Lagrangian Method for Q-Function Learning (with Applications to Machine Translation)

Huang Bojun [1]

Abstract

This paper discusses a new approach to the fundamental problem of learning optimal Q-functions. In this approach, optimal Q-functions are formulated as saddle points of a *nonlinear* Lagrangian function derived from the classic Bellman optimality equation. The paper shows that the Lagrangian enjoys *strong duality*, in spite of its nonlinearity, which paves the way to a general Lagrangian method to Q-function learning. As a demonstration, the paper develops an imitation learning algorithm based on the duality theory, and applies the algorithm to a state-of-the-art machine translation benchmark. The paper then turns to demonstrate a symmetry breaking phenomenon regarding the optimality of the Lagrangian saddle points, which justifies a largely overlooked direction in developing the Lagrangian method.

1. Introduction

Machine learning methods can be broadly categorized into two classes: policy learning, and Q-function learning. In policy learning, the goal is to learn a policy function that directly encodes the desired mapping between task inputs and outputs. An imitation learning example is *behavior cloning*, which tunes a parametric policy toward the desired policy function based on example input-output pairs of the latter. A reinforcement learning example is *policy gradient*, which iteratively improves the parametric policy function following the gradient direction of the task performance.

Alternatively, one can try to learn a Q-function that encodes a preferential score for each candidate output given an input. An *optimal Q-function* would induce a *greedy policy* that identifies the optimal output simply by comparing the Q-values, rather than the precise consequences of the outputs. Implicitly or explicitly, the majority of statistical supervised learning practices nowadays fall in the category of Q-function learning, where the general strategy is to make the Q-value of each output equal to the conditional probability that an expert would choose this output. Note that although the learned Q-function represents a desired probability distribution, this distribution is not used to sample the outputs at decision time (which contrasts with behavior cloning), but is used to power a greedy and deterministic decision policy. The resulted policy is provably optimal in many simple yet common scenarios, such as for maximizing prediction accuracy in classification tasks (Bishop, 2006).

In more complex scenarios, such as in *sequential decision making* or *structured prediction* tasks, Q-functions that simply follow the expert probabilities are generally not optimal, and learning optimal Q-functions becomes a more challenging problem. In this case, the standard paradigm at the moment is to learn the *Bellman optimal value* function (Sutton and Barto, 2018), which is a special optimal Q-function that complies with a particular constraint known as the *Bellman equation*. The Bellman equation also immediately implies a *value iteration* method to find/approximate the optimal Q-function that the equation defined, leading to the many variants of *Q-Learning* algorithms (Watkins, 1989).

In this paper, we propose a new method for Q-function learning. Our method is based on a variational (re-)formulation of the Bellman equation for optimal Q-function. The variational formulation induces a nonlinear *Lagrangian function* whose saddle points correspond to a class of optimal Q-functions (but not necessarily the Bellman optimal value). The Q-function learning problem can then be reduced to saddle-point optimization for the Lagrangian.

This high-level idea is known as the *Lagrangian method* in a wide variety of engineering domains (Boyd et al., 2004), including machine learning (Goodfellow et al., 2014), and more specifically, has been recently applied to related value-function learning problems (Dai et al., 2018; Lee and He, 2018; Nachum et al., 2019; Yang et al., 2020). There is however a key obstacle to apply the Lagrangian method to Q-function learning: To admit effective and efficient algorithms, it is typically required that the Lagrangian function has a *strong duality* property. When the Lagrangian is linear,

[1] Rakuten Institute of Technology, Rakuten Group Inc., Japan. Correspondence to: Huang Bojun <bojhuang@gmail.com>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

the strong duality property is universally guaranteed, which is the theoretical foundation for most of its algorithmic applications. Unfortunately, for (optimal) Q-function learning, the Lagrangian derived from the Bellman equation is nonlinear, which prevents further algorithmic developments.

Our first main contribution is to prove that, for a large class of decision tasks, the nonlinear Lagrangian function for optimal Q-function learning turns out to be a special nonlinear function that actually enjoys the strong duality "as usual" (Section 3). This observation potentially opens a door to a new general approach to Q-function learning.

As a demonstration of this potential, we developed a simple imitation learning algorithm based on the Lagrangian duality theory. We presented two practical variants of the algorithm, and empirically applied them to Machine Translation (MT) as a case study. Our algorithms are able to train Transformer (Vaswani et al., 2017), a state-of-the-art neural network model, on large-scale MT benchmarks with real-world data, and lead to 1.4 BLEU (=5%) improvement over standard baseline (Section 4).

Thirdly, we discovered an unusual *symmetry breaking* phenomenon for the Lagrangian for optimal Q-function: From the same Bellman equation, there can be two mirrored ways to derive the same Lagrangian function, resulting in two kinds of saddle-point solutions, minimax points and maximin points. While previous research on the topic has been exclusively focusing on the former (i.e. minimax points), they can be sub-optimal in modern learning settings, as we show in Section 5. Intriguingly, we prove that the other class of saddle points, the maximin points, turn out to guarantee optimality. This observation points to a new direction that has perhaps been overlooked for a decades-old topic.

Lastly, our paper provides a new and more general theorem about the Bellman optimality structure (see Section 2), which enabled us to develop the theory on top of the (episode-wise) *total reward* optimality criterion. The latter is arguably a more accurate formulation than the canonical discounted-MDP setting for many real-world applications. We believe that our mathematical treatment contributes to fill this well-known gap between theoretical formulation and practical setup for Q-function learning.

2. Problem Formulation

In general, a decision task is mathematically a Markov Decision Process (MDP) $(\mathcal{S}, \mathcal{A}, R, P, \rho)$. $\mathcal{S}$ is the state space, $\mathcal{A}$ the action space. $R(s)$ is a bounded reward (possibly negative) associated to each state $s \in \mathcal{S}$. [1] $P(s'|s,a)$ specifies action-conditioned transition probabilities between states, and ρ is the initial-state distribution. Given an MDP, a policy function π specifies the action selection probabilities under each state, which induces a markov chain with $\mathbf{P}_\pi[S_1 = s] = \rho(s)$ and $\mathbf{P}_\pi[S_{t+1} = s'|S_t = s] = \sum_{a \in \mathcal{A}} P(s'|s,a) \cdot \pi(a|s)$. Running π in the markov chain $\mathbf{P}_\pi$ results in an infinite trajectory $\zeta = (S_1, A_1, S_2, A_2 \dots)$.

In this paper, we focus on *finite-time* decision tasks, in which there is a set of **terminal states** $\mathcal{S}_\perp \subseteq \mathcal{S}$ so that the trajectory ζ will run into a terminal state in finite steps under any policy π. Formally, let $T \doteq \inf\{t \geq 1 : S_t \in \mathcal{S}_\perp\}$ be the **termination time** (which is a random variable in the probability space of $\mathbf{P}_\pi$), a finite-time task is an MDP with $\mathbf{E}_\pi[T] < \infty, \forall \pi$. The goal is to find an **optimal policy** that maximizes the expected total reward until termination:

$$J(\pi) \doteq \mathop{\mathbf{E}}_{\zeta \sim \pi} \left[\sum_{t=1}^{T(\zeta)} R(S_t) \right] \tag{1}$$

Finite-time tasks are important for both empirical and theoretical reasons. Empirically, they account for most real-world tasks in current AI practice. See Section E.3 for a MDP formulation of *machine translation* as example. One should not confuse finite-time tasks with *finite-horizon* tasks. The latter is a special case of the former, in which T has a finite upper bound H (so $\mathbf{E}_\pi[T] \leq H$). Alternatively, a task in which every state has a non-zero probability $1 - \gamma$ to reach terminal states immediately (under any policy) does not have a finite horizon, yet it is a finite-time task because in such task T follows a geometric distribution with finite mean. Moreover, if in this case $R(s) = 0$ for all terminal states, maximizing the total-reward objective (1) with termination would be equivalent to maximizing the discounted reward $\mathbf{E}_\pi[\sum_{t=1}^{\infty} R(S_t) \cdot \gamma^{t-1}]$ without termination (see Appendix A). In this sense, the canonical discounted-MDP tasks can be recast into a special case of finite-time decision tasks with (undiscounted) total-reward objective (1) too.

Learning for finite-time decision task is typically based on trajectory data from *multiple* decision episodes of the task, each starting from a state following ρ and ending at a terminal state in $\mathcal{S}_\perp$. In imitation learning setting, the actions in the episodes may be provided by an expert policy; in this case the agent can learn from other's experiences. In reinforcement learning setting, the agent has to count on itself to generate the actions in its learning data.

This *learning process* – in which the agent looks at experience of either others or itself, episode after episode – can be formulated as a special family of finite-time MDPs where any terminal state transits back to a random state following the initial distribution ρ under whatever action. [2]

[1] Our state-based reward formulation follows (Schulman et al., 2015) and (Bojun, 2020), and is equivalent to, if not more general than, the various action-based reward formulations (that some readers might feel more familiar with). See Appendix A for details.

[2] The "transit back" setting is fully aligned with the definition of finite-time MDP, and with the objective (1) too, as neither of them prescribes what happens after the termination (except that the time homogeneity of MDP requires the transition to continue).

A rollout trajectory of such recurrent MDP consists of an *infinite* sequence of *finite* episodes. Following Bojun (2020), we call such a "MDP for learning", an *Episodic Learning Process* (ELP). Formally, an MDP is an ELP if (1) it is a finite-time MDP (i.e. $\mathbf{E}_\pi[T] < \infty, \forall \pi$), (2) all terminal states "reset" the MDP in a homogeneous manner: $P(s'|s_1, a_1) = P(s'|s_2, a_2) = \rho(s')$, $\forall s_1, s_2 \in \mathcal{S}_\perp$, $\forall a_1, a_2 \in \mathcal{A}$, $\forall s' \in \mathcal{S}$, and (3) for mathematical convenience, the MDP is set to start at step 0 from a terminal state $s_0 \in \mathcal{S}_\perp$ (in that case S_1 still follows ρ). The ELP thus defined formally characterizes a learning problem for finite-time decision task, in which the learning algorithm obtains a (multi-episode) trajectory from the ELP, via either observing a given expert policy or trying an evolving behavior policy of its own, and seeks to find a good policy with respect to objective function (1).

A Q-function assigns a real number to each state-action pair (s, a) as the "perceived benefit" of doing a under s. We use $\mathcal{Q} = \{\mathcal{S} \times \mathcal{A} \to \mathbb{R}\}$ to denote the set of all possible Q-functions. Each Q-function induces a greedy policy, denoted by π_Q, for which $\pi_Q(a|s) > 0$ only if $Q(s, a) = \max_{\bar{a}} Q(s, \bar{a})$. A Q-function is an **optimal Q-function** if the greedy policy π_Q is an optimal policy.

Bellman optimal value function is a special optimal Q-function that is characterized by the Bellman optimality operator. In its general form (Degris et al., 2012; Sutton et al., 2011; 2014; 2016), a generalized Bellman optimality operator $\mathcal{B}^\gamma : \mathcal{Q} \to \mathcal{Q}$ transforms a value function Q into another value function $\mathcal{B}^\gamma Q$ such that for all $(s, a) \in \mathcal{S} \times \mathcal{A}$,

$$\mathcal{B}^\gamma Q(s,a) \doteq \sum_{s' \in \mathcal{S}} P(s'|s,a) \cdot \Big(R(s') + \gamma(s') \cdot \max_{a' \in \mathcal{A}} Q(s', a')\Big) \tag{2}$$

where $\gamma : \mathcal{S} \to [0, 1]$ is a **discounting function** over the states. If $\gamma(s) < 1$ on every state s, the corresponding Bellman optimality operator has a unique fixed point in the Q-space of any MDP, and this fixed point is known as the Bellman optimal value function.

However, this fixed point under uniform discounting is generally not an optimal Q-function with respect to the undiscounted objective (1). In the following, we present a new theorem that generalizes the classic theory of Bellman optimal value to a more general class of discounting functions, among which a particular γ-function makes the Bellman optimal value precisely an optimal Q-function w.r.t. (1).

Theorem 1. *In any finite Episodic Learning Process* $(\mathcal{S}, \mathcal{A}, P, R, \rho)$*, let* γ *be any discounting function such that* $\gamma(s) < 1$ *at all terminal state* $s \in \mathcal{S}_\perp$*, then*

(1) $\mathcal{B}^\gamma$ *has a unique fixed point in* $\mathcal{Q}$*.*

(2) The fixed point of $\mathcal{B}^\gamma$ *is the limiting point of repeatedly applying* $\mathcal{B}^\gamma$ *to any* $Q \in \mathcal{Q}$*.*

(3) The fixed point of $\mathcal{B}^\gamma$ *is an optimal Q-function to* (1) *when* γ *is the following* ***episodic discounting function****:*

$$\gamma_{\mathit{epi}}(s) \doteq \mathbb{1}[s \notin \mathcal{S}_\perp] = \begin{cases} 1 & \textit{for } s \notin \mathcal{S}_\perp \\ 0 & \textit{for } s \in \mathcal{S}_\perp \end{cases}. \tag{3}$$

Different from the classic result which crucially relies on uniform discounting, both the uniqueness and optimality in Theorem 1 are rooted from an inherent graph property of ELP. See the proof in Appendix B.

Since we are focusing on optimizing the objective (1) only, in the rest of the paper: We will write $\mathcal{B} \doteq \mathcal{B}^{\gamma_{\text{epi}}}$, and simply call it the **Bellman operator**, for the particular Bellman optimality operator that uses the particular episodic discounting function (3). We call the fixed point of $\mathcal{B}$, the **Bellman value**, denoted by Q^*. Similarly, we call $Q = \mathcal{B}Q$, the **Bellman equation**, which is again assuming the particular episodic discounting function. More explicitly, Bellman equation refers to the following non-linear equation over $\mathcal{Q}$:

$$Q(s,a) = \sum_{s' \in \mathcal{S}} \max_{a' \in \mathcal{A}} P(s'|s,a) \cdot \Big(R(s') + \gamma_{\text{epi}}(s') \cdot Q(s', a')\Big) \tag{4}$$

It is worth noting that although the Bellman value is unique, there can be more than one optimal Q-functions for a given problem. For example, any Q-function that gives the same preferential order with the Bellman value is also optimal.

3. A Nonlinear Lagrangian Duality

In this and the next section, we discuss a variational treatment to the Bellman equation which converts (4) to a constrained *minimization* problem. We will demonstrate some nice theoretical properties of this variational problem in this section, and will present algorithmic applications of the theory in the next section.

Our idea is inspired by a long-known linear programming reformulation (Puterman, 1994) of the closely related Bellman equation for *state-value functions* $V : \mathcal{S} \to \mathbb{R}$,

$$V(s) = \max_a \sum_{s'} P(s'|s,a) \cdot \Big(R(s') + \gamma \cdot V(s')\Big). \tag{5}$$

Both the Q-form Bellman equation (4) and the V-form Bellman equation (5) are nonlinear due to the $\max$ operator inside. But the V-form equation (5) admits a *linear* variational re-formulation (Dai et al., 2018):

$$\begin{aligned} &\min_V \quad \sum_{s \in \mathcal{S}} \rho(s) \cdot V(s) \\ &V(s) \geq \sum_{s' \in \mathcal{S}} P(s'|s,a) \Big(R(s') + \gamma\, V(s')\Big), \forall (s,a) \end{aligned} \tag{6}$$

Thanks to its linearity, (6) enjoys the standard LP duality properties, and in particular has *minimax equality* for its corresponding Lagrangian function, which is the basis for a recently revived thread of research on the LP approach to MDP and RL (Chen and Wang, 2016; Wang, 2017; Cho and Wang, 2017; Dai et al., 2018; Nachum and Dai, 2020). In traditional settings, with the V-function solution of (5) or (6), one could construct an optimal Q-function by averaging the V-values over the transition probabilities, obtaining $Q(s, a) = \sum_{s' \in \mathcal{S}} P(s'|s, a)\, V(s')$.

In modern learning settings, however, directly applying this V-function based approach encounters additional obstacles. First, the transition probabilities $P(s'|s, a)$ are often unknown in learning settings, and learning these probabilities itself is a substantial challenge. Second, computing greedy actions from the V-function often requires significantly more computational resource when deep neural networks are used. Specifically, we need to collect the V-values of successor states for each action, which requires to evaluate the V-function under at least $|\mathcal{A}|$ different states even assuming deterministic transition (and the number can be much higher under stochastic transition). On the other hand, to decide the greedy action for given state s, we only need to evaluate the Q-function under one state, the state s. Further evaluating the Q-values for each action *under the same state* usually involves much less computation. [3]

For these reasons, most value-based learning algorithms focus on learning the Q-functions directly (Watkins, 1989; Hasselt, 2010; Mnih et al., 2015; Haarnoja et al., 2018). It is thus natural to ask if we can develop a variational treatment directly to the Q-form Bellman equation (4), similar to what we did to the V-form equations.

Indeed, the Q-form Bellman equation (4) can be similarly recast into a constrained optimization problem, such as in the following form:

$$\begin{aligned} \min_{Q} \quad & \mathop{\mathbf{E}}_{\zeta \sim \pi} \Big[Q(S_T, A_T) \Big] \\ \text{s.t.} \quad & Q(s, a) \geq \mathcal{B}Q(s, a) \quad , \quad \forall (s, a) \end{aligned} \tag{7}$$

where π is an arbitrary policy, and T is the termination time. Unfortunately, unlike the V-form Bellman equation for which the nonlinear operation $\max_a$ can be "unpacked" into $|\mathcal{A}|$ linear constraints (cf. (5) and (6)), the optimization problem (7) for the Q-form Bellman equation is still nonlinear, due to the $\max_{a'}$ "wrapped" inside $\sum_{s'}$. As a result, although (7) can still be written into the Lagrangian form, it is unclear if the nonlinear Lagrangian still enjoys strong duality, a key property for designing effective and principled learning algorithms (Wang, 2017; Dai et al., 2018).

In the following, we give an affirmative answer to this open question by proving a *minimax theorem* for the nonlinear Lagrangian of the Q-form Bellman equation (4).

Definition. *Given a finite ELP $(\mathcal{S}, \mathcal{A}, P, R, \rho)$, the* ***Lagrangian function*** *with* ***conjugate policy*** *π is $\mathcal{L}_\pi(Q, \boldsymbol{\lambda}) \doteq$*

$$\mathop{\mathbf{E}}_{\zeta \sim \pi} \Big[Q(S_T, A_T) \Big] + \sum_{s,a} \lambda(s, a) \Big(\mathcal{B}Q(s, a) - Q(s, a) \Big) \tag{8}$$

Note that the conjugate policy π of the Lagrangian only affects the first term of $\mathcal{L}_\pi$. The second term of $\mathcal{L}_\pi$ always uses the nonlinear *optimality* operator $\mathcal{B}$ which takes max over the Q-values, regardless of the conjugate policy π.

The first term of the Lagrangian takes average only over the terminal states, instead of over *all* states as in dynamic programming (Puterman, 1994; Wang, 2017), because the latter is typically intractable in learning settings – the learning agent who relies on the episodic rollout to obtain data cannot sweep through the whole state-action space itself, not even sample the space (near-)uniformly. [4] On the other hand, the Lagrangian averages over the *terminal* states, instead of over the *initial* states as in previous learning-oriented works (e.g. (Cho and Wang, 2017; Dai et al., 2018; Nachum and Dai, 2020)), because the distribution of initial states are fixed, while which terminal states a policy would reach depends on the behavior of the policy. It turns out that this policy dependency, as well as the mathematical construct of evaluating the Q-function over the terminal states (which is only meaningful in the ELP formulation), would play an important role in developing the main results we shall see.

Let us first confirm, with the following lemma, that the Bellman value function Q^* is a minimax solution of the Lagrangian function (8). See the proof in Appendix C.1.

Lemma 2.1. *In any finite ELP, for any conjugate policy π,*

$$Q^* \in \arg \min_{Q \in \mathcal{Q}} \max_{\boldsymbol{\lambda} \geq 0} \; \mathcal{L}_\pi(Q, \boldsymbol{\lambda}).$$

Now we show that $\mathcal{L}_\pi$ has a dual form when the Lagrangian multiplier is set to a special vector. Specifically, it is known that in any episodic learning process, every policy π has a unique **stationary distribution** $\rho_\pi(s)$ such that $\mathbf{P}_\pi[S_{t+1} = s] = \rho_\pi(s)$ if $\mathbf{P}_\pi[S_t = s] = \rho_\pi(s)$ (Bojun, 2020). The following lemma gives a transformation of the Lagrangian when the multiplier vector $\boldsymbol{\lambda}$ is proportional to the stationary distribution of the conjugate policy π (and is scaled by the average episode length of π).

