Learning Successor States and Goal-Dependent Values: A Mathematical Viewpoint

Léonard Blier, Corentin Tallec, Yann Ollivier

January 13, 2021

Abstract

In reinforcement learning, temporal difference-based algorithms can be sample-inefficient: for instance, with sparse rewards, no learning occurs until a reward is observed. This can be remedied by learning richer objects, such as a model of the environment, or *successor states*. Successor states model the expected future state occupancy from any given state [Day93, KSGG16], and summarize all paths in the environment for a given policy. They are related to *goal-dependent value functions*, which learn how to reach arbitrary states.

We formally derive the temporal difference algorithm for successor state and goal-dependent value function learning, either for discrete or for continuous environments with function approximation. Especially, we provide finite-variance estimators even in continuous environments, where the reward for exactly reaching a goal state becomes infinitely sparse.

Successor states satisfy more than just the Bellman equation: a *backward* Bellman operator and a *Bellman–Newton* (BN) operator encode path compositionality in the environment. The BN operator is akin to second-order gradient descent methods, and provides the "true" update of the value function when acquiring more observations from the environment, with explicit tabular bounds. In the tabular case and with infinitesimal learning rates, mixing the usual and backward Bellman operators provably improves eigenvalues for asymptotic convergence, and the asymptotic convergence of the BN operator is provably better than TD, with a rate independent from the environment. However, the BN method is more complex and less robust to sampling noise.

Finally, a *forward-backward* (FB) finite-rank parameterization of successor states enjoys reduced variance and improved samplability, provides a direct model of the value function, has fully understood fixed points corresponding to long-range dependencies (but ignores small-scale dependencies), approximates the BN method, and provides two canonical representations of states as a byproduct.

Contents

1 Introduction, Overview of Results

The *successor state operator* of a Markov reward process is an object that directly encodes the passage from a reward function to the corresponding

value function. In particular, it expresses the value functions of all possible reward functions for a given, fixed policy.

Goal-dependent value functions are a related object with many similar properties. They describe the optimal value functions and policies for a specific set of tasks: typically, for all rewards located at all possible target states. In this case, the policy depends on the target state.

Here we offer a formal treatment of these objects in both finite and continuous spaces. We present several learning algorithms and associated results. In particular, we focus on proper treatment of the infinitely-sparse reward problem encountered by TD-style approaches in continuous spaces if the reward is located at a precise state.

Possible advantages of working with these objects include:

- Contrary to TD, learning starts even before any rewards are observed. Sucessor state learning extracts information from every observed transition, by learning how to reach every visited state. Subsequent reward observations provide an instantaneous update to the value function via the successor state operator.

  This learning is done without reward signals, illustrating an "unsupervised reinforcement learning" approach. Successor state lie in between model-free and model-based reinforcement learning approaches, providing a representation of the future of a state without having to synthesize future states or unrolling synthetic trajectories. Algorithmically, they rely on having two states as inputs rather than generating a state.

- Successor states and goal-dependent values exploit relationships between how to reach different states. With function approximation, generalization occurs between different target states. But even in a tabular setting with no generalization, these objects satisfy more algebraic relations than the usual Bellman equation: a *backward* Bellman equation and a *Bellman–Newton* equation, expressing path compositionality in the Markov process (Fig 1). This leads to quantifiable asymptotic gains.

- Successor states and goal-dependent values can be used to solve several problems at once, such as learning to reach arbitrary states. Even for optimizing a single reward, they can be used for auxiliary tasks such as going to an arbitrary state, which could be useful for exploration, or to provide good state representations.

For learning value functions dependent on goal states, an obvious approach is to apply any standard reinforcement learning algorithm, with reward 1 when the visited state is equal to the goal state (e.g., [SHGS15]). But this breaks down in continuous spaces, as the reward function becomes infinitely sparse (a random trajectory is never going to reach any predefined goal

exactly). Even in discrete spaces, the reward becomes exponentially sparse as the number of components increase.

This problem is avoided by a suitable mathematical treatment. The intuition behind several of our results is the following: If the goal is to learn how to reach arbitrary states, then this is not a sparse reward problem, although straightforward TD implementations treat it as such; it is a problem with rewards everywhere. Approaches such as *hindsight experience replay* [AWR+17] attempt to exploit this intuition by resampling goals a posteriori in an off-policy algorithm, but it is unclear to us how much of the problem HER solves in continuous spaces. The mathematical treatment here proves that finite-variance algorithms exist for such problems, even in continuous spaces.

Overview of results. In a nutshell, successor states summarize all possible paths in the environment for a given policy (Section 4.3). For finite spaces, the entries $M_{ss'}$ of the successor state matrix describe the expected discounted time spent in state s' by a trajectory starting at s [Day93]: $M_{ss'} = \mathbb{E}\left[\sum_{t\geqslant 0} \gamma^t \mathbb{1}_{s_t=s'} | s_0 = s\right]$. The entry $M_{ss'}$ is also the value function at s if the reward is 1 at s' and 0 everywhere else. As such, M contains the information about reaching every state in the environment, not just those states providing a reward. For a fixed policy, the value function depends linearly on the reward: in a finite state space, for any reward function, represented as a vector R over states, its associated value function is $V = MR$.

The goal-dependent value function $V_{ss'}$ at state s for goal s' (another state) is defined as the value function at s of the optimal policy for reaching a unit reward located at s'. The difference with $M_{ss'}$ is now that the policy depends on s' instead of being fixed. Learning this object allows for learning how to reach different goal states. Contrary to $M_{ss'}$, $V_{ss'}$ does not contain information on how to optimize dense rewards (mixtures of goal states), only rewards located at a single state. It is also possible to define V for more general types of goals rather just a target state, although the goals must be predefined and mixtures of goals are not possible a posteriori.

The bulk of the text presents theoretically well-motivated algorithms to learn these objects directly for any two states s, s'. The main contributions of this text are the following.

- We formally define successor states and goal-dependent value (and Q) functions in general state spaces (Sections 3 and 5), extending the discrete case of [Day93]. For continuous states, this involves some measure theory (Section 3.2), but the intuition is clear from the discrete case (Section 3.1).

  Successor states are always well-defined for a given policy (Theorem 2). But goal-dependent value functions are generally not unique in con-

tinuous spaces (Section 5.3); still, there exists a canonical solution (Theorem 14), smaller than all others.

- We formally derive the temporal difference (TD) algorithm for successor state learning, both for discrete spaces, and for continuous spaces with function approximation (Theorem 6), beyond the tabular setting of [Day93]. A naive application of TD on a state-goal product space, with reward 1 when the state reaches the goal, degenerates in continuous spaces: the reward becomes infinitely sparse (it is 0 with probability 1 and ∞ with probability 0). Instead, the TD estimators we provide have finite variance (Section 4.1.4, Proposition 8).

  Known convergence results for TD extend to this setting: tabular case with any sampling policy, linear parameterization on-policy, arbitrary function approximation assuming reversibility of the Markov process (Section 4.1.5).

  Likewise, we formally derive the TD algorithm for goal-dependent Q and V functions (Section 5), with finite-variance estimators even in continuous spaces. The goals may be target states, or target values for some vector-valued function of the states (Section 5.2).

- Algorithmically, successor states and goal-dependent values are represented by function approximators depending on two states (the current state and a goal state) instead of one. TD learning works in a black-box environment by sampling from a set of observed transitions $s \to s'$ between states, and sampling goal states (typically from the same distribution). No reward signal is needed.

  Most variants of TD still apply: V or Q learning, target networks, multi-step returns... (Appendix A). Notably, Appendix A.6 describes *relative TD* to deal directly with a decay factor $\gamma = 1$ and to reduce variance for γ close to 1.

  Successor states and goal-dependent values can be used to learn an optimal policy for a particular reward, or to learn goal-dependent policies. Many different options are described in Section 8, such as Q-learning or policy gradient, with several ways to learn the value function from successor states.

- Successor states satisfy more than one Bellman equation: we introduce *backward TD* for successor states (Section 4.2, Theorem 9), and the corresponding parametric update (Theorem 26, Appendix A.4).

  Successor states encode all paths in the Markov process for a fixed policy (Section 4.3). The usual (forward) Bellman equation $M = \mathrm{Id} + \gamma PM$ adds a newly observed transition at the front of all known paths, while the backward Bellman equation $M = \mathrm{Id} + \gamma MP$ extends known paths

by adding newly observed transitions at the back. This backward equation exists for successor states but not goal-dependent value functions. In the tabular setting and with small learning rates, combining forward and backward TD turns out to improve the eigenvalues of the learning process (Section 9.2).

- We introduce "second-order" methods for learning successor states, which are to TD what Newton-type methods are to first-order gradient descent (Section 7). In addition to the usual (forward) and the backward Bellman equations, there is a third Bellman equation satisfied by M, which leads to the *Bellman–Newton operator* $M \leftarrow 2M - M^2 + \gamma MPM$ (Section 7.2). It also enjoys a path interpretation, learning by path concatenation and doubling the length of known paths (Proposition 20). The forward and backward Bellman operators only increase the length of known paths by 1 (Appendix C).

  Asymptotically and in the small learning rate limit, the Bellman–Newton operator converges provably faster than TD (Section 9.3), with an asymptotic rate *independent of the environment and policy*. However, in practice this method is less resistant to sample noise: smaller learning rates are necessary, so the comparison with TD is less clear. There is also a parametric version of the Bellman–Newton operator (Theorem 21), but it is numerically fickle.

  We also study the estimation of M by direct inversion of $\mathrm{Id} - \gamma\hat{P}$ using an empirical estimate $\hat{P}$ of the transition matrix P in a finite state space. The resulting update of M when adding each new observation is the same as a Bellman–Newton update with learning rate $1/t$ (Theorems 17 and 18). In finite spaces, we provide an explicit non-asymptotic bound for the convergence of M and the value function V based on these empirical estimates (Theorem 16).

- Representing the successor state operator as a dot product $F(s_1)^\top B(s_2)$ between features of the starting state s_1 and the target state s_2 has many nice properties (Section 6). Here, the "forward" and "backward" feature functions F and B are both learned to approximate M: this may have independent interest for representation learning.

  First, this method provides a direct representation of the value function without additional learning (Eq. 47).

  Second, when learning F and B by any of the algorithms above, in expectation the updates factorize between s_1 and s_2 (Proposition 15). This allows for variance reduction, and for purely trajectory-wise algorithms which only use the currently observed transition $s \to s'$ without sampling an additional target state s_2 (Section 6.2), in contrast to the general form of TD for M.

Third, this representation keeps some properties of the Bellman–Newton method without its shortcomings; they actually coincide when the transition matrix of the process is symmetric (Theorem 41).

Finally, the fixed points of TD for this representation can be fully characterized in the tabular and overparameterized cases (Propositions 35–39 in Appendix E, and Section 6.2). They are related to eigenspaces of the transition matrix P. Notably, in the tabular or overparameterized case, if forward TD is used to learn F and backward TD to learn B, then the fixed points are exactly local minimizers of the error between the $F^\top B$ model and the true successor state operator (Proposition 35). In contrast, for ordinary TD on the value function and a linear model, the fixed points are not minimizers of the error to the true value function.

Some related work on successor states. The successor state operator is linked to various existing objects under various names (fundamental matrix, occupation matrix, successor representations, successor features...). Successor states have even been identified in the neurosciences [SBG17].

For discount factor $\gamma = 1$, the successor matrix M is known as the *fundamental matrix* [KS60, Bré99, GS97] of a Markov process (up to subtracting the invariant measure). [1] The fundamental matrix encodes many properties of the Markov chain, such as value functions ([Ber12], as we use here) or hitting times [KS60]. In a reinforcement learning context, and with $\gamma < 1$, this matrix goes back at least to [Day93].

Learning successor states by temporal difference is mentioned in [Day93] for the tabular case and with linear approximations; the parametric case has never been derived as far as we know.

In a deep learning context, several recent works have used the related *successor representations* [KSGG16], e.g., for transfer [BDM+17, BBQ+18, ZSBB17, LTL17, MWB18, BHB+20], hierarchical RL [MRG+18] or exploration [MBB19].

In particular, the Deep Successor Representation algorithm [KSGG16] approximates successor states by learning a state representation $\varphi(s)$ together with a successor representation $m(s)$ defined as the expected discounted representation of future states from s: $m(s) = \mathbb{E}\left[\sum_{t\geqslant 0} \gamma^t \varphi(s_t) | s_0 = s\right]$. As $\varphi = m = 0$ is a fixed point of the method, a reconstruction loss must be used to prevent collapse. Here we directly learn the successor states $M_{ss'}$ for every pair of states in the original space.

[1] Namely, in Markov chain theory, the fundamental matrix is defined with an additional rank-one term which avoids all problems with $\gamma = 1$ and is analogous to *relative TD*. The case $\gamma < 1$ is obtained from it [Ber12, §5.1.1]. In this introduction, to stay closer to RL practice, we take $\gamma < 1$ and define M without this term. The case $\gamma = 1$ is treated in Appendix A.6 (relative TD for M).

Successor states provide the value function for every goal state: this is related to learning multiple RL tasks [SMD+11, SHGS15, JKSY20, PG17] which performs joint V- or Q-learning for a set of goals. To some extent, this makes it possible to reach or transfer to previously unseen goals [SHGS15].

Recently, [vHMH+20] proposed an algorithm to learn a model of eligibility traces; we prove in Appendix D that the expected eligibility traces at each state is proportional to the transpose of the successor state matrix ("predecessor" states).

