pp. 993–1000.

[32] H. R. Maei, Gradient temporal-difference learning algorithms, Phd thesis, University of Alberta, 2011.

[33] S. Zhang, S. Whiteson, Truncated emphatic temporal difference methods for prediction and control, Journal of Machine Learning Research 23 (2022) 1–59.

[34] H. van Hasselt, S. Madjiheurem, M. Hessel, D. Silver, A. Barreto, D. Borsa, Expected eligibility traces, in: Proceedings of the 35th AAAI Conference on Artificial Intelligence, 2021, pp. 9997–10005.

[35] A. Hallak, S. Mannor, Consistent on-line off-policy evaluation, in: Proceedings of the 34th International Conference on Machine Learning, 2017, pp. 1372–1383.

[36] Q. Liu, L. Li, Z. Tang, D. Zhou, Breaking the curse of horizon: Infinite-horizon off-policy estimation, in: Advances in Neural Information Processing Systems, 2018, pp. 5361–5371.

[37] C. Gelada, M. G. Bellemare, Off-policy deep reinforcement learning by bootstrapping the covariate shift, in: Proceedings of the 33th AAAI Conference on Artificial Intelligence, 2019, pp. 3647–3655.

[38] O. Nachum, Y. Chow, B. Dai, L. Li, Dualdice: Behavior-agnostic estimation of discounted stationary distribution corrections, in: Advances in Neural Information Processing Systems, 2019, pp. 2315–2325.

[39] S. Zhang, H. Yao, S. Whiteson, Breaking the deadly triad with a target network, in: Proceedings of the 38th International Conference on Machine Learning, 2021, pp. 12621–12631.

[40] S. Zhang, Y. Wan, R. S. Sutton, S. Whiteson, Average-reward off-policy policy evaluation with function approximation, in: Proceedings of the 38th

International Conference on Machine Learning, 2021, pp. 12578–12588.

[41] T. Wang, M. Bowling, D. Schuurmans, D. J. Lizotte, Stable dual dynamic programming, in: Advances in neural information processing systems, 2008, pp. 1569–1576.

[42] A. Hallak, S. Mannor, Consistent on-line off-policy evaluation, in: Proceedings of the 34th International Conference on Machine Learning, 2017, pp. 1372–1383.

[43] S. Zhang, V. Veeriah, S. Whiteson, Learning retrospective knowledge with reverse reinforcement learning, in: Advances in Neural Information Processing Systems, 2020, pp. 19976–19987.

[44] D. Precup, R. S. Sutton, S. Dasgupta, Off-policy temporal difference learning with function approximation, in: Proceedings of the 18th International Conference on Machine Learning, 2001, pp. 417–424.

[45] S. Zhang, W. Boehmer, S. Whiteson, Generalized off-policy actor-critic, in: Advances in Neural Information Processing Systems, 2019, pp. 1999–2009.

[46] H. Robbins, S. Monro, A stochastic approximation method, The annals of mathematical statistics (1951) 400–407.

[47] H. Yu, On convergence of emphatic temporal-difference learning, in: Proceedings of The 28th Conference on Learning Theory, 2015, pp. 1724–1751.

[48] H. Yu, Weak convergence properties of constrained emphatic temporal-difference learning with constant and slowly diminishing stepsize, The Journal of Machine Learning Research 17 (2016) 7745–7802.

[49] D. A. Levin, Y. Peres, Markov chains and mixing times, volume 107, American Mathematical Soc., 2017.

[50] S. Ghiassian, A. Patterson, S. Garg, D. Gupta, A. White, M. White, Gradient temporal-difference learning with regularized corrections, in: Proceedings of the 37th International Conference on Machine Learning, 2020, pp. 3524–3534.

[51] D. Bertsekas, J. Tsitsiklis, Parallel and distributed computation: Numeral methods (1989).

[52] J. Z. Kolter, The fixed points of off-policy TD, in: Advances in Neural Information Processing Systems, 2011, pp. 2169–2177.

[53] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, W. Zaremba, Openai gym, arXiv preprint arXiv:1606.01540 (2016).

[54] R. A. Horn, C. R. Johnson, Matrix analysis, 2nd ed., Cambridge university press, 2012.

[55] V. S. Borkar, S. P. Meyn, The ode method for convergence of stochastic approximation and reinforcement learning, SIAM Journal on Control and Optimization 38 (2000) 447–469.