Lemma 2.2. *In any finite ELP, let $\mathcal{L}_\pi$ be the Lagrangian with conjugate policy π, and let $\boldsymbol{\lambda}_\pi$ be the particular Lagrangian multiplier with $\boldsymbol{\lambda}_\pi(s, a) = \rho_\pi(s) \cdot \pi(a|s) \cdot \mathbf{E}_\pi[T]$, then $\mathcal{L}_\pi(Q, \boldsymbol{\lambda}_\pi) =$*

[3] For very large action spaces, such as continuous spaces, optimizing over even state-conditioned Q-values can be non-trivial (Gu et al., 2016), but in that case optimizing over the V-values under different states would be only more forbiddingly expensive.

[4] Mathematically, running a policy with uniform (or positive) distribution over the actions would have non-zero probability to reach any state; but typically in reality, such explorations will "almost surely" be stuck in a very limited region of the state space.

$$J(\pi) + \sum_{s\notin\mathcal{S}_\perp}\sum_{a\in\mathcal{A}} \boldsymbol{\lambda}_\pi(s,a)\Big(\max_{\bar{a}} Q(s,\bar{a}) - Q(s,a)\Big) \tag{9}$$

Proof idea: Applying a known *ELP ergodic theorem* (Bojun, 2020), we can transform the first term of (8) from an average over trajectories to an average over the state-action space:

$$\mathop{\mathbf{E}}_{\zeta\sim\pi}\Big[Q(S_T,A_T)\Big] = \mathop{\mathbf{E}}_{\zeta\sim\pi}[T]\cdot \mathop{\mathbf{E}}_{S,A\sim\rho_\pi}\Big[\Big(1-\gamma_{\text{epi}}(s)\Big)\,Q(S,A)\Big]$$

Substituting the above into (8) and re-arranging would give (9). To derive the $J(\pi)$ term in (9) (which is an average over the trajectory space, see (1)), we will need to use the ELP ergodic theorem again to transform things back to the trajectory space. See Appendix C.2 for the complete proof. □

The first term in the dual form of the Lagrangian, i.e. in (9), is the *true* performance of the conjugate policy π, which is a constant with respect to the Q-function. Utilizing this fact, we can prove the strong duality property for Lagrangians with optimal conjugate policy.

Theorem 2. *In any finite ELP $(\mathcal{S},\mathcal{A},P,R,\rho)$, if μ is an optimal policy, then its conjugate Lagrangian $\mathcal{L}_\mu$ has strong duality property, for which*

$$\min_{Q\in\mathcal{Q}}\ \max_{\boldsymbol{\lambda}\geq 0}\ \mathcal{L}_\mu(Q,\boldsymbol{\lambda}) = \max_{\boldsymbol{\lambda}\geq 0}\ \min_{Q\in\mathcal{Q}}\ \mathcal{L}_\mu(Q,\boldsymbol{\lambda}) = J(\mu) \tag{10}$$

Proof idea: For any conjugate policy π, due to Lemma 2.1, Q^* is a minimax solution of $\mathcal{L}_\pi$, from which we can obtain $\min_Q \max_{\boldsymbol{\lambda}} \mathcal{L}_\pi(Q,\boldsymbol{\lambda}) = \mathbf{E}_\pi[Q_{\pi^*}(S_T,A_T)] = J(\pi_{Q^*})$, where π_{Q^*} is the Q^*-greedy policy.

On the other hand, again for any conjugate policy π, due to Lemma 2.2, the dual form (9) of the Lagrangian, if seen as a function of Q, attains its minimum when Q achieves complementary slackness with $\boldsymbol{\lambda}_\pi$ in the second term, in which case the Lagrangian dual equals $J(\pi)$. Thus, for the particular multiplier $\boldsymbol{\lambda}_\pi$ we have $\min_Q \mathcal{L}_\pi(Q,\boldsymbol{\lambda}_\pi) = J(\pi)$, which is a lower bound of $\max_{\boldsymbol{\lambda}} \min_Q \mathcal{L}_\pi(Q,\boldsymbol{\lambda}_\pi)$.

Now, set the conjugate policy π to be the optimal policy μ, as assumed in the theorem, we would have

$$\begin{aligned}\max_{\boldsymbol{\lambda}}\min_Q \mathcal{L}_\mu(Q,\boldsymbol{\lambda}) &\geq \min_Q \mathcal{L}_\mu(Q,\boldsymbol{\lambda}_\mu) = J(\mu) = J(\pi_{Q^*})\\ &= \min_Q \max_{\boldsymbol{\lambda}} \mathcal{L}_\mu(Q,\boldsymbol{\lambda}).\end{aligned} \tag{11}$$

Because of the *weak duality* of the Lagrangian (which universally holds for any function), we also have $\max_{\boldsymbol{\lambda}} \min_Q \mathcal{L}_\mu(Q,\boldsymbol{\lambda}) \leq \min_Q \max_{\boldsymbol{\lambda}} \mathcal{L}_\mu(Q,\boldsymbol{\lambda})$, which means (11) can only be an equality, thus gives (10) as desired. See Appendix C.3 for the complete proof. □

Theorem 2 is a minimax theorem for the Q-functions and the Lagrangian multipliers. But from the proof idea above, we can also see that π_{Q^*}, the greedy policy of the Bellman value, serves as the counterpart of Q^* to form a minimax point of $\mathcal{L}_\mu$. In fact, we can judge if any pair of Q-function and policy form a Lagrangian minimax point based on complementary slackness (see the proof in Appendix C.4):

Proposition 3. *Given a finite ELP, for any Q-function Q and any policy π, let $\rho_\pi(s,a) = \rho_\pi(s)\ \pi(a|s)$ and $\boldsymbol{\lambda}_\pi(s,a) = \rho_\pi(s,a)\ \mathbf{E}_\pi[T]$, we have*

$$\mathcal{L}_\pi(Q,\bar{\boldsymbol{\lambda}}) \leq \mathcal{L}_\pi(Q,\boldsymbol{\lambda}_\pi) \leq \mathcal{L}_\pi(\bar{Q},\boldsymbol{\lambda}_\pi) \quad , \quad \forall \bar{Q},\bar{\boldsymbol{\lambda}}$$

if and only if

(1) $\mathcal{B}Q(s,a) - Q(s,a) \leq 0 \qquad , \ \forall (s,a)$

(2) $\rho_\pi(s,a)\Big(\mathcal{B}Q(s,a) - Q(s,a)\Big) = 0 \qquad , \ \forall (s,a)$

(3) $\rho_\pi(s,a)\Big(\max_{\bar{a}} Q(s,\bar{a}) - Q(s,a)\Big) = 0$, $\forall s \notin \mathcal{S}_\perp, \forall a$

4. Lagrangian Minimization Algorithms

From the last section we know that the Q-form Lagrangian $\mathcal{L}_\mu$ has strong duality (where μ is optimal policy), and that the special Lagrangian multiplier $\boldsymbol{\lambda}_\mu$ constitutes a minimax saddle point of the Lagrangian, together with the solution of (7). In other words, $\boldsymbol{\lambda}_\mu$ must have maximized the *Lagrangian dual* function $\min_Q \mathcal{L}_\mu(Q,\boldsymbol{\lambda})$, that is,

$$\begin{aligned}\max_\pi J(\pi) = \min_Q \max_{\boldsymbol{\lambda}\geq 0} \mathcal{L}_\mu(Q,\boldsymbol{\lambda}) &= \max_{\boldsymbol{\lambda}\geq 0} \min_Q \mathcal{L}_\mu(Q,\boldsymbol{\lambda})\\ = \min_Q \mathcal{L}_\mu(Q,\boldsymbol{\lambda}_\mu).&\end{aligned} \tag{12}$$

In light of (12), a simple idea to solve the variational Bellman value problem (7) is to find the Q-function that minimizes the Lagrangian at $\boldsymbol{\lambda}_\mu$, that is, minimizes $\mathcal{L}_\mu(Q,\boldsymbol{\lambda}_\mu)$. Such a Q-function would be, by (12), the counterpart of $\boldsymbol{\lambda}_\mu$; the two together form a minimax point of the Lagrangian, and thus this Q-function would be a solution of (7).

Although the closed form of $\mathcal{L}_\mu(Q,\boldsymbol{\lambda}_\mu)$ depends on an optimal policy μ (which might appear impossible to solve at a first glance, as coming up with such an optimal policy was our original goal of learning), notice that estimating the gradient of $\mathcal{L}_\mu(Q,\boldsymbol{\lambda}_\mu)$ only requires sampling data from the optimal policy μ, rather than explicit knowledge about how μ is constructed. Specifically, for a parameterized model $Q(s,a;\boldsymbol{w})$, we have

$$\begin{aligned}\nabla_{\boldsymbol{w}}\mathcal{L}_\mu(Q(\boldsymbol{w}),\boldsymbol{\lambda}_\mu) = \mathop{\mathbf{E}}_{\zeta\sim\mu}\Big[\nabla_{\boldsymbol{w}} Q(S_T,A_T;\boldsymbol{w})\Big] + \\ \mathop{\mathbf{E}}_{\zeta\sim\mu}[T] \mathop{\mathbf{E}}_{S,A\sim\rho_\mu}\Big[\nabla_{\boldsymbol{w}}[\mathcal{B}Q - Q](S,A;\boldsymbol{w})\Big]\end{aligned} \tag{13}$$

where $[\mathcal{B}Q - Q](s,a;\boldsymbol{w}) \doteq$

$$\mathop{\mathbf{E}}_{S'\sim P(s,a)}\left[R(S') + \gamma_{\text{ epi}}(S')\ \max_{a'} Q(S',a';\boldsymbol{w})\right] - Q(s,a;\boldsymbol{w})$$

in which the optimal policy μ only plays a role in determining the data distributions of the terminal state-actions (S_T, A_T), of the termination time T, and of the transition variable (S, A, S').

This observation inspires a general imitation learning approach, named *LAgrangian MINimization* (LAMIN) here, in which we try to collect some demonstration data from an optimal policy,[5] based on which we construct an estimator of the Lagrangian gradient (13), then apply standard stochastic gradient procedures to find a local minimum of $\mathcal{L}_\mu(Q(\boldsymbol{w}), \boldsymbol{\lambda}_\mu)$. The obtained $Q(\boldsymbol{w})$ is an approximation to the variational solution (7) of the Bellman equation.

4.1. Practical algorithms

One may design different estimators of (13) to power the LAMIN optimization. In the following we discuss two (families of) such gradient estimators, as demonstrating examples; they both perform reasonably and similarly well in our validation experiments presented in Section 4.2.

LAMIN1: A technical challenge of estimating (13) is to deal with the max operation in the Bellman operator $\mathcal{B}$. One popular trick is to "soften" the max operation with a Boltzmann distribution with small temperature β:

$$[\mathcal{B}Q - Q](s,a;\boldsymbol{w}) \approx \mathop{\mathbf{E}}_{S'\sim P(s,a)}\ \mathop{\mathbf{E}}_{A'\sim\pi^\beta_{\boldsymbol{w}}(S')}\left[\delta(s,a,S',A';\boldsymbol{w})\right] \tag{14}$$

where $\pi^\beta_{\boldsymbol{w}}(a|s) \doteq \frac{\exp\left(Q(s,a;\boldsymbol{w})/\beta\right)}{\sum_b \exp\left(Q(s,b;\boldsymbol{w})/\beta\right)}$, and δ is the temporal difference error with $\delta(s,a,s',a';\boldsymbol{w}) \doteq R(s') + \gamma_{\text{ epi}}(s')Q(s',a';\boldsymbol{w}) - Q(s,a;\boldsymbol{w})$.

Let $\mathcal{L}^\beta_\mu$ denote the smoothed Lagrangian that uses the right-hand side of (14). $\mathcal{L}^\beta_\mu$ is readily differentiable *and* can be arbitrarily close to the exact Lagrangian $\mathcal{L}_\mu$ with $\beta \to 0$ (e.g. with $\beta = 0.01$, $\pi_1/\pi_2 > 20,000$ if $Q_1 - Q_2 > 0.1$).

The LAMIN1 algorithm simply seeks to minimize the smoothed Lagrangian $\mathcal{L}^\beta_\mu(Q(\boldsymbol{w}), \boldsymbol{\lambda}_\mu)$ with SGD, based on unbiased estimator of $\nabla_{\boldsymbol{w}}\mathcal{L}^\beta_\mu(Q(\boldsymbol{w}), \boldsymbol{\lambda}_\mu)$ (which can be analytically computed). See pseudo-code in Appendix E.1.

As with all SGD algorithms with unbiased gradient estimator, LAMIN1 is guaranteed to converge to a local optimum of the smoothed Lagrangian for any differentiable Q-model (subject to standard assumptions (Goodfellow et al., 2016)). Moreover, LAMIN1 can converge to globally *optimal Q-function* in some "simple" cases, such as when tabular models are used (even though the smoothed Lagrangian function is *not* convex in that case):

Proposition 4. *LAMIN1 converges to an optimal Q-function with respect to objective* (1) *if $Q(\boldsymbol{w})$ is a tabular model with $Q(s,a;\boldsymbol{w}) = \boldsymbol{w}_{s,a}$.*

Proposition 4 implies that the β-smoothing trick in LAMIN1 is more than a heuristic for approximate optimization, but may also serve as a practical *correction* to an inherent bias of the minimax points of the Lagrangian that we will discuss in Section 5. See Appendix E.1 for more nuanced discussion on LAMIN1 and its optimality property.

LAMIN2: Another common trick is to construct "local" gradient estimator for the (original) Lagrangian $\mathcal{L}_\mu$, where the estimation is unbiased, in a per-step sense, for "most" of the optimization steps. Specifically, let $\boldsymbol{w}_t$ be the parameter vector that is to be updated at SGD step t, we can estimate the value of $\nabla_{\boldsymbol{w}}[\mathcal{B}Q - Q](s,a;\boldsymbol{w})\,\big|_{\boldsymbol{w}=\boldsymbol{w}_t}$, for this particular parameter $\boldsymbol{w}_t$, with

$$\begin{aligned}&\nabla_{\boldsymbol{w}}\,[\mathcal{B}Q - Q](s,a;\boldsymbol{w})\,\big|_{\boldsymbol{w}=\boldsymbol{w}_t}\\ \approx &\mathop{\mathbf{E}}_{S'\sim P(s,a)}\ \mathop{\mathbf{E}}_{A'\sim\pi_{Q(\boldsymbol{w}_t)}(S')}\left[\nabla_{\boldsymbol{w}}\,\delta(s,a,S',A';\boldsymbol{w})\right]\big|_{\boldsymbol{w}=\boldsymbol{w}_t}\end{aligned} \tag{15}$$

where $\pi_{Q(\boldsymbol{w}_t)}$ is the $Q(\boldsymbol{w}_t)$-greedy policy. Note that the greedy policy $\pi_{Q(\boldsymbol{w}_t)}$ in (15) does not depend on $\boldsymbol{w}$ and thus is invariant to $\nabla_{\boldsymbol{w}}$; in contrast, the Boltzmann policy $\pi^\beta_{\boldsymbol{w}}$ in LAMIN1 (see (14)) will be differentiated by $\nabla_{\boldsymbol{w}}$.

We remark that despite the approximation sign in (15), the LAMIN2 estimator is expected to enjoy *exact* consistency for "most" of the time in the SGD process. Specifically,

Proposition 5. (15) *becomes exact equality if for the given Q-function parameter $\boldsymbol{w}_t$, $Q(\boldsymbol{w}_t)$ suggests a unique best action $a_{\max}(s;\boldsymbol{w}_t) \doteq \arg\max_{a\in\mathcal{A}}\ Q(s,a;\boldsymbol{w}_t)$ for state s.*

See the proof in Appendix E.2. In practice, the condition in Proposition 5 – i.e. that $Q(\boldsymbol{w}_t)$ gives unique best actions – shall be the normal case for large-scale and continuously-valued models, as it should be rare that the real-numbered Q-values of two actions would happen to be *identical* under a model parameter that is being stochastically optimized. [6]

Finally, the local gradient estimator (15) can be further combined with the β-smoothing trick, that is, replacing the greedy policy $\pi_{Q(\boldsymbol{w}_t)}$ in (15) with the Boltzmann policy $\pi^\beta_{Q(\boldsymbol{w}_t)}$. See the resulted pseudo-code in Appendix E.2. Empirically, modestly higher temperatures help slightly improve the performance of LAMIN2 in our experiments.

[5] When only data from a sub-optimal policy is available, the policy (although sub-optimal with respect to the true task performance) *is* optimal to an shaped imitation reward, so that our algorithm implicitly learns for this shaped reward, leading to policies as good as the given policy (in the best case).

[6] The same argument applies to all ReLU neural networks too.

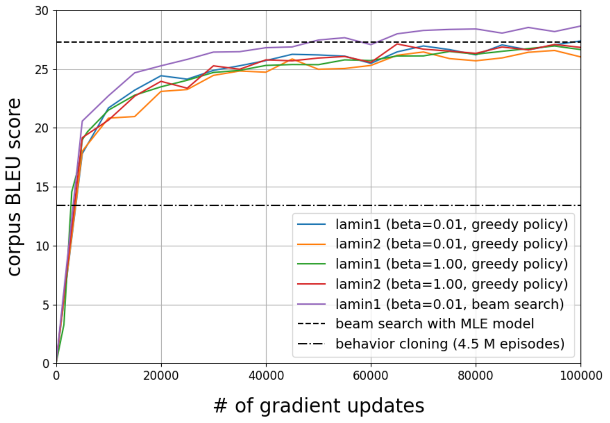

Figure 1. Learning curves of LAMIN. BLEU scores are measured on held-out test set. The dashed line indicates the standard baseline performance of 27.3, achieved by Vaswani et al. (2017) with a beam-search policy (beam size =4) using standard supervised learning. The purple line runs the same beam-search policy but with model trained by LAMIN1. Other lines use greedy policy, which is about 2x faster than search. The dash-dot line indicates the performance of behavior cloning, which only achieves 13.4.

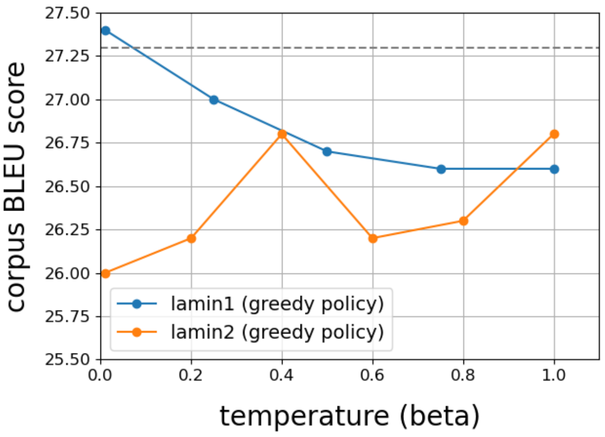

Figure 2. Performance of LAMIN1 and LAMIN2 under different temperature β. Greedy policy is used in all cases.

4.2. Applications to Machine Translation

As a case study, we now apply LAMIN algorithms to Machine Translation (MT), which is an application with significant real-world impact (Wu et al., 2016), and is also an excellent example of episodic learning problem: A translation episode starts with a given sentence of a source language, and the agent takes actions to generate translation tokens one by one sequentially. The episode terminates when the agent outputs a special end-of-sentence (EOS) token, at which point the translation quality is evaluated. See Appendix E.3 for the complete ELP formulation of machine translation.

Given a source sentence X, most MT metrics measure the quality of a generated translation Y by comparing the similarity between Y and some *reference translation* Z of X, where Z is provided by human expert. It follows that the policy used by the human expert (which maps X to Z) must be an optimal policy under such metric. A trajectory of this optimal policy, in the form of a sequence of source-reference sentence pairs, is indeed widely available in standard MT benchmarks, which can be readily used to power LAMIN algorithms. In general, the same idea applies to all machine learning problems where the performance metric is a similarity evaluation against reference/groundtruth outputs.