Our second-order algorithms in Section 7 are based on an implicit process estimation approach. Process estimation is also used in [PW19] to obtain convergence bounds for the value function in finite MDPs, under a "synchronous" setting (a transition is observed from every state at every step). They prove that process estimation is minimax-optimal for this setting.

More generally, successor state learning comes in the context of *unsupervised RL*, in which relevant features of the environment are learned without the supervision of a reward signal. Many works have suggested that unsupervised RL improves sample efficiency [SJK+19]. Notably, this includes model-based methods [FLHI+18]. Contrary to the latter, successor state learning does not require synthesizing accurate future states; to some extent, a transition model is implicitly learned via a function $m(s, s')$ that describes how much s' lies in the future of s with the current policy.

2 Notation for Markov Reward Processes

We consider a *Markov reward process* (MRP) $\mathcal{M} = \langle \mathcal{S}, P, r, \gamma \rangle$ with state space $\mathcal{S}$ (discrete or continuous), transition probabilities $P_{ss'}$ from s to s', random reward signal r_s at state s, and discount factor $0 \leqslant \gamma < 1$ [SB18]. We do not assume that the state space $\mathcal{S}$ is finite.

In the finite case, $P_{ss'}$ can be viewed as a matrix. In the general case, for each $s \in \mathcal{S}$, $P(s, \mathrm{d}s')$ is a *probability measure* on s' that depends on s. From now on, we use the notation $P(s, \mathrm{d}s')$ to cover both cases. [2]

A Markov *decision* process, with a given policy, with actions $a \in \mathcal{A}$, transition probabilities $P(s, a, \mathrm{d}s')$, and policy $\pi(s, a)$, defines two Markov reward processes: one on states via $P(s, \mathrm{d}s') := \sum_a \pi(s, a) P(s, a, \mathrm{d}s')$, and another on state-action pairs via $P((s, a), (\mathrm{d}s', \mathrm{d}a')) := P(s, a, \mathrm{d}s')\, \pi(s', a')$. (start at (s, a), get s', then choose the next action at s'). Thus, we work on states and value functions, but all results extend to state-action pairs and Q functions.

[2] Formally, we take the setting from [Hai10]. The state space $\mathcal{S}$ is assumed to be a complete, separable metric space (*Polish space*), such as a finite or countable space or $\mathbb{R}^n$. It is equipped with its Borel σ-algebra (the σ-algebra generated by all open sets). This guarantees that integration behaves as expected. $P(s, \mathrm{d}s')$ is assumed to be a *Markov kernel*, namely, a measurable map from $\mathcal{S}$ to probability measures over $\mathcal{S}$.

For now the policy is fixed: we deal with policy evaluation and successor states under that policy. Goal-dependent policies are treated in Section 5.

We denote $R(s) := \mathbb{E}[r_s]$ the expected reward at s. The *value function* V is $V(s_0) := \mathbb{E}\left[\sum_k \gamma^k r_{s_k}\right]$ where $s_0, s_1, \dots, s_t$ is a trajectory starting at s_0 sampled from the process. We denote by $\mathbb{1}_s$ the vector equal to 1 at the s coordinate and 0 elsewhere.

Data model. We assume access to observations from the Markov reward process, such as a fixed dataset of stored transitions, or some sample trajectories. Each observation is a triplet (s, s', r_s) with $s' \sim P(s, \mathrm{d}s')$ and r_s the associated reward. Consecutive observations need not be independent. We denote by $\rho(\mathrm{d}s)$ be the distribution of states s coming from the observations. We cannot choose the states s: ρ is unknown and we do not make any assumptions on it. For instance, if we have access to trajectories from the process, obtained by some exploration policy, then ρ would be the law of states visited under that policy. If we just have a finite dataset of transitions, ρ would be the (unknown) law from which this dataset was sampled.

Markov kernels as operators. Interpreting P and the successor state as operators on functions over $\mathcal{S}$ clarifies the statements of the results below. We follow the standard theory of Markov kernels [Hai10, Hai06]. We denote by $B(\mathcal{S})$ the set of bounded measurable functions on S. P acts on such functions as follows. If f is a function in $B(\mathcal{S})$, Pf is defined as $(Pf)(s) := \mathbb{E}_{s' \sim P(s, \mathrm{d}s')}[f(s')]$. This is compatible with the matrix notation Pf in the finite case, viewing f as a vector. In the text, we freely identify Markov kernels with the corresponding operators.

If P_1 and P_2 are two such Markov kernel operators, their composition $P_1 P_2$ is again a Markov kernel operator, and coincides with matrix multiplication in the finite case. In particular, P^n represents n steps of P. The identity operator Id corresponds to always staying in the same state, namely, a transition operator $P(s, \mathrm{d}s') = \delta_s(\mathrm{d}s')$ with δ_s the Dirac measure at s.

We denote $\Delta := \mathrm{Id} - \gamma P$, the discrete Laplace operator of the Markov process. Finally, if A is an operator acting on functions over $\mathcal{S}$, we denote its inverse by A^{-1}, if it exists.

Norms. Both $P(s, \mathrm{d}s')$ and the successor state operator $M(s, \mathrm{d}s')$ are measures on s' that depend on s. We will use the following norms on such objects: if $\rho(\mathrm{d}s)$ is some reference probability measure on $\mathcal{S}$, and $M_1(s, \mathrm{d}s')$ and $M_2(s, \mathrm{d}s')$ are two such objects, we define

$$\|M_1 - M_2\|_\rho^2 := \mathbb{E}_{s \sim \rho,\, s' \sim \rho} \left(m_1(s, s') - m_2(s, s')\right)^2 \tag{1}$$

where $m_1(s, s') := M_1(s, \mathrm{d}s')/\rho(\mathrm{d}s')$ is the density of M_1 with respect to ρ (if it exists; if not, the norm is infinite), and likewise for M_2. We will also

use the *total variation* norm

$$\|M_1 - M_2\|_{\rho,\mathrm{TV}} := \mathbb{E}_{s\sim\rho} \|M_1(s,\cdot) - M_2(s,\cdot)\|_{\mathrm{TV}} \tag{2}$$

with $\|p_1 - p_2\|_{\mathrm{TV}} := \sup_{A\subset\mathcal{S}} |p_1(A) - p_2(A)|$ the usual total variation distance between two measures.

3 The Successor State Operator of a Markov Process

As an introduction before defining successor states over general state spaces, we start with the case of finite state spaces, for which all the objects can be seen as vectors matrices. This is the case treated in [Day93].

3.1 The Successor State Matrix in a Finite State Space

Informally, for finite state spaces, given two states s_1 and s_2 in a Markov process, the successor state matrix M is a matrix whose entry $M_{s_1 s_2}$ is the expected discounted time spent at s_2 if starting the process at s_1 [Day93].

$M_{s_1 s_2}$ is also the value function at s_1 if the reward is located at s_2 ($R_s = \mathbb{1}_{s=s_2}$). Thus, columns of M contain the value functions of all single-target rewards. For a fixed Markov process (e.g., fixed environment and policy), the value function is a linear function of the reward. Thus, by linearity, for any reward, the associated value function is $V = MR$. Namely, M contains information about the value function of *every* reward.

We gather several equivalent definitions of the matrix M in the following Proposition. Since this is a particular case of the more general results below, we do not include a proof.

Proposition 1 (Successor state matrix of a finite Markov process). *Consider a Markov process on a finite state space, with transition matrix P. The following definitions of the* successor state matrix M *are equivalent:*

1. *M is the inverse of the Laplace operator $\Delta = \mathrm{Id} - \gamma P$,*

$$M = (\mathrm{Id} - \gamma P)^{-1} = \sum_{n\geqslant 0} \gamma^n P^n. \tag{3}$$

2. *M is the matrix that transforms a reward function into the corresponding value function: for any reward function R, the associated value function is*

$$V = MR. \tag{4}$$

3. *For each state s, the column s of the matrix M represents the value function of a Markov reward process whose reward is 1 when at state s and 0 everywhere else ($R = \mathbb{1}_s$).*

4. *M is the unique fixed point of the Bellman operator*

$$M \leftarrow \mathrm{Id} - \gamma P M \tag{5}$$

or equivalently of the backward *Bellman operator*

$$M \leftarrow \mathrm{Id} - \gamma M P. \tag{6}$$

5. *For each state s, the row s of the matrix M represents the expected occupation time at each state, for trajectories starting at s, with discounting γ:*

$$M_{ss'} = \sum_{t \geqslant 0} \gamma^t \mathbb{P}(s_t = s' | s_0 = s) \tag{7}$$

where $s_0, \ldots, s_t$ is a random trajectory in the Markov process.

6. *The entry ss' of the matrix M, is the number of paths from s to s', weighted by their probability in the process, and with decay γ according to their length:*

$$M_{ss'} = \sum_n \sum_{\substack{p=(s_0 s_1 \ldots s_n) \\ \text{path from} \\ s_0 = s \text{ to } s_n = s'}} \gamma^n \prod_{i=1}^{n} P(s_i | s_{i-1}). \tag{8}$$

3.2 The Successor State Operator in a General State Space

M is also well-defined in general state spaces, using the Markov process formalism of Section 2, as follows. This extends [Day93] to arbitrary $\mathcal{S}$. (All proofs are given in the Appendix.)

Theorem 2. *The* successor state operator *M of a Markov reward process is defined as*

$$M := \sum_{n \geqslant 0} \gamma^n P^n, \qquad M(s_1, \mathrm{d}s_2) = \sum_{n \geqslant 0} \gamma^n P^n(s_1, \mathrm{d}s_2). \tag{9}$$

where $P^0 := \mathrm{Id}$. Thus, for each s_1, $M(s_1, \mathrm{d}s_2)$ is a measure on s_2, with total mass $\frac{1}{1-\gamma}$.

Then M is a well-defined operator over the set $B(\mathcal{S})$ of bounded measurable functions on $\mathcal{S}$. Moreover,

$$M = (\mathrm{Id} - \gamma P)^{-1} \tag{10}$$

as operators over $B(\mathcal{S})$, and

$$\text{and} \quad V = MR \tag{11}$$

for any reward function R. (Note that M does not depend on R.)

M can be interpreted as *paths* in the Markov process: $M(s_1, \mathrm{d}s_2)$ represents the number of paths from s_1 to s_2, weighted by their probability and discounted by their length. This will be relevant to compare the algorithms below. Indeed, in the finite-state case and using matrix notation, $P^n_{ss'}$ is the probability to go from s to s' in n steps; therefore

$$M_{ss'} = \sum_{n \geqslant 0} \gamma^n (P^n)_{ss'} = \sum_{n \geqslant 0} \gamma^n \sum_{s=s_0, s_1, \dots, s_{n-1}, s_n = s'} P_{s_0 s_1} \cdots P_{s_{n-1} s_n} \tag{12}$$

$$= \sum_{p \text{ path from } s \text{ to } s'} \gamma^{|p|} \, \mathbb{P}(p) \tag{13}$$

where, if $p = (s_0, ..., s_n)$ is a path, $\mathbb{P}(p) = P_{s_0 s_1} \cdots P_{s_{n-1} s_n}$ is its probability and $|p| = n$ its length. The same holds with integrals instead of sums in continuous spaces.

Yet another interpretation of M is via expected eligibility traces: indeed, when visiting a state s, the expectation of the eligibility trace vector $(e_{s'})$ is directly related to $M_{s's}$. The details are given in Appendix D; see also the discussion of "predecessor features" in [vHMH+20].

Successor states and successor representations. Given a function φ over the state space $\mathcal{S}$, the expectation of the cumulated, discounted future values of φ given the starting point s_0 of a trajectory (s_t) is

$$\mathbb{E}\left[\sum_{t \geqslant 0} \gamma^t \varphi(s_t)\right] = \sum_{t \geqslant 0} \gamma^t (P^t \varphi)(s_0) = (M\varphi)(s_0). \tag{14}$$

Thus, the successor representation (e.g., in the sense of [KSGG16]) of a state s is obtained by applying M to some user-chosen function φ.

Representing and learning the successor state operator. With continuous states, M cannot be represented as a matrix. Instead, we will learn a function of a pair of states. Namely, we will learn a parametric model of M via its density with respect to the data distribution ρ over states (this choice makes every algorithm samplable from the data). We present two versions of this. The first version represents M as

$$M(s_1, \mathrm{d}s_2) \approx \tilde{m}_\theta(s_1, s_2) \rho(\mathrm{d}s_2) \tag{15}$$

and the second version as

$$M(s_1, \mathrm{d}s_2) \approx \delta_{s_1}(\mathrm{d}s_2) + m_\theta(s_1, s_2) \rho(\mathrm{d}s_2) \tag{16}$$

where δ_{s_1} is the Dirac measure at s_1, and where $\tilde{m}_\theta$ and m_θ are functions over pairs of states, depending smoothly on some parameter θ. We will derive well-principled algorithms to learn the functions $\tilde{m}(s_1, s_2)$ and $m(s_1, s_2)$ from

observations of the Markov process. The data distribution ρ is unknown, but all algorithms below only require the ability to sample states from ρ, which we can do by definition since ρ is the distribution of states in the dataset. These two models correspond, respectively, to

$$V(s_1) = \mathbb{E}_{s_2 \sim \rho}[\tilde{m}_\theta(s_1, s_2) R(s_2)] \tag{17}$$

and

$$V(s_1) = R(s_1) + \mathbb{E}_{s_2 \sim \rho}[m_\theta(s_1, s_2) R(s_2)]. \tag{18}$$

The first version is simpler. The motivation for the second version is as follows. In continous spaces, M has a singular part, corresponding to the immediate reward in V, and to the term Id in the series for M: for each s_1, the measure $M(s_1, \cdot)$ comprises a Dirac mass at s_1. In continuous spaces, this singular part cannot be represented as $\tilde{m}(s_1, s_2)\rho(\mathrm{d}s_2)$ for a smooth function $\tilde{m}$. But since this singular part δ_{s_1} is known, we can just parameterize and learn the absolutely continuous part $m(s_1, s_2)$. Thus, the second version may represent M exactly (at least if P is smooth), while in general the first version cannot. Still, the first version may provide useful approximations.

