Trust Region Policy Optimization with Optimal Transport Discrepancies: Duality and Algorithm for Continuous Actions

Antonio Terpin[*]
Automatic Control Laboratory
ETH Zürich
aterpin@ethz.ch

Nicolas Lanzetti[*]
Automatic Control Laboratory
ETH Zürich
lnicolas@ethz.ch

Batuhan Yardim
Dept. of Computer Science
ETH Zürich
alibatuhan.yardim@ethz.ch

Florian Dörfler
Automatic Control Laboratory
ETH Zürich
dorfler@ethz.ch

Giorgia Ramponi
Dept. of Computer Science
ETH AI Center
giorgia.ramponi@ai.ethz.ch

Abstract

Policy Optimization (PO) algorithms have been proven particularly suited to handle the high-dimensionality of real-world continuous control tasks. In this context, Trust Region Policy Optimization methods represent a popular approach to stabilize the policy updates. These usually rely on the Kullback-Leibler (KL) divergence to limit the change in the policy. The Wasserstein distance represents a natural alternative, in place of the KL divergence, to define trust regions or to regularize the objective function. However, state-of-the-art works either resort to its approximations or do not provide an algorithm for continuous state-action spaces, reducing the applicability of the method. In this paper, we explore optimal transport discrepancies (which include the Wasserstein distance) to define trust regions, and we propose a novel algorithm – Optimal Transport Trust Region Policy Optimization (OT-TRPO) – for continuous state-action spaces. We circumvent the infinite-dimensional optimization problem for PO by providing a one-dimensional dual reformulation for which strong duality holds. We then analytically derive the optimal policy update given the solution of the dual problem. This way, we bypass the computation of optimal transport costs and of optimal transport maps, which we implicitly characterize by solving the dual formulation. Finally, we provide an experimental evaluation of our approach across various control tasks. Our results show that optimal transport discrepancies can offer an advantage over state-of-the-art approaches.

1 Introduction

Reinforcement Learning (RL) has achieved outstanding results in numerous fields, from resource management [16], recommendation systems [42], and optimization of chemical reactions [44], to video-games [18, 43, 10] and board games [39], without sparing the world's champion of GO [32]. Many of these successful applications rely on Policy Optimization (PO) algorithms, a family of RL methods that are particularly suited to handle the high-dimensionality of real-world control tasks. PO algorithms approach the RL setting as an optimization problem in the policy space. In this context, the main challenge is to provide policy improvement guarantees. One remarkable option in this direction

[*]Equal contribution.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

is represented by Trust Region Policy Optimization (TRPO) [29], which constrains the optimization problem to policies that are "close" to the current one, whereby the Kullback-Leibler (KL) divergence is used as a similarity measure. Nevertheless, "closeness" in the policy space can also be quantified via other functions. Recent work [22, 33, 20] proposed to replace the KL divergence with the Wasserstein distance, a particular instance of optimal transport discrepancy (or cost). Besides being very natural and expressive, optimal transport discrepancies enjoy powerful topological, differential, geometrical, computational, and statistical features and guarantees [37, 3, 14]. In particular, (i) optimal transport discrepancies allow us to compare probability measures (and thus policies) not sharing the same support (for which the KL divergence is infinity); and (ii) they encapsulate the geometry encoded by the transport cost in the action space: the discrepancy between two actions coincides with the discrepancy between the corresponding deterministic policies (whereas the KL divergence is again infinity). These reasons make optimal transport discrepancies particularly attractive for RL. However, the mere evaluation of optimal transport discrepancies entails solving a transportation problem (e.g., see [23, 34]), which poses significant computational challenges for its deployment. Most of the previous work on the topic [22, 20] overcomes the computational burden via approximation, effectively changing the original problem. Conversely, [33] proposes two algorithms to solve the PO problem exactly, studying the trust regions described via the Wasserstein distance and the Sinkhorn divergence. However, their analysis is limited to discrete (and finite) settings. In our work, we consider optimal transport discrepancies to construct the trust region in settings where actions and states take value in general compact Polish spaces. This allows tackling many applications of interest involving continuous domains such as physical control tasks. We derive and leverage a dual reformulation of the PO problem to ensure an optimal policy update within the trust region, without any additional need for line searches (conversely to [29]). We circumvent the computation of optimal transport discrepancies via an analytical expression of the transport maps, which are characterized thanks to the dual reformulation. Notably, our analysis enables a practical and efficient algorithm that encompasses both discrete and continuous settings.

Contributions. Our contributions are summarized as follows:

1. We derive the dual of the optimal transport trust region policy optimization problem and we show that strong duality holds for general compact metric state-action spaces. We further characterize the optimal policy update given the solution to the dual problem. We show that policy updates can result in monotonic improvement of the performance function.
2. We propose a novel PO algorithm for continuous spaces, **O**ptimal **T**ransport **T**rust **R**egion **P**olicy **O**ptimization (OT-TRPO). Herein, we leverage the derived duality theory to provide policy updates that satisfy the optimal transport discrepancy constraint while circumventing its computation.
3. We conduct experiments in several RL benchmarks in both discrete and continuous state-action spaces, comparing our method to state-of-the-art approaches. Our results show the effectiveness of our approach for PO and the benefits of using optimal transport discrepancies.

2 Related Works

Optimal transport, and in particular the Wasserstein distance, has found various applications in RL and in particular in PO algorithms. In this section, we discuss the most relevant for our work; a broader overview is postponed to Appendix A.1. In [22], the authors propose Behavior Guided Policy Gradient (BGPG), whereby they replace the KL divergence trust region from TRPO [29] by a Wasserstein distance penalty in a behavioral space. Although our alternating procedure in Section 5 may resemble the approach of [22] in spirit, our approach is fundamentally different; cfr. [22, Algorithms 1 and 3] with Algorithm 1. In [20], the authors further suggested incorporating additional information about the local behavior of policies encapsulated in the so-called Wasserstein Information Matrix, in the attempt to speed up the PO using a Wasserstein Natural Policy Gradient (WNPG). However, these approaches are relatively slow, compared to the traditional Proximal Policy Optimization (PPO) and TRPO. Conversely to our work, they do not build on the idea of trust regions: we instead guarantee that the policy update is "close" to the previous one, where the "closeness" is defined via an optimal transport discrepancy (e.g., the Wasserstein distance). Accordingly, the closest related work to ours is the recent paper [33] which studied Wasserstein Policy Optimization (WPO) for discrete action spaces. In contrast to [33], the present work addresses the general setting of compact Polish spaces encompassing the cases of continuous and discrete state-action spaces as particular cases. While we also adopt a duality approach, our level of generality induces many challenges compared to the discrete action space setting (see Remark 2), and it is compatible even with non-direct policy

parametrizations. Finally, our work is closely connected with Wasserstein Distributionally Robust Optimization (DRO) [8, 19, 4, 25, 13]. Albeit our duality results are inspired from this literature, DRO is concerned with quantifying the worst-case risk of a cost functional over an ambiguity set of probability measures, which is a fundamentally different setting; see Remark 1. Conversely to all the previous work, we show how to perform exact optimal transport-based TRPO in continuous settings: we exploit optimal transport theory to circumvent the computational burden of evaluating optimal transport discrepancies while still performing exact policy updates within the trust regions. Accordingly, to the best of our knowledge, our practical algorithm is completely novel.

3 Preliminaries

We briefly introduce useful background and notation for the remainder of the paper.

Notation. For every Polish space $\mathcal{X}$ (i.e., completely metrizable separable topological space), the set of Borel probability measures on $\mathcal{X}$ is denoted by $\mathcal{P}(\mathcal{X})$. The Dirac measure at some point $x \in \mathcal{X}$ is denoted by δ_x. Given two Polish spaces $\mathcal{X}, \mathcal{Y}$, a Borel probability measures $\mu \in \mathcal{P}(\mathcal{X})$, and a Borel map $T : \mathcal{X} \to \mathcal{Y}$, the pushforward measure of μ, denoted by $T_{\#}\mu$, is defined by $(T_{\#}\mu)(A) \coloneqq \mu(T^{-1}(A))$ for all $A \in \mathcal{B}(\mathcal{Y})$, where $\mathcal{B}(\mathcal{Y})$ is the collection of Borel subsets of $\mathcal{Y}$. The set of probability measures on a finite set $\mathcal{X}$ coincides with the probability simplex and will also be denoted by $\mathcal{P}(\mathcal{X})$. For a given function $f : \mathcal{X} \to \mathbb{R}$, the notation $\|f\|_\infty$ refers to $\sup_{x\in\mathcal{X}} |f(x)|$.

Markov Decision Process. We consider an infinite-horizon discounted Markov Decision Process (MDP) [24] $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathrm{P}, r, \rho, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathrm{P} : \mathcal{S}\times\mathcal{A}\times\mathcal{S} \to \mathbb{R}_{\geq 0}$ is the state transition probability kernel, $r : \mathcal{S}\times\mathcal{A} \to \mathbb{R}$ is the reward function, ρ is the initial state probability distribution, and $\gamma \in [0, 1)$ is the discount factor. A randomized stationary Markovian policy, which we will simply call a policy in the rest of the paper, is a mapping $\pi : \mathcal{S} \to \mathcal{P}(\mathcal{A})$ specifying for each $s \in \mathcal{S}$ a probability measure over the set of actions $\mathcal{A}$ by $\pi(\cdot|s) \in \mathcal{P}(\mathcal{A})$. The set of all policies is denoted by Π. Each policy $\pi \in \Pi$ induces a discrete-time Markov reward process $\{(s_t, r(s_t, a_t))\}_{t\in\mathbb{N}}$, where $s_t \in \mathcal{S}$ represents the state of the system at time t and $r(s_t, a_t)$ corresponds to the reward received when executing action $a_t \in \mathcal{A}$ in state s_t. We denote by $\mathbb{P}_{\rho,\pi}$ the probability distribution of the Markov chain (s_t, a_t) issued from the MDP controlled by the policy π with initial state distribution ρ. The associated expectation is denoted by $\mathbb{E}_{\rho,\pi}$ and the notation $\mathbb{E}_\pi$ is used whenever there is no dependence on ρ. The state-value function $V^\pi : \mathcal{S} \to \mathbb{R}$ and the action-value function $Q^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ are defined for all $s \in \mathcal{S}, a \in \mathcal{A}$ by $V^\pi(s) \coloneqq \mathbb{E}_\pi[\sum_{t=0}^\infty \gamma^t r(s_t, a_t)|s_0 = s]$ and $Q^\pi(s, a) \coloneqq \mathbb{E}_\pi[\sum_{t=0}^\infty \gamma^t r(s_t, a_t)|s_0 = s, a_0 = a]$. We also define the advantage function $A^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ by $A^\pi(s, a) \coloneqq Q^\pi(s, a) - V^\pi(s)$. Given an initial state probability distribution ρ, our goal is to find a policy π maximizing the expected long-term return

$$J(\pi) \coloneqq \mathbb{E}_{\rho,\pi}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right],$$

which is well-defined when, e.g., the reward function is bounded. To solve this PO problem, we only have access to the observed state, action, and reward s_t, a_t, r_t at each time step t, whereas the state transition kernel P is unknown. When the state and action spaces ($\mathcal{S}$ and $\mathcal{A}$) are finite, an optimal policy π^* is guaranteed to exist. When $\mathcal{S}$ and $\mathcal{A}$ are continuous, a (measurable) optimal policy is also guaranteed to exist (see [24, Theorem 6.11.11, p. 262]) under appropriate assumptions on the state and action spaces, the reward function and the transition kernel; we will explicit these later on. In this paper, we focus on the continuous state-action space setting and comment on the discrete (non necessarily finite) setting as a special case.

Optimal transport. Consider a Polish space $\mathcal{X}$ and a continuous non-negative function $c : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$, referred to as *transport cost*. Let $\mu, \nu \in \mathcal{P}(\mathcal{X})$ and define the set of joint probability measures on $\mathcal{X} \times \mathcal{X}$ with marginals μ and ν:

$$\Gamma(\mu, \nu) \coloneqq \{\gamma \in \mathcal{P}(\mathcal{X} \times \mathcal{X}) : \gamma(A \times \mathcal{X}) = \mu(A), \gamma(\mathcal{X} \times B) = \nu(B)\ \forall\, A, B \in \mathcal{B}(\mathcal{X})\}.$$

We define the *optimal transport discrepancy* on $\mathcal{P}(\mathcal{X})$ for every probability measures μ and ν by

$$C(\mu, \nu) \coloneqq \min_{\gamma\in\Gamma(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{X}} c(x, x')\, \mathrm{d}\gamma(x, x'). \tag{1}$$

Notice that this definition is valid for both discrete and continuous measures. When $c = d^p$, where d is a distance on $\mathcal{X}$ and $p \geq 1$, then $C(\mu, \nu)^{1/p}$ reduces to the celebrated type-p Wasserstein

distance [37, 2]. In our PO context, we will use this discrepancy to compare two probability measures $\pi(\cdot|s) \in \mathcal{P}(\mathcal{A})$ and $\tilde{\pi}(\cdot|s) \in \mathcal{P}(\mathcal{A})$ for every $s \in \mathcal{S}$, where $\pi, \tilde{\pi} \in \Pi$ are two policies.

4 Optimal Transport for Trust Region Policy Optimization

In this section, we study the TRPO algorithm with a trust region defined using an optimal transport discrepancy as a measure of closeness between policies. We prove that the arising optimization problem admits an amenable dual reformulation. Importantly, we show that, given the dual optimal solution, the primal solution has an analytical expression, which can lead to monotonic improvements of the performance index.

4.1 Policy iteration algorithm with optimal transport-based trust regions

By the policy difference lemma [11, Lemma 6.1], the difference between the expected returns of two policies $\pi, \tilde{\pi} \in \Pi$ reads

$$J(\tilde{\pi}) = J(\pi) + \int_{\mathcal{S}} \int_{\mathcal{A}} A^{\pi}(s,a) \mathrm{d}\tilde{\pi}(a|s) \mathrm{d}\rho_{\tilde{\pi}}(s), \tag{2}$$

where $\rho_{\tilde{\pi}}$ is the discounted state-occupancy measure [11]. The complex dependency of the discounted visitation frequency $\rho_{\tilde{\pi}}$ on the policy $\tilde{\pi}$ hampers the direct optimization of (2); see [29, Sec. 2]. Following previous work, we consider instead a local approximation of the expected return J, defined by

$$L_{\pi}(\tilde{\pi}) \coloneqq J(\pi) + \int_{\mathcal{S}} \int_{\mathcal{A}} A^{\pi}(s,a) \mathrm{d}\tilde{\pi}(a|s) \mathrm{d}\rho_{\pi}(s). \tag{3}$$

Observe that this approximation uses the discounted state-occupancy measure ρ_{π} (which can be estimated) instead of $\rho_{\tilde{\pi}}$ (see (2)). In other words, the influence of a policy change on the discounted state-occupancy measure is neglected. Moreover, this surrogate function coincides with the expected return J when $\tilde{\pi} = \pi$. Then, (3) motivates a policy update rule maximizing at each time step the approximation $L_{\pi}(\tilde{\pi})$ over $\tilde{\pi}$, where π is the current policy that we want to improve upon (see also [33, Section 2, Eq. (1)]). To ensure stability of the update, we conservatively update the policy using a discrepancy constraint between the current and the new one. Unlike TRPO, we do not use the KL divergence to define the trust region, but instead an optimal transport discrepancy. Then, at each time step, our method solves

$$\begin{aligned} &\sup_{\tilde{\pi} \in \Pi} \int_{\mathcal{S}} \int_{\mathcal{A}} A^{\pi}(s,a) \mathrm{d}\tilde{\pi}(a|s) \mathrm{d}\rho_{\pi}(s), \\ &\text{s.t. } \tilde{\pi} \in \mathcal{T}_{\varepsilon}(\pi) \coloneqq \left\{ \tilde{\pi} \in \Pi : \int_{\mathcal{S}} C(\pi(\cdot|s), \tilde{\pi}(\cdot|s)) \mathrm{d}\rho_{\pi}(s) \leq \varepsilon \right\}, \end{aligned} \tag{P}$$

where $\varepsilon > 0$ is a parameter defining the radius of the trust region $\mathcal{T}_{\varepsilon}(\pi)$. Similarly to [29, Eq. (12)] and [33, Problem (4)], we consider the *average* optimal transport discrepancy over the state space as optimization constraint. Accordingly, the OT-TRPO policy optimization results from iteratively solving Problem (P).