We tested our algorithms using the WMT 2014 English→German dataset, one of the most influential MT benchmarks. We parameterized value function Q with the standard TransformerBase model (Vaswani et al., 2017), which contains about 65 million model parameters. The action space consists of 37000 *unstructured* actions in each decision step (corresponding to 37000 tokens in the vocabulary of the target language). The training data consists of 4.5 million translation episodes made by human translators in European Parliament and multi-lingual news websites (Bojar et al., 2014). We trained the Transformer model using LAMIN1 and LAMIN2, with varying temperature β, then tested the Q-greedy policy with the standard BLEU metric. See Appendix E.4 for more details on experiment setup.

Figure 1 shows the learning curves of LAMIN1 and LAMIN2, under temperature 0.01 (as a close approximation of the exact Lagrangian function) and 1.0 (corresponding to the softmax distribution). We see that all the variants demonstrate similar learning curves. Notably, LAMIN1 with $\beta = 0.01$, which is slightly better among the variants, attains a BLEU score of 27.4 *with greedy policy*. In comparison, with standard supervised learning, the same model famously attains 27.3 on the same data set *only if* further combined with a systematic beam search over the solution space (Vaswani et al., 2017). On the other hand, when also incorporating with the same beam search procedure, the Q-function trained by LAMIN1 (with $\beta = 0.01$) attains 28.7, which is 1.4 BLEU (=5%) higher.

Notably, policy-based learning algorithms do not seem to work well in this task. As Figure 1 shows, behavior cloning, as a popular imitation learning baseline that mimics the expert policy via cross-entropy loss then samples the actions by the learned policy (Ho and Ermon, 2016), only achieves 13.4 BLEU (even being fed with 4.5 million demonstrations, a data size that is significantly larger than the typical ones used in many imitation learning research (Garg et al., 2021)). On the other hand, policy-based RL algorithms such as

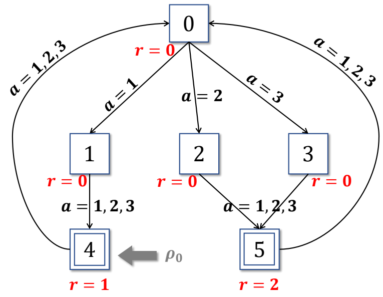

Figure 3. An example for the suboptimality of minimax Q-functions. State 4 and 5 are terminal states, from which any action leads to state 0. Choosing action $1, 2, 3$ under state 0 deterministically transits to state $1, 2, 3$, respectively. The agent only receives non-zero rewards at terminal states, with $R(4) = 1$, $R(5) = 2$.

policy gradient are known to have difficulties in getting effective learning on this benchmark (Choshen et al., 2020).

Finally, Figure 2 illustrates how the smoothing temperature β affects the performance of LAMIN1 and LAMIN2. Table 1 and 2 in Appendix E.4 reports the numerical scores. We see that LAMIN1 has slightly better performance than LAMIN2 under most temperatures. Interesting, higher temperature seems to help the performance of LAMIN2 but hurt that of LAMIN1 (albeit with limited margins in both cases).

5. Minimax Points vs Maximin Points

The Lagrangian method we explored so far has been focusing on a particular class of Lagrangian saddle points, the minimax points, i.e. $\arg\min_Q \max_{\boldsymbol{\lambda}} \mathcal{L}_\pi$. This follows a long tradition in the literature, dating back at least to (Puterman, 1994) but perhaps to (d'Epenoux, 1960; 1963), where the "habit" is to formulate the variational problem of Bellman equation into the minimization form, or the *primal-form* (Cho and Wang, 2017; Dai et al., 2018; Nachum and Dai, 2020), whose optimal solutions correspond to the minimax points of the Lagrangian. Let us call such Q-functions, **minimax Q-functions**.

Indeed, the Bellman value Q^*, as an optimal Q-function, is a minimax Q-function (Lemma 2.1), and approximate solutions to find minimax Q-functions appear to empirically perform well, as our results in Section 4.2 demonstrated. However, a minimax Q-function is not necessarily an optimal Q-function. Figure 3 gives an example. Clearly, an optimal policy in this ELP should only choose action 2 or 3 (or both), but not action 1, under state 0. On the other hand, one can verify that for this ELP, the constant Q-function $Q(s, a) \equiv 2$ is a minimax Q-function, which is certainly not optimal as it assigns the same Q-value to all actions under state 0. Moreover, the multiplier $\boldsymbol{\lambda}$ that counterparts with the constant Q-function (which together form a minimax point) needs to have $\boldsymbol{\lambda}(1, a) = 0$ for all a. Such a $\boldsymbol{\lambda}$ cannot encode a "dual policy" at all. See D.4 for more details.

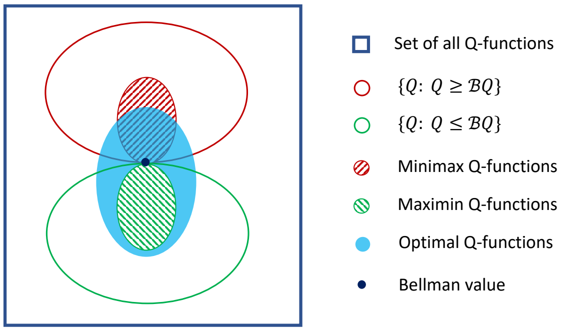

Figure 4. A landscape of the Q-space, which has combined conclusions from Lemma 2.1, Lemma 6.1, Figure 3, and Theorem 6.

In general, the problem with minimax Q-functions is that they only guarantee optimality for optimal actions, but do not enforce sub-optimality for sub-optimal actions (such as the action 1 under state 0 in Figure 3). We remark that this is not limited to Q-functions or episodic/finite-time settings, but is a general issue rooted from the primal-form variational formulation. For example, the minimax solutions of V-form Lagrangian in Discounted-MDPs, as studied in previous works (Cho and Wang, 2017; Dai et al., 2018), suffer from the same issue too (see Appendix D.4 for a counter-example). This deficiency of minimax Q-functions might explain our empirical observation in the last section that adding a small temperature in the Bellman operator of the Lagrangian helps with the learning (because the Boltzmann averaging allows, to some extent, the optimizer to downgrade sub-optimal actions to further lower the Lagrangian).

Interestingly enough, it turns out that another class of Lagrangian saddle points – the *maximin points* – can indeed guarantee the optimality of the corresponding Q-functions. Specifically, consider the following "mirrored" variational problem of the Bellman equation:

$$\begin{aligned} \max_{Q} \quad & \mathop{\mathbf{E}}_{\zeta\sim\pi} \Big[Q(S_T, A_T) \Big] \\ \text{s.t.} \quad & Q(s, a) \le \mathcal{B}Q(s, a) \quad , \quad \forall (s, a) \end{aligned} \tag{16}$$

Comparing with the primal-form variational problem (7), which asks to find a "minimal" Q from a set of "large" Q's (in the sense that $Q \ge \mathcal{B}Q$), the dual-form variational problem (16) asks to find a "maximal" Q from a set of "small" Q's (in the sense that $Q \le \mathcal{B}Q$). It is easy to see that the Bellman value Q^* is still an optimal solution of the dual-form problem, as the following lemma confirms (see the proof in Appendix D.1).

Lemma 6.1. *In any finite ELP, for any conjugate policy π, Q^* is an optimal solution of problem* (16)*, or equivalently,*

$$Q^* \in \arg \max_{Q\in\mathcal{Q}} \min_{\boldsymbol{\lambda}\ge 0} \mathcal{L}_\pi(Q, \boldsymbol{\lambda}).$$

We call an optimal solution of (16), a **maximin Q-function**. Note that minimax and maximin Q-functions, as defined by (7) and (16) (resp.), correspond to different types of saddle points of the *same* function $\mathcal{L}_\pi$. But different from minimax Q-functions, it turns out that a maximin Q-function always enforces sub-optimality for all (truly) sub-optimal actions *and* at the same time can guarantee optimality for at least one (truly) optimal action. As a result, a maximin Q-function *always and only* induces optimal policy.

Theorem 6. *In any finite ELP, for any conjugate policy π, let $Q_{\max}$ be a maximin Q-function with respect to the Lagrangian $\mathcal{L}_\pi$, then $Q_{\max}$ is an optimal Q-function.*

Proof idea: Because $Q_{\max}$ is a feasible solution of (16), we have $Q_{\max} \leq \mathcal{B}Q_{\max} \leq \mathcal{B}\mathcal{B}Q_{\max} \cdots \leq Q^*$, so $Q_{\max} \leq Q^*$, which implies that a sub-optimal action for Q^* (under a state) can only have even lower Q-values for $Q_{\max}$. Therefore, it is enough to prove that $\max_a Q_{\max}(s,a) = \max_a Q^*(s,a)$ at every non-terminal state *s that is reachable by a $Q_{\max}$-greedy policy* – in that case under any non-terminal state that the $Q_{\max}$-greedy policy may encounter, $Q_{\max}$ is giving the same Q-values with Q^* for optimal actions *and* is giving even lower Q-values for sub-optimal actions, so the induced greedy policy must always choose the optimal actions. Note that $Q_{\max}$'s values on terminal states do not affect its footprint, due to the ELP conditions.

The value equality at (reachable) optimal actions can be proved by induction, starting from the terminal states for which the equality of $Q_{\max}$ and Q^* is guaranteed by the objective function of (16). Then to enable a chain of induction, we will need to prove Proposition 13 in Appendix D.2, and use it as induction rule. See D.2 for the complete proof. □

Theorem 6 reveals an interesting symmetry breaking in the Lagrangian method for Q-function learning: From the same Bellman equation, there are two symmetric ways to form the variational problems, both leading to the same Lagrangian function, yet only one gives optimal Q-functions, the other does not (necessarily). Figure 4 gives a diagrammatic summary of this phenomenon. The asymmetry appears to be rooted from the asymmetric Bellman operator $\mathcal{B}$ which is only taking max, not min, over the Q-values. The maximin Q-functions have been largely overlooked in existing literature, but in light of their better optimality property as shown in this section, we believe that the maximin Q-functions, as well as the corresponding *dual-form* variational formulation (16), deserve more attentions from the community in future.

6. Related works

A main challenge of this work comes from the non-linearity in both the Bellman optimality equation and the associated Q-form Lagrangian being studied. In contrast, most related research focused on linear treatments to related objects. For a linear *policy-specific* Bellman operator $\mathcal{B}_\pi$ (which replaces $\max_a$ with a linear average), White (2017) proved that $\mathcal{B}_\pi$ has unique fixed point in episodic learning setting. The linear operator $\mathcal{B}_\pi$ leads to a LP re-formulation, based on which the "DICE" family of policy evaluation algorithms were developed (Nachum et al., 2019; Yang et al., 2020). On the policy optimization side, an active thread of research used saddle-point optimization to solve the linear V-form Lagrangian, again relying on the generic LP duality inherited from the linear treatment (Chen and Wang, 2016; Wang, 2017; Cho and Wang, 2017; Dai et al., 2018; Chen et al., 2018; Serrano and Neu, 2020; Si et al., 2004). The underlying techniques in these linear settings are not directly applicable to the nonlinear problems studied in this work.

A popular Inverse Reinforcement Learning (IRL) framework of imitation learning also uses minimax saddle-point optimization to learn optimal policy and value functions (Ho and Ermon, 2016). (Garg et al., 2021) developed a Q-function learning algorithm based on the IRL framework, which bears some similarity with the LAMIN algorithm developed in our paper. We however note that the rationale behind the two algorithms are completely different, and the Lagrangian method we discussed is not limited to the IRL problem.

The WMT machine translation benchmark used in this paper is a fruitful driver behind the rapid technical advances in Neural Machine Translation recent years (Wu et al., 2016; Koehn and Knowles, 2017; Vaswani et al., 2017). MDP-based techniques have been actively studied as a promising method for this problem (Ranzato et al., 2016; Edunov et al., 2018; Bahdanau et al., 2017; Wu et al., 2018), but with relatively limited empirical gain observed so far (Choshen et al., 2020). To our best knowledge, the LAMIN algorithm is one of the first MDP-based solutions that is able to train Transformer-scale neural networks *independently* (without the aid of other major techniques, such as pretraining or ensemble learning) to attain competitive performance on the WMT benchmark.

The disparity between the canonical discounted-reward formulation and common learning practice is a well recognized issue in reinforcement learning. The RL textbook (Sutton and Barto, 2018) devoted its Section 10.4 to the issue of deprecating the discounted formalism. The DP textbook (Bertsekas and Tsitsiklis, 1996) subsumed the discounted setting as a special case of an finite-termination setting. A special case of Theorem 1 dedicated to episodic discounting was proved by (Bertsekas and Tsitsiklis, 1991).

Acknowledgments

The author of the paper would like to thank Fei Yuan and Longtu Zhang for inspiring discussions on this work, as well as for their help in preparing the WMT experiment.

References

Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron Courville, and Yoshua Bengio. An actor-critic algorithm for sequence prediction. 2017.

Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005.

Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 2nd edition, 2000. ISBN 1886529094.

Dimitri P Bertsekas and John N Tsitsiklis. An analysis of stochastic shortest path problems. *Mathematics of Operations Research*, 16(3):580–595, 1991.

Dimitri P Bertsekas and John N Tsitsiklis. *Neuro-dynamic programming*. Athena Scientific, 1996.

Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006. ISBN 978-0387-31073-2.

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, et al. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the ninth workshop on statistical machine translation*, pages 12–58, 2014.

Huang Bojun. Steady state analysis of episodic reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 9335–9345, 2020.

Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Yichen Chen and Mengdi Wang. Stochastic primal-dual methods and sample complexity of reinforcement learning. *arXiv preprint arXiv:1612.02516*, 2016.

Yichen Chen, Lihong Li, and Mengdi Wang. Scalable bilinear pi learning using state and action features. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 834–843. PMLR, 10–15 Jul 2018.

Woon Sang Cho and Mengdi Wang. Deep primal-dual reinforcement learning: Accelerating actor-critic using bellman duality. *arXiv preprint arXiv:1712.02467*, 2017.

Leshem Choshen, Lior Fox, Zohar Aizenbud, and Omri Abend. On the weaknesses of reinforcement learning for neural machine translation. In *International Conference on Learning Representations*, 2020.

Bo Dai, Albert Shaw, Niao He, Lihong Li, and Le Song. Boosting the actor with dual critic. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=BkUp6GZRW.

Thomas Degris, Martha White, and Richard S Sutton. Off-policy actor-critic. *arXiv preprint arXiv:1205.4839*, 2012.

F d'Epenoux. Sur un probleme de production et de stockage dans l'aléatoire. *Revue Française de Recherche Opérationelle*, 14:3–16, 1960.

Francois d'Epenoux. A probabilistic production and inventory problem. *Management Science*, 10(1):98–108, 1963.

Sergey Edunov, Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. Classical structured prediction losses for sequence to sequence learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 355–364, 2018.

Divyansh Garg, Shuvam Chakraborty, Chris Cundy, Jiaming Song, and Stefano Ermon. Iq-learn: Inverse soft-q learning for imitation. *Advances in Neural Information Processing Systems*, 34, 2021.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

Shixiang Gu, Timothy Lillicrap, Ilya Sutskever, and Sergey Levine. Continuous deep q-learning with model-based acceleration. In *International Conference on Machine Learning*, pages 2829–2838, 2016.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pages 1861–1870, 2018.

Hado V Hasselt. Double q-learning. In *Advances in Neural Information Processing Systems*, pages 2613–2621, 2010.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems*, pages 4565–4573, 2016.

Philipp Koehn and Rebecca Knowles. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, 2017.

Donghwan Lee and Niao He. Stochastic primal-dual q-learning. *arXiv preprint arXiv:1810.08298*, 2018.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

Ofir Nachum and Bo Dai. Reinforcement learning via fenchel-rockafellar duality. *arXiv preprint arXiv:2001.01866*, 2020.

Ofir Nachum, Yinlam Chow, Bo Dai, and Lihong Li. Dualdice: Behavior-agnostic estimation of discounted stationary distribution corrections. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/file/cf9a242b70f45317ffd281241fa66502-Paper.pdf.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

Matt Post. A call for clarity in reporting bleu scores. In *Proceedings of the Third Conference on Machine Translation (WMT)*, pages 186–191, 2018.

Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. 1994.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. In *International Conference on Learning Representations*, 2016.

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

Joan Bas Serrano and Gergely Neu. Faster saddle-point optimization for solving large-scale markov decision processes. In Alexandre M. Bayen, Ali Jadbabaie, George Pappas, Pablo A. Parrilo, Benjamin Recht, Claire Tomlin, and Melanie Zeilinger, editors, *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, volume 120 of *Proceedings of Machine Learning Research*, pages 413–423, The Cloud, 10–11 Jun 2020.

Jennie Si, Andrew G. Barto, Warren Buckler Powell, and Don Wunsch. *The Linear Programming Approach to Approximate Dynamic Programming*, pages 153–178. 2004. doi: 10.1109/9780470544785.ch6.

Felix Stahlberg and Bill Byrne. On nmt search errors and model errors: Cat got your tongue? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3347–3353, 2019.

Rich Sutton, Ashique Rupam Mahmood, Doina Precup, and Hado Hasselt. A new q (lambda) with interim forward view and monte carlo equivalence. In *International Conference on Machine Learning*, pages 568–576, 2014.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Richard S Sutton, Joseph Modayil, Michael Delp, Thomas Degris, Patrick M Pilarski, Adam White, and Doina Precup. Horde: a scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 761–768, 2011.

Richard S Sutton, A Rupam Mahmood, and Martha White. An emphatic approach to the problem of off-policy temporal-difference learning. *The Journal of Machine Learning Research*, 17(1):2603–2631, 2016.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, L ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010, 2017.

Mengdi Wang. Randomized linear programming solves the discounted markov decision problem in nearly-linear (sometimes sublinear) run time. *Mathematics of Operations Research*, 2017.

CJCH Watkins. Learning from delayed rewards. *PhD thesis, King's College, University of Cambridge*, 1989.

Martha White. Unifying task specification in reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3742–3750. JMLR. org, 2017.

Lijun Wu, Fei Tian, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. A study of reinforcement learning for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3612–3621, 2018.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, L ukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144, 2016. URL http://arxiv.org/abs/1609.08144.

Mengjiao Yang, Ofir Nachum, Bo Dai, Lihong Li, and Dale Schuurmans. Off-policy evaluation via the regularized lagrangian. *Advances in Neural Information Processing Systems*, 33, 2020.

Fei Yuan, Longtu Zhang, Huang Bojun, and Yaobo Liang. Simpson's bias in nlp training. In *AAAI 2021*, 2021.

A. On the MDP Formulation

In our problem formulation, the rewards are conditioned on states, not explicitly on actions. However, the action dependency is implicitly captured by the action-conditioned state transition. In fact, given an MDP M with action-conditioned state-reward transition function $P(s', r'|s, a)$ (from which all kinds of action-based rewards can be defined(Sutton and Barto, 2018)), we can construct an MDP $\tilde{M}$ with states $\tilde{s} \doteq (s, r)$, state(-only) transition function $\tilde{P}(\tilde{s}'|\tilde{s}, a) = \tilde{P}\big((s', r')|(s, r), a\big) \doteq P(s', r'|s, a)$, and state-based reward function $\tilde{R}(\tilde{s}) = \tilde{R}\big(s, r\big) \doteq r$. It is clear that M and $\tilde{M}$ are encoding the same probability space of state-action-reward trajectories. The choice of state-based reward formulation is mostly for mathematical convenience.

Similarly, although the objective function assumed in this paper does not explicitly encompass temporal discounting, the canonical exponential reward discounting – if indeed required as a formulation of something real – can be equivalently captured as a discounting implicitly from stochastic termination, as many works have pointed out (e.g. see (Sutton et al., 2011; Bertsekas, 2000)). In the following we nevertheless provide a self-contained argument regarding the generality of our objective formulation.