The function $m_\theta(s_1, s_2)$ can be interpreted as a (directed) similarity measure between s_1 and s_2, coming from the structure of the Markov process.

In this text, we define several algorithms for learning m_θ: the extension of temporal difference (TD) to successor states (Section 4.1); backward TD for successor states (Section 4.2); and second-order-type methods (Section 7). The matrix-factorized forward-backward parameterization $\tilde{m}_\theta(s_1, s_2) = F_\theta(s_1)^\top B_\theta(s_2)$ has many additional properties and is treated in Section 6.

A learned model of M can be used in several ways:

- M may be used to improve learning for a given reward. For instance, with a sparse reward located at a *known* target state s_{tar}, then $V(s) = M(s, \mathrm{d}s_{\mathrm{tar}})$. In that case, learning M directly provides the value function, while ordinary TD would not work because of the sparse reward. With dense rewards, M can be used in the learning of the value function (Section 8).

- Objects similar to M may be used to learn goal-dependent policies, such as learning how to reach any arbitrary state. This does not cover dense rewards, but extends to reaching states with arbitrary values for some features. This is covered in Section 5.

Section 8 gives more details about the ways to use M to learn value functions and policies.

4 TD Algorithms for Deep Successor State Learning

4.1 The (Forward) TD Algorithm for Successor States

4.1.1 The Forward Bellman Equation

Theorem 3 (Bellman equation for successor states). *The successor state operator M is the only operator which satisfies the Bellman equation $M = \mathrm{Id} + \gamma PM$.*

This Bellman equation makes sense, as operators, on any state space, discrete or continuous. In finite spaces, each column of the matrix M contains the value function for a reward located at a specific target state, and the Bellman equation for M is just the collection of the standard Bellman equations for every target state; the Id term is the reward for reaching state s when the target is s.

This Bellman operator on M has the same contractivity properties as the usual Bellman operator.

Proposition 4 (Contractivity of the Bellman operator on M). *Equip the space of functions $B(\mathcal{S})$ with the sup norm $\|f\|_\infty := \sup_{s\in S} |f(s)|$. Equip the space of bounded linear operators from $B(\mathcal{S})$ to $B(\mathcal{S})$ with the operator norm $\|M\|_{\mathrm{op}} := \sup_{f\in B(\mathcal{S}),\, f\neq 0} \|Mf\|_\infty / \|f\|_\infty$.*

Then the Bellman operator $M \mapsto \mathrm{Id} + \gamma PM$ is γ-contracting for this norm.

Consequently, for any learning rate $\eta \leqslant 1$, iterated application of the Bellman operator $M \leftarrow (1-\eta)M + \eta(\mathrm{Id} + \gamma PM)$ converges to the successor state operator.

4.1.2 Forward TD for Successor States: Tabular Case

Given that the Bellman equation on M is a collection of ordinary Bellman equations for every target state, an obvious algorithm to learn M in finite state spaces is to perform ordinary TD in parallel for all these single-state rewards, as in [Day93]. Let s_{tar} be some target state and consider the reward $\mathbb{1}_{s_{\mathrm{tar}}}$. Upon observing a transition $s \to s'$, ordinary TD for this reward updates V by $V_s \leftarrow V_s + \eta\, \delta V_s$, where η is some learning rate and $\delta V_s = \mathbb{1}_{s=s_{\mathrm{tar}}} + \gamma V(s') - V(s)$. Performing TD in parallel for every column of M with target state s_{tar} is equivalent to the following [Day93].

Definition 5 (Tabular temporal difference for successor states). *The TD algorithm for M, in a finite state space, maintains M as a matrix. Upon observing a transition $s \to s'$ in the Markov process, M is updated by $M \leftarrow M + \eta\, \delta M$ where η is a learning rate and δM has entries*

$$\delta M_{ss_2} := \mathbb{1}_{s=s_2} + \gamma M_{s's_2} - M_{ss_2} \qquad \forall s_2 \tag{19}$$

In the tabular case and with deterministic rewards, learning M via TD, then estimating V via the matrix product $V = MR$, is equivalent to directly learning V via tabular TD (Appendix A.3): tabular TD on M treats all target states s_2 as independent learning problems, and no learning gain is achieved.

However, this equivalence does not hold with function approximation, which introduces generalization between states. Since any target state is reached with zero probability, applying parametric TD naively in parallel for every target state would always provide reward 0 in continuous environments. The parametric TD updates we present below are not equivalent to this naive TD: they have the same expectation but avoid the zero-reward problem.

4.1.3 Forward TD for Successor States: Function Approximation

In continuous environments, it is not possible to store M as a matrix. But we can maintain a model m_θ of the density of M, as explained in Section 3.2. As in usual parametric TD, we learn θ by defining an "ideal" update given by the Bellman equation, and update θ so that M gets closer to it.

Theorem 6 (TD for successor states with function approximation). *Maintain a parametric model of M as in Eq. 16 via $M_{\theta_t}(s_1, \mathrm{d}s_2) = \delta_{s_1}(\mathrm{d}s_2) + m_{\theta_t}(s_1, s_2)\rho(\mathrm{d}s_2)$, with θ_t the value of the parameter at step t, and with m_θ some smooth family of functions over pairs of states.*

Define a target update of M via the Bellman equation, $M^{\mathrm{tar}} := \mathrm{Id} + \gamma P M_{\theta_t}$. Define the loss between M and M^{tar} via $J(\theta) := \frac{1}{2}\left\|M_\theta - M^{\mathrm{tar}}\right\|_\rho^2$ using the norm (1). Then the gradient step on θ to reduce this loss is

$$-\partial_\theta J(\theta)_{|\theta=\theta_t} = \mathbb{E}_{s\sim\rho,\, s'\sim P(s,\mathrm{d}s'),\, s_2\sim\rho}\big[\gamma\, \partial_\theta m_{\theta_t}(s, s') + \partial_\theta m_{\theta_t}(s, s_2)\left(\gamma m_{\theta_t}(s', s_2) - m_{\theta_t}(s, s_2)\right)\big]. \quad (20)$$

For the model variant in Eq. 15, $M_{\theta_t}(s_1, \mathrm{d}s_2) = \tilde{m}_{\theta_t}(s_1, s_2)\rho(\mathrm{d}s_2)$, the gradient step on θ to reduce the loss $J(\theta)$ is

$$-\partial_\theta J(\theta)_{|\theta=\theta_t} = \mathbb{E}_{s\sim\rho,\, s'\sim P(s,\mathrm{d}s'),\, s_2\sim\rho}\big[\partial_\theta \tilde{m}_{\theta_t}(s, s) + \partial_\theta \tilde{m}_{\theta_t}(s, s_2)\left(\gamma \tilde{m}_{\theta_t}(s', s_2) - \tilde{m}_{\theta_t}(s, s_2)\right)\big]. \quad (21)$$

This gradient step is "samplable". Namely, we can define a stochastic update $\widehat{\delta\theta_{\mathrm{TD}}}$ with expectation (20): sample a transition $s \to s'$ from the dataset of transitions, and *another* independent "destination" state s_2 from the dataset, then set

$$\widehat{\delta\theta_{\mathrm{TD}}} = \gamma\, \partial_\theta m_\theta(s, s') + \partial_\theta m_\theta(s, s_2)\left(\gamma m_\theta(s', s_2) - m_\theta(s, s_2)\right) \quad (22)$$

or likewise for $\tilde{m}$ (only the first term is different).

This algorithm uses a transition $s \to s'$ and one additional random state s_2, independent from s and s'. The Bellman–Newton update (Section 7.3) will use two additional random states s_1 and s_2 (but no additional transition). The law of s_2 is ρ, which means s_2 is just another state sampled from the dataset. For instance, if the dataset consists of a sampled trajectory trajectory $(s_t)_{t\geqslant 0}$, when observing a transition $s_t \to s_{t+1}$, additional independent state samples can be obtained by using states $s_{t'}$ at times t' independent from t (such as a random $t' \leqslant t$). This requires maintaining a replay buffer of observed states.

Several variants avoid having to sample s_2 independently from $s \to s'$. In the FB representation of M (Section 6), the expectation over s_2 can be estimated online using just the observed transition $s \to s'$, with no additional state. Appendix G also describes the possibility of using a "cheap" source for the additional states s_2 instead of actual states, as long as the transitions $s \to s'$ come from the true process. Finally, Theorem 13 makes it possible to use a joint rather than independent distribution for s and s_2 (such as choosing a target state s_2 and following an s_2-dependent policy for some time).

4.1.4 Infinitely Sparse Rewards and Forward TD vs TD on State-Goal Pairs

Why don't the Dirac rewards show up in the parametric TD algorithm of Theorem 6? Why don't the rewards become infinitely sparse with continuous states?

The tabular TD algorithm (19) for M features a sparse reward $\mathbb{1}_{s=s_2}$. Why don't these sparse rewards vanish completely in the continuous state limit, where an equality of states never occurs? This is simply because we know exactly when these terms make a contribution: namely, we know we can just take $s_2 = s$. In the continuous case, with a model $\tilde{m}(s, s_2)$, the sparse reward is a Dirac $\delta_s(\mathrm{d}s_2)$, and it shows up in TD as a term $\partial_\theta \tilde{m}(s, s_2)\delta_s(\mathrm{d}s_2)$. When integrated over s_2, this term is just $\partial_\theta \tilde{m}(s, s)$. Thus the contribution from the infinitely sparse Dirac term is actually finite and nonzero.

Intuitively, we are solving RL problems with an infinity of infinitely sparse target states s_2. But at every time step, when we visit state s, we know that we just visited the target state $s_2 = s$: every step brings a reward. This knowledge is exploited in the expressions we give for TD, resulting in a finite contribution $\gamma\, \partial_\theta m_{\theta_t}(s, s')$ in (20).

Algorithmically, it is quite important to use this. In algorithms that sample a target state s_2 fully independently from s (such as picking a random goal g in [SHGS15]), the contribution from the reward $\mathbb{1}_{s=s_2}$ is sometimes nonzero in the tabular case, but gets infinitely sparse and eventually vanishes in the continuous case. We provide more details in Section 4.1.4 (see also Section 5 for a discussion of state-goal resampling strategies such as hindsight

experience replay [AWR+17]).

On the other hand, successor states learned via Theorem 6 can in principle learn an infinite number of infinitely sparse rewards, with every transition being informative.

The state-goal process. In expectation, one can view forward TD for M as ordinary TD on the space of pairs (s, s_2), as follows. For the tabular case this holds without expectations, but for the parametric case, this equivalence holds only in expectation: ordinary parametric TD on pairs (s, s_2) would have infinite variance on continuous spaces due to the Dirac reward $\delta_s(\mathrm{d}s_2)$, but the successor state update in Theorem 6 avoids this infinite variance, as discussed above.

In the tabular case, the equivalence is a direct consequence of the Definition 5 for tabular forward TD on M.

Proposition 7 (Tabular forward TD on M as ordinary TD on state-goal pairs). *Let P be the transition matrix of the Markov process on state space S. We call* state-goal Markov process *the Markov process on $S \times S$ whose transition matrix is $P \otimes \mathrm{Id}$, namely (s, s_2) goes to (s', s_2) with $s' \sim P(\mathrm{d}s'|s)$.*

Let S be discrete. Then tabular TD for successor states on S (Definition 5) is equivalent to ordinary tabular TD on the value function of the state-goal process for the reward function $R(s, s_2) = \mathbb{1}_{s=s_2}$.

The parametric case is handled as follows. In discrete or continuous state spaces, the successor state operator $M(s, \mathrm{d}s_2)$ satisfies the Bellman equation $M(s, \mathrm{d}s_2) = \delta_s(\mathrm{d}s_2) + \gamma \mathbb{E}_{s' \sim P(\mathrm{d}s'|s)} M(s', \mathrm{d}s_2)$ as measures over s_2. Consider the parameterization (15), $M(s, \mathrm{d}s_2) = \tilde{m}_\theta(s, s_2)\rho(\mathrm{d}s_2)$ where m_θ is some parametric function (the parameterization (16) with m_θ is similar). The Bellman equation rewrites as $\tilde{m}_\theta(s, s_2)\rho(\mathrm{d}s_2) = \delta_s(\mathrm{d}s_2) + \gamma \mathbb{E}_{s' \sim P(\mathrm{d}s'|s)} \tilde{m}_\theta(s, s_2)\rho(\mathrm{d}s_2)$. If S is discrete, the ratio of measures $\delta_s(\mathrm{d}s_2)/\rho(\mathrm{d}s_2)$ is an ordinary function and we can rewrite the successor state Bellman equation as

$$\tilde{m}_\theta(s, s_2) = \frac{\delta_s(\mathrm{d}s_2)}{\rho(\mathrm{d}s_2)} + \gamma \mathbb{E}_{s' \sim P(\mathrm{d}s'|s)} \tilde{m}_\theta(s, s_2). \tag{23}$$

This is the Bellman equation over state-goal pairs (s, s_2) for the reward function $R(s, s_2) := \delta_s(\mathrm{d}s_2)/\rho(\mathrm{d}s_2)$ and transition matrix $P \otimes \mathrm{Id}$. It is similar to goal-dependent value functions (as in, e.g., [SHGS15]), up to the $1/\rho(\mathrm{d}s_2)$ factor necessary to turn measures into functions. Parametric TD using this equation is just the average of parametric TD for the individual value functions associated to each goal s_2.