4.2 Dual of the trust-region constrained problem (P)

Problem (P) is intractable for two main reasons. First, as soon as the state or action space is continuous, it is an infinite-dimensional optimization problem. Second, the mere evaluation of the trust-region constraint (e.g., for line search as in TRPO [29]) needs (possibly) infinitely many computations of the optimal transport discrepancy, which is itself already challenging to estimate. However, inspired by prior works on Wasserstein DRO [8, 19, 41, 4], we show that problem (P) admits a tractable one-dimensional convex dual reformulation. This duality theorem is the cornerstone of the design of our algorithm. Before stating the result, we make the following assumptions.

Assumption 1. The state space $\mathcal{S}$ is a compact subset of an Euclidean space, the action space $\mathcal{A}$ is a compact subset of a Polish space, the reward function r is a continuous function and for every continuous function w on $\mathcal{S}$, $\int_{\mathcal{S}} w(u) \mathrm{dP}(u|s,a)$ is continuous in both s and a.

Under this assumption, there exists an optimal measurable (stationary) policy to the PO problem formulated in Section 3. We refer the reader to [24, Theorem 6.11.11, p. 262] for a statement of this result and milder assumptions. In particular, our duality result continues to hold if $\mathcal{S}$ is a compact Polish space (i.e., not necessarily Euclidean).

Assumption 2. For every policy $\pi \in \Pi$, the advantage function $A^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is continuous. Moreover, the transport cost $c : \mathcal{A} \times \mathcal{A} \to \mathbb{R}_{\geq 0}$ is continuous and satisfies $c(a, a) = 0$ for all $a \in \mathcal{A}$.

In the next theorem, we show that under these assumptions Problem (P) admits a dual reformulation for which strong duality holds.

Theorem 1 (Dual formulation)**.** *For every $\varepsilon > 0$ and for every policy $\pi \in \Pi$, under Assumptions 1 and 2 the following strong duality result holds:*

$$\max_{\tilde{\pi} \in \Pi} \left\{ \int_{\mathcal{S}} \int_{\mathcal{A}} A^\pi(s, a) \, \mathrm{d}\tilde{\pi}(a|s) \mathrm{d}\rho_\pi(s) : \int_{\mathcal{S}} C(\pi(\cdot|s), \tilde{\pi}(\cdot|s)) \mathrm{d}\rho_\pi(s) \leq \varepsilon \right\} \tag{P}$$

$$= \min_{\lambda \geq 0} \left\{ \lambda \varepsilon + \int_{\mathcal{S}} \int_{\mathcal{A}} \max_{a' \in \mathcal{A}} \{A^\pi(s, a') - \lambda c(a, a')\} \mathrm{d}\pi(a|s) \mathrm{d}\rho_\pi(s) \right\}. \tag{D}$$

Moreover, the primal and dual problems (P) *and* (D) *admit a maximizer and a minimizer, respectively.*

Remarkably, Problem (D) is one-dimensional and convex, and it only involves the current policy π, advantage function A^π, and visitation frequency ρ_π. The proof of Theorem 1 is constructive. In particular, we derive a closed-form solution of problem (P) as a function of the optimal Lagrange multiplier λ^* solving problem (D) and the policy π defining the problem. Even if the closed-form policy is part of Theorem 1 and its proof, we present it separately for clarity and for later reference. To do so, we introduce some additional notation, which is instrumental to derive a practical algorithm (similarly to [33]).

Under Assumptions 1 and 2, define for every $\lambda \geq 0$ the λ-regularized advantage $\Phi_\lambda : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ and its associated set of maximizers for every $s \in \mathcal{S}, a \in \mathcal{A}$ as follows:

$$\begin{aligned} \Phi_\lambda(s, a) &\coloneqq \max_{a' \in \mathcal{A}} \{A^\pi(s, a') - \lambda c(a, a')\}, \\ \mathcal{D}_\lambda(s, a) &\coloneqq \arg\max_{a' \in \mathcal{A}} \{A^\pi(s, a') - \lambda c(a, a')\}. \end{aligned} \tag{4}$$

Corollary 2 (Optimal policy)**.** *Under the setting and assumptions of Theorem 1, for any policy $\pi \in \Pi$, let $\lambda^* \geq 0$ be the minimizer of the dual problem* (D)*. Then, the following statements hold:*

1. *For every $\lambda \geq 0$, there exist two measurable selection maps $\underline{T}_\lambda : \mathcal{S} \times \mathcal{A} \to \mathcal{A}$ and $\overline{T}_\lambda : \mathcal{S} \times \mathcal{A} \to \mathcal{A}$ such that for every $s \in \mathcal{S}, a \in \mathcal{A}$*

$$\underline{T}_\lambda(s, a) \in \arg\min_{a' \in \mathcal{D}_\lambda(s,a)} c(a, a'), \quad \overline{T}_\lambda(s, a) \in \arg\max_{a' \in \mathcal{D}_\lambda(s,a)} c(a, a'). \tag{5}$$

2. *If $\lambda^* > 0$, there exists $t^* \in [0, 1]$ such that*

$$t^* \int_{\mathcal{S}} \int_{\mathcal{A}} c(a, \underline{T}_{\lambda^*}(s, a)) \mathrm{d}\pi(a|s) \mathrm{d}\rho_\pi(s) + (1 - t^*) \int_{\mathcal{S}} \int_{\mathcal{A}} c(a, \overline{T}_{\lambda^*}(s, a)) \mathrm{d}\pi(a|s) \mathrm{d}\rho_\pi(s) = \varepsilon. \tag{6}$$

3. *There exists an optimal feasible policy $\tilde{\pi}$ for problem* (P) *defined for every $s \in \mathcal{S}$ by*

$$\tilde{\pi}(\cdot|s) \coloneqq t^* \underline{T}_{\lambda^*}(s, \cdot)_\# \pi(\cdot|s) + (1 - t^*) \overline{T}_{\lambda^*}(s, \cdot)_\# \pi(\cdot|s), \tag{7}$$

where t^* results from (6) *if $\lambda^* > 0$ and $t^* = 0$ if $\lambda^* = 0$.*

Intuitively, Corollary 2 suggests that the optimal policy results from displacing the probability mass $\pi(a|s)$ to the maximizers of the λ^*-regularized advantage $\Phi_{\lambda^*}(s, a)$, where $\lambda^* \geq 0$ is the optimal dual solution. Since maximizers are generally not unique (i.e., $\mathcal{D}_{\lambda^*}(s, a)$ is not a singleton), one needs to balance between the *closest* (i.e., $\underline{T}_{\lambda^*}(s, a)$) and the *furthest apart* (i.e., $\overline{T}_{\lambda^*}(s, a)$) to satisfy the trust region constraint. In the special case $\lambda^* = 0$, the trust region constraint is either not active (i.e., the optimal policy lies within the trust region) or it does not affect the optimal trust region constraint (i.e., the optimal policy would lie at the boundary of the trust region even if the constraint is removed). In this case, it suffices to displace all probability mass to the closest maximizer of the advantage functions (i.e., $\underline{T}_0(s, a) \in \mathcal{D}_0(s, a)$). The complete proof of the results of this section is deferred to Appendix B.1.

Remark 1. Similar results were previously established in the literature in the context of DRO (e.g., see [8, Theorem 1] and [4, Theorem 1]). While these results closely inspire our proof, there is a major difference: in DRO, one seeks to evaluate the worst-case cost over an ambiguous set of probability distributions, expressed in terms of the optimal transport discrepancy. As such, the *average* optimal transport discrepancy in the trust region constraint is replaced by a single optimal transport discrepancy. Thus, one does not need to ensure the regularity of the problem with respect to the state (e.g., measurability of $\underline{T}_\lambda$ w.r.t. s). This is reflected in our assumption of *joint* continuity of the advantage in state and action and the state space being compact. To readily deploy existing results in DRO, one needs (i) to consider a single state only (i.e., $\mathcal{S} = \{s\}$) or (ii) to define a trust region *for each* state (at the price of infinitely many constraints). To transform (P) into a DRO, one might alternatively identify a policy as a probability measure over $\prod_{s\in\mathcal{S}} \mathcal{A}$, and hope to deploy standard duality arguments in DRO. However, the uncountable product of Polish spaces is not a Polish space, which makes all results in DRO, and more generally in optimal transport [37], inapplicable.

Remark 2. A similar result in the discrete case was presented in [33]. We highlight four major differences. First, in the discrete setting, problem (P) is a finite-dimensional linear optimization problem, for which strong duality holds. Thus, linear programming arguments can be used to derive the dual reformulation. In the continuous setting, the same proof strategy would require to mobilize the abstract machinery of infinite-dimensional linear programming [15]. Second, in the discrete setting, continuity and measurability of all functions are "for free". On the contrary, the continuous case imposes a careful analysis of these issues. Third, the optimal policy update [33] implicitly assumes that the set $\mathcal{D}_\lambda(s,a)$ (see (4)) is a singleton, which is rarely satisfied in practice. Fourth, as a byproduct of our proof, we show that (D) is a (one-dimensional) *convex* optimization problem, which can be solved efficiently via off-the-shelf solvers. This way, we do not need to resort to approximation techniques [33, Section 6.1] for the optimal dual multiplier.

Discrete state-action spaces. In the remainder of this section, we specialize our results to discrete (finite) state-action spaces (which trivially satisfy Assumptions 1 and 2). Without loss of generality, we represent the state and action spaces by $\mathcal{S} = \{s_1, \ldots, s_M\}$ and $\mathcal{A} = \{a_1, \ldots, a_N\}$ where M and N are two positive integers, and we describe any policy $\pi \in \Pi$ and its corresponding state-occupancy measure ρ_π as discrete measures:

$$\pi(\cdot|s_i) = \sum_{j=1}^N \pi_{i,j}\delta_{a_j} \qquad \forall i \in \{1,\ldots,M\}, \qquad \rho_\pi = \sum_{i=1}^M \rho_i \delta_{s_i}, \tag{8}$$

where $\rho_i, \pi_{i,j} \geq 0$ for every $i \in \{1,\ldots,M\}$, $j \in \{1,\ldots,N\}$, $\sum_{i=1}^M \rho_i = 1$ and $\sum_{j=1}^N \pi_{i,j} = 1$ for every $i \in \{1,\ldots,M\}$.[2] The analogous results to Theorem 1 and Corollary 2 are as follows.

Corollary 3 (Dual formulation - discrete setting)**.** *Let $\varepsilon > 0$. For every policy $\pi \in \Pi$, the following strong duality result holds:*

$$\begin{aligned}\max_{\substack{t\in[0,1],\underline{b}_{i,j},\overline{b}_{i,j}\in\mathcal{A},\\ i\in\{1,\ldots M\},\, j\in\{1,\ldots,N\}}} & \left\{ \sum_{i=1}^M \rho_i \sum_{j=1}^N \pi_{i,j}\left(tA^\pi(s_i,\underline{b}_{i,j}) + (1-t)A^\pi(s_i,\overline{b}_{i,j})\right) : \right. \qquad \text{(discrete-P)}\\ & \qquad \left. \sum_{i=1}^M \rho_i \sum_{j=1}^N \pi_{i,j}\left(tc(a_j,\underline{b}_{i,j}) + (1-t)c(a_j,\overline{b}_{i,j})\right) \leq \varepsilon \right\}\\ & = \min_{\lambda\geq 0}\left\{ \lambda\varepsilon + \sum_{i=1}^M \rho_i \sum_{j=1}^N \pi_{i,j}\Phi_\lambda(s_i,a_j)\right\}. \qquad \text{(discrete-D)}\end{aligned}$$

In particular, let $\lambda^* \geq 0$ be a solution to (discrete-D)*, and given for every $i \in \{1,\ldots,M\}$, $j \in \{1,\ldots,N\}$, select any*

$$\underline{b}^*_{i,j} \in \operatorname*{arg\,min}_{a'\in\mathcal{D}_{\lambda^*}(s_i,a_j)} c(a_j,a'), \qquad \overline{b}^*_{i,j} \in \operatorname*{arg\,max}_{a'\in\mathcal{D}_{\lambda^*}(s_i,a_j)} c(a_j,a'), \tag{9}$$

and let

$$\underline{c} \coloneqq \sum_{i=1}^M \rho_i \sum_{j=1}^N \pi_{i,j} c(a_j,\underline{b}^*_{i,j}), \qquad \overline{c} \coloneqq \sum_{i=1}^M \rho_i \sum_{j=1}^N \pi_{i,j} c(a_j,\overline{b}^*_{i,j}).$$

[2] This representation is also valid beyond the finite state-action space setting when the policies and the state-occupancy measures are empirical distributions with finitely many samples.

Then, an optimal policy $\tilde{\pi}$ is given by

$$\tilde{\pi}(\cdot|s_i) = \sum_{j=1}^{N} \pi_{i,j} \left(t^* \delta_{\underline{b}^*_{i,j}} + (1-t^*) \delta_{\overline{b}^*_{i,j}} \right), \qquad \forall i \in \{1, \dots, M\}, \tag{10}$$

with $t^* = (\overline{c} - \varepsilon)/(\overline{c} - \underline{c}) \in [0,1]$ (and $t^* \in [0,1]$ if $\underline{c} = \overline{c} = \varepsilon$) if $\lambda^* > 0$ and $t^* = 0$ if $\lambda^* = 0$.

The proof of this result stems from substituting the discrete measures as defined in (8) in problems (P) and (D), and observing that the images of the mappings $\underline{T}_{\lambda^*}$ and $\overline{T}_{\lambda^*}$ have finite support in the current setting. Notably, Corollary 3 directly provides an implementable algorithm for the policy update, circumventing the difficulty of the mixed-integer optimization problem (discrete-P): solving the one-dimensional convex program (discrete-D) provides the optimal Lagrange multiplier associated to the trust region constraint of the primal problem which can be directly used to compute the actions $\underline{b}^*_{i,j}$, $\overline{b}^*_{i,j}$ via (9), and thus the policy $\tilde{\pi}$ via (10).