Given a discounted-MDP M that never terminates (i.e. $\mathcal{S}_\perp = \emptyset$) and that uses the discounted-reward objective $\mathbf{E}_\pi[\sum_{t=1}^{\infty} R(S_t) \cdot \gamma^{t-1}]$, we can construct an MDP $\tilde{M}$ by adding a zero-reward terminal state $s_\perp$, with $R(s_\perp) = 0$, and modifying the transition function so that every state in M transits to $s_\perp$ with probability $1 - \gamma$ under any action. The modified MDP $\tilde{M}$ has finite average termination time $\mathbf{E}_\pi[\tilde{T}] = \sum_{l=2}^{\infty} l \cdot \gamma^{l-2}\,(1-\gamma) = \frac{\gamma}{1-\gamma} + 2$, thus $\tilde{M}$ is a finite-time MDP. Moreover, the total-reward objective (1) of $\tilde{M}$ equals the discounted-reward objective of M:

$$
\begin{aligned}
\mathbf{E}_\pi\Big[\sum_{t=1}^{T} R(\tilde{S}_t)\Big] =& \sum_{l=1}^{\infty}\Big(\,\mathbf{E}_\pi\Big[\sum_{t=1}^{l} R(\tilde{S}_t)|\tilde{T} = l\Big] \cdot \mathbf{P}_\pi[\tilde{T} = l]\,\Big) \\
=& \sum_{l=2}^{\infty}\Big(\,\underline{\mathbf{E}_\pi\Big[\sum_{t=1}^{l-1} R(\tilde{S}_t)|\tilde{T} = l\Big]} \cdot \mathbf{P}_\pi[\tilde{T} = l]\,\Big) \\
=& \sum_{l=2}^{\infty}\Big(\,\underline{\mathbf{E}_\pi\Big[\sum_{t=1}^{l-1} R(S_t)\Big]} \cdot \mathbf{P}_\pi[\tilde{T} = l]\,\Big) \\
=& \sum_{l=2}^{\infty}\Big(\,\mathbf{E}_\pi\Big[\sum_{t=1}^{l-1} R(S_t)\Big] \cdot \gamma^{l-2}\,(1-\gamma)\,\Big) \\
=& \mathbf{E}_\pi\Big[\sum_{m=1}^{\infty}\sum_{t=1}^{m} R(S_t) \cdot \gamma^{m-1}\,(1-\gamma)\Big] \\
=& \mathbf{E}_\pi\Big[\sum_{t=1}^{\infty}\Big(\sum_{m=t}^{\infty}\gamma^{m-1}\,(1-\gamma)\Big) \cdot R(S_t)\Big] \\
=& \mathbf{E}_\pi\Big[\sum_{t=1}^{\infty}\gamma^{t-1} \cdot R(S_t)\Big]
\end{aligned}
\tag{17}
$$

Therefore, a policy that maximizes the total reward in the finite-time MDP $\tilde{M}$ will also maximize the discounted reward in the original MDP M.

Note that in (17), we have used the fact that $\mathbf{P}_\pi[\tilde{T} = 1] = 0$ and that $R(s_\perp) = 0$. In the key step highlighted by the underlines, we have $\mathbf{E}_\pi[\sum_{t=1}^{l} R(\tilde{S}_t)|\tilde{T} = l + 1] = \mathbf{E}_\pi[\sum_{t=1}^{l} R(S_t)]$ because the condition $\tilde{T} = l + 1$ only rules out early-terminating trajectories – that is, rules out the possibility that some state $\tilde{S}_t$ for $t \leq l$ equals $s_\perp$ – but the condition does not alter the probability ratios between trajectories that do not early terminate. More specifically, let $s_{1..l}$ denote an arbitrary trajectory segment $s_1 \ldots s_l$ that does not early terminate (i.e. $s_t \neq s_\perp$ for $1 \leq t \leq l$), then

$$
\begin{aligned}
\mathbf{P}_\pi[\tilde{S}_{1..l} = s_{1..l}|\tilde{T} = l + 1] &= \frac{\mathbf{P}_\pi[\tilde{T} = l + 1|\tilde{S}_{1..l} = s_{1..l}]}{\mathbf{P}_\pi[\tilde{T} = l + 1]} \cdot \mathbf{P}_\pi[\tilde{S}_{1..l} = s_{1..l}] \\
&= \frac{1-\gamma}{\gamma^{l-1} \cdot (1-\gamma)} \cdot \mathbf{P}_\pi[\tilde{S}_{1..l} = s_{1..l}]
\end{aligned}
$$

$$= \frac{1}{\gamma^{l-1}} \cdot \mathbf{P}_\pi[\tilde{S}_t \neq s_\perp \text{ for } t = 1..l] \cdot \mathbf{P}_\pi[\tilde{S}_{1..l} = s_{1..l} | \tilde{S}_t \neq s_\perp \text{ for } t = 1..l]$$
$$= \mathbf{P}_\pi[\tilde{S}_{1..l} = s_{1..l} | \tilde{S}_t \neq s_\perp \text{ for } t = 1..l]$$
$$= \mathbf{P}_\pi[S_{1..l} = s_{1..l}]$$

B. Proofs of the Bellman Optimality under Generalized Discounting

In this section we prove Theorem 1, which asserts the uniqueness and optimality of the solution of generalized Bellman equation in episodic learning setting.

B.1. Properties of Episodic Learning Process

We start with presenting a series of mathematical properties of ELP; the first three are known, the rest are new. Most of our mathematical results in this paper are based on these properties.

Proposition 7. *((Bojun, 2020), Lemma 1.1) For any policy π in any ELP, the Markov chain induced by policy π is irreducible: Let s and s' be any two states reachable by π, $\sum_{\tau=1}^{\infty} \mathbf{P}_\pi[S_{t+\tau} = s' | S_t = s] > 0$.*

Proposition 8. *((Bojun, 2020), Lemma 1.2) For any policy π in any ELP, the Markov chain induced by policy π is positive recurrent: Let s be any state reachable by π, $\mathbf{E}_\pi[T_s] < \infty$, where T_s is the recurrent time of s in the markov chain of π.*

Proposition 9. *((Bojun, 2020), Theorem 4) For any policy π in any ELP, let $f : \mathcal{S} \to \mathbb{R}$ be a real-valued function over the states, we have*

$$\underset{S\sim\rho_\pi}{\mathbf{E}} \Big[f(S) \Big] = \underset{\zeta\sim\pi}{\mathbf{E}} \Big[\sum_{t=1}^{T} f(S_t) \Big] \Big/ \underset{\zeta\sim\pi}{\mathbf{E}} \Big[T \Big] \tag{18}$$

For two Q-functions $Q_1, Q_2 \in \mathcal{Q}$, we write $Q_1 \geq Q_2$ iff $Q_1(s, a) \geq Q_2(s, a), \forall (s, a) \in \mathcal{S} \times \mathcal{A}$. The following proposition confirms that the generalized Bellman optimality operator (2) is *monotonic*.

Proposition 10. *For any discounting function $\gamma : \mathcal{S} \to [0, 1]$, the generalized Bellman optimality operator $\mathcal{B}^\gamma$ is a monotonic operator over $\mathcal{Q}$, that is, $Q_1 \geq Q_2 \Rightarrow \mathcal{B}^\gamma Q_1 \geq \mathcal{B}^\gamma Q_2$ for all $Q_1, Q_2 \in \mathcal{Q}$.*

Proof. Rewrite (2) as $\mathcal{B}^\gamma Q(s, a) = \sum_{s'} \Big(P(s'|s, a) \cdot \gamma(s') \Big) \cdot \max_{a'} Q(s', a') + \sum_{s'} P(s'|s, a) R(s')$. As $Q_1(s', a') \geq Q_2(s', a')$ for all (s', a'), we have $\max_{a'} Q_1(s', a') \geq \max_{a'} Q_2(s', a')$ for all s', and thus $\sum_{s'} \Big(P(s'|s, a) \cdot \gamma(s') \Big) \cdot \max_{a'} Q_1(s', a') \geq \sum_{s'} \Big(P(s'|s, a) \cdot \gamma(s') \Big) \cdot \max_{a'} Q_2(s', a')$, where $P(s'|s, a) \cdot \gamma(s') \geq 0$ for all s'. □

Now, as a well-known special case, if the discounting function is a constant c less than 1, the corresponding Bellman optimality operator $\mathcal{B}_c$ is a *contraction mapping* with respect to the maximum-norm distance over $\mathcal{Q}$, which guarantees, by the Banach fixed-point theorem, that there is a special Q-function Q^*_c which is both the unique fixed point and the unique limiting point of the Bellman optimality operator (i.e. $Q^*_c = \mathcal{B}_c Q^*_c$ and $Q^*_c = \lim_{n\to\infty} (\mathcal{B}_c)^n Q$, $\forall Q \in \mathcal{Q}$).

Unfortunately, $\mathcal{B}^\gamma$ loses the above contraction property under general discounting.

Proposition 11. *In any ELP where there is a single state s^* and a single action a^* such that doing a^* under s^* only goes to non-terminal states s', i.e. $\sum_{s' \notin \mathcal{S}_\perp} P(s'|s^*, a^*) = 1$, the Bellman operator $\mathcal{B}$ with episodic discounting (3) is **not** a contraction mapping with respect to the maximum-norm distance: For some $Q_1, Q_2 \in \mathcal{Q}$, $\max_{s,a} |Q_1(s, a) - Q_2(s, a)| = \max_{s,a} |\mathcal{B}Q_1(s, a) - \mathcal{B}Q(s, a)|$.*

Proof. Consider two Q-functions with constant difference everywhere: $Q_1(s, a) \equiv Q_2(s, a) + \delta$, for some $\delta > 0$. Clearly, $\max_{s,a} |Q_1(s, a) - Q_2(s, a)| = \delta$. On the other hand, for any (s, a) we have $|\mathcal{B}Q_1(s, a) - \mathcal{B}Q_2(s, a)| = \sum_{s' \in \mathcal{S}} P(s'|s, a) \cdot \gamma(s') \cdot \delta \leq \delta$, which equals δ at the particular (s^*, a^*) because $\sum_{s' \in \mathcal{S}} P(s'|s^*, a^*) \cdot \gamma(s') \cdot \delta = \sum_{s' \notin \mathcal{S}_\perp} P(s'|s^*, a^*) \cdot 1 \cdot \delta = \delta$. □

Proposition 11 is a bad news for most non-trivial ELPs: It says that the generalized Bellman operators $\mathcal{B}^\gamma$ – and the episodic operator $\mathcal{B}$ in particular – is not a contraction mapping unless we always has a chance to immediately terminate an episode no matter where we are (even when we are at terminal states, at which point the episode has not effectively started yet!).

However, it turns out that for a large class of the generalized Bellman operators (including the episodic operator $\mathcal{B}$), they still enjoy unique fixed and limiting point in *all* finite ELPs, not because of the contraction property as in discounted-MDPs, but because of a graph property dedicated to the family of ELPs:

Proposition 12. *Given an Episodic Learning Process* $(\mathcal{S}, \mathcal{A}, P, R, \rho_0)$ *and a policy* π *in it, for any subset of states* $\Omega \subseteq \mathcal{S}$, *let* $\mathcal{C}_\pi(\Omega) \doteq \{\ s' :\ \exists s \in \Omega,\ \mathbf{P}_\pi[S_{t+1} = s' | S_t = s] > 0\ \}$ *be the set of all successor states that are one-step reachable from* Ω *under* π, *and let* $\mathcal{S}_\pi$ *be the set of states that are ever reachable under* π *(from initial states, in finite steps), then* $\mathcal{C}_\pi(\Omega) \subseteq \Omega$ *only if* $\mathcal{S}_\pi \subseteq \Omega$.

Proof. For contradiction suppose $\mathcal{C}_\pi(\Omega) \subseteq \Omega$ and yet there is a π-reachable state $s^* \in \mathcal{S}_\pi$ that is outside the given subset Ω. We will show that in this case, it is possible to construct a policy μ (that is possibly different from π) such that s^* is also reachable under μ, and that μ admits an infinite trajectory that passes through s^* and never return back to s^* (see Figure 5 below). This would contradict with Proposition 8 above which asserts that a μ-reachable state s^* must have finite mean recurrence time under μ.

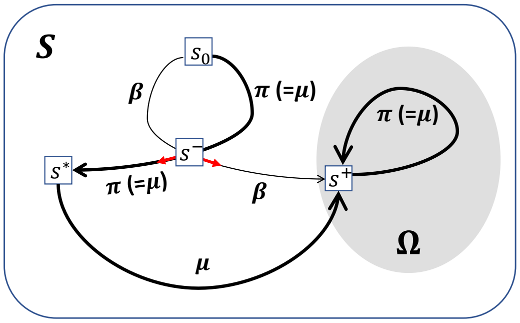

Figure 5. An μ-admissible trajectory that goes through s^* but never returns.

Specifically, first observe that $\mathcal{C}_\pi(\Omega) \subseteq \Omega$ means Ω is an absorbing subset under π so that once we get into Ω we would never get out. In this case, the existence of a π-reachable state $s^* \notin \Omega$ entails that Ω cannot contain all initial states, as otherwise from no initial state (in Ω) we can go outside the absorbing subset Ω to reach s^*. Let s_0 be such an initial state that is outside Ω, from which we can reach s^* under π (as assumed) without reaching any state in Ω in the middle (otherwise we never reach s^*).

Now, pick an arbitrary state s^+ in Ω, there must be some policy β (not necessarily π) under which we can reach s^+ from s_0 (as states unreachable under *any* policy should not be included in $\mathcal{S}$ in the first place, see Section 2). Without loss of generality we can again assume that we can reach s^+ from s_0 under β without going through any other state in Ω in the middle because otherwise we can simply re-define s^+ to be the first state in Ω that we have encountered on the path (from s_0 to the "old s^+").

So far we have obtained an initial state s_0 outside Ω, from which there is an admissible path $s_0 \xrightarrow{\pi} s^*$ for policy π, and an admissible path $s_0 \xrightarrow{\beta} s^+$ for policy β. Both paths only contain states outside Ω (except s^+). Now we construct policy μ as follows: we ask μ to copy π on states in the path $s_0 \xrightarrow{\pi} s^*$, and ask μ to copy β on states in the path $s_0 \xrightarrow{\beta} s^+$. If a state shows up in both paths – such as the state s^- in Figure 5 – we ask μ to be a (probability) mixture of both π and β on that state. Clearly, we can reach both s^* and s^+ from s_0 under μ (note that the policy mixing only decreases the probabilities to reach s^* and s^+ but does not change their reachability).

Since both s^* and s^+ are in $\mathcal{S}_\mu$, by Proposition 7, policy μ must admit a finite path from s^* to s^+ too. Note that so far we have only prescribed μ's behavior *outside* Ω. Our final step is to ask μ to copy π for all states in Ω, so that Ω is also an absorbing subset for μ, meaning that once we reach s^+ (from s^*), we will be stuck in Ω without going back to s^* (which is outside Ω by our assumption at the beginning of the proof). In this way, we have constructed a policy μ, under which we can first go from s_0 to s^*, then go from s^* to s^+, and then be stuck in Ω forever without returning to s^*. A possibility of such an infinite trajectory under μ directly contradicts with Proposition 8. □

B.2. Proof of Theorem 1 (1) (2)

Proposition 12 says that in ELPs, a subset of $\mathcal{S}$ can be an absorbing set under a policy only if it is "big" enough to have contained all reachable states of this policy. Consequently, there cannot exist an absorbing subset *outside* the reachable set $\mathcal{S}_\pi$. Utilizing this fact, we can prove the first two statements of Theorem 1.

(Theorem 1 (1) (2))**.** *In any ELP $(\mathcal{S}, \mathcal{A}, P, R, \rho)$ with finite state space $\mathcal{S}$ and finite action space $\mathcal{A}$, let γ be any discounting function such that $\gamma(s) < 1$ for all terminal state $s \in \mathcal{S}_\perp$, then*

(1) $\mathcal{B}^\gamma$ has a unique fixed point, i.e., the equation $Q = \mathcal{B}^\gamma Q$ has a unique solution.

(2) The fixed point of $\mathcal{B}^\gamma$ is also the limiting point of repeatedly applying $\mathcal{B}^\gamma$ to any $Q \in \mathcal{Q}$.

Proof. For any two Q-functions Q_1 and Q_2, consider their L_∞-distance

$$d(Q_1, Q_2) \doteq \max_{s \in \mathcal{S}} \; d_s(Q_1, Q_2)$$

where

$$d_s(Q_1, Q_2) \doteq \max_{a \in \mathcal{A}} \; |Q_1(s, a) - Q_2(s, a)|.$$

As usual, we have

$$\begin{aligned} d(\mathcal{B}^\gamma Q_1, \mathcal{B}^\gamma Q_2) &= \max_{s \in \mathcal{S}} \; \max_{a \in \mathcal{A}} \; \Big| \sum_{s' \in \mathcal{S}} P(s'|s, a) \cdot \gamma(s') \cdot \Big(\max_{a'_1} \; Q_1(s', a'_1) - \max_{a'_2} \; Q_2(s', a'_2) \Big) \Big| \\ &\leq \max_{s \in \mathcal{S}} \; \max_{a \in \mathcal{A}} \; \sum_{s' \in \mathcal{S}} P(s'|s, a) \cdot \gamma(s') \cdot \Big| \max_{a'_1} \; Q_1(s', a'_1) - \max_{a'_2} \; Q_2(s', a'_2) \Big| \\ &\leq \max_{s \in \mathcal{S}} \; \max_{a \in \mathcal{A}} \; \sum_{s' \in \mathcal{S}} P(s'|s, a) \cdot \gamma(s') \cdot \max_{a'} \Big| Q_1(s', a') - Q_2(s', a') \Big| \\ &= \max_{s \in \mathcal{S}} \; \max_{a \in \mathcal{A}} \; \sum_{s' \in \mathcal{S}} P(s'|s, a) \cdot \gamma(s') \cdot d_{s'}(Q_1, Q_2). \end{aligned} \tag{19}$$

Traditionally, it was assumed that $\gamma(s') \equiv \gamma_c < 1$ for all states, thus the $\gamma(s')$ term in (19) can be readily moved out of the sum, immediately yielding $d(\mathcal{B}Q_1, \mathcal{B}Q_2) \leq \gamma_c \cdot d(Q_1, Q_2)$. When $\gamma(s')$ is not constant and is allowed to be 1 for non-terminal states, applying the operator $\mathcal{B}^\gamma$ to Q_1 and Q_2 cannot guarantee to reduce $d(Q_1, Q_2)$, as discussed in Proposition 11.

However, by utilizing the graph property as proved in Proposition 12, we can show that $\mathcal{B}^\gamma$ guarantees to reduce the per-state distance $d_s(Q_1, Q_2)$ at some "support dimension" s, so that if we repeatedly apply $\mathcal{B}^\gamma$, the set of "support states" will become smaller and smaller and eventually become empty at which point the overall L_∞-distance gets reduced (by the composite operator "repeatedly applying $\mathcal{B}^\gamma$").

Specifically, for given $Q_1, Q_2 \in \mathcal{Q}$, we will identify a sequence of *proper* subsets of states

$$\mathcal{S} = \texttt{d-support}(0) \supset \texttt{d-support}(1) \supset \texttt{d-support}(2) \supset \cdots \supset \texttt{d-support}(|\mathcal{S}|) \tag{20}$$

such that

$$s \notin \texttt{d-support}(k) \;\Rightarrow\; \forall i \geq k, \; d_s\Big((\mathcal{B}^\gamma)^i Q_1, (\mathcal{B}^\gamma)^i Q_2 \Big) < d(Q_1, Q_2) \tag{21}$$

for all $k \geq 0$. The construction of the subsets is by induction, and is based on the following insight:

Lemma 2.1. *Under the condition of Theorem 1 , if (21) holds for k − 1, then*

$$\exists s^* \in \texttt{d-support}(k), \;\; \textit{such that} \; \forall i \geq k, \; d_{s^*}\Big((\mathcal{B}^\gamma)^i Q_1, (\mathcal{B}^\gamma)^i Q_2 \Big) < d(Q_1, Q_2)$$

Proof. We first refactor (19) a little bit, which actually holds in a per-state sense, so

$$\begin{aligned} d_s(\mathcal{B}^\gamma Q_1, \mathcal{B}^\gamma Q_2) &\leq \max_{a \in \mathcal{A}} \; \sum_{s' \in \mathcal{S}} P(s'|s, a) \cdot \gamma(s') \cdot d_{s'}(Q_1, Q_2) \\ &\leq \max_{s' \in \mathcal{S}} \; d_{s'}(Q_1, Q_2) \;\; = \;\; d(Q_1, Q_2). \end{aligned} \tag{22}$$

Recursively applying (22) gives

$$
\begin{aligned}
d_s\Big((\mathcal{B}^\gamma)^k Q_1, (\mathcal{B}^\gamma)^k Q_2\Big) &\le \max_{a\in\mathcal{A}} \sum_{s'\in\mathcal{S}} P(s'|s,a)\cdot\gamma(s')\cdot d_{s'}\Big((\mathcal{B}^\gamma)^{k-1}Q_1, (\mathcal{B}^\gamma)^{k-1}Q_2\Big)\\
&= \sum_{s'\in\mathcal{S}} P(s'|s,a_{\max}(s))\cdot\gamma(s')\cdot d_{s'}\Big((\mathcal{B}^\gamma)^{k-1}Q_1, (\mathcal{B}^\gamma)^{k-1}Q_2\Big)\\
&\le \sum_{s'\in\mathcal{S}} P(s'|s,a_{\max}(s))\cdot\gamma(s')\cdot d(Q_1,Q_2) \le d(Q_1,Q_2)
\end{aligned}
\tag{23}
$$

where $a_{\max}(s) \doteq \arg\max_a \ \sum_{s'\in\mathcal{S}} P(s'|s,a)\cdot\gamma(s')\cdot d_{s'}\Big((\mathcal{B}^\gamma)^{k-1}Q_1, (\mathcal{B}^\gamma)^{k-1}Q_2\Big)$.