Naive TD on this state-goal Bellman equation does not behave well due to the sparse reward $\delta_s(\mathrm{d}s_2)$: most pairs have reward 0 and this induces high variance. In continuous spaces, TD on this equation degenerates: the reward

is 0 with probability 1 but its variance is infinite due to the infinite Dirac function $\delta_s(\mathrm{d}s_2)/\rho(\mathrm{d}s_2)$. However, the *expected* TD update can be computed algebraically and results in the finite-variance update for successor states. Thus we have the following result.

Proposition 8 (Parametric TD on M as finite-variance version of parametric TD on goal-state pairs). *Let the state space S be discrete. Then the Bellman equation $M = \mathrm{Id} + \gamma PM$ for successor states is equivalent to the ordinary Bellman equation (23) for the state-goal process on pairs (s, s_2) with reward function $R(s, s_2) := \delta_s(\mathrm{d}s_2)/\rho(\mathrm{d}s_2)$.*

Moreover, in expectation over state-goal samples $(s, s_2) \sim \rho \otimes \rho$, the ordinary parametric TD update for the Bellman equation (23) of the state-goal process is equal to the parametric TD update for successor states from Theorem 6, both for the parameterizations m_θ and $\tilde{m}_\theta$.

Let the state space S be continuous, with ρ covering the whole space. Then ordinary parametric TD for the Bellman equation (23) on the state-goal process is undefined: the reward term is 0 with probability 1 but has infinite variance. On the other hand, its expectation is well-defined, and the parametric TD update for successor states from Theorem 6 has the same expectation but finite variance (for smooth and bounded m_θ).

4.1.5 Convergence properties for TD on successor states

Forward TD for M converges in the same conditions as ordinary TD for the value function. This is obtained by viewing forward TD for M as ordinary TD on the space of pairs (s, s_2), as in Section 4.1.4. Thus, interpreting TD for successor states as TD on the state-goal process immediately transfers existing convergence results for ordinary TD to successor states.

We consider three such results: convergence of tabular TD, convergence of TD on-policy with a linear parameterization, and convergence of TD on-policy for any parameterization if the random walk is reversible. In each case, we refer to the original works for additional technical conditions (learning rates, smoothness...)

- In the tabular case, forward TD on M (Definition 5) converges, with pairs (s, s_2) sampled at each step from essentially any selection scheme (stochastic or deterministic) that ensures every pair is selected infinitely often, and with suitable learning rates [Tsi94].
- TD with linear parameterization on discrete spaces is known to converge *on-policy* [TVR97], namely, with states sampled according to a steady-state distribution of the Markov process (assumed to be nonzero on every state). For successor states this translates to the following. Assume S is discrete and the successor state operator is parameterized

as

$$M(s, \mathrm{d}s_2) \approx \sum_i \theta_i \, \varphi_i(s, s_2) \rho(\mathrm{d}s_2) \tag{24}$$

or equivalently $\tilde{m}_\theta(s, s_2) = \sum_i \theta_i \, \varphi_i(s, s_2)$, where $\theta = (\theta_1, \ldots, \theta_k)$ is the parameter to be learned, and $\varphi_1, \ldots, \varphi_k$ are fixed functions. Assume ρ is a positive steady-state distribution of the Markov operator P, and let ρ_2 be any positive distribution over S. Then $\rho \otimes \rho_2$ is a steady-state distribution of the Markov operator $P \otimes \mathrm{Id}$ over state-goals, and parametric TD for the Bellman equation (23) with pairs (s, s_2) sampled from $\rho \otimes \rho_2$, is convergent for suitable learning rates. This also covers the parametric update in Theorem 6, which has the same expectation by Proposition 8.

- For TD with arbitrary parametric families, convergence is known assuming that the Markov operator P is *reversible*, namely, that ρ is its steady-state distribution and further satisfies the *detailed balance* condition $\rho(\mathrm{d}s)P(\mathrm{d}s'|s) = \rho(\mathrm{d}s')P(\mathrm{d}s|s')$, in other words, steady-state flows from state s to s' and s' to s are equal. Then, parametric TD is a stochastic gradient descent of a global loss between the approximate and true value function [Oll18]. This result extends to MDPs which are "reversible enough" [BB19]. Applying the result of [Oll18] to successor states via the state-goal process yields the following. Assume that the space S is finite and that the Markov operator P is reversible. Let $\tilde{m}_\theta$ be any smooth parametric model for successor states. Let $\tilde{m}^*$ be the true value, namely, let the true successor state operator be $M(s, s_2) = \tilde{m}^*(s, s_2)\rho(\mathrm{d}s_2)$. Define the loss function

$$\ell(\theta) := (1 - \gamma) \, \|\tilde{m}_\theta - \tilde{m}^*\|^2_{\rho\otimes\rho} + \gamma \, \|\tilde{m}_\theta - \tilde{m}^*\|^2_{\mathrm{Dir}} \tag{25}$$

where $\|f\|^2_{\rho\otimes\rho} := \mathbb{E}_{s\sim\rho,\, s_2\sim\rho} f(s, s_2)^2$ and the *Dirichlet norm* is

$$\|f\|^2_{\mathrm{Dir}} := \frac{1}{2}\mathbb{E}_{s\sim\rho,\, s_2\sim\rho,\, s'\sim P(\mathrm{d}s'|s)}(f(s', s_2) - f(s, s_2))^2. \tag{26}$$

Then the parametric TD step for M (Theorem 6) is equal to the gradient of this loss, $-\frac{1}{2}\partial_\theta \ell(\theta)$. (This is a global loss between the parametric model and the true value $\tilde{m}^*$, contrary to the loss in Theorem 6 which uses a loss with respect to the right-hand-side of the Bellman equation, which depends on the current estimate.)

Thus, in the reversible case with ρ the stationary distribution, parametric TD for M converges to a local minimum of the global loss (25), under the general conditions for convergence of stochastic gradient descent.

4.1.6 Variants of Forward TD: Target Networks, Multi-Step Returns, $\gamma = 1$, Using Features as Targets...

The variants of TD used in practice also exist for successor states.

In Appendix A we provide the parametric updates for two variants: using a *target network* (namely, performing several gradient steps toward $\mathrm{Id} + \gamma P M^{\mathrm{tar}}$ without updating M^{tar}), and using *multi-step returns*.

Appendix A.6 describes *relative TD* for successor states: this makes it possible to deal directly with $\gamma = 1$ and to reduce variance for γ close to 1.

Appendix G deals with using different probability distributions for s and s_2 (e.g., using synthetic states for s_2 to have more samples), and using a different reference measure for the parameterization of M (e.g., representing M by its density with respect to the uniform measure rather than the unknown distribution ρ in (16) and (15)).

Appendix A.5 mentions situations where the reward is known to depend only on some features $\varphi(s)$ of the state s (such as a subset of coordinates of s). A typical example would be a specific target value for $\varphi(s)$. In that case, it is enough to learn the successor state operator $M(s, \mathrm{d}g)$ with the second argument in the space of features, $g = \varphi(s)$. Then $M(s, \mathrm{d}g)$ directly provides the value function of the problem with a reward when $\varphi(s)$ is equal to g. It can be used to express the value function of any reward that depends only on g. The forward TD updates for $M(s, \mathrm{d}g)$ are similar to the case of $M(s, \mathrm{d}s_2)$.

4.2 Backward TD for Successor States

Theorem 9. *The successor state operator M is the only operator which satisfies the backward Bellman equation, $M = \mathrm{Id} + \gamma M P$.*

This equation has no analogue on V. It is similar to an update of expected eligibility traces (see Appendix D). The resulting operator has the same contractivity properties as the usual (forward) Bellman operator.

Proposition 10 (Contractivity of the backward Bellman operator on M). *Equip the space of functions $B(\mathcal{S})$ with the sup norm $\|f\|_\infty := \sup_{s \in S} |f(s)|$. Equip the space of bounded linear operators from $B(\mathcal{S})$ to $B(\mathcal{S})$ with the operator norm $\|M\|_{\mathrm{op}} := \sup_{f \in B(\mathcal{S}),\, f \neq 0} \|Mf\|_\infty / \|f\|_\infty$.*

Then the backward Bellman operator $M \mapsto \mathrm{Id} + \gamma M P$ is γ-contracting for this norm.

The corresponding parametric update to bring M closer to $\mathrm{Id} + \gamma M P$, similar to Theorem 6, is

$$\widehat{\delta\theta_{\mathrm{BTD}}} = \gamma\, \partial_\theta m_\theta(s, s') + m_\theta(s_1, s) \left(\gamma\, \partial_\theta m_\theta(s_1, s') - \partial_\theta m_\theta(s_1, s)\right) \tag{27}$$

for the model (32) using m_θ, and

$$\widehat{\delta\theta_{\mathrm{BTD}}} = \partial_\theta \tilde{m}_\theta(s, s) + \tilde{m}_\theta(s_1, s) \left(\gamma\, \partial_\theta \tilde{m}_\theta(s_1, s') - \partial_\theta \tilde{m}_\theta(s_1, s)\right) \tag{28}$$

for the model (33) using $\tilde{m}_\theta$. Here a transition $s \to s'$ and another, independent state s_1 are both sampled from the dataset. A precise statement is given in Theorem 26 (Appendix A.4).

Backward TD for M is not structurally different from forward TD: it corresponds to forward TD for the "time-reversed" Markov process (Appendix D). But since states are typically observed in a time-ordered sequence, this might produce a difference. In general, the backward TD update (27) does not look like a time-reversal of the forward TD update (37): (27) involves Bellman gaps of gradients ∂m while (37) involves Bellman gaps of m. This difference is superficial and disappears in expectation under the stationary distribution: if we assume that ρ is the stationary distribution of the process, then (27) is equal in expectation to

$$\gamma\, \partial_\theta m_\theta(s, s') + \partial_\theta m_\theta(s_1, s') \left(\gamma m_\theta(s_1, s) - m_\theta(s_1, s')\right) \tag{29}$$

which looks more like a time-reversal of the forward TD update (37), with (time-reversed) Bellman gaps of m. [3]

Moreover, contrary to forward TD, learning M by backward TD then setting $V = MR$ is *not* equivalent to learning V via TD in the tabular case.

Mixing forward and backward TD can change the learning of M in various ways. In the tabular case and in the infinitesimal learning rate limit, such mixing substantially reduces the dimension of the subspace of M where convergence is slowest (Section 9.2). With the matrix-factorized parameterization of Section 6, using forward, or backward TD, or a mixture of the two, provides approximations of M using slightly different criteria (Appendix E).

There is no version of backward TD for the goal-dependent optimal Q function of Section 5.1. Performing a random step on the goal state does not commute with optimizing an action depending on the goal state. With a fixed policy, backward TD is forward TD on a time-reversed Markov process, but when choosing actions, time reversal is not possible: in the expectimax problem (45), each action choice may depend both on previous actions and on the goal state, and reversing time is not possible. Similarly, there is no backward TD for the target-feature version of Section A.5, as the features do not generally contain full information about the next transition and the future features.

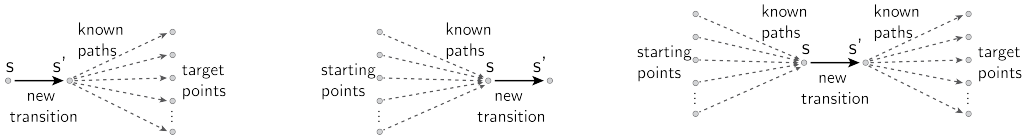

Figure 1: Combining paths: forward TD, backward TD, and path composition (Bellman–Newton).

4.3 Path Combinatorics Interpretation: Incorporating Newly Observed Transitions

The difference between forward and backward TD for M is best understood in the path viewpoint on M (Eq. 12). Indeed, the current estimate of $M_{s_1 s_2}$ contains a current estimation on the number of paths from s_1 to s_2, weighted by their estimated probabilities in the Markov process. TD replaces M with PM, and adds Id, which represents the trivial paths from s to s. Backward TD uses MP instead. In both cases, the operator P is sampled via an observed transition $s \to s'$. Thus, PM builds new known paths by taking all paths contained in M and adding the transition $s \to s'$ at the front of each path, while MP adds the transition $s \to s'$ at the back of each path in M (Fig. 1). Forward TD reasons at fixed *target states* (rewards) [GO19], while backward TD reasons at fixed *starting points*.

Thus, TD and backward TD on M differ in how they learn new paths from known paths when each new transition is observed. Arguably, both are reasonable ways to update a mental model of paths in an environment when discovering new transitions (e.g., if a new street $s \to s'$ opens in a city).

There is a third way to build new paths when observing a new transition $s \to s'$: take all known paths to s, all known paths from s', and insert $s \to s'$ in the middle (Fig. 1). This exploits path concatenation, roughly doubling the length of known paths. This operation is involved in the way that M actually changes when the process is changed by increasing $P(s, \mathrm{d}s')$ (the way possible paths actually change when a new street opens). This is the basis of the "second-order" algorithms we present for M in Section 7.