Remark 3. The policy update suggested by (10) differs from the one in [33] (see [33, Theorem 1, (5)] where $f^*_s(i,j) \in \{0,1\}$ with their notation). Indeed, our policy update relies on "splitting the probability mass": the probability mass $\pi(\cdot|s_i)$ is displaced to $\underline{b}^*_{i,j}$ and $\overline{b}^*_{i,j}$ with weights t^* and $1-t^*$, respectively. This result is consistent with the Wasserstein DRO literature (e.g., see [4, Remark 2]). The result provided in [33] corresponds to the particular case where $\underline{b}^*_{i,j} = \overline{b}^*_{i,j}$ which amounts to supposing that the set $\mathcal{D}_\lambda(s_i, a_j)$ defined in (4) is a singleton. We provide further comments and examples in Appendix A.2 to illustrate the importance of this "mass splitting".

4.3 Policy improvement

In the next result, we show that our policy update leads to a monotonic improvement of the performance function J up to the advantage function estimation error.

Proposition 4 (Performance improvement)**.** *Let $\pi \in \Pi$. Consider solutions $\tilde{\pi}^* \in \Pi$ and $\lambda^* \geq 0$ of problems* (P) *and* (D)*, respectively. If the true advantage function A^π is approximated by some estimated advantage function $\hat{A}^\pi$ such that $\|A^\pi - \hat{A}^\pi\|_\infty < \infty$, then the following bound holds:*

$$J(\tilde{\pi}^*) \geq J(\pi) + \frac{\lambda^*}{1-\gamma} \int_{\mathcal{S}} C(\pi(\cdot|s), \tilde{\pi}^*(\cdot|s)) \mathrm{d}\rho_{\tilde{\pi}^*}(s) - \frac{2\|A^\pi - \hat{A}^\pi\|_\infty}{1-\gamma}. \tag{11}$$

Proposition 4 indicates that optimal transport-based trust region policy optimization leads to monotonic improvement of the performance function when we have access to the true advantage function. The proof, postponed to Appendix B.2, of this result builds on the performance difference lemma (see (2)) and uses the closed-form expression of the optimal policy solving problem (P) as constructed in the proof of Theorem 1 (see Corollary 2). The analog of this result for a finite action space was proved in [33, Theorem 2, p. 5]. To the best of our knowledge, this result is novel for the continuous state-action space setting.

5 Practical Optimal Transport Trust Region Policy Optimization Algorithm

In this section, we use the duality results on Section 4 to derive a practical algorithm for OT-TRPO. Herein, we restrict the policy search set Π to the set of policies π_θ parametrized by a vector $\theta \in \mathbb{R}^d$ for some integer $d > 0$. We require the policy parametrization to be continuously differentiable with respect to θ (for every state and action). This way, we simultaneously cover the direct parametrization (for which Corollary 3 directly provides a policy update) as well as commonly used policy parametrizations (e.g., softmax and the Gaussian policies). Accordingly, the dual problem (D) can be reformulated as follows for every $\theta \in \mathbb{R}^d$:

$$\min_{\lambda \geq 0} G(\lambda, \theta) \coloneqq \lambda\varepsilon + \int_{\mathcal{S}} \int_{\mathcal{A}} \max_{a' \in \mathcal{A}} \{A^{\pi_\theta}(s,a') - \lambda c(a,a')\} \mathrm{d}\pi_\theta(a|s) \mathrm{d}\rho_{\pi_\theta}(s). \tag{12}$$

Given a current policy represented by the vector θ, we first solve the one-dimensional convex problem (12) to obtain its solution λ^*. Then, we use the optimal dual multiplier λ^* to derive the optimal policy update within the trust region. The procedure is summarized in Algorithm 1. Depending on the parametrization of the policy, the steps of Section 5 are as follows:

Algorithm 1 OT-TRPO.

1: Initialize π_{θ_0}
2: **for all** $t = 0, 1, \dots$ **do**
3: Estimate $A^{\pi_{\theta_t}}$ and $\rho_{\pi_{\theta_t}}$.
4: Compute $\lambda^* \in \operatorname{argmin}_{\lambda \geq 0} G(\lambda, \theta_t)$.
5: Update θ_t to θ_{t+1} using λ^*.
6: **end for**

Algorithm 1 - step 3. In the discrete setting, the visitation frequency is estimated via Monte Carlo methods. In the continuous case, we weight every visited state equally. We propose three ways to estimate the unknown advantage function via samples[3]:

1. Monte Carlo methods or TD-learning (for discrete settings only);
2. General Advantage Estimation (GAE) [30], using a neural network to approximate the value function like in standard actor-critic methods; and
3. Direct estimation via non-linear approximators (e.g., using directly a neural network for the advantage function).

Algorithm 1 - step 4 (evaluation of G). Depending on the setting, we propose various ways to evaluate $G(\lambda, \theta)$. They all apply to both continuous and discrete states.

1. *Finite actions*: Since the maximization in (12) is over finitely many actions, we can directly evaluate (12) for any $\lambda \geq 0$.
2. *Gaussian policy parametrization*: With $m(s)$ being the mean of the Gaussian policy (with fixed variance), we approximate $G(\lambda, \theta)$ by

$$G(\lambda, \theta) \approx \begin{cases} \lambda\varepsilon + \sum_{s \in \hat{\mathcal{S}}} \max_{a' \in \{a, m(s)\}} \{A^{\pi_\theta}(s, a') - \lambda c(m_\theta(s), a')\} \rho_{\pi_\theta}(s) & \text{if } A^{\pi_\theta} \text{ via GAE,} \\ \lambda\varepsilon + \sum_{s \in \hat{\mathcal{S}}} \max_{a' \in \hat{\mathcal{A}}(s)} \{A^{\pi_\theta}(s, a') - \lambda c(m_\theta(s), a')\} \rho_{\pi_\theta}(s) & \text{if } A^{\pi_\theta} \text{ via NN,} \end{cases}$$

   where $\hat{\mathcal{S}}$ are the states visited in the trajectory and $\hat{\mathcal{A}}(s)$ is a (possibly state-dependent) collection of samples from $\mathcal{A}$.
3. *Arbitrary policy parametrization*: For a neural network approximation of the advantage function, we approximate $G(\lambda, \theta)$ by

$$G(\lambda, \theta) \approx \lambda\varepsilon + \sum_{s \in \hat{\mathcal{S}}} \sum_{a \in \hat{\mathcal{A}}_1(s)} \max_{a' \in \hat{\mathcal{A}}_2(s)} \{A^{\pi_\theta}(s, a') - \lambda c(a, a')\} \pi_\theta(a|s) \rho_{\pi_\theta}(s),$$

   where $\hat{\mathcal{S}}$ are the states visited in the trajectory and $\hat{\mathcal{A}}_i(s)$ are (possibly state-dependent) collections of samples from $\mathcal{A}$.

Algorithm 1 - step 4 (solving for λ^*). Since (D) is a one-dimensional convex optimization problem, λ^* can be found using any solver for convex optimization problems.

Algorithm 1 - step 5. Update the parameter vector θ.

1. *Direct parametrization (finite spaces)*: Update θ according to (10) and (9).
2. *Direct parametrization via policy network (continuous states, discrete actions)*: Use (10) and (9) to compute the optimal policy update at the visited states, denoted by $\pi^*_{\theta_t}$. Then, update the policy network by performing gradient descent on the loss $L(\theta) = \sum_{s \in \hat{\mathcal{S}}} \rho_{\pi_{\theta_t}}(s) \left\| \pi_\theta(\cdot|s) - \pi^*_{\theta_t}(\cdot|s) \right\|^2$ to steer π_θ towards the optimal policy update $\pi^*_{\theta_t}$ *within* the trust region.
3. *Arbitrary policy parametrization*: Since there are infinitely many actions, the computation of the maximization is computationally demanding, and so Corollary 2 cannot be directly utilized for the policy update. Yet, we can update the policy via gradient ascent. The intuition is as follows: according to Corollary 2, the optimal policy update attains the maximum $\max_{a' \in \mathcal{A}} \{A^{\pi_{\theta_t}}(s, a') - \lambda^* c(a, a')\}$ at each state. Thus, we can steer the policy π_θ to maximize

$$\theta \mapsto \sum_{s \in \hat{\mathcal{S}}} \int_{\mathcal{A}} \max_{a' \in \mathcal{A}} \{A^{\pi_{\theta_t}}(s, a') - \lambda^* c(a_\theta, a')\} \mathrm{d}\pi_\theta(a_\theta|s) \rho_{\pi_{\theta_t}}(s).$$

   In the particular case of a Gaussian policy with parametrized mean (and fixed variance), combined with GAE estimate of the advantage function, one can maximize

$$\theta \mapsto \sum_{s \in \hat{\mathcal{S}}} \max\{A^{\pi_{\theta_t}}(s, a') - \lambda^* c(m_\theta(s), a'), 0\} \rho_{\pi_{\theta_t}}(s).$$

[3] In the experiments reported in the main paper we used the first and the second method for discrete and continuous environments, respectively. In Appendix A.5 we further comment on the different methods.

Intuitively, this update implicitly estimates the transport maps $\overline{T}_{\lambda^*}$ and $\underline{T}_{\lambda^*}$, which are needed for the optimal policy update. This way, we steer the policy network towards the optimal policy update *within* the trust region. Among others, this policy update allows for the following interpretation: imposing an optimal transport-based trust region constraints is, at least formally, equivalent to maximizing a *regularized* advantage function, where the value of the regularization $\lambda^* \geq 0$ is based on the transport cost c and the radius of the trust region ε.

6 Experiments and Insights

In this section, we evaluate the performance of OT-TRPO across a variety of environments [5, 36] of increasing complexity, ranging from discrete to continuous settings. We compare it to the classical TRPO [29, 9] and PPO [31, 9], with Advantage Actor Critic (A2C) [17, 9], with the recent approaches leveraging the Wasserstein distance, BGPG [22] and WNPG [20] (in continuous settings), and with WPO [33] (in discrete settings). The training curves are shown in Fig. 1; see Appendix A.3 for implementation details, Appendix A.6 for further details on the experimental results, and Appendix A.4 for an ablation study on our algorithm.

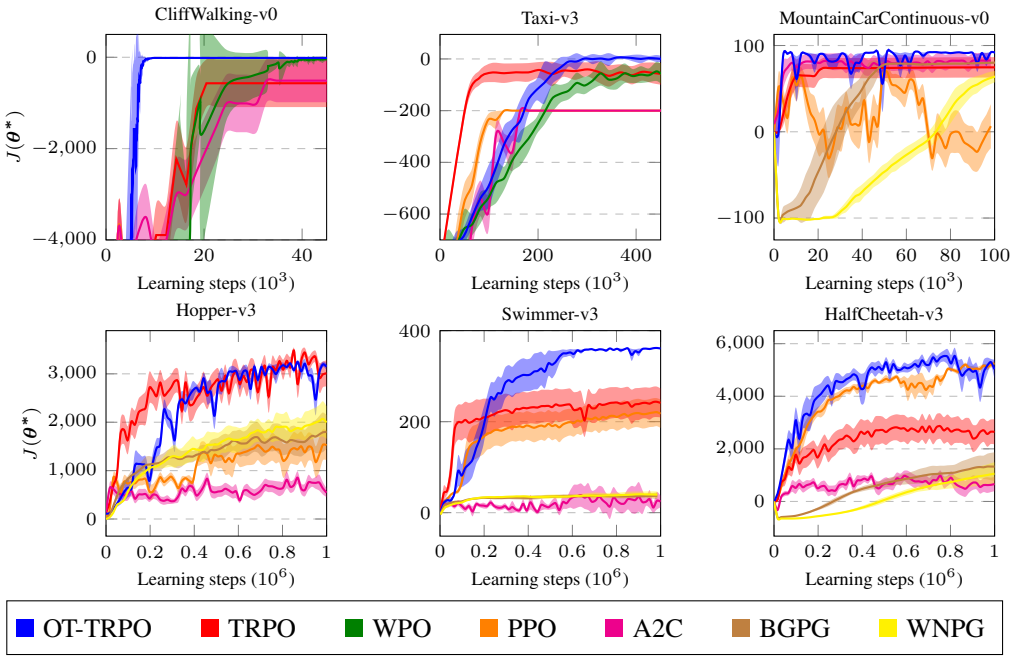
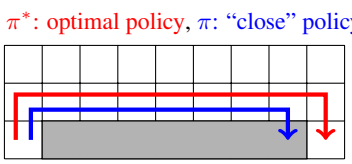


Figure 1: Cumulative rewards during the training process in different environments. The shaded area represents the mean $\pm$ the standard deviation across 10 independent runs. Every policy evaluation in each run is averaged over 10 sampled trajectories.

Our approach is shown to consistently improve over the other algorithms: OT-TRPO leads to larger final returns, with lower variance, and only in few cases at the expense of a slightly slower learning curve. Four remarks are in order. First, the performance gain of OT-TRPO compared to BGPG and WNPG confirms that trust regions help stabilize training, as already observed in [29]. Second, optimal transport discrepancies induce a more natural notion of "closeness" between policies compared to the KL divergence (e.g., see [3, Example 2.1]). For instance, in CliffWalking-v0, consider the optimal policy π^* and the candidate policy π depicted below, which differ at one state only (see figure). The optimal transport discrepancy between π and π^* is $\rho_\pi(s)C(\pi(\cdot|s), \pi^*(\cdot|s)) = \rho_\pi(s)c(\text{Down}, \text{Right})$. When using the KL divergence, instead, the discrepancy is infinite, since the two policies do not share the same support. In particular, if initialized with π, TRPO cannot converge to the optimal policy, regardless of the radius of the trust region. Third, OT-TRPO improves on WPO, in two ways: (i) it leads to superior performances of the trained policies and (ii) it does not violate



the trust region constraint (which, e.g., in Taxi-v3 is the case for 72% of the updates of WPO). This performance improvement results from the "mass splitting" (see Remark 3), which is the only difference between the two algorithms (in discrete settings). Fourth, in continuous settings, the performance of OT-TRPO is aligned with the best-performing alternative approach. In the environment Swimmer-v3, it even yields an improvement of more than 50% in the performance of the trained agent.

7 Conclusion and Future Work

We studied trust region policy optimization for continuous state-action spaces whereby the trust region is defined in terms of a general optimal transport discrepancy. Our analysis bases on a one-dimensional convex dual reformulation of the optimization problem for the policy update which (i) enjoys strong duality and (ii) directly characterizes the optimal policy update, bypassing the computational burden of evaluating optimal transport discrepancies. Moreover, we show that the policy update can yield a monotonic improvement of the performance index. Empowered by our theoretic results, we propose a novel algorithm, OT-TRPO, for trust region policy optimization with optimal transport discrepancies. We evaluate its performance across several environments. Our results reveal that trust regions defined by optimal transport discrepancies can offer advantages over the KL divergence or non-trust region methods.

There are several research directions that merit further investigation. We highlight two. First, transport costs provide us actionable knobs to shape the geometry of the trust region, and can be used to encode prior knowledge on the environment or preferred exploration strategies. Second, we would like to study the convergence properties of the proposed algorithm.