Now we prove that the inequality (23) must be strict for some support-state $s^* \in \texttt{d-support}(k)$. In particular, we will prove

$$
\sum_{s'\in\mathcal{S}} P(s'|s^*,a_{\max}(s^*))\cdot\gamma(s')\cdot d_{s'}\Big((\mathcal{B}^\gamma)^{k-1}Q_1, (\mathcal{B}^\gamma)^{k-1}Q_2\Big) < d(Q_1,Q_2) \tag{24}
$$

Case 1: There is an $s^* \in \texttt{d-support}(k)$ with $\sum_{s'\in\texttt{d-support}(k)} P\big(s'|s^*,a_{\max}(s^*)\big) < 1$. In this case it's possible to go from s^* to some s' outside the subset of $\texttt{d-support}(k)$. For such s' we have $\gamma(s')\cdot d_{s'}\Big((\mathcal{B}^\gamma)^{k-1}Q_1, (\mathcal{B}^\gamma)^{k-1}Q_2\Big) < d(Q_1,Q_2)$ (because (21) holds for $k-1$ as assumed), thus

$$
\begin{aligned}
& \sum_{s'\in\mathcal{S}} P(s'|s^*,a_{\max})\cdot\gamma(s')\cdot d_{s'}\Big((\mathcal{B}^\gamma)^{k-1}Q_1, (\mathcal{B}^\gamma)^{k-1}Q_2\Big)\\
< & \sum_{s'\notin\texttt{d-support}(k)} P(s'|s^*,a_{\max})\cdot d(Q_1,Q_2) \ + \\
& \sum_{s'\in\texttt{d-support}(k)} P(s'|s^*,a_{\max})\cdot\gamma(s')\cdot d_{s'}\Big((\mathcal{B}^\gamma)^{k-1}Q_1, (\mathcal{B}^\gamma)^{k-1}Q_2\Big)\\
\le & \sum_{s'\notin\texttt{d-support}(k)} P(s'|s^*,a_{\max})\cdot d(Q_1,Q_2) + \sum_{s'\in\texttt{d-support}(k)} P(s'|s^*,a_{\max})\cdot d(Q_1,Q_2)\\
= \ & d(Q_1,Q_2).
\end{aligned}
$$

Case 2: For all $s \in \texttt{d-support}(Q_1,Q_2)$, $\sum_{s'\in\texttt{d-support}(Q_1,Q_2)} P\big(s'|s,a_{\max}(s)\big) = 1$. This is equivalent to say that there exists a policy – which would choose $a_{\max}(s)$ under the corresponding s – such that it is *impossible* to move from any state in $\texttt{d-support}(k)$ to a state outside $\texttt{d-support}(k)$ under this policy. In other words, let μ be such a policy, we have

$$
\mathcal{C}_\mu\Big(\texttt{d-support}(k)\Big) \subseteq \texttt{d-support}(k)
$$

which, by Proposition 12, entails

$$
\mathcal{S}_\mu \subseteq \texttt{d-support}(k), \tag{25}
$$

(25) literally says that the set of support-states $\texttt{d-support}(k)$, as an absorbing subset under μ as assumed in case 2, must contain all reachable states under μ. By the definition of ELP, these μ-reachable states must in turn contain at least one terminal state (otherwise we would not have finite episode under μ at all). Let $s_\perp \in \mathcal{S}_\mu \subseteq \texttt{d-support}(k)$ be such a reachable terminal state under μ. Since $s_\perp$ is reachable under μ (and μ chooses $a_{\max}(s)$ under each s), there must also be an $s^* \in \mathcal{S}_\mu$ such that $P(s_\perp|s^*,a_{\max}(s^*)) > 0$. Because $\gamma(s_\perp) < 1$ as assumed as the general condition of Theorem 1, we

have

$$\begin{aligned}
& \sum_{s' \in \mathcal{S}} P(s'|s^*, a_{\max}) \cdot \gamma(s') \cdot d_{s'}\Big((\mathcal{B}^\gamma)^{k-1} Q_1, (\mathcal{B}^\gamma)^{k-1} Q_2 \Big) \\
\le \; & \sum_{s' \in \mathcal{S}} P(s'|s^*, a_{\max}) \cdot \gamma(s') \cdot d(Q_1, Q_2) \\
= \; & \Big(P(s_\perp|s^*, a_{\max}) \cdot \gamma(s_\perp) + \sum_{s' \in \mathcal{S} \backslash \{s_\perp\}} P(s'|s^*, a_{\max}) \cdot \gamma(s') \Big) \cdot d(Q_1, Q_2) \\
< \; & \Big(P(s_\perp|s^*, a_{\max}) + \sum_{s' \in \mathcal{S} \backslash \{s_\perp\}} P(s'|s^*, a_{\max}) \cdot \gamma(s') \Big) \cdot d(Q_1, Q_2) \\
\le \; & \Big(P(s_\perp|s^*, a_{\max}) + \sum_{s' \in \mathcal{S} \backslash \{s_\perp\}} P(s'|s^*, a_{\max}) \Big) \cdot d(Q_1, Q_2) \\
= \; & d(Q_1, Q_2).
\end{aligned}$$

Now we have proved that $\exists s^* \in \texttt{d-support}(k),\ d_{s^*}\Big((\mathcal{B}^\gamma)^k Q_1, (\mathcal{B}^\gamma)^k Q_2 \Big) < d(Q_1, Q_2)$. It is straightforward to verify that the same proof idea applies to all $i > k$ too (in case 1, we still have $\gamma(s') \cdot d_{s'}\Big((\mathcal{B}^\gamma)^{i-1} Q_1, (\mathcal{B}^\gamma)^{i-1} Q_2 \Big) < d(Q_1, Q_2)$ because (21) holds for $k-1$; in case 2, the existence of $s_\perp$ is a graph property that is independent of how many times $\mathcal{B}^\gamma$ is applied). □

With Lemma 2.1, we can construct each subset $\texttt{d-support}(k)$ in (20) by removing the s^* from $\texttt{d-support}(k-1)$. It's clear that (21) will hold for the sequence of support subsets thus constructed. In particular, note that Lemma 2.1 holds for $k = 0$ without the inductive condition (that (21) holds for $k-1$) because in this case it's impossible to go outside $\texttt{d-support}(0) = \mathcal{S}$ as in Case 1, so only Case 2 is possible (and in this case the proof does not need the inductive condition).

Now, (20) implies that $\texttt{d-support}(|\mathcal{S}|)$ is empty set. Substituting this observation into (21), yields $d_s\Big((\mathcal{B}^\gamma)^{|\mathcal{S}|} Q_1, (\mathcal{B}^\gamma)^{|\mathcal{S}|} Q_2 \Big) < d(Q_1, Q_2)$ for all $s \in \mathcal{S}$, which means $d\Big((\mathcal{B}^\gamma)^{|\mathcal{S}|} Q_1, (\mathcal{B}^\gamma)^{|\mathcal{S}|} Q_2 \Big) < d(Q_1, Q_2)$. In other words, the existence of the d-support sequence satisfying (20) and (21) means that the composite operator of "repeatedly applying B^γ for $|\mathcal{S}|$ times" (i.e. $(\mathcal{B}^\gamma)^{|\mathcal{S}|}$) guarantees to reduce the L_∞-distance of every Q-function pair in $\mathcal{Q}$.

Moreover, note that the initial values of Q_1 and Q_2 do not determine how much percentage the distance between them will reduce – the values of Q_1 and Q_2 only affect what $a_{\max}$ is in (24), There are $|\mathcal{S}| \cdot |\mathcal{A}| \cdot |\mathcal{S}|$ possible transition probabilities in total, and $|\mathcal{S}|$ possible γ-values in (24), so no matter how $a_{\max}$ (which is a policy) and $\arg\max \mathbf{d}$ change over iterations, they just select a different subset from the $|\mathcal{S}|^3 \cdot |\mathcal{A}|$ possible terms. There are a (possibly very large yet) finite number of such subsets, thus when the sum of the subset is less than 1, there must be an absolute upper bound $1 - r_{\min}$ for the subset sum. This upper bound ratio of distance reduction may be extremely close to 1, especially after repeating the process for $|\mathcal{S}|$ times, but still, it is a definite upper bound smaller than 1, which is enough to make the composite operator $(\mathcal{B}^\gamma)^{|\mathcal{S}|}$ a (possibly very weak) contraction mapping, and thus admits a unique fixed point.

Our last step is to confirm that the unique fixed point of the composite operator $(\mathcal{B}^\gamma)^{|\mathcal{S}|}$ is also the *unique* fixed (and limiting) point of the original Bellman operator $\mathcal{B}^\gamma$. Clearly $\mathcal{B}^\gamma$ cannot have two fixed points, as otherwise their distance could not get reduced by repeatedly applying $\mathcal{B}^\gamma$ for $|\mathcal{S}|$ times, so the key is to show that $\mathcal{B}^\gamma$ does have *a* fixed point. In fact, we will prove the following slightly stronger result:

Lemma 2.2. *Under the condition of Theorem 1, let Q^* be the unique fixed point of $(\mathcal{B}^\gamma)^{|\mathcal{S}|}$,*

$$Q^* = \lim_{n \to \infty} (\mathcal{B}^\gamma)^n Q \;, \;\; \forall Q \in \mathcal{Q} \tag{26}$$

and

$$Q^* = \mathcal{B}^\gamma Q^* \tag{27}$$

Proof. For brevity we use $\mathcal{B}$ to denote the generalized Bellman operator in this proof.

We first prove that (26) holds for a special family of Q-functions, that is, for all Q-functions Q^- with $Q^- \leq \mathcal{B}Q^-$. [7] Specifically, because $\mathcal{B}$ is monotonic operator, for any such Q^- we have

$$Q^- \leq \mathcal{B}Q^- \leq (\mathcal{B})^2 Q^- \cdots \leq (\mathcal{B})^{|\mathcal{S}|}Q^- \cdots \leq (\mathcal{B})^{2\cdot|\mathcal{S}|}Q^- \cdots \leq Q^* \tag{28}$$

where Q^* is an supremum (but not necessarily the limit) of the sequence above because of the contraction property of $\mathcal{B}^{|\mathcal{S}|}$. Recall that $Q_1 \leq Q_2$ means $\forall s, a \;\; Q_1(s,a) \leq Q_2(s,a)$, so (28) implies that the Q-functions in it must have non-increasing distances to Q^*. On the other hand, as the subsequence $Q^-,\ (\mathcal{B})^{|\mathcal{S}|}Q^-,\ (\mathcal{B})^{2\cdot|\mathcal{S}|}Q^-,\ (\mathcal{B})^{3\cdot|\mathcal{S}|}Q^-, \ldots$ converges to Q^*, we know that for any $\epsilon > 0$, there exists an i^* such that $d\Big((\mathcal{B})^{i^*\cdot|\mathcal{S}|}Q^-, Q^*\Big) < \epsilon$, and thus $d\Big((\mathcal{B})^{i}Q^-, Q^*\Big) < \epsilon$ for all integer $i > i^*$ due to the monotonicity, which literally means that the overall sequence (28) also converges to Q^*.

Now we have obtained a Q-function (i.e. Q^-) starting from which (and only from which, for now) repeatedly applying $\mathcal{B}$ (rather than $(\mathcal{B})^{|\mathcal{S}|}$) will converge to Q^*. In other words, we have $Q^* = \lim_{n\to\infty} (\mathcal{B})^n Q^-$. As a result, we must also have $\mathcal{B}Q^* = \mathcal{B} \lim_{n\to\infty} (\mathcal{B})^n Q^- = \lim_{n\to\infty} (\mathcal{B})^n Q^- = Q^*$, thus we have proved $\mathcal{B}Q^* = Q^*$ (i.e. Q^* is *a* fixed point of $\mathcal{B}$).

The final step of the proof for Lemma 2.2 is to show that $\mathcal{B}$ has a limiting point regardless of the initial Q-function (previously we have only proved this for the special initial Q-function Q^-): For any $Q \in \mathcal{Q}$, because $Q^* = \mathcal{B}Q^*$, we have $d(\mathcal{B}Q, Q^*) = d(\mathcal{B}Q, \mathcal{B}Q^*) \geq d(Q, Q^*)$, where the inequality is by (19). Again, $d(\mathcal{B}Q, Q^*) \geq d(Q, Q^*)$ means the sequence (28) has non-increasing distances to Q^*, but this time for all $Q \in \mathcal{Q}$. So, by the same logic as before, Q^* must be the limit of the sequence $Q, \mathcal{B}Q, (\mathcal{B})^2 Q, \ldots,$ for all $Q \in \mathcal{Q}$. □

The specific form of the γ-function does not play a role in the proof of Lemma 2.2 as presented above, so the conclusion of the lemma – i.e. (26) and (27) – apply to all $\mathcal{B}^\gamma$ that comply with the condition of Theorem 1, which marks the completion of the proof for statement (1) and (2) in Theorem 1. □

In comparison, the classic Bellman optimality property requires $\gamma(s) < 1$ at *every* state $s \in \mathcal{S}$, while Theorem 1 only requires $\gamma(s) < 1$ at terminal states. On the other hand, the classic result applies to all MDPs, while Theorem 1 is fundamentally based on unique structures of episodic learning process. Importantly, the episodic discounting function that we use in ELP – i.e. (3) – does satisfy the condition of Theorem 1.

B.3. Proof of Theorem 1 (3)

Now we prove statement (3) of Theorem 1 which asserts that Q^*, as the unique fixed point of the Bellman operator under episodic discounting, is indeed an optimal Q-function:

(Theorem 1 (3)). *In any finite ELP, let Q^* be the fixed point of $\mathcal{B}$ (i.e. the solution of (4)), let π^* be a policy such that $\pi^*(a|s) > 0$ only if $Q^*(s,a) = \max_{\bar{a}} Q^*(s,\bar{a})$, then $J(\pi^*) = \max_\pi J(\pi)$, where J is the episodic-reward objective (1).*

Proof. Every policy π is coupled with a (unique) **on-policy value function** Q_π which is a special Q-function that assigns Q-values according to the conditional expectations of the episode-wise total reward under the policy π:

$$Q_\pi(s,a) \doteq \mathop{\mathbf{E}}_{\{S_t,A_t\}\sim\pi} \left[\sum_{t=1}^{T} R(S_t) \Big| S_0 = s, A_0 = a \right] \quad , \quad \forall (s,a) \in \mathcal{S}\times\mathcal{A} \tag{29}$$

Comparing (29) with the definition of the episodic-reward objective J (i.e. with (1)), and with the ELP conditions, one can find that for any policy π, its performance score equals its on-policy value *at terminal states* (and all terminal states must have the same on-policy value):

$$J(\pi) = Q_\pi(s_\perp, a) \quad , \quad \forall s_\perp \in \mathcal{S}_\perp \, , \, \forall a \in \mathcal{A}. \tag{30}$$

In the following we will prove that for the Q^*-greedy policy π^*, we have $Q_{\pi^*} \geq Q_\pi$ for all π, which by (30) entails that $J(\pi^*) \geq J(\pi)$.

[7] Such Q^- guarantees to exist. In fact, every *on-policy value function* is such a Q^-; see B.3 for details.

First observe that $Q_{\pi^*} = Q^*$, that is, the Q-function Q^* is the on-policy value function of the greedy policy induced by itself. Specifically, with the episodic γ-function (3), for any (s,a), by the definition of π^* we have

$$\begin{aligned}
Q^*(s,a) &= \mathcal{B}Q^*(s,a) \\
&= \mathop{\mathbf{E}}_{S'\sim P(s,a)} \max_{a'} \Big[R(S') + \gamma(S')\cdot Q^*(S',a')\Big] \\
&= \mathop{\mathbf{E}}_{S'\sim P(s,a)} \mathop{\mathbf{E}}_{A'\sim\pi^*(S')} \Big[R(S') + \gamma(S')\cdot Q^*(S',A')\Big] && (31)\\
&= \mathop{\mathbf{E}}_{\{S_1,A_1\}\sim\pi^*} \Big[R(S_1) + \gamma(S_1)\cdot Q^*(S_1,A_1)\Big| S_0=s, A_0=a\Big] \\
&= \mathop{\mathbf{E}}_{\{S_1,A_1,S_2,A_2\}\sim\pi^*} \Big[R(S_1) + \gamma(S_1)R(S_2) + \gamma(S_1)\gamma(S_2)\cdot Q^*(S_2,A_2) \;\Big|\; S_0=s, A_0=a\Big] \\
&= \mathop{\mathbf{E}}_{\{S_t,A_t\}\sim\pi^*} \Big[\sum_{t=1}^{T} R(S_t)\Big| S_0=s, A_0=a\Big] \\
&= Q_{\pi^*}(s,a) && (32)
\end{aligned}$$

In above, the equations from (31) to (32) apply to not only π^* but to any policy π too. This means

$$\begin{aligned}
Q_\pi(s,a) &= \mathop{\mathbf{E}}_{S'\sim P(s,a)} \mathop{\mathbf{E}}_{A'\sim\pi(S')} \Big[R(S') + \gamma(S')\cdot Q_\pi(S',A')\Big] \\
&\le \mathop{\mathbf{E}}_{S'\sim P(s,a)} \max_{a'} \Big[R(S') + \gamma(S')\cdot Q_\pi(S',a')\Big] \\
&= \mathcal{B}Q_\pi(s,a)
\end{aligned}$$

So, for any policy π, we have $Q_\pi \le \mathcal{B}Q_\pi$. In other words, $\{Q : \exists\pi, Q = Q_\pi\}$, the set of all on-policy value functions, is a subset of the set $\{Q : Q \le \mathcal{B}Q\}$.

Now we have known that Q^* is a maximum Q-function of the set $\{Q \le \mathcal{B}Q\}$, and that this set contains the set of on-policy value functions as a subset. Because Q^* itself is an on-policy value function (as $Q^* = Q_{\pi^*}$), it follows that Q^* must also be a maximum Q-function of the subset $\{Q_\pi\}$. Thus we have proved that $Q_{\pi^*} = Q^* \ge Q_\pi$ for all π, as desired. □

C. Proofs of the Nonlinear Lagrangian Duality (Section 3)

C.1. Proof of Lemma 2.1

(Lemma 2.1). *In any finite ELP, for any conjugate policy π,* $Q^* \in \arg\min_{Q\in\mathcal{Q}} \max_{\boldsymbol{\lambda}\ge 0} \mathcal{L}_\pi(Q,\boldsymbol{\lambda})$.

Proof. Because $\mathcal{B}$ is monotonic (Proposition 10), for any Q with $Q \ge \mathcal{B}Q$ we have $Q \ge \mathcal{B}Q \ge (\mathcal{B})^2 Q \cdots \ge Q^*$, thus $Q^* \le Q$ for all $Q \in \mathcal{Q}$. On the other hand, the objective function of the variational problem (7) is a probabilistic average over the Q-values at terminal states/actions, which must attain its minimum at Q^* because Q^* is per-state-action minimal. In other words, Q^* is an optimal solution of the variational problem (7).