[3] The difference between (29) and (27) just lies in shifting terms around along a trajectory $s \to s' \to s'' \to \dots$: in one case, the term $m_\theta(s_1, s')\partial_\theta m_\theta(s_1, s')$ is grouped with the previous transition $s \to s'$, in the other case, with the next transition $s' \to s''$. Thus the difference is minor if working online along trajectories, but (27) is valid even if ρ is not the stationary distribution.

5 Multiple Policies: Goal-Dependent Q and V functions

The principles above can be used to learning goal-dependent policies and a goal-dependent value or Q function, just by letting the policy be goal-dependent in the results above. This option only covers rewards located at a given target state, not dense rewards; it can also cover target *features* of states rather than a fully specified target state (Section 5.2), namely, having target values for some function φ of the state. A first application is to learn all optimal policies to reach any goal state s_2, either via Q-learning (Section 5.1) or V-learning (Section 5.2).

This approach partially solves the well-known sparse reward problem in goal-dependent learning. For instance, let us consider the goal-dependent value function $V(s, \mathrm{d}s_2)$: for every target state s_2, it solves the s_2-dependent Bellman equation

$$V(s, \mathrm{d}s_2) = \delta_s(\mathrm{d}s_2) + \gamma \mathbb{E}_{s' \sim P(\mathrm{d}s'|s,s_2)} V(s', \mathrm{d}s_2) \tag{30}$$

with reward when $s = s_2$, and s_2-dependent policy $P(\mathrm{d}s'|s, s_2)$. (In the continuous case, the goal-dependent value function is a *measure* on s_2, because the probability to exactly reach a state is usually 0. We will learn its density with respect to a reference measure.)

Using TD directly on this equation leads to sparse reward problems: in continuous state spaces, a reward is never observed (and rarely observed in large discrete spaces).

However, the contribution of the reward $\delta_s(\mathrm{d}s_2)$ to the TD update can be computed exactly in expectation. The resulting update does not involve sparse rewards anymore: every transition is informative because it shows how to reach the currently visited state (as discussed in Section 4.1.3). This update is the same as with successor states: the update for $\tilde{m}$ in Theorem 6 can be directly used to train goal-dependent policies by seeing $\tilde{m}(s, s_2)$ as the value function at s when the goal state is s_2 (see the example after Theorem 13).

Existing workarounds for this sparse reward issue include strategies for resampling state-goal pairs that more frequently lead to nonzero rewards, such as *Hindsight Experience Replay* (HER) [AWR+17], which works with any Q-learning method, assuming knowledge of the reward function associated to each goal. It is not clear to us whether HER actually solves the infinitely-sparse-reward issue or not. [4] The results described here are not mutually exclusive with using HER: HER is a sampling strategy for transitions in

[4] For instance, with noisy dynamics in a continuous space, the probability to reach a state exactly is always 0, so if the reward is 1 when reaching the state, the Q-function computed by HER would be 0. Here we have used infinite (Dirac) rewards when reaching a goal: this leads to a well-defined, nonzero Q function, but rescaling the reward by an infinite factor would result in infinite HER updates. On the other hand, in some non-noisy

the training set, which can be used with any Q-learning method, such as those presented here; so in principle HER could be used as the state-goal distribution ρ_{SG} for Q-learning in Theorem 13.

We start with learning the optimal Q function for every target state (Section 5.1). We first describe the precise meaning of goal-dependent Bellman equations such as (30), and present the resulting parametric update.

Next we turn to a more general statement involving either the V or Q function, and target *features* instead of target states (Section 5.2). We discuss three use cases: Q-learning with any goal feature function, V-learning conditioned to goal states, and V-learning conditioned to goal features, which presents some subtleties. The goal-dependent V function can be used to train goal-dependent policies by any policy gradient method.

In Section 5.3 we provide mathematical details for the existence and uniqueness of goal-dependent Bellman equations, in the case of the Q function. Having to work with measures of potentially infinite mass results in non-uniqueness of the solution, but there is still a "natural" solution, equal both to the smallest solution and to the limit of the finite-horizon solution.

5.1 The Optimal Q-function for Every Goal State

Several works have attempted to learn optimal Q functions indexed by an additional "goal" which encodes a variable reward. The simplest case is when the reward is located at a single goal state g. Computing the Q function $Q(s, a, g)$ for every goal state g fully solves the navigation problem in an environment, although this function does not provide the optimal policies for "mixed" rewards, only for single-state rewards.

The viewpoint presented here allows for a more principled approach to this object Q; notably, it can avoid the sparse reward problem of algorithms that sample a state s and a goal state g independently, with reward $\mathbb{1}_{s=g}$. This is avoided thanks to the direct algebraic treatment of Diracs or sparse rewards discussed above.

So far, the successor state operator was defined for a given, fixed policy. The goal-dependent Q function uses a different (optimal) policy for every goal. It can be defined through the optimal Bellman operator.

Definition 11 (Optimal Bellman operator for successor states). *Let $Q(s, a, \mathrm{d}s_2)$ be a measure on s_2 depending on a state-action pair (s, a). Define the* optimal Bellman operator *T via*

$$(TQ)(s, a, \mathrm{d}s_2) := \delta_s(\mathrm{d}s_2) + \gamma\, \mathbb{E}_{s' \sim P(s'|s,a)} \sup_{a'} Q(s', a', \mathrm{d}s_2). \tag{31}$$

continuous MDPs with continuous actions, it is possible to reach a state exactly, and in that instance HER would work without modification. This point needs more investigation.

In the discrete case, this is just the usual optimal Bellman operator in parallel for every goal state s_2, namely, $(TQ)(s,a,s_2) = \mathbb{1}_{s=s_2} + \gamma\, \mathbb{E}_{s'\sim P(s'|s,a)} \max_{a'} Q(s',a',s_2)$. In the continuous case, for each state-action (s,a), $Q(s,a,\cdot)$ is a measure over the state space, and the supremum $\sup_{a'} Q(s',a',\mathrm{d}s_2)$ is a supremum of measures over s_2. [5]

In the discrete case, a fixed point exists by standard contractivity arguments; however, with continuous states, the situation is tricky, see Section 5.3. In particular, with continuous states the measure Q may have either finite or infinite mass; intuitively, the total mass of Q indicates how many distinct policies we can follow to reach different states. The total mass of $Q(s,a)$ is the total number of distinct points that can be reached from (s,a) by taking different action sequences, weighted by the probability and discounted by time. In contrast, the successor state operator of a single fixed policy (Sections 3–4) always has total mass $\sum \gamma^t = \frac{1}{1-\gamma}$: there is no choice of actions so the total probability of states is 1 at each time step.

To see this, consider two extreme examples. In the first, the environment just ignores every action and sends the agent to a random uniform state at each time step. Then for any (s,a), $Q(s,a,\mathrm{d}s_2)$ is $\delta_s(\mathrm{d}s_2) + \frac{\gamma}{1-\gamma}\,\mathrm{d}s_2$, with total mass $\frac{1}{1-\gamma}$. In the second example, for every state we have an action that sends us directly to that state. Then $Q(s,a,\mathrm{d}s_2)$ is a measure for which *every* single state $s_2 \neq s$ has mass $\frac{\gamma}{1-\gamma}$, and the total mass of $Q(s,a,\cdot)$ is infinite. This can be arranged even with finite action spaces: generally, at horizon t the mass may be as large as $\gamma^t(\#A)^t$ if every action sequence leads to a different part of the state (examples in Appendix B.4). In the fixed-policy case, the mass at horizon t was always γ^t and the total mass was always finite.

Parametric goal-dependent Q-learning. Let us consider parametric models for Q. As before, we consider two models given by

$$Q_\theta(s,a,\mathrm{d}s_2) := \delta_s(\mathrm{d}s_2) + q_\theta(s,a,s_2)\rho(\mathrm{d}s_2) \tag{32}$$

and

$$Q_\theta(s,a,\mathrm{d}s_2) := \tilde{q}_\theta(s,a,s_2)\rho(\mathrm{d}s_2) \tag{33}$$

respectively, and we will learn q_θ and $\tilde{q}_\theta$. For instance, up to the factor ρ, the models in [SHGS15] correspond to $\tilde{q}_\theta(s,a,s_2) = h(\varphi_\theta(s,a), \psi_\theta(s_2))$. [6]

[5] In general, the supremum of k measures $\mu_1,\dots,\mu_k$ is defined as follows: for every measurable set A, $(\sup_i \mu_i)(A) := \sup_{(B_i)} \sum \mu_i(B_i)$ where the supremum is taken over all partitions of $A = B_1 \sqcup B_2 \sqcup \cdots \sqcup B_k$ into disjoint measurable sets (B_i). This is also the smallest measure that is larger than every μ_i. Each B_i is the set where μ_i is the largest measure in the family. This means that at each point in state space, we select the measure with the highest value; thus, the sup over actions in (31) depends on the goal states s_2.

The definition assumes that the set of actions is countable; otherwise, additional smoothness assumptions are required for existence.

[6] The factor ρ, or some other measure, is needed to get a well-defined object in continuous state spaces. In discrete spaces, it results in an s_2-dependent scaling of the Q function,

The resulting parametric update is as follows. The update is off-policy: we assume access to a dataset of transitions (s, a, s') in a Markov decision process. Let ρ_{SA} be the distribution of the state-action pair (s, a) in the dataset; its marginal over s is ρ as before. Given a measure-valued function of (s, a), such as $Q(s, a, \mathrm{d}s_2)$, we define its norm similarly to (1) as

$$\|Q\|^2_{\rho_{\mathrm{SA}},\rho} := \mathbb{E}_{(s,a)\sim\rho_{\mathrm{SA}},\, s_2\sim\rho}[q(s, a, s_2)^2] \tag{34}$$

where $q(s, a, s_2) := Q(s, a, \mathrm{d}s_2)/\rho(\mathrm{d}s_2)$ is the density of Q with respect to ρ, if it exists (otherwise the norm is infinite).

Theorem 12 (Parametric Q-learning for every goal state). *Consider a parametric model of Q given by (32) or (33), where $q_\theta(s, a, s_2)$ or $\tilde{q}(s, a, s_2)$ are smooth functions depending on the parameter θ.*

Let θ_0 be some value of the parameter. Define a target update Q^{tar} of Q via the optimal Bellman operator (31) applied to Q_{θ_0}, namely, $Q^{\mathrm{tar}}(s, a, \mathrm{d}s_2) := (TQ_{\theta_0})(s, a, \mathrm{d}s_2) = \delta_s(\mathrm{d}s_2) + \gamma\, \mathbb{E}_{s'\sim P(s'|s,a)} \sup_{a'} Q_{\theta_0}(s', a', \mathrm{d}s_2)$. Define the loss between Q_θ and Q^{tar} via $J(\theta) := \frac{1}{2}\left\|Q_\theta - Q^{\mathrm{tar}}\right\|^2_{\rho_{\mathrm{SA}},\rho}$ using the norm (34).

Then the gradient step on θ to reduce this loss is

$$\begin{aligned} -\partial_\theta J(\theta) = \mathbb{E}_{(s,a)\sim\rho_{\mathrm{SA}},\, s'\sim P(s'|s,a),\, s_2\sim\rho} \big[&\gamma\, \partial_\theta q_\theta(s, a, s') \\ &+ \partial_\theta q_\theta(s, a, s_2) \left(\gamma \sup_{a'} q_{\theta_0}(s', a', s_2) - q_\theta(s, a, s_2)\right)\big] \end{aligned} \tag{35}$$

for the model (32) using q_θ, and

$$\begin{aligned} -\partial_\theta J(\theta) = \mathbb{E}_{(s,a)\sim\rho_{\mathrm{SA}},\, s'\sim P(s'|s,a),\, s_2\sim\rho} \big[&\partial_\theta \tilde{q}_\theta(s, a, s) \\ &+ \partial_\theta \tilde{q}_\theta(s, a, s_2) \left(\gamma \sup_{a'} \tilde{q}_{\theta_0}(s', a', s_2) - \tilde{q}_\theta(s, a, s_2)\right)\big] \end{aligned} \tag{36}$$

for the model (33) using $\tilde{q}_\theta$.

Here we have presented the update using a fixed "target network" with parameter θ_0 (typically a previous value of θ), a common practice for parametric Q-learning.

This update is "samplable": sample a transition (s, a, s') from the dataset, another independent transition (s_2, a_2, s_2') from the dataset (a_2 and s_2' are discarded), and estimate the gradient by

$$\widehat{\delta\theta} = \gamma\, \partial_\theta q_\theta(s, a, s') + \partial_\theta q_\theta(s, a, s_2) \left(\gamma \sup_{a'} q_{\theta_0}(s', a', s_2) - q_\theta(s, a, s_2)\right) \tag{37}$$

or likewise for $\tilde{q}$ (only the first term is different).

which still has the same optimal policy for each s_2.

This update is perfectly analogous to the successor state updates for m_θ and $\tilde{m}_\theta$ in Theorem 6, except that q_θ and $\tilde{q}_\theta$ depend on the actions, and that the policy follows a supremum over actions instead of being fixed.

As before, the infinite, infinitely sparse rewards $\delta_s(\mathrm{d}s_2)$ of the every-goal problem produce the finite contribution $\gamma\,\partial_\theta q_\theta(s,a,s')$ or $\partial_\theta \tilde{q}(s,a,s)$ in this parametric update. Sampling two independent states s and s_2 is still needed, but for the Bellman gap term, not for the reward term.