Acknowledgements

This project has received funding from Google Brain, Swiss National Science Foundation under the NCCR Automation (grant agreement 51NF40_180545), and it was partially supported by the ETH AI Center.

References

[1] Charalambos D. Aliprantis and Kim C. Border. *Infinite Dimensional Analysis: a Hitchhiker's Guide*. Springer, Berlin; London, 2006.

[2] Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré. *Gradient flows: In Metric Spaces and in the Space of Probability Measures*. Springer, 2008.

[3] Liviu Aolaritei, Nicolas Lanzetti, Hongruyu Chen, and Florian Dörfler. Uncertainty propagation via optimal transport ambiguity sets. *arXiv preprint arXiv:2205.00343*, 2022.

[4] Jose Blanchet and Karthyek Murthy. Quantifying distributional model risk via optimal transport. *Mathematics of Operations Research*, 44(2):565–600, 2019.

[5] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI Gym. *arXiv preprint arXiv:1606.01540*, 2016.

[6] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.

[7] Rui Gao, Xi Chen, and Anton J. Kleywegt. Wasserstein distributionally robust optimization and variation regularization. *arXiv preprint arXiv:1712.06050*, 2017.

[8] Rui Gao and Anton J. Kleywegt. Distributionally robust stochastic optimization with Wasserstein distance. *arXiv preprint arXiv:1604.02199*, 2016.

[9] Ashley Hill, Antonin Raffin, Maximilian Ernestus, Adam Gleave, Anssi Kanervisto, Rene Traore, Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, and Yuhuai Wu. Stable baselines. `https://github.com/hill-a/stable-baselines`, 2018.

[10] Łukasz Kaiser, Mohammad Babaeizadeh, Piotr Miłos, Błażej Osiński, Roy H Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, et al. Model based reinforcement learning for Atari. In *International Conference on Learning Representations*, 2019.

[11] Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *International Conference on Machine Learning*, pages 267–274. PMLR, 2002.

[12] Achim Klenke. *Probability Theory: A Comprehensive Course*. Springer, 2008.

[13] Daniel Kuhn, Peyman Mohajerin Esfahani, Viet Anh Nguyen, and Soroosh Shafieezadeh-Abadeh. Wasserstein distributionally robust optimization: Theory and applications in machine learning. In *Operations research & management science in the age of analytics*, pages 130–166. Informs, 2019.

[14] Nicolas Lanzetti, Saverio Bolognani, and Florian Dörfler. First-order conditions for optimization in the Wasserstein space. *arXiv preprint arXiv:2209.12197*, 2022.

[15] David G. Luenberger. *Optimization by vector space methods*. John Wiley & Sons, 1997.

[16] Hongzi Mao, Mohammad Alizadeh, Ishai Menache, and Srikanth Kandula. Resource management with deep reinforcement learning. In *15th ACM workshop on hot topics in networks*, pages 50–56, 2016.

[17] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*, pages 1928–1937. PMLR, 2016.

[18] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[19] Peyman Mohajerin Esfahani and Daniel Kuhn. Data-driven distributionally robust optimization using the Wasserstein metric: performance guarantees and tractable reformulations. *Mathematical Programming*, 171(1):115–166, 2018.

[20] Ted Moskovitz, Michael Arbel, Ferenc Huszar, and Arthur Gretton. Efficient Wasserstein natural gradients for reinforcement learning. *arXiv preprint arXiv:2010.05380*, 2020.

[21] James R. Munkres. *Topology*. Featured Titles for Topology. Prentice Hall, Incorporated, 2000.

[22] Aldo Pacchiano, Jack Parker-Holder, Yunhao Tang, Anna Choromanska, Krzysztof Choromanski, and Michael I. Jordan. Learning to score behaviors for guided policy optimization. In *International Conference on Machine Learning*, pages 7401–7410. PMLR, 2020.

[23] Gabriel Peyré and Marco Cuturi. Computational optimal transport. *Foundations and Trends in Machine Learning*, 11(5-6):1–257, 2019.

[24] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2005.

[25] Hamed Rahimian and Sanjay Mehrotra. Distributionally robust optimization: A review. *arXiv preprint arXiv:1908.05659*, 2019.

[26] Pierre H. Richemond and Brendan Maginnis. On Wasserstein reinforcement learning and the Fokker-Planck equation. *arXiv preprint arXiv:1712.07185*, 2017.

[27] Ralph Tyrell Rockafellar. *Convex Analysis*. Princeton University Press, 2015.

[28] Walter Rudin. *Real and Complex Analysis, 3rd Ed.* McGraw-Hill, Inc., USA, 1987.

[29] John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel. Trust region policy optimization. In *International Conference on Machine Learning*, pages 1889–1897. PMLR, 2015.

[30] John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *International Conference on Learning Representations*, 2016.

[31] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[32] D. Silver, A. Huang, and C. et al. Maddison. Mastering the game of go with deep neural networks and tree search. *Nature 529, 484–489*, 2016.

[33] Jun Song, Chaoyue Zhao, and Niao He. Efficient Wasserstein and Sinkhorn policy optimization, 2022.

[34] Bahar Taskesen, Soroosh Shafieezadeh-Abadeh, and Daniel Kuhn. Semi-discrete optimal transport: Hardness, regularization and numerical solution. *arXiv preprint arXiv:2103.06263*, 2021.

[35] Dávid Terjék and Diego González-Sánchez. Optimal transport with f-divergence regularization and generalized Sinkhorn algorithm. In *International Conference on Artificial Intelligence and Statistics*, pages 5135–5165. PMLR, 2022.

[36] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012.

[37] Cédric Villani. *Optimal Transport: Old and New*. Springer-Verlag Berlin Heidelberg, 2008.

[38] Patrick Nadeem Ward, Ariella Smofsky, and Avishek Joey Bose. Improving exploration in soft-actor-critic with normalizing flows policies. *arXiv preprint arXiv:1906.02771*, 2019.

[39] Konstantia Xenou, Georgios Chalkiadakis, and Stergos Afantenos. Deep reinforcement learning in strategic board game environments. *Springer International Publishing*, 2019.

[40] Ruiyi Zhang, Changyou Chen, Chunyuan Li, and Lawrence Carin. Policy optimization as wasserstein gradient flows. In *International Conference on Machine Learning*, pages 5737–5746. PMLR, 2018.

[41] Chaoyue Zhao and Yongpei Guan. Data-driven risk-averse stochastic optimization with Wasserstein metric. *Operations Research Letters*, 46(2):262–267, 2018.

[42] Guanjie Zheng, Fuzheng Zhang, Zihan Zheng, Yang Xiang, Nicholas Jing Yuan, Xing Xie, and Zhenhui Li. DRN: A deep reinforcement learning framework for news recommendation. In *World Wide Web Conference*, pages 167–176, 2018.

[43] Yue Zheng. Reinforcement learning and video games. *arXiv preprint arXiv:1909.04751*, 2019.

[44] Zhenpeng Zhou, Xiaocheng Li, and Richard N Zare. Optimizing chemical reactions with deep reinforcement learning. *ACS Central Science*, 3(12):1337–1344, 2017.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] All claims are in line with the paper and its contributions.
   (b) Did you describe the limitations of your work? [Yes] We discuss both benefits and drawbacks of our methodology.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] We do not believe our work entails potential negative societal impacts.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes] We carefully read ethics review guidelines.
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] All assumptions are stated before the corresponding theoretic results.
   (b) Did you include complete proofs of all theoretical results? [Yes] All proofs are included in the supplementary material.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Our code, with instructions, is part of the supplementary material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] All training details are specified in the supplementary material.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Our plots include error bars (mean $\pm$ standard deviation).
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] We comment on the computational time in the supplementary material.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We cited the creators of OpenAI, Mujoco (which we used for experiments), stable baselines (which provided algorithms), and the creators of state-of-the-art algorithms.
   (b) Did you mention the license of the assets? [N/A] The license of the assets is easily verifiable following its reference.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] Our code is included in the supplementary material.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] All assets are publicly available or asked to the authors.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] The data we used does not contain personal information or offensive content.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] We did not use crowdsourcing or conducted research with human subjects.
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] We did not use crowdsourcing or conducted research with human subjects.
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] We did not use crowdsourcing or conducted research with human subjects.

Appendix

A Appendix

A.1 More details on related work

Optimal transport and in particular the Wasserstein distance has found various applications in RL, despite the computational challenge raised by its evaluation.

Optimal transport for PO. In [26], the authors established a connection between policy gradient in Wasserstein trust regions and variational optimal transport, suggesting to solve the Fokker-Planck equations and to use diffusion processes. Concurrently, [40] formulated the PO problem as a gradient descent flow on the space of probability measures using Wasserstein Gradient Flows [2] which is then solved approximately via particles.

BGPG [22]. In [22], the authors proposed to replace the KL divergence trust region from TRPO [29] by a Wasserstein distance penalty in a behavioral space; the proposed approach embeds the policies in a latent behavioral space via a map acting on the trajectories induced by the policies, and leverages the Wasserstein distance to compare two embeddings. To approximate the Wasserstein distance, they exploit the dual formulation of the entropy-regularized Wasserstein distance. Although our alternating procedure in Section 5 may formally resemble the approach of [22] in spirit, our approach (i) builds on the idea of trust regions and (ii) does not consider policy embeddings or entropy regularization. Not surprisingly, our Algorithm 1 is fundamentally different from Algorithms 1 and 3 in [22]. If BGPG can be seen as a variant of TRPO with a Wasserstein regularization, we propose a novel distinct variant of OT-based TRPO fully based on trust regions.

WNPG [20]. While [22] proposed to use the Wasserstein distance as a global penalty to the objective, [20] further suggested to incorporate additional information about the local behavior of policies encapsulated in the so-called Wasserstein Information Matrix. They proposed accordingly WNPG to speed up policy optimization using a Wasserstein natural gradient. The cornerstone of this algorithm is the estimation of a Wasserstein natural gradient stemming from a second-order expansion of the 2-Wasserstein distance between two (parametric) behavioral embedding distributions of two parameterized policies.

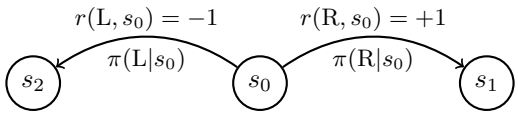
WPO [33]. The closest related work to ours is the recent paper [33] which studied PO with Wasserstein distance and Sinkhorn divergence-based trust regions for discrete (and finite) action spaces. In contrast to [33], the present work addresses the general setting of compact subsets of Polish spaces encompassing the cases of continuous and discrete state-action spaces as particular cases. While we also adopt a duality approach, our level of generality induces many challenges compared to the discrete action space setting (see Remark 2). Moreover, the methods proposed in [33] do not allow to restrict the set of policies to a particular family of distributions, which might lead to very large models. Conversely, the novel algorithm we propose can handle policy parametrization (including direct parametrization). We refer the reader to Section 4 for more detailed comparisons with [33].

A.2 Comments on "mass splitting"

In this section, we elaborate on the concept of "mass splitting". Specifically, we propose two examples showing that splitting the probability mass is in general required to construct optimal policy updates.

Example 1. Let $\mathcal{S} = \{s_1\}$, $\mathcal{A} = \{a_1, a_2\}$, and $\varepsilon \in (0, 1)$. Suppose $c(a_1, a_2) = 1, c(a_1, a_1) = c(a_2, a_2) = 0$, $\pi(\cdot|s_1) = \delta_{a_1}$, $A^\pi(s_1, a_1) = 0$ and $A^\pi(s_1, a_2) = 1$. Clearly, it is optimal to assigns as much probability mass as possible to action a_2; i.e., $\tilde{\pi}(\cdot|s_1) = (1-\varepsilon)\delta_{a_1} + \varepsilon\delta_{a_2}$, which corresponds to $t^* = 1 - \varepsilon$, $\underline{b}_{1,1} = a_1$ and $\underline{b}_{1,2} = a_2$ using the notation of Corollary 3. However, if an optimal policy was to be described by a single transport map (as in [33, Theorem 1, Eq. (5), p. 4]), the only possible solutions would be $\pi_1(\cdot|s_1) = \delta_{a_1}$ and $\pi_2(\cdot|s_1) = \delta_{a_2}$, which are respectively sub-optimal and infeasible.

Example 2. Consider an agent in an initial state s_0 who can move left (L) or right (R). The rewards are $r(s_0, \mathrm{L}) = -1$, and $r(s_0, \mathrm{R}) = +1$, and the task terminates whenever the agents reaches s_1 or at s_2, as shown below.



Consider the initial policy $\pi_0(\mathrm{L}|s_0) = 1$ and $\pi_0(\mathrm{R}|s_0) = 0$, and trust region defined by the binary distance, and let $\varepsilon \in (0, 1)$. Then, Corollary 3 yields the new policy $\pi_1(\mathrm{L}|s) = 1 - \varepsilon$ and $\pi_1(\mathrm{R}|s) = \varepsilon$. Note that this update requires "mass splitting". A second update yields $\pi_2(\mathrm{L}|s) = 1 - 2\varepsilon$ and $\pi_2(\mathrm{R}|s) = 2\varepsilon$, and after $1/\varepsilon$ the routine converges to the optimal policy $\pi^*(\mathrm{L}|s) = 0$ and $\pi^*(\mathrm{R}|s) = 1$. Conversely, the update of [33] (i.e., *without* "mass splitting") leads to either $\pi_1 = \pi_0$, which is suboptimal, or to $\pi_1 = \pi^*$, which violates the trust region constraint.

A.3 Implementation details

In this section, we present the implementation details supporting our experimental results.

A.3.1 OT-TRPO details

We now discuss the details concerning the implementation and tuning of our algorithm. Our code, along with the instructions to set up the environment and run it, is available at `https://gitlab.ethz.ch/lnicolas/ot-trpo`.

Discrete settings. For the discrete settings, we developed a custom training and testing framework. We used TD-learning to estimate the advantage function, with learning rate α and discount factor γ. Namely, we first collect a set $\mathcal{T}$ of trajectory. Then, we initialize $Q(s, a) = 0$ for all $(s, a) \in \mathcal{S} \times \mathcal{A}$ and for all $\tau \in \mathcal{T}$ and every $(s_t, a_t, r_t, s_{tt}, a_{tt}) \in \tau$ we update Q as

$$Q(s_t, a_t) = (1 - \alpha)Q(s_t, a_t) + \alpha(r_t + \gamma Q(s_{tt}, a_{tt})).$$

The transport cost used is the binary distance[4] (i.e., $c(a, a') = 0$ if $a = a'$ and $c(a, a') = 1$ otherwise), and the trust region radius is ε. Every n full environment simulations the policy is updated. The parameters for the different environments are reported in Table 1.

Parameter	*CliffWalking-v0*	*Taxi-v3*
α	0.999999	0.9
γ	0.2	0.5
ε	0.01	0.01
n	1	32

Table 1: Hyperparameters for the discrete environments.

With respect to the computational complexity of the training process, one can assess the following about the main steps of Algorithm 1:

Step 3. The complexity of TD-learning is linear in the number N of samples collected during the rollout of the current policy; i.e., $\mathcal{O}(N)$.