By standard Lagrangian duality theory, a Q-function is an optimal solution of (7) if and only if it is a minimax solution of the Lagrangian (8). This is because only Q-functions with $Q \ge \mathcal{B}Q$ can prevent $\max_{\boldsymbol{\lambda}\ge 0}\mathcal{L}_\pi(Q,\boldsymbol{\lambda})$ from being tuned arbitrarily large (by $\boldsymbol{\lambda}$), and for those Q's that satisfy the constraint (i.e. the complementary slackness condition), the second term in the Lagrangian would equal zero, rendering $\max_{\boldsymbol{\lambda}\ge 0}\mathcal{L}_\pi(Q,\boldsymbol{\lambda}) = \mathbf{E}_{\zeta\sim\pi}\Big[Q(S_T,A_T)\Big]$, which attains its minimum at Q^* as just proved. □

C.2. Proof of Lemma 2.2

(Lemma 2.2). *In any finite ELP, let $\mathcal{L}_\pi$ be the Lagrangian with conjugate policy π, and let $\boldsymbol{\lambda}_\pi$ be the particular Lagrangian multiplier with $\boldsymbol{\lambda}_\pi(s,a) = \rho_\pi(s)\cdot\pi(a|s)\cdot\mathbf{E}_\pi[T]$, where ρ_π is the stationary distribution of π, then*

$$\mathcal{L}_\pi(Q,\boldsymbol{\lambda}_\pi) = J(\pi) + \sum_{s\notin\mathcal{S}_\perp}\sum_{a\in\mathcal{A}} \boldsymbol{\lambda}_\pi(s,a)\left(\max_{\bar{a}} Q(s,\bar{a}) - Q(s,a)\right)$$

Proof. Define $f(s) \doteq \mathbf{E}_{A\sim\pi(s)}\left[\mathbb{1}[s \in \mathcal{S}_\perp]\cdot Q(s,A)\right]$, by Proposition 9 we have

$$\mathop{\mathbf{E}}_{\zeta\sim\pi}\left[Q(S_T,A_T)\right] = \mathop{\mathbf{E}}_{S_{1..T}\sim\pi}\left[\sum_{t=1}^{T} f(S_t)\right] = \mathbf{E}_{\zeta\sim\pi}[T]\cdot \mathop{\mathbf{E}}_{S\sim\rho_\pi}\mathop{\mathbf{E}}_{A\sim\pi(S)}\left[\mathbb{1}[S\in\mathcal{S}_\perp]\cdot Q(S,A)\right].$$

So, for any $Q \in \mathcal{Q}$, we have

$$\begin{aligned}
&\mathcal{L}_\pi(Q,\boldsymbol{\lambda}_\pi)\\
=&\mathbf{E}_{\zeta\sim\pi}[T]\cdot \mathop{\mathbf{E}}_{S,A\sim\rho_\pi}\left[\mathbb{1}[S\in\mathcal{S}_\perp]\cdot Q(S,A)\right] + \mathbf{E}_{\zeta\sim\pi}[T]\cdot \mathop{\mathbf{E}}_{S,A\sim\rho_\pi}\left[\mathcal{B}Q(S,A) - Q(S,A)\right]\\
=&\mathbf{E}_{\zeta\sim\pi}[T]\cdot \mathop{\mathbf{E}}_{S,A\sim\rho_\pi}\left[\mathbb{1}[S\in\mathcal{S}_\perp]\cdot Q(S,A) - Q(S,A)\right]+\\
&\mathbf{E}_{\zeta\sim\pi}[T]\cdot \mathop{\mathbf{E}}_{S,A\sim\rho_\pi}\left[\mathop{\mathbf{E}}_{S'\sim P(S,A)}\left[R(S') + \gamma(S')\cdot\max_{a'}Q(S',a')\right]\right]\\
=&-\mathbf{E}_{\zeta\sim\pi}[T]\cdot \mathop{\mathbf{E}}_{S,A\sim\rho_\pi}\left[\mathbb{1}[S\notin\mathcal{S}_\perp]\cdot Q(S,A)\right] + \underline{\mathbf{E}_{\zeta\sim\pi}[T]\cdot \mathop{\mathbf{E}}_{S'\sim\rho_\pi}}\left[R(S') + \gamma(S')\cdot\max_{a'}Q(S',a')\right]\\
=&\underline{\mathbf{E}_{\zeta\sim\pi}[T]\cdot \mathop{\mathbf{E}}_{S\sim\rho_\pi}\left[R(S)\right]} + \mathbf{E}_{\zeta\sim\pi}[T]\cdot \mathop{\mathbf{E}}_{S,A\sim\rho_\pi}\left[\mathbb{1}[S\notin\mathcal{S}_\perp]\cdot\left(\max_a Q(S,a) - Q(S,A)\right)\right]\\
=&\underline{J(\pi)} + \mathbf{E}_{\zeta\sim\pi}[T]\cdot\sum_{s\in\mathcal{S},a\in\mathcal{A}}\rho_\pi(s)\cdot\pi(a|s)\cdot\mathbb{1}[s\notin\mathcal{S}_\perp]\cdot\left(\max_{\bar a}Q(s,\bar a) - Q(s,a)\right)\\
=&J(\pi) + \mathbf{E}_{\zeta\sim\pi}[T]\cdot\sum_{s\in\mathcal{S}\setminus\mathcal{S}_\perp}\sum_{a\in\mathcal{A}}\rho_\pi(s)\cdot\pi(a|s)\cdot\left(\max_{\bar a}Q(s,\bar a) - Q(s,a)\right)
\end{aligned}$$

In above, $\mathbf{E}_{\zeta\sim\pi}[T]\cdot\mathbf{E}_{S\sim\rho_\pi}\left[R(S)\right] = J(\pi)$ is obtained by applying the transformation of Proposition 9 again, this time with $f(s)\doteq R(s)$. □

C.3. Proof of Theorem 2

(ELP Minimax Theorem)**.** *In any finite ELP $(\mathcal{S},\mathcal{A},P,R,\rho)$, if μ is an optimal policy, then its conjugate Lagrangian $\mathcal{L}_\mu$ has strong duality property, for which*

$$\min_{Q\in\mathcal{Q}}\ \max_{\boldsymbol{\lambda}\geq 0}\ \mathcal{L}_\mu(Q,\boldsymbol{\lambda}) = \max_{\boldsymbol{\lambda}\geq 0}\ \min_{Q\in\mathcal{Q}}\ \mathcal{L}_\mu(Q,\boldsymbol{\lambda}) = J(\mu)$$

Proof. Let π^* be a Q^*-greedy policy, which is thus an optimal policy. For any conjugate policy π, since Q^* is a minimax solution of $\mathcal{L}_\pi$ (Lemma 2.1), we have

$$\min_Q\max_{\boldsymbol{\lambda}\geq 0}\mathcal{L}_\pi(Q,\boldsymbol{\lambda}) = \max_{\boldsymbol{\lambda}\geq 0}\mathcal{L}_\pi(Q^*,\boldsymbol{\lambda}) = \mathop{\mathbf{E}}_{\zeta\sim\pi}[Q^*(S_T,A_T)].$$

By (32) in B.3, we have $\mathbf{E}_\pi[Q^*(S_T,A_T)] = \mathbf{E}_\pi[Q_{\pi^*}(S_T,A_T)]$. By (30) in B.3, we further have $\mathbf{E}_\pi[Q_{\pi^*}(S_T,A_T)] = J(\pi^*)$, even for $\pi\neq\pi^*$. Connecting these equations together, gives

$$\min_Q\max_{\boldsymbol{\lambda}\geq 0}\ \mathcal{L}_\pi(Q,\boldsymbol{\lambda}) = J(\pi^*)\quad,\ \forall\pi. \tag{33}$$

Again, for any conjugate policy π, due to Lemma 2.2, the Lagrangian function has the dual form (9) under the particular multiplier $\boldsymbol{\lambda}_\pi$, where (9) is copied below for convenience of presentation:

$$\mathcal{L}_\pi(Q,\boldsymbol{\lambda}_\pi) = J(\pi) + \sum_{s\notin\mathcal{S}_\perp}\sum_{a\in\mathcal{A}}\boldsymbol{\lambda}_\pi(s,a)\cdot\left(\max_{\bar a}Q(s,\bar a) - Q(s,a)\right)$$

In the second term above, both $\boldsymbol{\lambda}_\pi(s,a)$ and $\max_{\bar a}Q(s,\bar a) - Q(s,a)$ are non-negative for any Q and π, so it attains its minimum, which is zero, when Q achieves complementary slackness with $\boldsymbol{\lambda}_\pi$. On the other hand, the first term above, i.e.

$J(\pi)$, does not change with Q. So, the sum of the two terms, i.e. $\mathcal{L}_\pi(Q, \boldsymbol{\lambda}_\pi)$, will attain its minimum when the second term is zero, that is,

$$\min_{Q\in\mathcal{Q}} \; \mathcal{L}_\pi(Q, \boldsymbol{\lambda}_\pi) = J(\pi) \quad , \quad \forall\pi. \tag{34}$$

Now set the conjugate policy π in (33) and (34) to the optimal policy μ, as assumed in the theorem, we have

$$\max_{\boldsymbol{\lambda}\geq 0}\min_{Q\in\mathcal{Q}} \; \mathcal{L}_\mu(Q, \boldsymbol{\lambda}) \geq \min_{Q\in\mathcal{Q}} \; \mathcal{L}_\mu(Q, \boldsymbol{\lambda}_\mu) = J(\mu) = J(\pi^*) = \min_{Q}\max_{\boldsymbol{\lambda}\geq 0} \; \mathcal{L}_\mu(Q, \boldsymbol{\lambda}).$$

Because of the *weak minimax duality* (which universally holds for any function), we also have $\max_{\boldsymbol{\lambda}} \min_Q \mathcal{L}_\mu(Q, \boldsymbol{\lambda}) \leq \min_Q \max_{\boldsymbol{\lambda}} \mathcal{L}_\mu(Q, \boldsymbol{\lambda})$, which means the above inequality must actually be an equality, as desired. □

C.4. Proof of Proposition 3

(Proposition 3). *Given a finite ELP, for any Q-function Q and any policy π, let $\rho_\pi(s,a) = \rho_\pi(s)\;\pi(a|s)$ and $\boldsymbol{\lambda}_\pi(s,a) = \rho_\pi(s,a)\;\mathbf{E}_\pi[T]$, we have*

$$\mathcal{L}_\pi(Q, \bar{\boldsymbol{\lambda}}) \leq \mathcal{L}_\pi(Q, \boldsymbol{\lambda}_\pi) \leq \mathcal{L}_\pi(\bar{Q}, \boldsymbol{\lambda}_\pi) \quad , \quad \forall \bar{Q}, \bar{\boldsymbol{\lambda}}$$

if and only if

(1) $\mathcal{B}Q(s,a) - Q(s,a) \leq 0 \qquad , \quad \forall(s,a) \in \mathcal{S}\times\mathcal{A}$

(2) $\rho_\pi(s,a) \cdot \Big(\mathcal{B}Q(s,a) - Q(s,a)\Big) = 0 \qquad , \quad \forall(s,a) \in \mathcal{S}\times\mathcal{A}$

(3) $\rho_\pi(s,a) \cdot \Big(\max_{\bar{a}} Q(s,\bar{a}) - Q(s,a)\Big) = 0 \quad , \quad \forall s \notin \mathcal{S}_\perp, a \in \mathcal{A}$

The "if" part is straightforward: Condition (1) and (2) immediately gives $\mathcal{L}_\pi(Q, \boldsymbol{\lambda}_\pi) = \mathbf{E}_{\zeta\sim\pi}[Q(S_T, A_T)] = \max_{\bar{\boldsymbol{\lambda}}\geq 0} \mathcal{L}_\pi(Q, \bar{\boldsymbol{\lambda}})$. On the other hand, condition (3) means that the second term in the dual-form Lagrangian (9) is zero, so $\mathcal{L}_\pi(Q, \boldsymbol{\lambda}_\pi) = J(\pi)$. By (34) in C.3, we have $\min_{\bar{Q}} \; \mathcal{L}_\pi(\bar{Q}, \boldsymbol{\lambda}_\pi) = J(\pi)$, thus $\mathcal{L}_\pi(Q, \boldsymbol{\lambda}_\pi) = \min_{\bar{Q}} \mathcal{L}_\pi(\bar{Q}, \boldsymbol{\lambda}_\pi)$.

Now we prove the "only if" part, for which we resort to the general saddle-point condition: Under *given* conjugate policy π, a $(Q, \boldsymbol{\lambda})$ pair is a minimax saddle-point of function $\mathcal{L}_\pi(Q, \boldsymbol{\lambda})$ if and only if

(i) $\min\limits_{\bar{Q}\in\mathcal{Q}} \max\limits_{\bar{\boldsymbol{\lambda}}\geq 0} \mathcal{L}_\pi(\bar{Q}, \bar{\boldsymbol{\lambda}}) = \max\limits_{\bar{\boldsymbol{\lambda}}\geq 0} \min\limits_{\bar{Q}\in\mathcal{Q}} \mathcal{L}_\pi(\bar{Q}, \bar{\boldsymbol{\lambda}}) = \mathcal{L}_\pi(Q, \boldsymbol{\lambda})$,

(ii) $Q \in \arg\min\limits_{\bar{Q}\in\mathcal{Q}} \max\limits_{\bar{\boldsymbol{\lambda}}\geq 0} \mathcal{L}_\pi(\bar{Q}, \bar{\boldsymbol{\lambda}})$,

(iii) $\boldsymbol{\lambda} \in \arg\max\limits_{\bar{\boldsymbol{\lambda}}\geq 0} \min\limits_{\bar{Q}\in\mathcal{Q}} \mathcal{L}_\pi(\bar{Q}, \bar{\boldsymbol{\lambda}})$.

By condition (ii), if $(Q, \boldsymbol{\lambda}_\pi)$ form a minimax equilibrium, then Q must be a minimax solution of $\mathcal{L}_\pi$. From C.1 we know that such minimax Q-function must have $Q \geq \mathcal{B}Q$, which gives condition (1).

By condition (i), and by (33) in C.3, we have $\mathcal{L}_\pi(Q, \boldsymbol{\lambda}_\pi) = \min_{\bar{Q}\in\mathcal{Q}} \max_{\bar{\boldsymbol{\lambda}}\geq 0} \mathcal{L}_\pi(\bar{Q}, \bar{\boldsymbol{\lambda}}) = J(\pi^*) = \mathbf{E}_{\zeta\sim\pi}[Q(S_T, A_T)]$. In other words, $\sum_{s,a} \lambda_\pi(s,a) \; \big(\mathcal{B}Q(s,a) - Q(s,a)\big)$, as the second term in $\mathcal{L}_\pi(Q, \boldsymbol{\lambda}_\pi)$, must be zero in this circumstance. Because $\lambda_\pi(s,a) = \rho_\pi(s,a) \cdot \mathbf{E}_{\zeta\sim\pi}[T]$, where $\mathbf{E}_{\zeta\sim\pi}[T] > 0$, and further because $\mathcal{B}Q \leq Q$ as just proved, the only way to make the term zero is to have $\rho_\pi(s,a) \cdot \big(\mathcal{B}Q(s,a) - Q(s,a)\big) = 0$ at every (s,a) pair, which gives condition (2).

Moreover, since $\mathcal{L}_\pi(Q, \boldsymbol{\lambda}_\pi) = \min_{\bar{Q}} \mathcal{L}_\pi(\bar{Q}, \boldsymbol{\lambda}_\pi)$ as assumed, and $\mathcal{L}_\pi(Q, \boldsymbol{\lambda}_\pi) = J(\pi)$ due to (34) in C.3, we know that the second term in the dual form of $\mathcal{L}_\pi(Q, \boldsymbol{\lambda}_\pi)$ must be zero, that is, $\sum_{s\notin\mathcal{S}_\perp} \sum_{a\in\mathcal{A}} \rho_\pi(s,a) \cdot \mathbf{E}_{\zeta\sim\pi}[T] \cdot \big(\max_{\bar{a}} Q(s,\bar{a}) - Q(s,a)\big) = 0$. Again, because $\mathbf{E}_{\zeta\sim\pi}[T] > 0$ and $\max_{\bar{a}} Q(s,\bar{a}) - Q(s,a) \geq 0$ for all (s,a), the only possibility is to have $\rho_\pi(s,a) \cdot \big(\max_{\bar{a}} Q(s,\bar{a}) - Q(s,a)\big) = 0$ for each $(s,a) \in \mathcal{S} \setminus \mathcal{S}_\perp \times \mathcal{A}$, which gives condition (3).

D. Proofs of the Minimax-Maximin Symmetry Breaking (Section 5)

D.1. Proof of Lemma 6.1

(Lemma 6.1). *In any finite ELP, Q^* is an optimal solution of*

$$\max_{Q} \quad \mathop{\mathbf{E}}_{\zeta\sim\pi}\Big[Q(S_T, A_T)\Big] \quad \textit{s.t.} \quad Q(s,a) \leq \mathcal{B}Q(s,a) \ \ , \ \ \forall(s,a)$$

for any conjugate policy π. Equivalently, $Q^* \in \arg\max_{Q\in\mathcal{Q}} \min_{\boldsymbol{\lambda}\geq 0} \mathcal{L}_\pi(Q,\boldsymbol{\lambda})$.

The proof is by a symmetric argument with the one for Lemma 2.1 (see C.1): Because $\mathcal{B}$ is monotonic, for Q with $Q \leq \mathcal{B}Q$ we have $Q \leq \mathcal{B}Q \leq (\mathcal{B})^2 Q \cdots \leq Q^*$, thus Q^* maximizes the objective $\mathbf{E}_{\zeta\sim\pi}[Q(S_T, A_T)]$ in a per-state-action manner.

D.2. Proof of Theorem 6

(Theorem 6). *In any finite ELP, for any conjugate policy π, let $Q_{\max}$ be an maximin Q-function with respect to the Lagrangian $\mathcal{L}_\pi$, – i.e. let $Q_{\max}$ be an optimal solution of* (16) *– then $Q_{\max}$ is an optimal Q-function, in the sense that $Q_{\max}$-greedy policy maximizes the total-reward objective* (1).

Proof. First observe that

$$Q_{\max}(s,a) \leq Q^*(s,a) \quad , \ \forall(s,a)\in\mathcal{S}\times\mathcal{A} \tag{35}$$

which is because $Q_{\max}$, as a feasible solution of (16), has $Q_{\max} \leq \mathcal{B}Q_{\max} \leq \mathcal{B}\mathcal{B}Q_{\max} \cdots \leq Q^*$. Let μ be a $Q_{\max}$-greedy policy, so $\mu(a|s) > 0$ only if $Q_{\max}(s,a) = \max_{\bar{a}} Q_{\max}(s,\bar{a})$. Let $\mathcal{S}_\mu \subseteq \mathcal{S}$ be the set of states reachable by policy μ. As described in the proof idea, we will focus on proving that

$$\max_a Q_{\max}(s,\bar{a}) = \max_a Q^*(s,a) \quad , \quad \forall s\in\mathcal{S}_\mu\setminus\mathcal{S}_\perp \tag{36}$$

which would necessarily imply that

$$\arg\max_a Q_{\max}(s,a) \subseteq \arg\max_a Q^*(s,a) \quad , \quad \forall s\in\mathcal{S}_\mu\setminus\mathcal{S}_\perp . \tag{37}$$

Note that (36) entails (37) because, by (35), for any action a sub-optimal to Q^*, it can only have even lower Q-value in $Q_{\max}$, with $Q_{\max}(s,a) \leq Q^*(s,a) < \max_{\bar{a}} Q^*(s,\bar{a}) = \max_{\bar{a}} Q_{\max}(s,\bar{a})$, where $Q_{\max}(s,a) < \max_{\bar{a}} Q_{\max}(s,\bar{a})$ (for the Q^*-suboptimal action a) guarantees that such an a cannot be $Q_{\max}$-optimal either. (37) guarantees that μ, as a $Q_{\max}$-greedy policy, will only choose Q^*-optimal actions at every non-terminal state it may encounter since time $t \geq 1$. Such a μ is equivalently to a Q^*-greedy policy, thus is also an optimal policy. Note that μ's choices on terminal states does not matter here as state-transitions in terminal steps are action-agnostic, due to the ELP condition.