5.2 Value and Q Functions with State Features as Goals

We now turn to a general result covering both value and Q functions (Q functions are obtained as the value function of the state-action Markov process, as explained in Section 2). We also cover target *features* rather than target states: namely, we are given a feature function φ on state space, and the reward is nonzero on states s such that $\varphi(s)$ achieves a particular goal value g. Target states correspond to $\varphi = \mathrm{Id}$.

Covering V functions requires the ability to work on-policy. Thus, we assume that goal-dependent policies are given, yielding goal-dependent transitions $s \to s'|g$ defined by their transition probabilities $P(\mathrm{d}s'|s,g)$.

Thus we wish to find solutions to the goal-dependent Bellman equation

$$V(s,\mathrm{d}g) = \delta_{\varphi(s)}(\mathrm{d}g) + \gamma\,\mathbb{E}_{s'\sim P(\mathrm{d}s'|s,g)} V(s',\mathrm{d}g) \tag{38}$$

with reward on states such that the features $\varphi(s)$ are equal to g. Full target states amount to $\varphi = \mathrm{Id}$: a nonzero reward when $s = g$. This can be used in turn to train the goal-dependent policies, for instance by policy gradient. (The technical meaning of this equation is similar to the case of Q above. For a discussion on existence and uniqueness we refer to Section 5.3.)

Here the training dataset is made of triplets $(s \to s'|g)$: transitions indexed by a goal. For the Q function this is not restrictive: working on state-action pairs, given a state-action (s,a,s') it is always possible to sample a goal g a posteriori, and to define the next action a' according to policy g in state s'. For the V function this is more restrictive: typically, the training set would be made of trajectories where a goal is selected at random and kept for some time. This results in some empirical distribution over state-goal pairs (s,g) in the training set, with s and g not independent.

A major issue is to avoid using the sparse rewards $\delta_{\varphi(s)}(\mathrm{d}g)$. Indeed, the most obvious approach to the Bellman equation (38) is to view this problem as an ordinary Markov process on the augmented state space of state-goal pairs (s,g). The TD update for this problem is

$$\delta\theta = \mathbb{E}_{(s,g)\sim\rho_{SG},\, s'\sim P(\mathrm{d}s'|s,g)} \left[\partial_\theta v_\theta(s,g) \left(\frac{\delta_s(\mathrm{d}g)}{\tau(\mathrm{d}g)} + \gamma v_\theta(s',g) - v_\theta(s,g) \right) \right]$$

where ρ_{SG} is the distribution of state-goal pairs (s,g) in the training set, and where the V function has been parameterized as $V_\theta(s,\mathrm{d}g) = v_\theta(s,g)\tau(\mathrm{d}g)$

for some arbitrary measure $\tau(\mathrm{d}g)$ on goal space. In a continuous state space, no reward would ever be observed.

Sparse rewards can be avoided by just using the goal $g = \varphi(s)$ for the sparse term: $\partial_\theta v_\theta(s,g)\,\delta_s(\mathrm{d}g) \rightsquigarrow \partial_\theta v_\theta(s,\varphi(s))$. The price to pay is computing the value function only *up to a goal-dependent scaling*. Namely, there is a simple sparsity-free TD update for the related problem

$$V(s,\mathrm{d}g) = \alpha(s,g)\,\delta_{\varphi(s)}(\mathrm{d}g) + \gamma\,\mathbb{E}_{s'\sim P(\mathrm{d}s'|s,g)} V(s',\mathrm{d}g). \tag{39}$$

Here the reward is nonzero only if $\varphi(s) = g$, but with an unknown factor $\alpha(s,g)$ that depends on the solution reached.

If $\alpha(s,g)$ depends only on g, then optimal policies are not affected: for every goal g, we just compute the correct value function for this goal up to a g-dependent scaling. This happens in many use cases, notably for Q-learning or if $\varphi = \mathrm{Id}$ (goals are full states), as shown below.

The least favorable use case is V-learning with $\varphi \neq \mathrm{Id}$. Then the scaling α may also vary among the states s which achieve $\varphi(s) = g$: this may result in policies which do solve the problem of finding a state s with $\varphi(s) = g$, but not necessarily in an optimal way. (In that case, another option is to explicitly provide a full state s_g such that $\varphi(s_g) = g$ and use the full state s_g as the goal instead, thus going back to $\varphi = \mathrm{Id}$.)

We now turn to the technical, general statement and discuss some explicit use cases. The theorem is stated for V functions; the case of Q functions follows by applying it to the state-action Markov process.

Theorem 13 (Goal-dependent TD). *Let $\varphi\colon S \to \mathcal{G}$ be a function from the state space to some goal space $\mathcal{G}$ (discrete if S is discrete and continuous if S is continuous).*

Assume that the training set consists of transitions $(s \to s'|g)$ indexed by a goal. Let the joint distribution of state-goal pairs in the training set be $\rho_{SG}(\mathrm{d}s,\mathrm{d}g)$. Let ρ_S and ρ_G be its marginals over s and g, namely, the distributions of states and of goals in the training set. Assume that the density of ρ_{SG} with respect to $\rho_S\rho_G$ is nonzero everywhere (every state-goal pair appears with some positive probability).

Parameterize the value function as $V_\theta(s,\mathrm{d}g) = v_\theta(s,g)\rho_G(\mathrm{d}g)$.

Then the parameter update

$$\delta\theta = \mathbb{E}_{(s,g)\sim\rho_{SG},\, s'\sim P(\mathrm{d}s'|s,g)} \left[\, \partial_\theta v_\theta(s,\varphi(s)) + \partial_\theta v_\theta(s,g)\,\left(\gamma v_\theta(s',g) - v_\theta(s,g)\right)\right] \tag{40}$$

is the TD update associated with the Bellman equation for the goal-dependent value function

$$V(s,\mathrm{d}g) = \alpha(s,g)\,\delta_{\varphi(s)}(\mathrm{d}g) + \gamma\,\mathbb{E}_{s'\sim P(\mathrm{d}s'|s,g)} V(s',\mathrm{d}g). \tag{41}$$

where $\alpha(s,g) := \rho_S(\mathrm{d}s)\rho_G(\mathrm{d}g)/\rho_{SG}(\mathrm{d}s,\mathrm{d}g)$. (Note that the value of $\alpha(s,g)$ is used only on states such that $\varphi(s) = g$.)

In the following cases, α depends only on g:

- *If the distributions of states and goals are independent in the training set, namely, if* $\rho_{SG}(s, g) = \rho_S(s)\rho_G(g)$*, then* $\alpha(s, g) = 1$*.*
- *If* $\varphi = \mathrm{Id}$ *(goals are full states) then the statement also holds with* $\alpha(g, g)$ *instead of* $\alpha(s, g)$ *in* (41)*.*

Concretely, a stochastic update $\delta\theta$ is obtained by sampling from the dataset a transition $(s \to s'|g)$ indexed by a goal g, and then updating by

$$\delta\theta = \partial_\theta v_\theta(s, \varphi(s)) + \partial_\theta v_\theta(s, g) \left(\gamma v_\theta(s', g) - v_\theta(s, g)\right). \tag{42}$$

This is similar to the update of $\tilde{m}_\theta$ in successor states (Theorem 6), except here the policy depends on the goal. This can be used in turn to train a goal-depedent policy (Section 8).

This theorem can work out in three different ways:

- Q-learning works for any goal features φ, using an ordinary off-policy training set of transitions $((s, a) \to s')$. In that case, there is no need for transitions to be indexed by a goal. This follows from the theorem applied to the state-action process, and yields $\alpha = 1$.
- V-learning works best with full goal states ($\varphi = \mathrm{Id}$). This requires a training set of transitions $(s \to s'|g)$ each indexed by a goal state (such as exploring with a given goal for some time). A goal-dependent policy can be trained by any policy gradient method. In that case, α depends only on g, thus, computing the value function for every g up to a g-dependent scaling that does not affect the optimal policy for g.
- V-learning can be applied with any goal features φ, but the resulting algorithm implicitly reweights the rewards among those states which achieve a given goal. Goal-dependent policies training by policy gradient will still reach a state such that $\varphi(s) = g$, but not necessarily in an optimal way, with certain states implicitly preferred.

Let us discuss the first two cases in more detail.

With Q-learning, it is possible to pick any goal a posteriori for any observed transition $((s, a) \to s')$. So goals and states can be picked independently, resulting in $\alpha = 1$. This plays out as follows: Assume the training set is made of transitions $((s, a) \to s')$, that we have a set of goals $g \in \mathcal{G}$, and that we maintain the value function $v_\theta((s, a), g)$ over state-action pairs. Assume we have g-dependent policies π_g, such as the greedy policy obtained from the Q-function $v_\theta((s, a), g)$. Then the expected TD update (40) can be realized by picking at random a transition $((s, a) \to s')$ in the dataset, picking at random a goal $g \sim \rho_G(\mathrm{d}g)$ according to any user-chosen distribution on goals, picking an action $a' \sim \pi_g(s')$, and updating the parameter via

$$\delta\theta = \partial_\theta v_\theta((s, a), \varphi(s, a)) + \partial_\theta v_\theta((s, a), g) \left(\gamma v_\theta((s', a'), g) - v_\theta((s, a), g)\right). \tag{43}$$

With V-learning, it is unreasonable to assume that goals and states are independent in the training set: this would require an exploration policy which randomly changes goals at every step. A more reasonable exploration policy would pick a goal $g \sim \rho_G$ and keep it for some time, using the goal-dependent policy π_g. This results in a set of transitions $(s \to s'|g)$ indexed by their goals, with a non-independent distribution of goals and visited states, thus $\alpha \neq 1$. If $\varphi = \mathrm{Id}$ the theorem states that α only depends on g so that optimal policies are not affected. The expected TD update (40) can be realized by picking at random a transition $(s \to s'|g)$ in the dataset and updating the parameter via

$$\delta\theta = \partial_\theta v_\theta(s, s) + \partial_\theta v_\theta(s, g) \left(\gamma v_\theta(s', g) - v_\theta(s, g)\right). \tag{44}$$

This update of v_θ is identical to the update of $\tilde{m}_\theta$ in successor states (Theorem 6), except here the transitions (or policy) depends on the goal.

There is no variance from sparse rewards in these expressions: the reward term produces the term $\partial_\theta v_\theta(s, \varphi(s))$, namely, a term directly evaluated at the goal $g = \varphi(s)$ associated with the currently visited state s. (But there is still some variance from the Bellman gap part of the expression.) Thus, when learning goal-dependent value or Q functions with sparse rewards, it is possible to avoid the sparse reward problem by directly setting the goal $g = \varphi(s)$ for the reward term in the TD update.

For comparison, algorithms such as hindsight experience replay store a mixture of state-related and state-independent goals in a training dataset of transitions, to be used with any off-policy learning algorithm. As in our setting, they assume knowledge of the reward function (such as $\delta_{\varphi(s)}(\mathrm{d}g)$) and access to a way to build goals from states, such as φ. This provides a strategy for building a relevant state-goal distribution in the training set. Such an approach is independent from our results, which directly reduce the variance in the Q-learning update. Thus in principle both approaches can be used simultaneously.

Multi-step, horizon-k versions of TD (Appendix A.2) do not seem to be available in the goal-dependent setting in a version that avoids the infinitely-sparse Dirac reward problem.

5.3 Existence and Uniqueness of Optimal Successor States

We now turn to finding a solution to the optimal goal-dependent Bellman equation $TQ = Q$.

For discrete, infinite Markov reward processes, the value function that solves the Bellman equation is in general not unique; it is unique under additional constraints such as boundedness. [7]

[7] For instance, consider the simple random walk on the state space $\mathbb{Z}$, which goes right

For the optimal goal-dependent Q-function, we cannot impose boundedness, since the solution sometimes has infinite mass. [8] Instead, we prove that the solution for the horizon-t problem exists and converges to the *smallest* solution of the Bellman equation when $t \to \infty$.

Let Q_t be the goal-dependent Q-function at bounded horizon t, obtained by expanding the expectimax problem at horizon t, namely

$$Q_t(s_1, a_1, \mathrm{d}s_g) := \delta_{s_1}(\mathrm{d}s_g) + \mathbb{E}_{s_2 \sim P(s_2|s_1,a_1)} \sup_{a_2} \left[\gamma \delta_{s_2}(\mathrm{d}s_g) + \cdots \mathbb{E}_{s_t \sim P(s_t|s_{t-1},a_{t-1})} \sup_{a_t} \left[\gamma^t \delta_{s_t}(\mathrm{d}s_g) \right] \cdots \right]. \tag{45}$$

This Q_t can also be described via the optimal Bellman operator T as $Q_t = T^t \mathbf{0}$, with $\mathbf{0}$ the zero measure. (In the following, "Q is a measure" is short for "for every state-action (s, a), $Q(s, a, \cdot)$ is a measure".)

Theorem 14. *Let T be the optimal Bellman operator of Definition 11. Let $\mathbf{0}$ be the measure with mass 0.*

Let $Q_t := T^t \mathbf{0}$ be the goal-dependent Q-function at horizon t. Then when $t \to \infty$, Q_t converges strongly [9] to a measure Q^*. This limit solves the Bellman equation $TQ^* = Q^*$, and is the smallest such solution. In finite state spaces, it is the only solution with finite mass.

The solution is never unique: the measure that gives infinite mass to every set is another. Hence the interest of considering the *smallest* solution, where the values come from rewards actually picked at some time t.