Step 4. This step boils down to `scipy.optimize.minimize_scalar`[5]. The complexity of each function call can be shown to be $\mathcal{O}(|\mathcal{S}||\mathcal{A}|^2)$.

Step 5. The complexity of this step can be shown to be $\mathcal{O}(|\mathcal{S}||\mathcal{A}|^2)$.

[4] As a result, the optimal transport discrepancy corresponds to the total variation between the probability distributions over the actions.

[5] `https://scipy.org`

Continuous settings. For the continuous settings, we developed an agent which can be interfaced with stable baselines [9]. In the benchmark provided, we estimate the advantage function via the GAE, with coefficient `gae_lambda`. We use the policy and value network provided in the `ActorCriticPolicy` class from [9], with network sizes $[64, 64]$ and the activation `activation_fn`. Accordingly, the policy is a Gaussian policy with fixed standard deviation whose mean is expressed by the policy network. That is, we do not use a neural network to approximate the advantage function. We perform stochastic gradient descent in batches of size `batch_size`, every `n_steps` timesteps, for `n_epochs` epochs, with learning rate `learning_rate`, and with maximum gradient norm `max_grad_norm`. The value function loss is multiplied by `vf_coef`. The transport cost is the square Euclidean distance[6], and the trust region radius is ε. Along the lines of the implementation of PPO [31, 9], we introduce a Z-normalization of the advantage estimates if `normalize` is `True`, a state dependent exploration with sample frequency `sde_sample_freq` if this value is not -1, and an orthogonal initialization if `ortho_init` is `True`. The logarithm of the standard deviation of the Gaussian policy network is initialized to `log_std_init`. The parameters for the different environments are reported in Table 2.

Parameter	*MountainCarCont.-v0*	*Hopper-v3*	*Swimmer-v3*	*HalfCheetah-v3*
ε	8.9919	0.4	0.2	0.0548
`n_steps`	512	512	1024	1024
`batch_size`	256	512	64	256
`n_epochs`	10	10	4	20
`learning_rate`	0.0029	0.0008	0.0003	0.0003
`max_grad_norm`	0.7	0.1	0.5	0.8
`activation_fn`	`ReLU`	`Tanh`	`Tanh`	`LeakyReLU`
`vf_coef`	0.6143	0.6349	0.5	0.0070
`gae_lambda`	0.95	0.92	0.98	0.9
`gamma`	0.999	0.995	0.999	0.99
`normalize`	`True`	`True`	`False`	`True`
`sde_sample_freq`	128	16	-1	128
`ortho_init`	`True`	`True`	`True`	`False`
`log_std_init`	0.0	-0.3619	0.0	-2.0291

Table 2: Hyperparameters and network sizes for the continuous environments. The values are rounded to the fourth decimal digit.

With respect to the computational complexity of the training process, one can assess the following about the main steps of Algorithm 1:

Step 3. The complexity of using a GAE is linear in the number N of samples collected during the rollout of the current policy; i.e., $\mathcal{O}(N)$.

Step 4. This step boils down to `scipy.optimize.minimize_scalar`. The complexity of each function call can be shown to be $\mathcal{O}(N)$.

Step 5. This corresponds to a gradient descent step. Therefore, the computational costs are in line with the algorithms to which we compare.

A.3.2 Compared algorithms details

We compare our algorithm to several baselines, which we implemented and tuned as follows:

- The implementation, hyperparameters, and network sizes for TRPO [29], PPO [31], and A2C [17] can be found in stable baselines [9].
- We thank the authors of [22] for privately providing us the code and tuning for BGPG.
- For WNPG [20], we used the code and tuning publicly available at `https://github.com/tedmoskovitz/WNPG`.
- We thank the authors of [33] for pointing us to their implementation (with tuning parameters) of WPO, available at `https://github.com/efficientwpo/EfficientWPO`. This code has

[6]Note that the corresponding optimal transport discrepancy is not a distance (but itself a square distance).

inspired in some parts our training and testing framework. Their primal update has been ported in our code to properly compare the only difference in the algorithms (in discrete settings). We use the same parameters for both WPO and OT-TRPO.

To the best of our knowledge, we used the hyperparameters and network sizes with the best performances for every algorithm in every environment. The results obtained for the baselines considered are comparable or better to the ones found in the literature; e.g., see [33]. We refrained from reporting the results of the algorithms whenever they were not reproducible or not comparable to the others.

For BGPG [22] and WNPG [20], we used `tensorflow.compat` to ensure compatibility of their code (written with TensorFlow 1.x) with our installation of TensorFlow 2.x. The code reproduces the results reported in these works, but the comparisons provided in this paper refer to the updated environments; e.g., *Hopper-v3* instead of *Hopper-v2*.

Finally, to the best of our knowledge, the timescale of the results obtained with the code in `https://github.com/efficientwpo/EfficientWPO` and the results ported in our work might differ. In the former, the evaluation is performed after a certain number of episodes (full environment simulation), rather than timesteps, and the conversion is not clear from the code (as every episode might differ in length). We instead adhere to the standard provided in [9].

A.4 Ablation study

We investigate how different (i) trust region radii ε and (ii) transportation costs, affect the learning performances in the *Taxi-v3* environment. These effects are shown in Fig. 2.

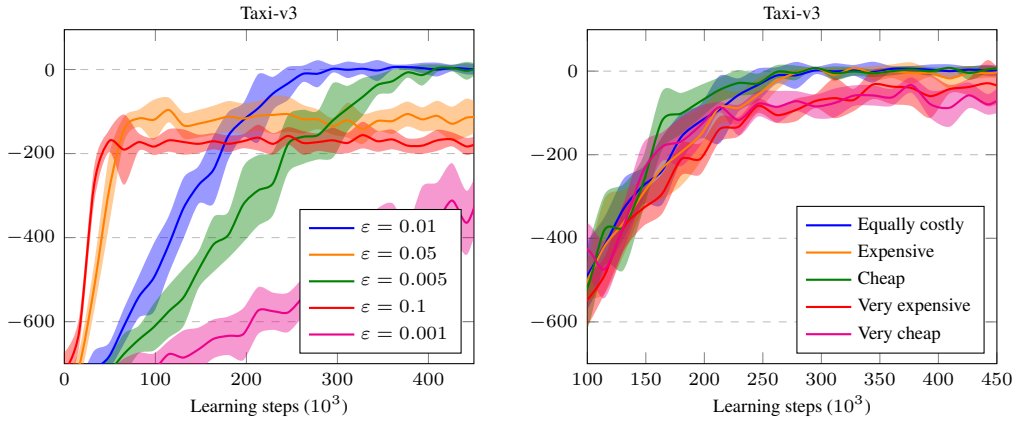


Figure 2: Ablation study. The shaded area represents the mean $\pm$ the standard deviation across 10 independent runs. Every policy evaluation in each run is averaged over 10 sampled trajectories.

Different trust region radii ε. As expected, larger radii lead to significantly steeper learning curves, at the expense of converging to suboptimal policies. Conversely, a smaller radius results in slower learning trajectories. Our trust region radius $\varepsilon = 0.01$ seems to balance the two effects.

Different transportation costs. The *Taxi-v3* environment is characterized by two radically different set of actions: $\text{Move} = \{\text{Up}, \text{Left}, \text{Right}, \text{Down}\}$ and $\text{Passenger} = \{\text{PickUp}, \text{DropOff}\}$. We use this insight to provide a simple example of how transport costs can affect the geometry of the action space and in turn affect the learning process. Specifically, we consider different cost functions. Let $m, m' \in \text{Move}, p, p' \in \text{Passenger}$. Consider now the following transport costs (for all of them we set $c(x, x) = 0 \; \forall x \in \text{Move} \cup \text{Passenger}$):

Equally costly. In this cost term, $c(m, m') = c(m', m) = c(p, p') = c(p', p) = c(m, p) = c(p, m)$. This is also a distance.

Cheap [Expensive]. In this cost term, $c(m, m') = c(m', m)$, $c(p, p') = c(p', p)$, but $c(m, p) > c(p, m)$ [$c(m, p) < c(p, m)$]. In particular, $c(p, m) = c(m, p) - 0.2$ [$c(p, m) = c(m, p) + 0.2$]. Since it is not symmetric, this is not a distance.

Very cheap [expensive]. In this cost term, $c(m, m') = c(m', m)$, $c(p, p') = c(p', p)$, but $c(m, p) \gg c(p, m)$ [$c(m, p) \ll c(p, m)$]. In particular, $c(p, m) = c(m, p) - 0.5$ [$c(p, m) = c(m, p) + 0.5$]. Since it is not symmetric, this is not a distance.

Notice that a "local minimum" in the *Taxi-v3* environment is to never play actions in the $\mathrm{Passenger}$ set: such a strategy yields a score of -200. Indeed, performing a $\mathrm{DropOff}$ leads to large penalty when not at the right location. As a result, carrying a passenger is "risky". However, these are also the actions that are required to cross the -200 bar and learn the optimal policy. We can interpret the aforementioned costs as follows:

Equally costly. This cost function ignores the distinction between the two classes of actions: moving from one to the other is equally costly.

Cheap [Expensive]. Moving mass towards "risky" actions is incentivized [penalized]. This cost function shapes the landscape of the action space so that there is a downhill [uphill] from Move to $\mathrm{Passenger}$, and withing these sets the geometry is analogous to the *Equally costly* one.

Very cheap [expensive]. Analogous to *Cheap* [*Expensive*].

We observe that with *Very cheap* [*expensive*] the agent learns apparently worse than with the others. Intuitively, this is due to the fact that the *Very cheap* cost incentivizes too much the agent to drop off the passenger; as a result, it completes the task, but it also incurs many illegal $\mathrm{DropOff}$s, which yield the penalty. Oppositely, the *Very expensive* cost is too conservative, and the agent learns to perform too few $\mathrm{DropOff}$s, being unable to complete the task correctly consistently.

On the other hand, the *Cheap* [*Expensive*] cost trains the agent slightly faster [slower] than the *Equally costly* option. Namely, these costs balance in a different way the trade-off between exploration and exploitation.

This example highlights the impact of the geometry of the action space, which, in turn, is another tuning parameter available to practitioners. For instance, one could consider transport costs which embed an exploration penalties (e.g., moving mass from actions that are already rarely used or to others that are almost always used can be penalized), or physical considerations (e.g., an expert supervisor might know that certain actions are most likely the correct ones given certain configurations of the agent).

Sensitivity to the advantage estimation Both the dual problem (D) and the "regularized" policy loss rely on the accuracy of the advantage function estimation, as well as on the solution of a maximization problem over a continuous space, which is approximated by finitely many evaluated points. That is, when training on continuous actions spaces, we rely on the approximation

$$\max_{a' \in \mathcal{A}} \{A^\pi(s, a') - \lambda c(a, a')\} \approx \max_{a' \in \mathcal{A}'} \{\hat{A^\pi}(s, a') - \lambda c(a, a')\}, \tag{13}$$

where $\mathcal{A}' \subset \mathcal{A}$ is a (possibly state-dependent) finite set of actions. While smoothness of the objective guarantees convergence as $|\mathcal{A}'| \to \infty$ with uniformly chosen actions in $\mathcal{A}$, we empirically investigate the sensitivity of the method to various practical advantage estimation schemes.

In the experimental results so far presented, we use a GAE based estimate of the advantage function. In continuous action spaces, we explore a neural network approximation for the advantage function as well. Overall, this approach does not perform comparably to the single sample estimation of the objective with GAE. The training performances in the *Hopper-v3* environment for different cardinalities of $\mathcal{A}'$ are juxtaposed in Fig. 3, and the trained agents scores are summarized in Table 3. In Fig. 4 we compare the performances of the neural network approximation of the advantage function with the GAE in the environments *HalfCheetah-v3*, *Hopper-v3*, and *MountainCarContinuous-v0*. The number of sampled actions is $|\mathcal{A}'| = 4$, $|\mathcal{A}'| = 2$, and $|\mathcal{A}'| = 16$, respectively.

In general, we observe that (i) simultaneous training of an advantage network and the OT-TRPO policy update could be unstable, and (ii) in the case of convergence, OT-TRPO with neural advantage function approximation converges to a suboptimal policy. We hypothesize that the OT-TRPO objective is sensitive to biased value estimates in the neighborhood of the mean of the Gaussian policy network. It will be subject of future research if and to what extent these results can be improved, for instance via normalizing flow policy parametrizations [38].

To support our observation regarding the failure mode of biased advantage estimates, we provide an ablation study for different λ parameters for the GAE. The training curves are shown in Fig. 5.

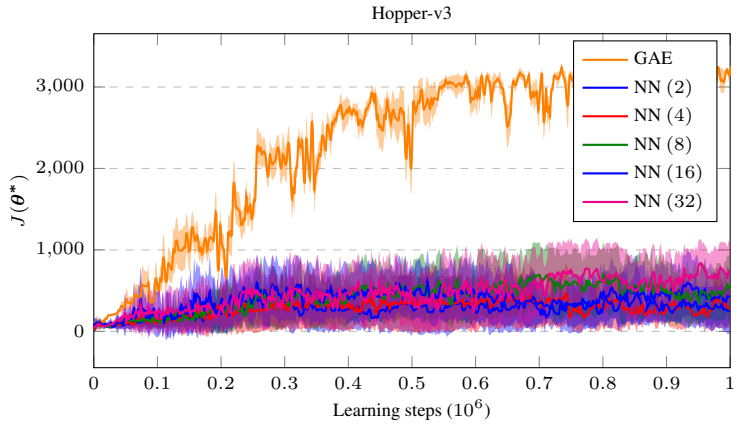


Figure 3: Ablation study. The advantage function is estimated either via GAE or via a neural network approximation "NN ($|\mathcal{A}'|$)". The shaded area represents the mean $\pm$ the standard deviation across 10 independent runs. Every policy evaluation in each run is averaged over 10 sampled trajectories.

Advantage estimator method	Trained agent scores
GAE	2939 ± 162
NN (2)	327 ± 96
NN (4)	305 ± 72
NN (8)	452 ± 136
NN (16)	387 ± 94
NN (32)	497 ± 150

Table 3: Trained agents scores when the advantage function is estimated either via GAE or via a neural network approximation "NN ($|\mathcal{A}'|$)".

It is known that for $\lambda = 1$, the GAE is equivalent to an unbiased Monte Carlo estimate of the advantage [30]. The observation that state-of-the-art performances are achieved with a `gae_lambda` hyperparameter very close to 1 (i.e., an unbiased Monte Carlo estimate) supports the idea that approximating the maximization with a biased advantage estimator is prone to instability.

A.5 More details on the policy update

We shall now discuss how the implementation affects the exactness guarantees of Theorem 1. Indeed, albeit theoretically the algorithm presented guarantees an optimal policy update at every step, the practical implementation might resort to approximations. We consider as running example the simple setting in Example 2, where all the computations can be carried out analytically. We consider a generic current policy with $\pi_\theta(L|s_0) = \theta, \pi_\theta(R|s_0) = 1 - \theta$.