Now, to prove (36), we first prove the following induction rule:

Proposition 13. *Under the context of Theorem 6, let (s,a) be an arbitrary state-action pair, and let*

$$(\mathcal{S}\times\mathcal{A})_{\textit{next}} \doteq \{(s',a')\in\mathcal{S}\times\mathcal{A}:\ s'\notin\mathcal{S}_\perp \ \ \mathtt{and} \ \ P(s'|s,a)\cdot\mu(a'|s') > 0\}$$

denote the set of all the non-terminal (s',a') pairs that can directly follow (s,a) under the $Q_{\max}$-greedy policy μ, then

$$\begin{aligned} & Q_{\max}(s,a) = Q^*(s,a) \\ \Rightarrow\ & Q_{\max}(s',a') = Q^*(s',a') \quad \mathtt{and} \ \ \max_{\bar{a}} Q_{\max}(s',\bar{a}) = \max_{\bar{a}} Q^*(s',\bar{a}) \ \ , \ \ \forall(s',a')\in(\mathcal{S}\times\mathcal{A})_{\textit{next}} \end{aligned} \tag{38}$$

Proof. Because $Q_{\max} \leq Q^*$ and $Q_{\max} \leq \mathcal{B}Q_{\max}$, we have

$$\begin{aligned} Q_{\max}(s,a) &\leq \mathcal{B}Q_{\max}(s,a) \\ &= \mathbf{E}_{S'}[R(S')] + \sum_{s'\in\mathcal{S}} P(s'|s,a)\cdot\gamma(s')\cdot\max_{\bar{a}\in\mathcal{A}} Q_{\max}(s',\bar{a}) \\ &\leq \mathbf{E}_{S'}[R(S')] + \sum_{s'\in\mathcal{S}} P(s'|s,a)\cdot\gamma(s')\cdot\max_{\bar{a}\in\mathcal{A}} Q^*(s',\bar{a}) \\ &= \mathcal{B}Q^*(s,a) \\ &= Q^*(s,a) \end{aligned} \tag{39}$$

The premise $Q_{\max}(s,a) = Q^*(s,a)$ in the induction rule (38) entails that the two inequality signs in above must be equality, among which the second one – i.e. the one leading (39) – can be equality only if

$$\sum_{s' \in (\mathcal{S}\times\mathcal{A})_{\text{next}}} P(s'|s,a) \cdot \max_{\bar{a}\in\mathcal{A}} Q_{\max}(s',\bar{a}) = \sum_{s' \in (\mathcal{S}\times\mathcal{A})_{\text{next}}} P(s'|s,a) \cdot \max_{\bar{a}\in\mathcal{A}} Q^*(s',\bar{a}) \tag{40}$$

where $s' \in (\mathcal{S}\times\mathcal{A})_{\text{next}}$ is a slight abuse of notation which means s' shows up in $(\mathcal{S}\times\mathcal{A})_{\text{next}}$ (in the form of a (s',a') pair, with some a'), or equivalently, $s' \in (\mathcal{S}\times\mathcal{A})_{\text{next}}$ means that $s' \notin \mathcal{S}_\perp$ and $P(s'|s,a) > 0$).

Now observe that for (40) to hold, the only possibility is that

$$\max_{\bar{a}\in\mathcal{A}} Q_{\max}(s',\bar{a}) = \max_{\bar{a}\in\mathcal{A}} Q^*(s',\bar{a}) \quad , \quad \forall s' \in (\mathcal{S}\times\mathcal{A})_{\text{next}} \tag{41}$$

as otherwise for those s' on which (41) do not hold, it can only be $\max_{\bar{a}\in\mathcal{A}} Q_{\max}(s',\bar{a}) < \max_{\bar{a}\in\mathcal{A}} Q^*(s',\bar{a})$ (because $Q_{\max} \le Q^*$); those s' must all have positive weights in (40) (by definition of $(\mathcal{S}\times\mathcal{A})_{\text{next}}$), and thus will cause a real loss at the LHS of (40) (and importantly, no other state in $(\mathcal{S}\times\mathcal{A})_{\text{next}}$ could claim a "gain" to compensate this loss, again because $Q_{\max} \le Q^*$). (41) is exactly the second consequence in the induction rule (38).

Next, to prove $Q_{\max}(s',a') = Q^*(s',a')$ for all $(s',a') \in (\mathcal{S}\times\mathcal{A})_{\text{next}}$, the first consequence in the induction rule (38)), we notice that for any of such (s',a') we have

$$\max_{\bar{a}\in\mathcal{A}} Q_{\max}(s',\bar{a}) = Q_{\max}(s',a') \le Q^*(s',a') \le \max_{\bar{a}\in\mathcal{A}} Q^*(s',\bar{a}) \tag{42}$$

in which $\max_{\bar{a}\in\mathcal{A}} Q_{\max}(s',\bar{a}) = Q_{\max}(s',a')$ is because a' is by definition a $Q_{\max}$-greedy action under s', and $Q_{\max}(s',a') \le Q^*(s',a')$ is (once again) because $Q_{\max} \le Q^*$.

By (41) we know that the two ends of (42) actually equal to each other, so the inequalities in between must also be equality, and in particular $Q_{\max}(s',a') = Q^*(s',a')$, as desired. □

Proposition 13 enables us to prove (36) by induction (which is enough to prove the whole theorem, as argued above). Specifically, because both $Q_{\max}$ and Q^* are optimal solutions of (16), and because the objective in (16) is a distribution over only the terminal states, it follows that $Q_{\max}$ and Q^* must be equal on at least one terminal state $s_\perp$. Starting from this terminal state $s_\perp$ – as well as an arbitrary action $a_\perp$ under it – we have $Q_{\max}(s_\perp,a_\perp) = Q^*(s_\perp,a_\perp)$, thus by the induction rule of Proposition 13 we obtain $\max_{\bar{a}} Q_{\max}(s',\bar{a}) = \max_{\bar{a}} Q^*(s',\bar{a})$ and $Q_{\max}(s',a') = Q^*(s',a')$ for all (s',a') in the $(\mathcal{S}\times\mathcal{A})_{\text{next}}$ set with respect to $(s,a) = (s_\perp,a_\perp)$; the latter enables us to expand the induction proof to all non-terminal states that are reachable by μ. □

D.3. An counter-example showing that a minimax Q-function can be sub-optimal in ELPs

In this subsection we elaborate more about the counter-example as illustrated by Figure 3 in Section 5 (the figure is copied above). In this ELP, $\mathcal{S} = \{0,1,2,3,4,5\}$, $\mathcal{A} = \{1,2,3\}$. State 4 and 5 are terminal states, from which any action leads to state 0. Choosing action $1,2,3$ under state 0 deterministically transits to state $1,2,3$, respectively. All actions under state 1 lead to state 4, and all actions under state 2 and 3 lead to state 5. The agent only receives non-zero rewards at terminal states, with $R(4) = 1$, $R(5) = 2$. The initial state at time 0 is set to state 4 (i.e. $\rho_0(s) > 0$ only if $s = 4$).

The Bellman fixed-point Q^* for this ELP is as follows:

- $Q^*(0,1) = 1, \;\; Q^*(0,2) = 2, \;\; Q^*(0,3) = 2$
- $Q^*(1,a) = 1, \; \forall a$
- $Q^*(2,a) = Q^*(3,a) = 2, \; \forall a$
- $Q^*(4,a) = Q^*(5,a) = 2, \; \forall a$

An optimal policy of this ELP should only choose action 2 or 3, but not action 1, under state 0.

For minimax Q-functions, denoted by $Q_{\min}$, as they are optimal solutions of (7), we have

$$Q_{\min}(4,a) \quad = \quad 2 \quad \ge \quad \max \left\{ \begin{array}{lclcl} Q_{\min}(0,1) & \ge & \max_a Q_{\min}(1,a) & \ge & 1 \\ Q_{\min}(0,2) & \ge & \max_a Q_{\min}(2,a) & \ge & 2 \\ Q_{\min}(0,3) & \ge & \max_a Q_{\min}(3,a) & \ge & 2 \end{array} \right.$$

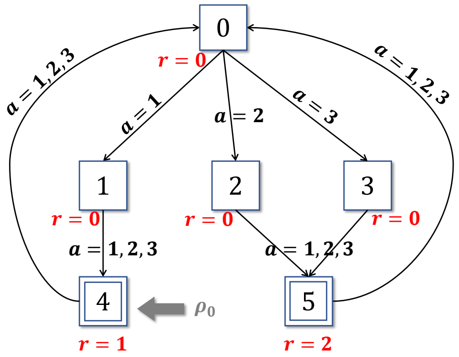

Figure 6. A copy of Figure 3

The above minimax condition only imposes tight bounds for Q^*-optimal actions (e.g. action 2 and 3 under state 0), but leaves "flexibility" for actions sub-optimal to Q^* (e.g. action 1 under state 0) as well as for *all* state-action pairs that follow an sub-optimal action (e.g. all actions under state 1). The consequence is that even constant value function $Q_{\min}(s,a) \equiv 2$ can be a minimax value function in this example, which is clearly sub-optimal, as discussed in Section 5.

In contrast, for maximin Q-functions, denoted by $Q_{\max}$, they are optimal solutions of (16), thus

$$Q_{\max}(4,a) \quad = \quad 2 \quad \leq \quad \max \begin{cases} Q_{\max}(0,1) & \leq & \max_a Q_{\max}(1,a) & \leq & 1 \\ Q_{\max}(0,2) & \leq & \max_a Q_{\max}(2,a) & \leq & 2 \\ Q_{\max}(0,3) & \leq & \max_a Q_{\max}(3,a) & \leq & 2 \end{cases}$$

We see that the maximin condition manages to imposes tight bounds for *at least one* Q^*-optimal action while at the same time can enforce *all* Q^*-sub-optimal actions to be still suboptimal to $Q_{\max}$. For example under state 0, $Q_{\max}(0,1)$ cannot exceed 1, while either $Q_{\max}(0,2)$ or $Q_{\max}(0,3)$ needs to be tight (i.e. $Q_{\max}(0,3) = 2$, or $Q_{\max}(0,2) = 2$, or both) so as to keep the maximum of the three no less than 2, as required.

Note that for both $Q_{\min}$ and $Q_{\max}$, the Lagrangian multiplier $\boldsymbol{\lambda}$ that forms equilibrium/saddle points with each of them (resp.) may not encode a policy, in general. In this example, for the constant $Q_{\min}$, we have $2 = Q_{\min}(0,1) = 0 + 1 \cdot Q_{\min}(1,a) > 1 + 0 \cdot Q_{\min}(4,a) = 1$, so $Q_{\min}(1,a) > \mathcal{B}Q_{\min}(1,a) = 1$ for all a, in which case its equilibrium multiplier $\boldsymbol{\lambda}$ has to have $\boldsymbol{\lambda}(1,a) = 0$ for all a, due to the equilibrium condition (2) proved in Proposition 3. Such an "all-zero" $\boldsymbol{\lambda}$ (on state 1) cannot be normalized into a policy.

Similarly, for the following specific maximin Q-function

- $Q_{\max}(4,a) = Q_{\max}(5,a) = 2$
- $Q_{\max}(0,1) = Q_{\max}(1,a) = 1$
- $Q_{\max}(0,2) = Q_{\max}(2,a) = 2$
- $Q_{\max}(0,3) = 1$
- $Q_{\max}(3,a) = 1.5$

we have $1 = Q_{\max}(0,3) < 0 + 1 \cdot Q_{\max}(3,a) < 2 + 0 \cdot Q_{\max}(5,a) = 2$, so $Q_{\max}(3,a) < \mathcal{B}Q_{\max}(3,a) = 2$ again for all a, so the multiplier corresponding to this $Q_{\max}$ still needs to be all zero at state 3 due to the complementary slackness condition, thus cannot be normalized at state 3.

D.4. A counter-example showing that a minimax V-function can be sub-optimal in discounted-MDPs

Moreover, the problems with minimax points of the Lagrangian, as demonstrated above, are not limited to Q-functions or to ELPs only, but seem to be fundamental issues rooted from the minimax structure. To see this, consider the *discounted-MDP* as shown in Figure 7 below.

To be strictly aligned with the related literature (Dai et al., 2018; Cho and Wang, 2017), the rewards are assigned to state-action pairs in this example. In this discounted-MDP, $\mathcal{S} = \{0,1,2,3\}$, $\mathcal{A} = \{1,2\}$. The initial state is set to state 0

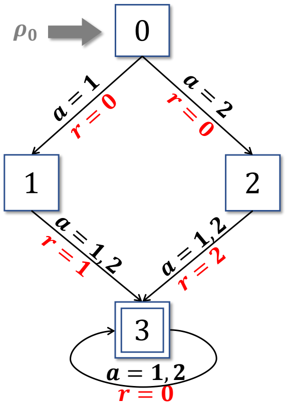

Figure 7. An example of discounted-MDP for studying saddle points of the V-form Lagrangian.

(i.e. $\rho_0(s) = 1$ if $s = 0$, otherwise $\rho_0(s) = 0$). From state 0, taking action $1, 2$ will deterministically goes to state $1, 2$, respectively, with zero reward obtained in this step. From state 1, any action leads to the absorbing state 3, with reward $R(1, a) = 1$ obtained (for all a). From state 2, any action leads to the same absorbing state 3, but with reward $R(2, a) = 2$ obtained (for all a). The absorbing state 3 will loop into itself forever, with zero reward obtained. An optimal policy in this discounted-MDP should choose action 2, not action 1, under state 0. The discounting constant γ is set to 0.5, so the optimal discounted-reward performance is $V^*(0) = 1$.

The V-form Lagrangian of the discounted-MDP above is: (Dai et al., 2018; Cho and Wang, 2017)

$$(1-\gamma) \cdot \mathop{\mathbf{E}}_{S_0 \sim \rho_0} \Big[V(S_0) \Big] + \sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} \lambda(s,a) \cdot \Big(R(s,a) + \gamma \cdot \mathop{\mathbf{E}}_{S' \sim P(s,a)} \Big[V(S') \Big] - V(s) \Big) \tag{43}$$

A minimax V-function $V_{\min}$ of the V-form Lagrangian (43) is an optimal solution of the Linear Programming problem (6), which has inspired some recently proposed RL algorithms (see Section 3). In this example, the LP can be more explicitly written as

$$\begin{aligned} \min_{V} \quad & 0.5 \cdot V(0) \\ \text{s.t.} \quad & V(0) \geq 0.5 \cdot V(1) \\ & V(0) \geq 0.5 \cdot V(2) \\ & V(1) \geq 0.5 \cdot V(3) + 1 \\ & V(2) \geq 0.5 \cdot V(3) + 2 \\ & V(3) \geq 0.5 \cdot V(3) \end{aligned}$$

for the LP above, a possible optimal solution is: $V_{\min}(0) = 1,\ \ V_{\min}(1) = 2,\ \ V_{\min}(2) = 2,\ \ V_{\min}(3) = 0$, which assigns the same value to action 1 and 2 under state 0, thus is not an optimal V-function.

E. Lagrangian Minimization for Machine Translation

E.1. The LAMIN1 Algorithm

As mentioned in Section 4, the idea of LAMIN1 is to minimize the smoothed Lagrangian $\mathcal{L}^{\beta}_{\mu}$ (with a small yet definite β) based on an unbiased gradient estimator of (14). For convenience, we copy (14) below:

$$\mathcal{L}^{\beta}_{\mu}(Q(\boldsymbol{w}), \boldsymbol{\lambda}_{\mu}) = \mathbf{E}_{\mu}[Q(S_T, A_T; \boldsymbol{w})] + \mathbf{E}_{\mu}[T] \cdot \mathop{\mathbf{E}}_{S,A,S' \sim \rho_{\mu}} \ \mathop{\mathbf{E}}_{A' \sim \pi^{\beta}_{Q(\boldsymbol{w})}(S')} \Big[\delta(S, A, S', A'; \boldsymbol{w}) \Big]$$

where $\pi^{\beta}_{Q(\boldsymbol{w})}(a|s) \doteq \frac{exp\left(Q(s,a;\boldsymbol{w})/\beta\right)}{\sum_b exp\left(Q(s,b;\boldsymbol{w})/\beta\right)}$ is the Boltzmann distribution with temperature β, and $\delta(s, a, s', a'; \boldsymbol{w}) \doteq R(s') + \gamma_{\text{epi}}(s')Q(s', a'; \boldsymbol{w}) - Q(s, a; \boldsymbol{w})$ is the temporal-difference error. Algorithm 1 gives the psuedo-code of such an algorithm.

Algorithm 1 The LAMIN1 algorithm.

```
Input: A piece of rollout data made by an optimal policy, in the form of $\{s_t, a_t, r_t\}_{0,1,\dots,n}$, in which $t = T_1, T_2, \dots, T_k$ are termination steps; A parametric $Q(\boldsymbol{w})$ model with initial weight $\boldsymbol{w}_0$; Learning rate $\alpha$; Boltzmann temperature $\beta$.
for gradient update $i = 0, 1, 2, \dots$ do
    $\Delta\boldsymbol{w} \leftarrow \frac{1}{k}\sum_{t=\{T_1 \dots T_k\}} \nabla_{\boldsymbol{w}} Q(s_t, a_t; \boldsymbol{w}) \Big|_{\boldsymbol{w}=\boldsymbol{w}_i} +$
        $\frac{n}{k} \cdot \frac{1}{n} \sum_{t=0}^{n-1} \gamma_{\text{epi}}(s_{t+1}) \, \nabla_{\boldsymbol{w}} \Big( \sum_a \pi^{\beta}_{\boldsymbol{w}}(s_{t+1}, a) \cdot Q(s_{t+1}, a; \boldsymbol{w}) \Big) - \nabla_{\boldsymbol{w}} Q(s_t, a_t; \boldsymbol{w}) \Big|_{\boldsymbol{w}=\boldsymbol{w}_i}$
    $\boldsymbol{w}_{i+1} \leftarrow \boldsymbol{w}_i - \alpha \cdot \Delta\boldsymbol{w}$
end for
Output: A $Q(\boldsymbol{w})$-greedy policy.
```

Algorithm 1 samples the stationary distribution ρ_μ in (14) by averaging over the rollout data of k episodes, which is unbiased thanks to the ergodicity of episodic learning (Bojun, 2020). The average episode length $\mathbf{E}_\mu[T]$ is estimated by n/k, the total number of steps in the data divided by the number of episodes. The overall algorithm is thus a standard unbiased SGD procedure, which shares the generic convergence property of all SGD procedures (i.e. convergence to local minimum of (14) is guaranteed under properly annealed learning rate (Goodfellow et al., 2016)). We also remark that the reward term $R(s')$ in the smoothed Lagrangian (more specifically, in the TD-error of (14)) is "differentiated out" in LAMIN1 because it is independent of the model parameter $\boldsymbol{w}$. As a result, Algorithm 1 does not use the reward data at all.