6 Matrix Factorization and the Forward-Backward (FB) Representation

6.1 Advantages of Matrix Factorization for M

In this section we study a specific parametric model for the successor state operator, which has many advantages: a "matrix-factorized" representation. We consider the model (15), namely $M(s_1, \mathrm{d}s_2) \approx \tilde{m}_\theta(s_1, s_2)\rho(\mathrm{d}s_2)$, with the particular choice

$$\tilde{m}_\theta(s_1, s_2) = F_{\theta_F}(s_1)^\top B_{\theta_B}(s_2) \tag{46}$$

or left with probability 1/2. Given $\gamma < 1$, let φ be the solution to $\varphi = \frac{\gamma}{2}(1 + \varphi^2)$ and define $f(s) := \varphi^s$ for $s \in \mathbb{Z}$. Then by construction, $f(s_t) = \gamma \mathbb{E}_{s_{t+1}|s_t} f(s_{t+1})$. Thus, if V is any solution to the Bellman equation, then $V + f$ is another solution. Such solutions "believe there is an infinite reward at infinity".

[8] For the same reason, contractivity arguments will not work in the proofs, as it is hard to find a norm that is finite and nonzero in every situation. The arguments rely on monotonicity of the Bellman operator.

[9] Namely, for every state-action (s, a) and for every measurable set A, $Q_t(s, a, A)$ converges to $Q(s, a, A)$.

where $F\colon S \to \mathbb{R}^r$ and $B\colon S \to \mathbb{R}^r$ are two learnable functions from the state space to some representation space $\mathbb{R}^r$, parameterized by $\theta = (\theta_F, \theta_B)$. This provides an approximation of M by a rank-r operator. Such a factorization is used for instance in [SHGS15] for the goal-dependent Q-function (up to the factor ρ).

Intuitively, F is a "forward" representation of states and B a "backward" representation: if the future of s_1 matches the past of s_2, then $M(s_1, \mathrm{d}s_2)$ is large. The learning algorithms presented above (forward and backward TD for M) can be directly applied to this parameterization, leading to explicit updates for F and B (Section 6.2).

This representation of M has a number of advantages and some shortcomings. (In this section we deal mostly with successor states; for goal-dependent value functions, this representation has fewer advantages.) The advantages are as follows.

- It provides a direct representation of the value function at every state, without learning an additional model of V. Namely,

$$V(s) \approx F(s)^\top B(R), \qquad B(R) := \mathbb{E}_{s\sim\rho}[r_s\, B(s)] \tag{47}$$

  where the "reward representation" $B(R)$ can be directly estimated by an online average of $B(s)$ weighted by the reward r_s at s. This is a direct consequence of (17). For instance, with sparse rewards, each time a reward is observed, the value function is updated everywhere. [10]

  This point applies to successor states, but not to goal-dependent value functions, which cannot handle arbitrary rewards.

- It simplifies the sampling of a pair of states (s, s_2) needed for forward TD. Indeed, the forward TD update (21) factorizes as an expectation over s, times an expectation over s_2 (Section 6.2 below), which can be estimated independently. The same applies to backward TD. This can potentially reduce variance a lot, and even allows for purely "trajectory-wise" online estimates using only the current transition $s \to s'$, without sampling of another independent state s_2. (Once more, this works for successor states but not for goal-dependent value functions, since in that case the transitions $s \to s'$ depend on s_2.)

- It produces two (policy-dependent) representations of states, a forward and a backward one, in a natural way from the dynamics of the MDP and the current policy. This could be useful for other purposes.

[10] The model of M using m instead of $\tilde{m}$ is less convenient for V, leading to $V(s) = R(s) + F(s)^\top B(R)$, thus requiring a model of the expected reward $R(s)$.

- Even in the tabular case, when the state space is discrete and unstructured, this provides a form of prior or generalization between states (based on a low-rank prior for the successor state operator). States that are linked by the MDP dynamics get representations F and B that are close.
- It has some of the properties of the second-order methods of Section 7, without their complexity. This is proved in Appendix F.1.

The shortcomings are as follows:

- It approximates the successor state operator by an operator of rank at most r. This is never an exact representation unless the representation dimension r is at least the number of distinct states.
- The best rank-r approximation of $(\mathrm{Id} - \gamma P)^{-1}$ erases the small singular values of P: thus this representation will tend to erase "high frequencies" in the reward and value function, and provide a spatially smoother approximation focusing on long-range behavior. This is fine as long as the reward is not a "fast-changing" function made up of high frequencies (such as a "checkerboard" reward).

  This can be expected: learning a reward-agnostic object such as M cannot work equally well for all rewards. For these reasons, it may be useful to use a mixed model for the value function with the FB model as a baseline, such as

$$V_\varphi(s) = F(s)^\top B(R) + v_\varphi(s) \tag{48}$$

  where F and B are learned via successor states, $B(R)$ is as in (47), and φ is learned via ordinary TD on the remainder. The FB part will catch reward-independent, long-range behavior, while the v_φ part will be needed to catch high frequencies in a particular reward function.

Why is a matrix-factorized form relevant for M? Small-rank approximations of a matrix are relevant when the matrix has a few large eigenvalues and many small eigenvalues (or singular values, depending on the precise criterion). Since the successor state operator is the inverse of $\mathrm{Id} - \gamma P$, this means the approximation is reasonable if $\mathrm{Id} - \gamma P$ has few small eigenvalues and many large eigenvalues.

The spectrum of Markov operators is a well-studied topic. For continuous-time operators associated with random diffusions, possibly with added drift, the spectrum generally follows *Weyl's law* [Wik]: in dimension d, the continuous-time analogue of $\mathrm{Id} - P$ has roughly $k^{d/2}$ eigenvalues of size $\leqslant k$, thus, few small and many large eigenvalues.

The simplest example is a random walk on a discrete torus $[1; n]$. The operator P is diagonal in Fourier representation, with eigenvectors $e^{2i\pi kx/n}$ with k an integer. The corresponding eigenvalue of P is $\cos(2\pi k/n)$, yielding an eigenvalue $(1-\gamma) + 2\gamma \sin^2(\pi k/n)$ for $\mathrm{Id} - \gamma P$. The largest eigenvalue of P is 1 (for $k = 0$) corresponding to the smallest eigenvalue $1-\gamma$ for $\mathrm{Id} - \gamma P$. For γ close to 1, $(\mathrm{Id} - \gamma P)^{-1}$ has a very large eigenvalue $1/(1-\gamma)$, then an eigenvalue of order $n^2/2\pi^2$, and the next eigenvalues behave like $n^2/2k^2\pi^2$, thus decreasing like $1/k^2$. In this case, a small-rank approximation is reasonable. A similar computation holds for periodic grids $[1; n]^d$ in higher dimension.

How general is this example? The best studied case is for continuous-time diffusions in continuous spaces such as a subset in $\mathbb{R}^d$. In continuous time, the analogue of the operator $\mathrm{Id} - \gamma P$ is the *infinitesimal generator* operator of the continuous-time Markov process. For the standard Brownian motion, this operator is the Laplacian $\Delta = \sum_{i=1}^d \partial^2/\partial x_i^2$. Its inverse Δ^{-1} plays the role of the successor state operator and provides the value function in continuous time. The spectrum of the Laplacian is well-known and follows Weyl's law: there are about $k^{d/2}$ eigenvalues of size $\leqslant k$ [Wik]. In particular, Δ has few small eigenvalues and many large eigenvalues, so that the successor state operator (given by Δ^{-1}, which provides the value function in continuous time) has few large eigenvalues and many small eigenvalues as needed.

This applies not only to Brownian motion, but to basically any diffusion with drift and variable coefficients on a subset of $\mathbb{R}^d$: indeed, in this case the infinitesimal generator is an *elliptic operator* and also follows Weyl's law [Gå53]. The same law also holds for diffusions on Riemannian manifolds, as the Riemannian Laplace operator also follows Weyl's law [Ber03, Chapter 9.7.2]. These continuous estimates are still valid when discretizing the state space [XZZ17]. So this situation is quite general.

6.2 The TD Updates for the FB Representation of M

We now describe the explicit parametric TD updates for the FB representation of successor states. These follow directly from the general expressions for forward TD and backward TD.

However, the particular factorized structure gives rise to more variants: pure forward (forward TD on F and B), forward-backward (forward TD update for F but backward TD update for B), etc. These will lead to slightly different fixed points and different dynamics for feature learning, as we will explore later.

Proposition 15 (Successor state TD updates in the FB representation). *Consider the parameterization $\tilde{m}_\theta(s_1, s_2) = F_{\theta_F}(s_1)^\top B_{\theta_B}(s_2)$ of the successor state operator M where F and B are two functions from S to $\mathbb{R}^r$, parameterized by $\theta = (\theta_F, \theta_B)$. Abbreviate F for F_{θ_F} and B for B_{θ_B}.*

Then the forward TD update (21) for F is equal to

$$\mathbb{E}_{s\sim\rho}\left(\partial_{\theta_F} F(s)^\top\right)B(s) + \mathbb{E}_{s\sim\rho, s'\sim P(s,\mathrm{d}s')}\left(\partial_{\theta_F} F(s)^\top\right)\Sigma_B(\gamma F(s') - F(s)) \quad (49)$$

where Σ_B is the matrix

$$\Sigma_B := \mathbb{E}_{s_2\sim\rho}\, B(s_2)B(s_2)^\top. \quad (50)$$

The forward TD update for B is equal to

$$\mathbb{E}_{s\sim\rho}\left(\partial_{\theta_B} B(s)^\top\right)F(s) + \mathbb{E}_{s_2\sim\rho}\left(\partial_{\theta_B} B(s_2)^\top\right) D_F B(s_2). \quad (51)$$

where D_F is the matrix

$$D_F := \mathbb{E}_{s\sim\rho,\, s'\sim P(s,\mathrm{d}s')}\, F(s)(\gamma F(s') - F(s))^\top. \quad (52)$$

The backward TD update for F is equal to

$$\mathbb{E}_{s\sim\rho}\left(\partial_{\theta_F} F(s)^\top\right)B(s) + \mathbb{E}_{s_1\sim\rho}\left(\partial_{\theta_F} F(s_1)^\top\right) D_B F(s_1), \quad (53)$$

where D_B is the matrix

$$D_B := \mathbb{E}_{s\sim\rho,\, s'\sim P(s,\mathrm{d}s')}(\gamma B(s') - B(s))B(s)^\top. \quad (54)$$

The backward TD update for B is equal to

$$\mathbb{E}_{s\sim\rho}\left(\partial_{\theta_B} B(s)^\top\right)F(s) + \mathbb{E}_{s\sim\rho,\, s'\sim P(s,\mathrm{d}s')}\left(\gamma\, \partial_{\theta_B} B(s')^\top - \partial_{\theta_B} B(s)^\top\right)\Sigma_F B(s) \quad (55)$$

where Σ_F is the matrix

$$\Sigma_F := \mathbb{E}_{s_1\sim\rho}\, F(s_1)F(s_1)^\top. \quad (56)$$

Proposition 34 (Appendix E) describes how these updates play out for finite spaces in the "tabular on FB" setting, in which a value of F and B is maintained for each state.

Forward or backward TD may be used separately for F and B, giving rise to four algorithms: forward on F and forward on B (ff-FB), forward on F and backward on B (fb-FB), backward on F and forward on B (bf-FB), and backward on F and backward on B (bb-FB). These algorithms behave quite differently on how they learn features and on the fixed points obtained, as discussed below.

Consequences for sampling and variance. A key feature of the FB updates is their decomposition as a product of an expectation over a transition $s \to s'$, times an expectation over another independent state s_2 or s_1.

This has important consequences algorithmically for variance reduction via minibatching. Indeed, a natural way to sample these updates would be

to sample a minibatch of transitions $s \to s'$, another minibatch of states s_2, and evaluate (49) on the minibatch $s \to s'$ with the value of Σ_B obtained on the minibatch s_2. This would not be possible for other parameterizations of M: in general, (21) would require to compute a separate quantity for each $(s \to s', s_2)$, thus requiring smaller minibatches in practice.

Furthermore, these updates lend themselves to a purely trajectory-wise online estimation, without even sampling another independent state s_2 or s_1: indeed, (49) can be estimated at the current transition $s \to s'$, while the matrices Σ_F etc., may be estimated online by an exponential moving average over past or recent states.

Fixed points, feature learning. With F and B of dimension r, each of the r components of F and B defines a function $F_i(s)$ or $B_i(s)$ on the state space. We call these functions *features*. The features of F provide a basis for approximating the value function V. In addition, the model for V ignores any part of the reward function that is uncorrelated to the features of B.

More precisely, the kernel of B and the image of $F^\top$ directly encode which features of states are ignored. Namely, if $R \in \operatorname{Ker} B\rho$ then the corresponding value function is estimated to 0: $MR = F^\top B\rho R = 0$. Thus $\operatorname{Ker} B\rho$ encodes the subspace of reward functions that is unseen by the model. $\operatorname{Ker} B\rho$ is exactly the space of functions which are $L^2(\rho)$-orthogonal to all the features in B. Likewise, for any reward function R, the approximate value function is $F^\top B\rho R = 0$ which lies inside $\operatorname{Im} F^\top$: thus $\operatorname{Im} F^\top$ is the space of features used to express the value functions.

The four algorithms ff-FB, fb-FB, bf-FB, and bb-FB greatly differ on how new features are learned:

- ff-FB learns new features by applying the operator P to existing features in F. These new features are put into both F and B. The fixed points of ff-FB correspond to eigenvectors of the matrix P and M (Proposition 36).