For this example, we can pedagogically solve both (P) and (D). The Q-function is $Q^{\pi_\theta}(s_0, L) = -1, Q^{\pi_\theta}(s_0, R) = 1$. Then, the value function at s_0 can be expressed in terms of the current policy as

$$V^{\pi_\theta}(s_0) = \theta Q^{\pi_\theta}(s_0, L) + (1 - \theta) Q^{\pi_\theta}(s_0, R) = 1 - 2\theta,$$

and the advantage reads

$$A^{\pi_\theta}(s_0, L) = Q^{\pi_\theta}(s_0, L) - V^{\pi_\theta}(s_0) = 2(\theta - 1), \quad A^{\pi_\theta}(s_0, R) = Q^{\pi_\theta}(s_0, R) - V^{\pi}(s_0) = 2\theta.$$

Trivially, the visitation frequency is fully described by $\rho_{\pi_\theta}(s_0) = 1$; we can ignore the terminal states as they do not affect the problem. We consider the binary transportation cost $c(L, R) = c(R, L) = 1, c(L, L) = c(R, R) = 0$, and we first solve (P). By direct inspection, it is easy to see that there are two cases. Let the new policy be $\pi_{\tilde{\theta}}(L|s_0) = \tilde{\theta}, \pi_{\tilde{\theta}}(R|s_0) = 1 - \tilde{\theta}$. If $\theta < \varepsilon$, then the solution is to move all the mass from action L to action R: $\tilde{\theta} = 0$. Otherwise, we can move at most ε mass, and this is the optimal solution: $\tilde{\theta} = \theta - \varepsilon$.

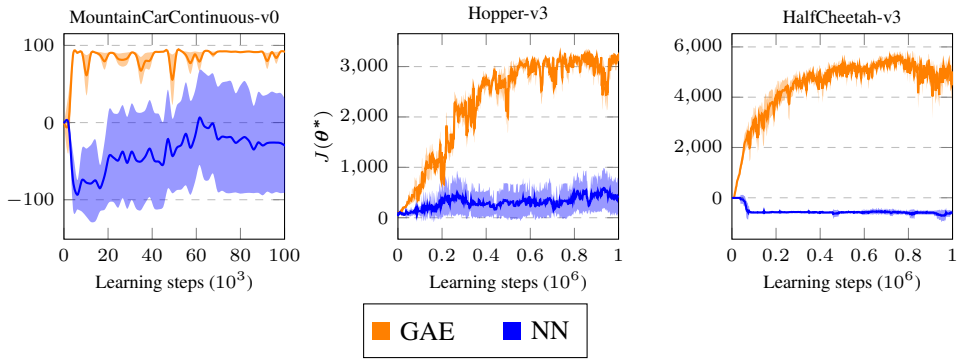


Figure 4: Ablation study. The advantage function is estimated either via GAE or via a neural network approximation. The shaded area represents the mean $\pm$ the standard deviation across 10 independent runs. Every policy evaluation in each run is averaged over 10 sampled trajectories.

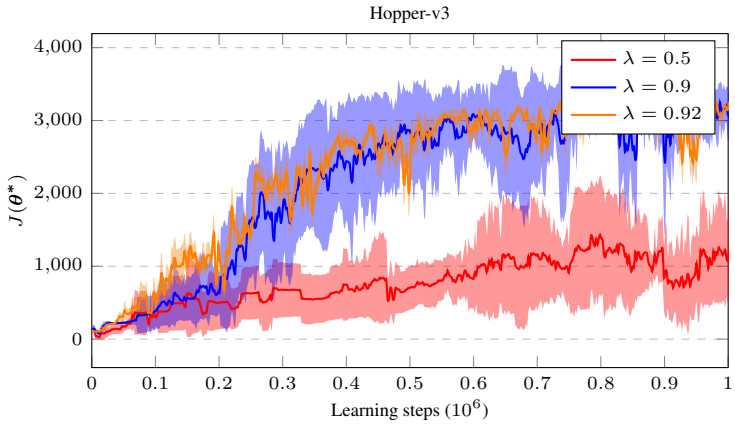


Figure 5: Ablation study on the λ hyperparameter of the GAE. The shaded area represents the mean $\pm$ the standard deviation across 10 independent runs. Every policy evaluation in each run is averaged over 10 sampled trajectories.

Second, we solve (D):

$$
\begin{aligned}
\lambda^* &= \arg\min_{\lambda \geq 0} \lambda\varepsilon + \int_{\mathcal{S}} \int_{\mathcal{A}} \max_{a' \in \mathcal{A}} \{A^{\pi_\theta}(s,a') - \lambda c(a,a')\} \, \mathrm{d}\pi_\theta(a|s) \mathrm{d}\rho_\pi(s) \\
&= \arg\min_{\lambda \geq 0} \lambda\varepsilon + \theta \max\{A^{\pi_\theta}(s_0,L), A^{\pi_\theta}(s_0,R) - \lambda\} + (1-\theta)\max\{A^{\pi_\theta}(s_0,L) - \lambda, A^{\pi_\theta}(s_0,R)\} \\
&= \arg\min_{\lambda \geq 0} \lambda\varepsilon + \theta \underbrace{\max\{2(\theta-1), 2\theta - \lambda\}}_{=2\theta - \min\{2,\lambda\}} + (1-\theta) \underbrace{\max\{2(\theta-1) - \lambda, 2\theta\}}_{=2\theta} \\
&= \arg\min_{\lambda \geq 0} \lambda\varepsilon - \theta \min\{2, \lambda\},
\end{aligned}
$$

which yields $\lambda^* = 0$ if $\theta < \varepsilon$, $\lambda^* = 2$ if $\theta \geq \varepsilon$. As expected, λ^* is not zero if the constraint is active, which is the case every time there is more mass in $\pi(L|s_0)$ than what we are allowed to move. In practice, we can solve this optimization problem to global optimality (with sufficient numerical precision). In discrete settings, the numerical approximations introduced by the solver are the only source of approximations.

In the considered example, with finite state-action spaces, one should follow the first approach discussed for the parameter update; namely, one should update θ according to (10) and (9). For the sake of providing insights and shading light on both the inner working of the proposed algorithm and its limitations, we now study all the proposed options for step 5 in Algorithm 1.

Direct parametrization (finite spaces) We distinguish two cases:

$\theta > \varepsilon$. The constraint is active: $\lambda^* = 2$. We obtain $\mathcal{D}_{\lambda^*}(s, L) = \{L, R\}$ and $\mathcal{D}_{\lambda^*}(s, R) = \{R\}$. The minimizers are not unique: we need "mass splitting". In particular, $\underline{T}_{\lambda^*}(s_0, L) = L$ (the "closest" minimizer is the one that not require any mass displacement), $\overline{T}_{\lambda^*}(s_0, L) = R$, and $\underline{T}_{\lambda^*}(s_0, R) = \overline{T}_{\lambda^*}(s_0, R) = R$. Finally, t^* is selected to "activate" the constraint. In this case, $t^* = \varepsilon$, and we obtain $\tilde{\pi}(L|s_0) = \theta - \varepsilon =: \tilde{\theta}$, as expected.

$\theta \leq \varepsilon$. In this case, $\lambda^* = 0$. Then we have $\mathcal{D}_{\lambda^*}(s, L) = \{R\}$ and $\mathcal{D}_{\lambda^*}(s, R) = \{R\}$. That is, the minimizers are unique, the unique transport map $\underline{T}_{\lambda^*}(s_0, \cdot) = \overline{T}_{\lambda^*}(s_0, \cdot) = T$ is the constant map $T(L) = T(R) = R$ and we do not need "mass splitting". The new policy is then trivially $\tilde{\pi}(L|s_0) = T_{\#}\pi(L|s_0) = 0$; that is, $\tilde{\theta} = 0$, again as expected.

Direct parametrization via policy network (continuous states, discrete actions) In this approach, we build on top of the result of the previous one, and we perform gradient descent on the loss $L(\theta) = \sum_{s \in \hat{\mathcal{S}}} \rho_{\pi_{\theta_t}}(s) \left\| \pi_\theta(\cdot|s) - \pi^*_{\theta_t}(\cdot|s) \right\|^2$ to steer π_θ. In particular, the minimization of L steers the policy network towards the optimal policy. Hence, if we were to perform gradient descent until convergence, the policy update would be exact up to numerical precision.

Arbitrary policy parametrization (continuous states, continuous actions) Armed with λ^*, we construct the loss function

$$L(\tilde{\theta}) = \sum_{s \in \mathcal{S}} \int_{\mathcal{A}} \max_{a' \in \mathcal{A}} \{A^{\pi_\theta}(s, a') - \lambda^* c(a, a')\} \mathrm{d}\pi_{\tilde{\theta}}(a|s) \rho_{\pi_\theta}(s).$$

In particular, $\tilde{\theta}$ affects only $\pi_{\tilde{\theta}}$. The intuition behind this loss function is to increase the probability mass where the regularized advantage function has its maxima. However, in general, the solutions are multiple, while the gradient descent will possibly converge to one. As such, one limitation of this cost function is related with the "mass splitting" issue. That is, if we were to perform gradient descent until convergence, the policy update might violate the trust-region constraint. Empirically, the proposed loss function performs well, and allows the deployment of optimal transport trust-region methods in continuous spaces. Moreover, the gradient descent is not iterated until convergence in practice. Nonetheless, the arising optimization procedure deserves further study in terms of convergence and optimality guarantees, but our duality results provide an intuition on the method. Future work will focus on understanding how to implicitly describe transport *plans*, rather than transport maps.

In the remainder of the discussion, we study the gradient for the running example, and we validate the intuition above:

$$\begin{aligned} \nabla_{\tilde{\theta}} L(\tilde{\theta}) &= \nabla_{\tilde{\theta}} \left[\tilde{\theta} \max\{2(\theta - 1), 2\theta - \lambda^*\} + (1 - \tilde{\theta}) 2\theta \right] \\ &= \max\{2(\theta - 1), 2\theta - \lambda^*\} - 2\theta \\ &= -\min\{2, \lambda^*\} \end{aligned}$$

We consider the cases $\theta > \varepsilon$ and $\theta \leq \varepsilon$ separately:

$\theta > \varepsilon$. The gradient reads $\nabla_{\tilde{\theta}} L(\tilde{\theta}) = -2$: we are steering θ to 0. Namely, the loss is embedding the transport map $\overline{T}_{\lambda^*}(s_0, L) = R$, $\overline{T}_{\lambda^*}(s_0, R) = R$. Following the gradient corresponds to interpolating between $\pi_\theta(\cdot|s_0)$ and $\overline{T}_{\lambda^*}(s_0, \cdot)_{\#}\pi_\theta(\cdot|s_0)$. However, as previously discussed, the optimal solution requires "mass splitting": if we perform gradient descent until convergence, the policy update yields $\tilde{\theta} = 0$, which violates the trust-region constraint.

$\theta \geq \varepsilon$. The gradient vanishes: $\nabla_{\tilde{\theta}} L(\tilde{\theta}) = -\lambda^* = 0$. This issue is related with the policy parametrization. The issue can be addressed with a decreasing trust-region radius $\varepsilon \to 0$. This highlights a potential drawback of the proposed practical implementation: albeit our theoretical results are valid and parametrization-independent, the choice of the network architecture affects the learning process. Aside from the possibility of a vanishing gradient, it is possible that the optimal policy is not contained in the family described by the parametrization.

A.6 Further details on the experimental results

A.6.1 Additional numerical details

In addition to the experimental results of Section 6, we provide the numerical results of the performance comparison among our algorithm and the baseline methods in Table 4. In particular, we average the expected scores obtained in the last 10% of the training episodes.

Algorithm	*CliffWalking-v0*	*Taxi-v3*	*MountainCarCont.-v0*
TRPO [29]	-568 ± 0	-51 ± 11	75 ± 0
PPO [31]	-5001 ± 0	-200 ± 0	-8 ± 13
A2C [17]	-512 ± 0	-200 ± 0	83 ± 0
WPO [33]	-23 ± 2	-65 ± 10	/
BGPG [22]	/	/	90 ± 3
WNPG [20]	/	/	88 ± 5
OT-TRPO	-14 ± 0	3 ± 3	88 ± 6

Algorithm	*Hopper-v3*	*Swimmer-v3*	*HalfCheetah-v3*
TRPO [29]	3130 ± 230	239 ± 4	2561 ± 114
PPO [31]	1388 ± 127	219 ± 2	5078 ± 126
A2C [17]	667 ± 73	24 ± 8	633 ± 70
BGPG [22]	1720 ± 59	37 ± 0	1309 ± 17
WNPG [20]	1998 ± 50	40 ± 0	985 ± 43
OT-TRPO	2939 ± 162	359 ± 2	4818 ± 4

Table 4: Averaged scores over last 10% episodes of the training process.

Moreover, in Table 5 we report the training time in the Mujoco [36] environments, which are the most computationally demanding. The values in Table 5 have been collected under comparable conditions and similar hardware. While the absolute values here provide little insight, the relative comparisons highlight that our algorithm has computational performances only slightly slower than TRPO [29], PPO [31], and A2C [17]. Instead, other methods levereging optiamal transport and, in particular, the Wasserstein distance (i.e., BGPG [22] and WNPG [20]) require a considerably longer training time.

Algorithm	*Hopper-v3*	*Swimmer-v3*	*HalfCheetah-v3*
TRPO [29]	5606 ± 2542	7882 ± 963	6719 ± 3517
PPO [31]	9967 ± 6702	7875 ± 1667	10088 ± 9966
A2C [17]	3190 ± 1210	7876 ± 2332	4295 ± 172
BGPG [22]	15009 ± 64	31402 ± 1491	50335 ± 112
WNPG [20]	29941 ± 12531	44774 ± 731	29620 ± 8687
OT-TRPO	8031 ± 931	16903 ± 2933	11276 ± 1410

Table 5: Training time in seconds of the different algorithms in the Mujoco [36] environments.

A.6.2 Some considerations on convergence speed

Finally, in some environments (e.g., *Taxi-v3*, *Hopper-v3*, and *Swimmer-v3*), TRPO seems to learn faster than the proposed algorithm (albeit not converging to a better solution). This is closely related to the choice of trust-region. With OT-TRPO, moving the "probability mass" is done at the cost $c(a_1, a_2)$. In TRPO, there is no notion of transport cost in the action space: all the actions are at the same "distance". That is, they differ only based on the probability of using them when at a state $s \in \mathcal{S}$;

namely $\log(\pi(a_1|s)/\pi(a_2|s))$. Whenever $c(a_1,a_2) > \log(\pi(a_1|s)/\pi(a_2|s))$, the convergence is faster with TRPO compared to OT-TRPO. Clearly, one could design a transport cost that speeds up the convergence (a "smaller" one), at the price of a potentially less stable and robust behavior of the algorithm. The study of the "optimal" choice of transport cost is indeed an interesting question for future research. As an example, in the *Taxi-v3* environment there are 6 actions. The proposed method with a binary distance incurs a cost of 1 to bring a state from a uniform probability distribution over the actions to a deterministic one (which for most states is the case in the optimal policy). Instead, TRPO considers such policies only $\log(6) < 1$ away. Similar (but more complicated) calculations can be carried out for the Mujoco environments. Another reason can be found in the lack of symmetry of the KL divergence. Such asymmetry can "enforce" a direction of exploration: going back to a previously explored point in the policy space might be more (or less) expensive. Oppositely, an optimal transport cost can be chosen to be symmetric (e.g., the Wasserstein distance and the costs used for our experiments). When combined with errors in the estimation of the advantage function, it can result in oscillations in the policy space, which might slow down the convergence.