As an implementation trick, the gradient computation in Algorithm 1, particularly for $\nabla_{\boldsymbol{w}}\Big(\sum_a \pi^{\beta}_{\boldsymbol{w}}(s_{t+1}, a) \cdot Q(s_{t+1}, a; \boldsymbol{w})\Big)$, can conveniently run on automatic differentiation libraries such as PyTorch by utilizing the following fact (for brevity and clarity, we omit the argument s_{t+1} in Q and π in the following):

$$
\begin{aligned}
\nabla \pi^{\beta}_{\boldsymbol{w}}(a) &= \nabla \exp\Big(\log \frac{e^{Q(a;\boldsymbol{w})/\beta}}{\sum_b e^{Q(b;\boldsymbol{w})/\beta}}\Big) = \pi^{\beta}_{\boldsymbol{w}}(a) \cdot \nabla\Big(\log \frac{e^{Q(a;\boldsymbol{w})/\beta}}{\sum_b e^{Q(b;\boldsymbol{w})/\beta}}\Big) \\
&= \pi^{\beta}_{\boldsymbol{w}}(a) \cdot \Big(\nabla Q(a;\boldsymbol{w})/\beta \;-\; \nabla \log \sum\nolimits_b e^{Q(b;\boldsymbol{w})/\beta}\Big) \\
&= \pi^{\beta}_{\boldsymbol{w}}(a) \cdot \Big(\nabla Q(a;\boldsymbol{w})/\beta \;-\; \frac{\sum_b e^{Q(b;\boldsymbol{w})/\beta} \cdot \nabla Q(b;\boldsymbol{w})/\beta}{\sum_c e^{Q(c;\boldsymbol{w})/\beta}}\Big) \\
&= \frac{1}{\beta} \cdot \Big(\pi^{\beta}_{\boldsymbol{w}}(a) \, \nabla Q(a;\boldsymbol{w}) - \pi^{\beta}_{\boldsymbol{w}}(a) \, \sum\nolimits_b \pi^{\beta}_{\boldsymbol{w}}(b) \, \nabla Q(b;\boldsymbol{w})\Big)
\end{aligned}
$$

and so

$$
\begin{aligned}
& \nabla\Big(\sum_a \pi^{\beta}_{\boldsymbol{w}}(a) \cdot Q(a;\boldsymbol{w})\Big) \\
= \;& \sum_a \pi^{\beta}_{\boldsymbol{w}}(a) \cdot \nabla Q(a;\boldsymbol{w}) + \sum_a Q(a;\boldsymbol{w}) \cdot \nabla \pi^{\beta}_{\boldsymbol{w}}(a) \\
= \;& \sum_a \pi^{\beta}_{\boldsymbol{w}}(a) \cdot \nabla Q(a;\boldsymbol{w}) \;+\; \frac{1}{\beta} \cdot \sum_a \pi^{\beta}_{\boldsymbol{w}}(a) \, Q(a;\boldsymbol{w}) \, \nabla Q(a;\boldsymbol{w}) \\
& - \frac{1}{\beta} \cdot \Big(\sum_a \pi^{\beta}_{\boldsymbol{w}}(a) \, Q(a;\boldsymbol{w})\Big) \cdot \Big(\sum_a \pi^{\beta}_{\boldsymbol{w}}(a) \, \nabla Q(a;\boldsymbol{w})\Big)
\end{aligned}
$$

Finally, we notice that the β-smoothing trick used in LAMIN1 is more than just an approximation heuristic, but may potentially play a role in correcting (to some extent) the sub-optimality bias of minimax Q-functions as discussed in Section 5. Specifically, as an inherent weakness, the original Lagrangian function $\mathcal{L}_\mu(Q, \boldsymbol{\lambda}_\mu)$ cannot distinguish minimax Q-functions that are optimal from minimax Q-functions that are sub-optimal, as the Lagrangian attains its global minimum in both cases. In contrast, the smoothed Lagrangian $\mathcal{L}^{\beta}_{\mu}(Q, \boldsymbol{\lambda}_\mu)$ tends to reach lower value at optimal minimax-Q-functions than at sub-optimal minimax-Q-functions. In fact, it can be proved that the sub-optimality bias is completely resolved by the β-smoothing trick in tabular settings.

As an illustration, consider the special case where there is only one non-terminal state, under which the agent can only choose between two actions, 1 and 2, and suppose action 1 is the truly optimal. In this simple case, a Q-function can be represented

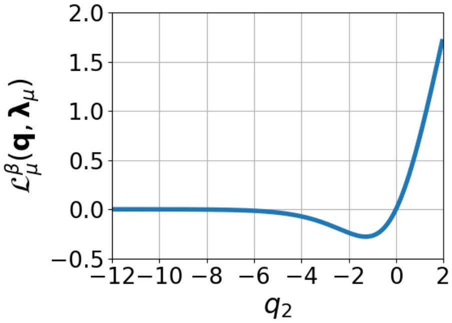

Figure 8. An illustration of the smoothed Lagrangian ($\beta = 1$) in the tabular setting with one state and two actions, with q_1 anchored at 0. The Lagrangian is not convex, yet it has only one local minimum, which is attained around $q_2 = -1.3$.

by a tuple (Q_1, Q_2), which indicating the value of action 1 and 2 respectively. A sub-optimal minimax-Q-function may have $Q_1 = Q_2$, as we showed in Section 5; in this case the smoothed Lagrangian $\mathcal{L}^{\beta=1}_{\mu}(Q, \boldsymbol{\lambda}_{\mu})$ would still equal $J(\mu)$, which can be seen from the dual form of the smoothed Lagrangian:

$$\mathcal{L}^{\beta=1}_{\mu}(Q, \boldsymbol{\lambda}_{\mu}) = J(\mu) + 1 \cdot \Big((\frac{e^{Q_1} \cdot Q_1}{e^{Q_1} + e^{Q_2}} + \frac{e^{Q_2} \cdot Q_2}{e^{Q_1} + e^{Q_2}}) - Q_1 \Big)$$

On the other hand, an optimal minimax-value can make $\mathcal{L}^{\beta=1}_{\mu}(Q, \boldsymbol{\lambda}_{\mu})$ lower than $J(\mu)$. For example, when $J(\mu) = Q_1 = 0$, we have $\mathcal{L}^{\beta=1}_{\mu}(Q, \boldsymbol{\lambda}_{\mu}) = \frac{e^{Q_2}}{1+e^{Q_2}} \cdot Q_2$, which attains -0.28 at $Q_2 = -1.3$. In other words, $\mathcal{L}^{\beta=1}_{\mu}(Q, \boldsymbol{\lambda}_{\mu}) < J(\mu) = 0$ under the optimal minimax-Q-function $(0, -1.3)$, which is thus distinguished from the sub-optimal minimax-Q-function $(0, 0)$ in smoothed Lagrangian minimization as LAMIN1 does. See Figure 8 for the shape of the smoothed Lagrangian in this case. Notice that the Lagrangian is *not* convex, despite the unique local minimum (which means LAMIN1 will converge to the global optimum in this case).

E.2. The LAMIN2 Algorithm

The idea of LAMIN2 is to minimize the original Lagrangian function $\mathcal{L}_{\mu}$ based on the "local" gradient estimator (15). The consistency of the LAMIN2 estimator (with respect to $\nabla \mathcal{L}_{\mu}$) is asserted by Proposition 5 in Section 4.1, which we now provide a proof.

Proposition 5 says that (15) becomes equality if the Q-model is unimodal. For convenience we copy (15) below (with $[\mathcal{B}Q - Q]$ and δ explicitly expanded, and the approximation sign replaced):

$$\begin{aligned} &\nabla_{\boldsymbol{w}} \mathop{\mathbf{E}}_{S' \sim P(s,a)} \Big[R(S') + \gamma_{\text{epi}}(S') \max_{a'} Q(S', a'; \boldsymbol{w}) \Big] - Q(s, a; \boldsymbol{w}) \, \Big|_{\boldsymbol{w} = \boldsymbol{w}_t} \\ = &\mathop{\mathbf{E}}_{S' \sim P(s,a)} \mathop{\mathbf{E}}_{A' \sim \pi_{Q(\boldsymbol{w}_t)}(S')} \Big[\nabla_{\boldsymbol{w}} \Big((R(S') + \gamma_{\text{epi}}(S') Q(S', A'; \boldsymbol{w}) - Q(s, a; \boldsymbol{w}) \Big) \Big] \, \Big|_{\boldsymbol{w} = \boldsymbol{w}_t} \end{aligned} \tag{44}$$

Clearly, (44) holds if and only if equation (45) in the following proposition holds.

Proposition 14. *Let $\mathcal{A}$ be a finite action space, and let $Q(s, a; \boldsymbol{w})$ be a differentiable parametric Q-model that suggests a single best action $a_{\max}(\boldsymbol{w}^*) \doteq \arg\max_{a \in \mathcal{A}} Q(s, a; \boldsymbol{w}^*)$ when evaluating the actions under a given state s with a given parameter vector $\boldsymbol{w}^*$, then*

$$\nabla_{\boldsymbol{w}} \max_{a \in \mathcal{A}} Q(s, a; \boldsymbol{w}) \, \Big|_{\boldsymbol{w} = \boldsymbol{w}^*} = \mathop{\mathbf{E}}_{a \sim \pi_{\max}(s; \boldsymbol{w}^*)} \Big[\nabla_{\boldsymbol{w}} Q(s, a; \boldsymbol{w}) \Big] \, \Big|_{\boldsymbol{w} = \boldsymbol{w}^*} \tag{45}$$

where $\pi_{\max}(\boldsymbol{w}^*)$ denote the $Q(\boldsymbol{w}^*)$-greedy policy.

Proof. We will prove that for any component of $\boldsymbol{w}$,

$$\frac{\partial}{\partial w_i} \max_{a \in \mathcal{A}} Q(s, a; \boldsymbol{w}) \, \Big|_{\boldsymbol{w} = \boldsymbol{w}^*} = \frac{\partial}{\partial w_i} Q(s, a_{\max}(\boldsymbol{w}^*); \boldsymbol{w}) \, \Big|_{\boldsymbol{w} = \boldsymbol{w}^*} \tag{46}$$

Algorithm 2 The LAMIN2 algorithm.

```
Input: A piece of rollout data made by an optimal policy, in the form of $\{s_t, a_t, r_t\}_{0,1,\ldots,n}$, in which $t = T_1, T_2, \ldots, T_k$ are termination steps; A parametric $Q(\boldsymbol{w})$ model with initial weight $\boldsymbol{w}_0$; Learning rate $\alpha$; Boltzmann temperature $\beta$.
for gradient update $i = 0, 1, 2, \ldots$ do
    $\Delta\boldsymbol{w} \leftarrow \frac{1}{k}\sum_{t=\{T_1\ldots T_k\}} \nabla_{\boldsymbol{w}} Q(s_t, a_t; \boldsymbol{w})\Big|_{\boldsymbol{w}=\boldsymbol{w}_i} +$
        $\frac{n}{k}\cdot\frac{1}{n}\sum_{t=0}^{n-1}\gamma_{\text{epi}}(s_{t+1})\left(\sum_a \pi^{\beta}_{\boldsymbol{w}_i}(s_{t+1}, a)\nabla_{\boldsymbol{w}} Q(s_{t+1}, a; \boldsymbol{w})\right) - \nabla_{\boldsymbol{w}} Q(s_t, a_t; \boldsymbol{w})\Big|_{\boldsymbol{w}=\boldsymbol{w}_i}$
    $\boldsymbol{w}_{i+1} \leftarrow \boldsymbol{w}_i - \alpha\cdot\Delta\boldsymbol{w}$
end for
Output: A $Q(\boldsymbol{w})$-greedy policy.
```

which readily gives (45).

To prove (46), notice that

$$\frac{\partial}{\partial w_i}\max_{a\in\mathcal{A}} Q(s, a; \boldsymbol{w})\Big|_{\boldsymbol{w}=\boldsymbol{w}^*} = \lim_{\Delta\to 0}\frac{Q(s, a_{\max}(\boldsymbol{w}^*+\Delta); \boldsymbol{w}^*+\Delta) - Q(s, a_{\max}(\boldsymbol{w}^*); \boldsymbol{w}^*)}{\Delta}$$

When there are a finite number of possible actions, there must be a non-zero gap between the best action and the second best action under $\boldsymbol{w}^*$. On the other hand, since Q is differentiable, the change of action-values becomes infinitely small as $\Delta \to 0$, thus the order between the two best actions will remain the same for small enough Δ, that is, $a_{\max}(\boldsymbol{w}^*+\Delta) = a_{\max}(\boldsymbol{w}^*)$ when $\Delta \to 0$. Therefore,

$$\begin{aligned}&\lim_{\Delta\to 0}\frac{Q(s, a_{\max}(\boldsymbol{w}^*+\Delta); \boldsymbol{w}^*+\Delta) - Q(s, a_{\max}(\boldsymbol{w}^*); \boldsymbol{w}^*)}{\Delta}\\ =\ &\lim_{\Delta\to 0}\frac{Q(s, a_{\max}(\boldsymbol{w}^*); \boldsymbol{w}^*+\Delta) - Q(s, a_{\max}(\boldsymbol{w}^*); \boldsymbol{w}^*)}{\Delta}\\ =\ &\frac{\partial}{\partial w_i} Q(s, a_{\max}(\boldsymbol{w}^*); w)\Big|_{\boldsymbol{w}=\boldsymbol{w}^*}\end{aligned}$$

□

Note that when $Q(\boldsymbol{w})$ is a sophisticated real-valued function, such as a deep neural network with scalar output, the chance that two actions have *precisely* the same value under $Q(\boldsymbol{w})$ should be rare. Also, continuous action space can be densely quantized into a finite action space with *arbitrarily* small quantizing error, thus (45) should still approximately hold even for continuous action spaces.

Algorithm 2 gives the pseudo-code of LAMIN2 in a further generalized form, where the greedy policy $\pi_{\boldsymbol{w}_i}$ is replaced with the Boltzmann policy $\pi^{\beta}_{\boldsymbol{w}_i}$. As mentioned in Section 4, higher temperatures, such as $\beta = 1.0$, can slightly improve performance in our WMT experiment (for $0.5 - 0.8$ BLEU score).

E.3. An Episodic Learning Formulation of Machine Translation

Many AI tasks are *sequence generation* problems, where we are given a context X, and are then asked to generate a sequence $Y = (\texttt{bos}, y_1 \ldots y_L, \texttt{eos})$ using *tokens* chosen from a given token space. Machine Translation (MT) is an example of such tasks, where the token space is the vocabulary of a target language, and the context X is a sentence (or a sequence of sentences) in a source language. The choice of each token y_t is conditioned on X and on the partial output $Y_{<t} \doteq (\texttt{bos}, y_1 \ldots y_{t-1})$. In particular, $Y_{<1} = (\texttt{bos})$, and $Y_{<L+1} = (\texttt{bos}, y_1 \ldots y_L)$, conditioned on which the algorithm will generate the first token y_1 and the last token $y_{L+1} \doteq \texttt{eos}$, respectively.

The ELP formulation exactly captures the real-world MT tasks as described above. In the MT context, an episode is the translation of a given sentence. The first episode effectively starts with $S_1 = (X^{(1)}, \texttt{bos})$ where $X^{(1)}$ is a full source sentence. A learning agent then chooses a token $A_1 = y_1^{(1)} \in \Sigma_{\text{target}} \cup \{\texttt{eos}\}$, after which the environment state transits, deterministically, to $S_2 = (X^{(1)}, Y^{(1)}_{<2}) = (X^{(1)}, \texttt{bos}, y_1^{(1)})$. The agent keeps generating actions $A_t = y_t^{(1)}$ under each $S_t = (X^{(1)}, Y^{(1)}_{<t})$ until it outputs $A_{T-1} = \texttt{eos}$ at some step $T-1$, leading to terminal state $S_T =$

$(X^{(1)}, Y^{(1)}) = (X^{(1)}, \texttt{bos}, y_1^{(1)} \dots y_{T-2}^{(1)}, \texttt{eos})$. The agent will then make a normal action A_T as in previous steps, which however makes no effect other than resetting the environment into $S_{T+1} = (X^{(2)}, \texttt{bos})$ from which the second translation episode begins. The process goes on episode after episode, generating a (theoretically infinite) sequence of translations $(X^{(1)}, Y^{(1)}), (X^{(2)}, Y^{(2)}), \dots$, which collectively serve as the training data for the agent to learn better translation policies.

An episode of length T as above results in a translation sentence $Y = (\texttt{bos}, y_1 \dots y_L, \texttt{eos})$ which contains $L = T - 2$ "normal" tokens from Σ_{target}. As a common and necessary practice, most real-world MT systems impose a maximum translation length H so that if an $\texttt{eos}$ action did not show up after H steps, the environment will transit to the terminal state $S_{H+2} = (X, \texttt{bos}, y_1 \dots y_H, \texttt{eos})$ even if the agent continues to output normal token $A_{H+1} \in \Sigma_{\text{target}}$ at step $H + 1$. When maximum translation length is applied, the corresponding episodic learning model of MT has bounded episode length, thus satisfies the ELP Condition (1) above. Such a formulation considers the mechanism of maximum translation length as a fundamental part of *learning-based* MT task specification that is essential in helping *learning* agents (which may not master when to output $\texttt{eos}$) to escape from long and meaningless translation episodes which may otherwise lead to ill-conditioned training data.

To comply with ELP Condition (3), we prescribe ρ_0 to be an arbitrary distribution over the set of terminal states, i.e. over all source-target sentence pairs (X, Y) where Y is complete sentence ending with $\texttt{eos}$. As with other terminal steps, no matter what S_0 and A_0 are, the next state S_1 will follow the same distribution, denoted as ρ_1, which specifies the distribution of the source sentences that the agent will receive in *each* episode (ELP Condition (2)).

Finally, the agent receives a scalar reward $R(X, Y) \in [0, 100]$ at each terminal state, based on the translation quality of Y (with respect to X). The reward is zero at all non-terminal states (which has only partial translations).[8] With this reward function, the total-reward objective $J(\pi)$ of a translation policy π corresponds to a sentence-averaged evaluation score over the corpus. For some MT metrics, this captures exactly the original metric; for example, the METEOR metric (Banerjee and Lavie, 2005) corresponds to $R(X, Y) = \texttt{METEOR}(Z(X), Y)$. For some other MT metrics, such as BLEU (Papineni et al., 2002), the sentence-level BLEU score needs to be properly smoothed to match the true corpus-level BLEU score (Yuan et al., 2021).

The ELP formulation of MT as discussed above can be formally summarized as follows:

- $\mathcal{S} = (\Sigma_{\text{source}})^H \times \{\texttt{bos}\} \times (\Sigma_{\text{target}})^H \times \{\text{␣}, \texttt{eos}\}$
- $\mathcal{A} = \Sigma_{\text{target}} \cup \{\texttt{eos}\}$
- $R(s) = \begin{cases} \texttt{metric}\big(\, X(s), Y(s) \,\big) & s \in \mathcal{S}_\perp \\ 0 & s \notin \mathcal{S}_\perp \end{cases}$, where $\mathcal{S}_\perp = \{(X, Y) : Y \text{ ends with } \texttt{eos}\}$
- $P(s'|s, a) = \begin{cases} \rho_1(s') & \text{if } s \in \mathcal{S}_\perp \\ \mathbb{1}[\, s' = (s, a) \,] & \text{if } s \notin \mathcal{S}_\perp \text{ and } |Y(s)| < H \\ \mathbb{1}[\, s' = (s, \texttt{eos}) \,] & \text{if } s \notin \mathcal{S}_\perp \text{ and } |Y(s)| = H \end{cases}$
- $\rho_0(s) > 0$ only if $s \in \mathcal{S}_\perp$

Note that in above both $\mathcal{S}$ and $\mathcal{A}$ are finite sets, in which case the model is a *finite* episodic learning process. For real-world machine translation, we typically have $|\mathcal{S}| < 40000^{2048} \times 40000^{2048} \times 2$ and $|\mathcal{A}| < 40000 + 1$.

The experimentation code in Supplementary Material contains a faithful implementation in Python of the formulation presented here.

E.4. Experiment Details

We tested our algorithmic idea using the WMT'14 NewsTest English→German (en2de) dataset [9]. The data was pre-processed and post-processed using the BPE tokenizer provided by YouTokenToMe [10], with shared vocabulary of size 37000.

[8] While it is certainly possible to make the rewards less sparse via reward shaping and engineering, we consider those reward variants as elements of a specific *solution method*, instead of as part of the *problem formulation* of MT.

[9] https://nlp.stanford.edu/projects/nmt/

[10] https://github.com/VKCOM/YouTokenToMe

A complete translation typically consists of 20-100 tokens (meaning that a translation episodes contains roughly 20-100 action steps). We used SacreBLEU (Post, 2018) to generate the BLEU scores, and trained the standard TransformerBase neural network (Vaswani et al., 2017), which is known to achieve a BLEU score of 27.3 on the WMT'14 dataset under the state-of-the-art method of MLE-based supervised learning (Vaswani et al., 2017).

We trained the model on the same 4.5 millions sentence pairs in the WMT'14 data set for $100,000$ gradient updates on a V100 GPU, with the same mini-batch size (and token-padding strategy) and learning rate schedule as recommended by Vaswani et al. (2017). A dropout rate of 0.1 is applied. The learned model is then used as search heuristic in the *vanilla-beam-search decoding* (e.g. see Algorithm 1 in (Stahlberg and Byrne, 2019)), with a beam size of 4. Empirically, we found that some more performance gain can be obtained by adding more tricks, such as modestly increasing the beam size (say, to 10), adding length penalty factor in the search heuristic (see (Wu et al., 2016)), and model averaging (see (Vaswani et al., 2017)), but we chose to exclude these tricks in our performance report so as to keep our algorithm simple and easy to implement.

The following tables give the numerical values of the performance scores shown in Figure 2.

β	BLEU@100k
0.01	27.4
0.25	27.0
0.5	26.7
0.75	26.6
1.0	26.6

Table 1. Corpus BLEU scores of LAMIN1 (i.e. Algorithm 1) under different temperature β.

β	BLEU@100k
0.01	26.0
0.2	26.2
0.4	26.8
0.6	26.2
0.8	26.3
1.0	26.8

Table 2. Corpus BLEU scores of LAMIN2 (i.e. Algorithm 2) under different temperature β.