B Proofs

B.1 Proof of Theorem 1

We start with weak duality:

Proposition 5 (Weak Duality)**.** *Weak duality holds. Namely,* (P) $\leq$ (D).

Proof. By minimax we have

$$\begin{aligned}
\text{(P)} &= \sup_{\tilde{\pi}\in\Pi}\inf_{\lambda\geq 0}\left\{\int_{\mathcal{S}}\int_{\mathcal{A}} A^{\pi}(s,a)\mathrm{d}\tilde{\pi}(a|s)d\rho_{\pi}(s) + \lambda\left(\varepsilon - \int_{\mathcal{S}} C(\pi(\cdot|s),\tilde{\pi}(\cdot|s))\mathrm{d}\rho(s)\right)\right\}\\
&\leq \inf_{\lambda\geq 0}\left\{\lambda\varepsilon + \sup_{\tilde{\pi}\in\Pi}\left\{\int_{\mathcal{S}}\int_{\mathcal{A}} A^{\pi}(s,a)\mathrm{d}\tilde{\pi}(a|s) - \lambda C(\pi(\cdot|s),\tilde{\pi}(\cdot|s))\mathrm{d}\rho_{\pi}(s)\right\}\right\}\\
&\leq \inf_{\lambda\geq 0}\left\{\lambda\varepsilon + \int_{\mathcal{S}}\underbrace{\sup_{\tilde{\pi}(\cdot|s)\in\Pi}\left\{\int_{\mathcal{A}} A^{\pi}(s,a)\mathrm{d}\tilde{\pi}(a|s) - \lambda C(\pi(\cdot|s),\tilde{\pi}(\cdot|s))\right\}}_{(\heartsuit)}\mathrm{d}\rho_{\pi}(s)\right\}.
\end{aligned}$$

Second, for all $s\in\mathcal{S}$, Kantorovich duality [2, Theorem 6.1.1] gives

$$(\heartsuit) = \sup_{\pi(\cdot|s)\in\Pi}\left\{\int_{\mathcal{A}} A^{\pi}(s,a)\mathrm{d}\tilde{\pi}(a|s) - \lambda\sup_{\phi+\psi\leq c}\left\{\int_{\mathcal{A}}\phi(a)\mathrm{d}\pi(a|s) + \int_{\mathcal{A}}\psi(a)\mathrm{d}\tilde{\pi}(a|s)\right\}\right\}.$$

Without loss of generality, we assume $\lambda > 0$; else, the result is straightforward. Then, we choose

$$\psi(\cdot) = A^{\pi}(s,\cdot)/\lambda \quad \text{and} \quad \phi(\cdot) = \inf_{a'\in\mathcal{A}}\left\{c(\cdot,a') - \psi(a')\right\}.$$

By Assumptions 1 and 2 ϕ,ψ are continuous on a compact space. Hence, they are also bounded [21, Theorem 27.4]. Moreover,

$$\phi(a_1) + \psi(a_2) = \inf_{a'\in\mathcal{A}}\{c(a_1,a') - \psi(a')\} + \psi(a_2) \leq c(a_1,a_2) - \psi(a_2) + \psi(a_2) \leq c(a_1,a_2).$$

Thus, (ϕ,ψ) is a valid, possibly sub-optimal, choices for the supremum. Overall, we get the upper bound

$$\begin{aligned}
(\heartsuit) &\leq \sup_{\pi(\cdot|s)\in\Pi}\left\{\int_{\mathcal{A}} -\inf_{a'\in\mathcal{A}}\left\{\lambda c(a,a') - A^{\pi}(s,a')\right\}\mathrm{d}\pi(a|s)\right\}\\
&= \int_{\mathcal{A}}\sup_{a'\in\mathcal{A}}\left\{A^{\pi}(s,a') - \lambda c(a,a')\right\}\mathrm{d}\pi(a|s).
\end{aligned}$$

Hence, we obtain

$$\text{(P)} \leq \inf_{\lambda\geq 0}\left\{\lambda\varepsilon + \int_{\mathcal{S}}\int_{\mathcal{A}}\sup_{a'\in\mathcal{A}}\left\{A^{\pi}(s,a') - \lambda c(a,a')\right\}\mathrm{d}\pi(a|s)\mathrm{d}\rho_{\pi}(s)\right\} = \text{(D)}. \qquad \square$$

The proof of strong duality (i.e., equality) is more delicate, and requires some preliminary results. First, we recall the definition the regularization operator Φ_λ, which represents intuitively a "regularized" advantage function:

$$\begin{aligned}\Phi_\lambda : \mathcal{S} \times \mathcal{A} &\to \mathbb{R} \\ (s,a) &\mapsto \sup_{a' \in \mathcal{A}} \left\{ A^\pi(s,a') - \lambda c(a,a') \right\}.\end{aligned}$$

The regularization operator Φ_λ is well defined by Assumptions 1 and 2. Indeed, for all $\lambda \in \mathbb{R}$, $s \in \mathcal{S}$, $a \in \mathcal{A}$, $a' \mapsto A^\pi(s,a') - \lambda c(a,a')$ is a continuous function on a compact space and thus it attains its maximum value (e.g., see [21, Theorem 27.4]). Accordingly, we recall the set of maximizers

$$\mathcal{D}_\lambda(s,a) \coloneqq \arg\max_{a' \in \mathcal{A}} \left\{ A^\pi(s,a') - \lambda c(a,a') \right\}.$$

Since minimizers are generally *not* unique, $\mathcal{D}_\lambda(s,a)$ is indeed a set. In fact, it is also closed:

Lemma 6. *For all $\lambda \in \mathbb{R}$, $s \in \mathcal{S}$, $a \in \mathcal{A}$, the set $\mathcal{D}_\lambda(s,a) \subset \mathcal{A}$ is non-empty and closed (and thus measurable).*

Proof. Since $a' \mapsto A^\pi(s,a') - \lambda c(a,a')$ is a continuous function on a compact space and thus it attains its maximum value (e.g., see [21, Theorem 27.4]), which directly implies that $\mathcal{D}_\lambda(s,a)$ is non-empty. To prove that it is closed, it suffices to show that $\mathcal{D}_\lambda(s,a)$ contains all its limit points (e.g., see [21, Corollary 17.7]). Let $(a_n)_{n\in\mathbb{N}}, a_n \in \mathcal{D}_\lambda(s,a)$ be a sequence converging to $\bar{a} \in \mathcal{A}$. Let $h^\lambda_{s,a} : \mathcal{A} \to \mathbb{R}$ be defined as $h^\lambda_s(a) = A^\pi(s,a') - \lambda c(a,a')$. Since $a_n \in \mathcal{D}_\lambda(s,a)$, $h^\lambda_{s,a}(a_n) = \Phi_\lambda(s,a)$ for all $n \in \mathbb{N}$. Moreover, A^π is continuous by Assumption 2, and thus so is h. By continuity,

$$h^\lambda_{s,a}(\bar{a}) = \lim_{n\to\infty} h^\lambda_{s,a}(a_n) = \Phi_\lambda(s,a),$$

and so $\bar{a} \in \mathcal{D}_\lambda(s,a)$. Measurability of $\mathcal{D}_\lambda(s,a)$ follows from closedness and $\pi(\cdot|s)$ being a Borel probability measure. □

Among all possible minimizers, the *closest* and *furthest apart* will play an important role in the proof of strong duality. Thus, we define

$$\overline{D}_\lambda(s,a) \coloneqq \max_{a' \in \mathcal{D}_\lambda(s,a)} c(a,a'), \qquad \underline{D}_\lambda(s,a) \coloneqq \min_{a' \in \mathcal{D}_\lambda(s,a)} c(a,a'),$$

where the $\max$ and $\min$ are well defined by Lemma 6, since $c(a,\cdot)$ is a continuous function for all $a \in \mathcal{A}$ and $\mathcal{D}_\lambda(s,a)$ is compact (being the closed subset of a compact space, [21, Theorem 26.2]).

These definitions allow us to study the properties of Φ more in detail:

Lemma 7 (Properties of Φ)**.** *The regularization operator Φ has the following properties:*

1. *It is uniformly (in s, a, and λ) lower and upper bounded by some constant.*
2. *It is lower semi-continuous, non-increasing, and convex in λ.*
3. *For all $\lambda > 0$ the left and right derivatives are*

$$\frac{\partial \Phi_\lambda(s,a)}{\partial \lambda -} = \overline{D}_\lambda(s,a), \qquad \frac{\partial \Phi_\lambda(s,a)}{\partial \lambda +} = \underline{D}_\lambda(s,a).$$

4. *There are measurable selections*

$$\overline{T}_\lambda(s,a) \in \arg\max_{a' \in \mathcal{D}_\lambda(s,a)} c(a,a'), \qquad \underline{T}_\lambda(s,a) \in \arg\min_{a' \in \mathcal{D}_\lambda(s,a)} c(a,a').$$

Proof. The proof follows closely [8, Lemma 3]. Specifically, we prove the steps separately:

1. Since A^π is continuous on a compact space, its absolute value is bounded by some C (e.g., see [21, Theorem 27.4]). For the upper bound: Φ_λ results from an infimum, $\lambda c(a,a') \geq 0$, and $a' = a$ is always a valid choice, yielding $\Phi_\lambda(s,a) \geq A^\pi(s,a) - \lambda c(a,a) \geq -C$ For the lower bound; since transport costs are non-negative, we also have $\Phi_\lambda(s,a) \leq \sup_{a' \in \mathcal{A}} A^\pi(s,a') \leq C$. Thus, $|\Phi_\lambda(s,a)| \leq C$ for all $\lambda \geq 0$, $s \in \mathcal{S}$, and $a \in \mathcal{A}$.
2. For all $s \in \mathcal{S}, a \in \mathcal{A}$, $\lambda \mapsto \Phi_\lambda(s,a)$ is the supremum of non-increasing and affine functions. As a result, it is non-increasing, convex and lower semi-continuous in λ.

3. The proof boils down to upper semi-continuity and lower semi-continuity of $\lambda \mapsto \overline{D}_\lambda(s,a)$ and $\lambda \mapsto \underline{D}_\lambda(s,a)$, respectively, whose proof is identical to the setting of DRO since both s and a are frozen; see [8, Corollary 1] and replace the distance d^p with c. Then, the results follows from [8, Lemma 4.iii], since, by definition, the growth rate is 0 for compact spaces and $\Phi > -\infty$.
4. The set of minimizers $\mathcal{D}_\lambda(s,a)$ is non-empty and closed by Lemma 6. Since it is a closed subset of a compact space, it is also compact [21, Theorem 26.2]. Moreover, $c(a,\cdot)$ is continuous for all $a \in \mathcal{A}$ and $c(\cdot,a')$ is continuous (and thus measurable) for all $a' \in \mathcal{A}$, thus the result follows from [1, Theorem 18.19]. □

We are now ready to prove strong duality:

Proof of Theorem 1. In view of Proposition 5, we only need to prove (P) $\geq$ (D). We split the proof in two steps. First, we show existence of an optimal multiplier $\lambda^* \in [0,\infty)$. Second, we study the case $\lambda^* > 0$, whereas $\lambda^* = 0$ is straightforward from direct inspection of (P) and (D).

Let $h : \mathbb{R}_{\geq 0} \to \mathbb{R}$ be defined as

$$h(\lambda) = \lambda\varepsilon + \int_{\mathcal{S}}\int_{\mathcal{A}} \Phi_\lambda(s,a)\mathrm{d}\pi(a|s)d\rho_\pi(s).$$

By Lemma 7.1-2, h is convex and lower semi-continuous in λ. Moreover, h is coercive:

$$\begin{aligned}\liminf_{\lambda\to\infty} h(\lambda) &= \liminf_{\lambda\to\infty} \lambda\varepsilon + \int_{\mathcal{S}}\int_{\mathcal{A}} \Phi_\lambda(s,a)\mathrm{d}\pi(a|s)d\rho_\pi(s)\\ &\geq \lim_{\lambda\to\infty} \lambda\varepsilon + \int_{\mathcal{S}}\int_{\mathcal{A}} A^\pi(s,a)\mathrm{d}\pi(s,a)\mathrm{d}\rho_\pi(s)\\ &= +\infty,\end{aligned}$$

since $\Phi_\lambda(s,a) \geq A^\pi(s,a)$ for all $a \in \mathcal{A}$ and $s \in \mathcal{S}$. Thus, there exists $\lambda^* \in \arg\min_{\lambda\geq 0} h(\lambda)$. Indeed, by definition of limit, there exists $\bar{\lambda}$ such that $h(\lambda) > h(0)$ for all $\lambda > \bar{\lambda}$. Hence, when looking for a minimum, we can restrict ourselves to $\lambda \in [0,\bar{\lambda}]$. Since h is lower semi-continuous on a compact space [21, Corollary 27.2], it attains its minimum at some $\lambda^* \in [0,\bar{\lambda}]$ [21, Theorem 27.4].

Assume now that $\lambda^* > 0$. We study the first-order optimality conditions on h. Since h is generally non-smooth (due to the maximum in Φ_λ), we express them via the left and right derivatives, well-defined for convex functions [27, Theorem 23.1], as $\frac{\partial h(\lambda)}{\partial \lambda -} \leq 0$ and $\frac{\partial h(\lambda)}{\partial \lambda +} \geq 0$.

Hence, we have

$$\begin{aligned}\frac{\partial h(\lambda)}{\partial \lambda -} \leq 0 \iff \varepsilon &\leq \frac{\partial}{\partial \lambda -}\left(\int_{\mathcal{S}}\int_{\mathcal{A}} \Phi_{\lambda^*}(s,a)\mathrm{d}\pi(a|s)\mathrm{d}\rho_\pi(s)\right)\\ &\overset{\heartsuit}{=} \int_{\mathcal{S}}\int_{\mathcal{A}} \frac{\partial}{\partial \lambda -}\Phi_{\lambda^*}(s,a)\mathrm{d}\pi(a|s)\mathrm{d}\rho_\pi(s)\\ &\overset{\diamondsuit}{=} \int_{\mathcal{S}}\int_{\mathcal{A}} \overline{D}_{\lambda^*}(s,a)\mathrm{d}\pi(a|s)\mathrm{d}\rho_\pi(s)\\ &\overset{\triangle}{=} \int_{\mathcal{S}}\int_{\mathcal{A}} c(a,\overline{T}_{\lambda^*}(s,a))\mathrm{d}\pi(a|s)\mathrm{d}\rho_\pi(s),\end{aligned} \tag{14}$$

and

$$\begin{aligned}\frac{\partial h(\lambda)}{\partial \lambda +} \geq 0 \iff \varepsilon &\geq \frac{\partial}{\partial \lambda +}\left(\int_{\mathcal{S}}\int_{\mathcal{A}} \Phi_{\lambda^*}(s,a)\mathrm{d}\pi(a|s)\mathrm{d}\rho_\pi(s)\right)\\ &\overset{\heartsuit}{=} \int_{\mathcal{S}}\int_{\mathcal{A}} \frac{\partial}{\partial \lambda +}\Phi_{\lambda^*}(s,a)\mathrm{d}\pi(a|s)\mathrm{d}\rho_\pi(s)\\ &\overset{\diamondsuit}{=} \int_{\mathcal{S}}\int_{\mathcal{A}} \underline{D}_{\lambda^*}(s,a)d\pi(a|s)d\rho_\pi(s)\\ &\overset{\triangle}{=} \int_{\mathcal{S}}\int_{\mathcal{A}} c(a,\underline{T}_{\lambda^*}(s,a))\mathrm{d}\pi(a|s)\mathrm{d}\rho_\pi(s),\end{aligned} \tag{15}$$

where in

$\heartsuit$ we use that, for all λ, $\overline{D}_\lambda(s,a) = c(a, \overline{T}_\lambda(s,a))$, $\underline{D}_\lambda(s,a) = c(a, \underline{T}_\lambda(s,a))$ result from the composition of a continuous (the cost) and a measurable function ($\overline{T}_\lambda(s,a)$ and $\underline{T}_\lambda(s,a)$), and thus they are measurable [28, Theorem 1.8]. Integrability follows directly from non-negativity [28, Theorem 1.17, Definition 1.23]. Hence, we can use the differentiation lemma [12, Theorem 6.28] together with Lemma 7;

$\diamondsuit$ we use Lemma 7.3; and

$\triangle$ we use the definition of $\underline{T}_{\lambda^*}$, $\overline{T}_{\lambda^*}$ and of $\underline{D}_{\lambda^*}(s,a)$, $\overline{D}_{\lambda^*}(s,a)$.

Overall, the inequalities in (14) and (15) lead to

$$\int_{\mathcal{S}} \int_{\mathcal{A}} c(a, \underline{T}_{\lambda^*}(s,a)) \mathrm{d}\pi(a|s) \mathrm{d}\rho_\pi(s) \leq \varepsilon \leq \int_{\mathcal{S}} \int_{\mathcal{A}} c(a, \overline{T}_{\lambda^*}(s,a)) \mathrm{d}\pi(a|s) \mathrm{d}\rho_\pi(s). \tag{16}$$

With this observation, we can construct a primal solution $\tilde{\pi}(\cdot|s)$ and show its optimality. First, by (16), there exists $t^* \in [0,1]$ such that

$$t^* \int_{\mathcal{S}} \int_{\mathcal{A}} c(a, \underline{T}_{\lambda^*}(s,a)) \mathrm{d}\pi(a|s) \mathrm{d}\rho_\pi(s) + (1-t^*) \int_{\mathcal{S}} \int_{\mathcal{A}} c(a, \overline{T}_{\lambda^*}(s,a)) \mathrm{d}\pi(a|s) \mathrm{d}\rho_\pi(s) = \varepsilon.$$

For all $s \in \mathcal{S}$ consider the policy

$$\tilde{\pi}(\cdot|s) \coloneqq t^* \underline{T}_{\lambda^*}(s,\cdot)_\# \pi(\cdot|s) + (1-t^*) \overline{T}_{\lambda^*}(s,\cdot)_\# \pi(\cdot|s).$$

By construction, $\pi(\cdot|s)$ is a feasible solution. Indeed, with Id being the identity mapping $\mathrm{Id}(a) = a$, we can consider the (possibly sub-optimal) transport plan $\gamma_s \in \Pi(\pi(\cdot|s), \tilde{\pi}(\cdot|s))$, defined as

$$\gamma_s \coloneqq t^* (\mathrm{Id}, \underline{T}_{\lambda^*}(s,\cdot))_\# \pi(\cdot|s) + (1-t^*)(\mathrm{Id}, \overline{T}_{\lambda^*}(s,\cdot)_\# \pi(\cdot|s).$$

Then, by definition of optimal transport discrepancy and monotonicity of the integral we have

$$\begin{aligned}
\int_{\mathcal{S}} C(\pi(\cdot|s), \tilde{\pi}(\cdot|s)) \mathrm{d}\rho_\pi(s) &\leq \int_{\mathcal{S}} \int_{\mathcal{A}\times\mathcal{A}} c(a,a') \mathrm{d}\gamma_s(a,a') \mathrm{d}\rho_\pi(s) \\
&= t^* \int_{\mathcal{S}} \int_{\mathcal{A}} c(a, \underline{T}_{\lambda^*}(s,a)) \mathrm{d}\pi(a|s) \mathrm{d}\rho_\pi(s) \\
&\quad + (1-t^*) \int_{\mathcal{S}} \int_{\mathcal{A}} c(a, \overline{T}_{\lambda^*}(s,a)) \mathrm{d}\pi(a|s) \mathrm{d}\rho_\pi(s) \\
&= \varepsilon.
\end{aligned} \tag{17}$$

To conclude, by definition, we have

$$\begin{aligned}
A^\pi(s, \overline{T}_{\lambda^*}(s,a)) &= \Phi_{\lambda^*}(s,a) + \lambda^* c(a, \overline{T}_{\lambda^*}(s,a)) \\
A^\pi(s, \underline{T}_{\lambda^*}(s,a)) &= \Phi_{\lambda^*}(s,a) + \lambda^* c(a, \underline{T}_{\lambda^*}(s,a))
\end{aligned} \tag{18}$$

and thus

$$\begin{aligned}
\text{(P)} &\geq \int_{\mathcal{S}} \int_{\mathcal{A}} A^\pi(s,a) \mathrm{d}\tilde{\pi}(a|s) \mathrm{d}\rho_\pi(s) \\
&= \int_{\mathcal{S}} \int_{\mathcal{A}} \left(t^* A^\pi(s, \underline{T}_{\lambda^*}(s,a)) + (1-t^*) A^\pi(s, \overline{T}_{\lambda^*}(s,a)) \right) \mathrm{d}\pi(a|s) \mathrm{d}\rho_\pi(s) \\
&= t^* \int_{\mathcal{S}} \int_{\mathcal{A}} A^\pi(s, \underline{T}_{\lambda^*}(s,a)) \mathrm{d}\pi(a|s) \mathrm{d}\rho_\pi(s) + (1-t^*) \int_{\mathcal{S}} \int_{\mathcal{A}} A^\pi(s, \overline{T}_{\lambda^*}(s,a)) \mathrm{d}\pi(a|s) \mathrm{d}\rho_\pi(s) \\
&\overset{(18)}{=} \lambda^* \int_{\mathcal{S}} \left(t^* c(a, \underline{T}_{\lambda^*}(s,a)) + (1-t^*) c(a, \overline{T}_{\lambda^*}(s,a)) \right) \mathrm{d}\rho_\pi(s) \\
&\quad + \int_{\mathcal{S}} \int_{\mathcal{A}} \Phi_{\lambda^*}(s,a) \mathrm{d}\pi(a|s) \mathrm{d}\rho_\pi(s) \\
&\overset{(17)}{=} \lambda^* \varepsilon + \int_{\mathcal{S}} \int_{\mathcal{A}} \Phi_{\lambda^*}(s,a) \mathrm{d}\pi(a|s) \mathrm{d}\rho_\pi(s) \\
&= \text{(D)},
\end{aligned}$$

proving (P) $\geq$ (D). In particular, we have (P) = (D) and π is the primal solution. Thus, the supremum over Π is attained. $\square$

The proofs of Corollaries 2 and 3 then follow directly.

Remark 4. Our theoretic results readily extends to transport costs $c_{s,\pi}$ parametrized by the state s or the policy π, as long as the mapping $(s, \pi, a_1, a_2) \mapsto c_{s,\pi}(a_1, a_2)$ is continuous.

A natural question is whether the results of Theorem 1 can be extended to Sinkhorn divergence (or entropy-regularized optimal transport discrepancies in general), in particular given the success of these in the context of RL [6, 23]. Our duality results do not directly apply to Sinkhorn divergence (or entropy-regularized optimal transport discrepancies in general). For instance, our proof of weak duality (Proposition 5) leverages Kantorovich duality, which is optimal transport specific. However, duality results for entropy-regularized optimal transport [35], and recent results in DRO [7] suggest that our proof can be adapted to such constraints. Having said this, we remark that Sinkhorn divergence was successful in mitigating the burden of optimal transport computations. The proposed algorithm does not rely on any of such, and it is thus unclear if regularized optimal transport discrepancies yield any benefit in this context.

B.2 Proof of Proposition 4

Proof. First, recall the performance difference lemma [11, Lemma 6.1]:

$$J(\tilde{\pi}^*) - J(\pi) = \frac{1}{1-\gamma} \int_{\mathcal{S}} \int_{\mathcal{A}} A^{\pi}(s,a) \mathrm{d}\tilde{\pi}^*(a|s) \mathrm{d}\rho_{\tilde{\pi}^*}(s).$$

Second, recall that from Corollary 2 that an optimal policy update is

$$\tilde{\pi}^*(\cdot|s) \coloneqq t^* \underline{T}_{\lambda^*}(s,\cdot)_{\#}\pi(\cdot|s) + (1-t^*)\overline{T}_{\lambda^*}(s,\cdot)_{\#}\pi(\cdot|s). \tag{19}$$

Then, the proof follows from the properties of $\tilde{\pi}^*$:

$$\begin{aligned}
J(\tilde{\pi}^*) - J(\pi) &= \frac{1}{1-\gamma} \int_{\mathcal{S}} \int_{\mathcal{A}} A^{\pi}(s,a) \mathrm{d}\tilde{\pi}^*(a|s) \mathrm{d}\rho_{\tilde{\pi}^*}(s) \\
&\overset{(19)}{=} \frac{1}{1-\gamma} \int_{\mathcal{S}} \int_{\mathcal{A}} t^* A^{\pi}(s, \underline{T}_{\lambda^*}(s,a)) + (1-t^*) A^{\pi}(s, \overline{T}_{\lambda^*}(s,a)) \mathrm{d}\pi(a|s) \mathrm{d}\rho_{\tilde{\pi}^*}(s) \\
&\overset{\heartsuit}{\geq} \frac{1}{1-\gamma} \int_{\mathcal{S}} \int_{\mathcal{A}} t^* \hat{A}^{\pi}(s, \underline{T}_{\lambda^*}(s,a)) + (1-t^*) \hat{A}^{\pi}(s, \overline{T}_{\lambda^*}(s,a)) \mathrm{d}\pi(a|s) \mathrm{d}\rho_{\tilde{\pi}^*}(s) \\
&\quad - \frac{\|A^{\pi} - \hat{A}^{\pi}\|_{\infty}}{1-\gamma} \\
&\overset{\diamondsuit}{=} \frac{\lambda^*}{1-\gamma} \int_{\mathcal{S}} \int_{\mathcal{A}} t^* c(a, \underline{T}_{\lambda^*}(s,a)) + (1-t^*) c(a, \overline{T}_{\lambda^*}(s,a)) \mathrm{d}\pi(a|s) \mathrm{d}\rho_{\tilde{\pi}^*}(s) \\
&\quad + \frac{1}{1-\gamma} \int_{\mathcal{S}} \int_{\mathcal{A}} \Phi_{\lambda^*}(s,a) \mathrm{d}\pi(a|s) \mathrm{d}\rho_{\tilde{\pi}^*}(s) - \frac{\|A^{\pi} - \hat{A}^{\pi}\|_{\infty}}{1-\gamma} \\
&\overset{\triangle}{\geq} \frac{\lambda^*}{1-\gamma} \int_{\mathcal{S}} C(\pi(\cdot|s), \tilde{\pi}^*(\cdot|s)) \mathrm{d}\rho_{\tilde{\pi}^*}(s) \\
&\quad + \frac{1}{1-\gamma} \int_{\mathcal{S}} \int_{\mathcal{A}} \hat{A}^{\pi}(s,a) \mathrm{d}\pi(a|s) \mathrm{d}\rho_{\tilde{\pi}^*}(s) - \frac{\|A^{\pi} - \hat{A}^{\pi}\|_{\infty}}{1-\gamma} \\
&\overset{\heartsuit}{\geq} \frac{\lambda^*}{1-\gamma} \int_{\mathcal{S}} C(\pi(\cdot|s), \tilde{\pi}^*(\cdot|s)) \mathrm{d}\rho_{\tilde{\pi}^*}(s) \\
&\quad + \frac{1}{1-\gamma} \int_{\mathcal{S}} \int_{\mathcal{A}} A^{\pi}(s,a) \mathrm{d}\pi(a|s) \mathrm{d}\rho_{\tilde{\pi}^*}(s) - \frac{2\|A^{\pi} - \hat{A}^{\pi}\|_{\infty}}{1-\gamma} \\
&\overset{\square}{=} \frac{\lambda^*}{1-\gamma} \int_{\mathcal{S}} C(\pi(\cdot|s), \tilde{\pi}^*(\cdot|s)) \mathrm{d}\rho_{\tilde{\pi}^*}(s) - \frac{2\|A^{\pi} - \hat{A}^{\pi}\|_{\infty}}{1-\gamma},
\end{aligned}$$

where in

$\heartsuit$ we use that, by assumption, $A^{\pi}(s,a) \geq \hat{A}^{\pi}(s,a) - \|A^{\pi} - \hat{A}^{\pi}\|_{\infty}$ and $\hat{A}^{\pi}(s,a) \geq A^{\pi}(s,a) - \|A^{\pi} - \hat{A}^{\pi}\|_{\infty}$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$;

$\diamondsuit$ by definition of the regularization operator Φ and of $\underline{T}_{\lambda^*}(s,a)$ and $\overline{T}_{\lambda^*}(s,a)$ (Lemma 7.4), for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$ it holds

$$A^\pi(s, \overline{T}_{\lambda^*}(s,a)) = \Phi_{\lambda^*}(s,a) + \lambda^* c(a, \overline{T}_{\lambda^*}(s,a)).$$

$\triangle$ as in (17), it holds

$$C(\pi^*(\cdot|s), \pi(\cdot|s)) \leq \int_{\mathcal{A}} t^* c(a, \underline{T}_{\lambda^*}(s,a)) + (1-t^*) c(a, \overline{T}_{\lambda^*}(s,a)) \mathrm{d}\pi(a|s)$$

and $A^\pi(s,a) \leq \Phi_{\lambda^*}(s,a)$, since for all $\lambda \in \mathbb{R}, a \in \mathcal{A}$, and $s \in \mathcal{S}$

$$\begin{aligned} \Phi_\lambda(s,a) &= \max_{a' \in \mathcal{A}} \{A^\pi(s,a') - \lambda c(a,a')\} \\ &\geq A^\pi(s,a) - c(a,a) \\ &= A^\pi(s,a); \end{aligned}$$

$\square$ the expected value of the advantage function vanishes, since, by definition, for all $s \in \mathcal{S}$

$$\int_{\mathcal{A}} A^\pi(s,a) \mathrm{d}\pi(a|s) = \int_{\mathcal{A}} Q^\pi(s,a) - V^\pi(s) \mathrm{d}\pi(a|s) = \int_{\mathcal{A}} Q^\pi(s,a) \mathrm{d}\pi(a|s) - V^\pi(s) = 0.$$

This concludes the proof. $\square$