- bb-FB learns new features by applying the operator $P^\top$ to existing features in B, and putting them into both F and B. [11] The fixed points of bb-FB correspond to eigen-probability densities of P and M (Remark 38, Proposition 37).

- fb-FB learns new features both by applying P to features in F, and $P^\top$ to features in B. The fixed points of fb-FB are the *rank-r truncated SVD decompositions* of the matrix M.

[11] The action of $P^\top$ on a positive vector v corresponds to the law of a state at time $t+1$ if the state at time t is distributed according to v. Thus, $P^\top$ naturally acts on probability distributions over states.

- bf-FB may not learn any features beyond the initialization of F and B. For any subspace of features, there is a fixed point of bf-FB which lies in that subspace.

We refer to Appendix E for precise statements of these properties. In addition, fb-FB and bf-FB preserve the symmetry with respect to time reversal of the process, while ff-FB and bb-FB do not.

Relationship with successor representation learning and with linear TD with learned features. To help interpreting these relations, we will relate them to objects from the literature. We make two claims. First, for fixed and orthonormal B, the forward update of F corresponds to standard successor representation learning with state representation (features) B. Second, for fixed F, the forward update of B corresponds to learning the value function for every target state via linear TD with fixed features F.

To simplify things, in this paragraph we consider the "tabular-FB" setting, in which F and B are parameterized just by listing the value of $F(s)$ and $B(s)$ on every state s, assuming a finite state space. [12] For instance, the forward TD update (49) for F, with learning rate $\eta > 0$, becomes $F(s) \leftarrow F(s) + \eta\, \delta F(s)$ where

$$\delta F(s) = B(s) + \Sigma_B(\gamma F(s') - F(s)) \tag{57}$$

upon sampling a transition $s \to s'$.

If B is a fixed, $L^2(\rho)$-orthonormal collection of feature functions (namely, if $\Sigma_B = \mathrm{Id}$), then this forward TD equation to learn F is identical to standard deep successor representation learning using B as the representation. Indeed, standard deep successor representation learning [KSGG16] starts with given features $\varphi(s)$ on the state space, and learns the successor features m as the expected discounted future value of φ along a trajectory (s_t): $m(s) = \sum_{t \geqslant 0} \gamma^t \mathbb{E}[\varphi(s_t) | s_0 = s]$. Such an m is the fixed point of the Bellman equation $m = \varphi + \gamma P m$. Via identifying $m = F$ and $\varphi = B$, ordinary TD for this Bellman equation is equivalent to (57) when $\Sigma_B = \mathrm{Id}$. However, this is not the case if $\Sigma_B \neq \mathrm{Id}$. This is because scalings are different: With the successor state operator, if B is doubled, then F is halved so that $M = F^\top B$ is fixed. With successor representations, if the state representation φ is doubled, then m is doubled as well.

Next, if F is fixed, we claim that the forward TD update for B corresponds to linear TD to learn all value functions corresponding to individual reward $\mathbb{1}_{s_2}$ at all target states s_2, with F as the feature basis. Indeed, if the reward function is $R = \mathbb{1}_{s_2}$, and the corresponding value function is represented as $V = F^\top w$ for some learned vector w (this is linear TD with feature

[12] This is different from a tabular setting for M, which would parameterize M by listing the values $M(s_1, s_2)$ for every pair of states.

basis F), then the TD update of w when observing a transition $s \to s'$ is $\delta w = F(s)\left(\mathbb{1}_{s=s_2} + \gamma F(s')^\top w - F(s)^\top w\right)$. Of course, the resulting w depends on s_2. Thus, if we learn a vector $w(s_2)$ this way for every s_2, we get an update $\delta w(s_2) = F(s)\left(\mathbb{1}_{s=s_2} + \gamma F(s')^\top w(s_2) - F(s)^\top w(s_2)\right)$. By identifying $w(s_2)$ and $B(s_2)$, then on expectation over s and s_2 sampled from ρ, this is equal to (51) for tabular B. Thus, for fixed F, the forward TD update (51) just puts into every $B(s_2)$ a representation of the value function for reward $\mathbb{1}_{s_2}$ as a linear combination of the features $F(s)$.

Thus, when learning F and B jointly, the "FB-tabular" forward TD update on F and B can be seen as a simultaneous learning of all value functions for all reward $\mathbb{1}_{s_2}$, by linear TD in a *learned* feature basis F.

7 Second-Order Methods for Successor States: Implicit Process Estimation and Bellman–Newton

We now turn to more complex, "second-order" algorithms for estimating successor states and value functions. First, we study the best online estimate of M and V in the tabular case, obtained by directly estimating the transition matrix and reward function, and exactly solving the Bellman equation in this estimated process. We provide a convergence theorem for this method (Theorem 16).

This provides an explicit online evolution equation for M and V from observed transitions, in which the transition matrix does not appear (*successor states via implicit process estimation*, Theorem 17). Interestingly, this "true" update of V is TD preconditioned by M (Theorem 18). This is related to viewing $M_{s_1 s}$ as an expectation of the eligibility trace at state s (Appendix D).

The resulting "true" update of M, taken in expectation, defines a *Bellman–Newton operator* (Definition 19), so called because it corresponds exactly to the Newton method for inverting the matrix M. Intuitively, this operator proceeds by concatenating known paths of the MDP, thus doubling the length of known paths, while TD and backward TD just add one transition to the set of known paths (see intuition in Section 4.3). This intuition is formalized in several ways (Proposition 20, Appendix C). This also translates as much better asymptotic convergence in the continuous-time limit (Section 9.3).

All these properties are exact analogues of the convergence properties of second-order Newton-like methods compared to simple first-order gradient descent. Thus, online estimation of M and the Bellman–Newton operator can be seen as "second-order" TD algorithms. Accordingly, they are also numerically trickier. Strengths and weaknesses are discussed in Section 7.4.

Finally, we derive the parametric version of the Bellman–Newton operator, extending it beyond full-matrix tabular updates to sampling in arbitrary

state spaces. However, this update has a large variance unless some kind of forward-backward (FB) representation is used.

7.1 Estimating a Markov Process Online

We now introduce estimates of M and V by online estimation of the Markov process, first in the tabular case, then via function approximation. The process estimation is *implicit*: it does not appear in the resulting algorithms for M. (In particular, we never store an estimated transition matrix $\hat{P}$, which would not make sense for continuous spaces; this excludes solving the problem by planning via the model $\hat{P}$.)

In a (small) finite state space, an obvious approach to learn M is to first learn an estimate $(\hat{P}, \hat{R})$ of the transition matrix P and reward vector R of the Markov reward process, by direct empirical averages; then set M and V to their true values in the estimated Markov process, namely, $\hat{M} = \sum_{n \geqslant 0} \gamma^n \hat{P}^n = (\mathrm{Id} - \gamma \hat{P})^{-1}$ and $\hat{V} = \hat{M}\hat{R}$.

The empirical averages $\hat{P}$ and $\hat{R}$ are updated for each new transition $s \to s'$ with reward r_s, by updating the row s of the transition matrix $\hat{P}$, and the value $\hat{R}_s$ at s:

$$\hat{P}_{ss_2} \leftarrow (1 - 1/n_s)\hat{P}_{ss_2} + (1/n_s)\mathbb{1}_{s_2 = s'} \quad \forall s_2, \qquad \hat{R}_s \leftarrow (1 - 1/n_s)\hat{R}_s + (1/n_s) r_s \tag{58}$$

with n_s the number of visits to state s up to time t. The initialization of $\hat{P}$ and $\hat{R}$ is forgotten after the first observation at each state ($n_s = 1$), but to fix ideas we initialize to $\hat{P} = \hat{R} = 0$. The corresponding estimates $\hat{M} = (\mathrm{Id} - \gamma \hat{P})^{-1}$ and $\hat{V} = \hat{M}\hat{R}$ converge to their true values, as shown by the following non-asymptotic bound.

Theorem 16 (Convergence bounds for process estimation). *Consider a finite Markov reward process with S states and E edges ((s, s') is an edge if $P_{ss'} > 0$), rewards almost surely bounded by $R_{\max}$, and stationary distribution ρ. Update $\hat{P}$ and $\hat{R}$ online via (58), initialized to $\hat{P} = \hat{R} = 0$.*

Then after t i.i.d. observations ($s \sim \rho$, $s' \sim P_{ss'}$), with probability $1 - \delta$, the estimates $\hat{M} = (\mathrm{Id} - \gamma \hat{P})^{-1}$ and $\hat{V} = \hat{M}\hat{R}$ satisfy

$$\|\hat{M} - M\|_{\rho, \mathrm{TV}} \leqslant \frac{2\gamma}{(1-\gamma)^2} \sqrt{\frac{2E}{t} \log \frac{2}{\delta}} \tag{59}$$

and

$$\sum_s \rho(s) \left| \hat{V}(s) - V(s) \right| \leqslant \frac{3 R_{\max}}{(1-\gamma)^2} \sqrt{\frac{2E}{t} \log \frac{4S}{\delta}}. \tag{60}$$

These bounds do not depend on the sampling measure ρ, although the norm used to define the error does. Thus, rarely visited points have no impact on these bounds.

Direct matrix inversion is inconvenient. But since (58) is a rank-one update of the matrix $\hat{P}$, one can compute the update of $\hat{M}$ resulting from (58); this update does not explicitly involve $\hat{P}$ anymore. This will form the basis for the parametric version.

We call the resulting algorithm *successor states via implicit process estimation* (SSIPE).

Theorem 17 (SSIPE: Tabular online update of M). *When a transition $s \to s'$ is added to an empirical estimate of a Markov reward process via (58), the successor state matrix $\hat{M}$ of the estimated process becomes $\hat{M} \leftarrow \hat{M} + \delta M$ with*

$$\delta M_{s_1 s_2} = \frac{1}{n_s} \hat{M}_{s_1 s} \frac{\mathbb{1}_{s_2=s} + \gamma \hat{M}_{s' s_2} - \hat{M}_{s s_2}}{1 - \frac{1}{n_s}(\gamma \hat{M}_{s's} - \hat{M}_{ss} + 1)} \qquad \forall s_1, s_2 \tag{61}$$

with n_s the number of times state s has been sampled. The estimated value function $\hat{V}$ becomes $\hat{V} \leftarrow \hat{V} + \delta V$ with

$$\delta V_{s_1} = \tfrac{1}{n_s}(r_s + \gamma \hat{V}_{s'} - \hat{V}_s)\, \hat{M}_{s_1 s} + o(1/n_s) \qquad \forall s_1 \tag{62}$$

where r_s is the observed reward.

This describes the "true" change of M when the Markov process changes by increasing $P_{ss'}$. This update contains a two-sided term $M_{s_1 s} M_{s' s_2}$: in terms of paths, this term combines all known paths from s_1 to s, the transition $s \to s'$, then all known paths from s' to s_2 (Fig. 1 and Appendix C).

The update of V has the form $\delta V = M \cdot$ (Bellman gap at s). The matrix M can be seen as a "credit assignment" to transfer the Bellman gap $R + \gamma PV - V$ observed at a state s to "predecessor" states.

The update (61) of M is also its TD update multiplied on the left by M (compare (61) and (19)). This is most clear when taking expectations over the next transition $s \to s'$, as follows.

Theorem 18 (The true change of M and V is TD preconditioned by M). *Estimate the successor matrix and value function of a finite MRP by $\hat{M} = (\mathrm{Id} - \gamma \hat{P})^{-1}$ and $\hat{V} = \hat{M}\hat{R}$ where $\hat{P}$ and $\hat{R}$ are estimated directly by the empirical averages (58).*

Consider the updates of these estimates after observing a new transition $s \to s'$. Then, in expectation over the transition $s \to s'$ sampled at time t, the update (61) of $\hat{M}$ is equal to

$$\mathbb{E}_{s \sim \rho,\, s' \sim P_{ss'}}[\delta M] = \tfrac{1}{t} \hat{M}(\mathrm{Id} + \gamma P \hat{M} - \hat{M}) + o(1/t) \tag{63}$$

when the number of observations t tends to infinity. The resulting update of the value function is $\hat{V} \leftarrow \hat{V} + \delta V$ with

$$\mathbb{E}_{s \sim \rho,\, s' \sim P_{ss'}}[\delta V] = \tfrac{1}{t} \hat{M}(R + \gamma P \hat{V} - \hat{V}) + o(1/t). \tag{64}$$

Thus, preconditioning the TD update by M itself produces an update that tracks the "true" value of M and V given all observations available so far. The learning rate $1/t$ is inherited from the direct estimate of P and R via empirical averages in (58).

This update of V is consistent with the view of M as an expected eligibility trace (Appendix D). Indeed, eligibility traces also update the value function at states s_1 that are connected to s via a trajectory. Actually, in expectation, these updates are the same: with $\lambda = 1$, the eligibility trace vector at a state s is an unbiased estimator of the column $M_{s_1 s}$ (Theorem 31 in Appendix D). From this viewpoint, learning M via a parametric model, or using TD(1), are both ways of estimating the "predecessor states" of a state s. Eligibility traces are unbiased but can have large variance, while the model of M has no variance but may have bias if not learned well.

Such a preconditioning is analogous to second-order methods in optimization using the inverse Hessian, which directly jump to the the location of the new optimum when one more data point becomes available. However, in second-order methods, the preconditioning matrix is symmetric definite positive, while this is not the case here; this can produce numerical problems.

In small-scale experiments, using the full matrix online update of $\hat{M}$ resulted in much faster convergence of the value function than TD, consistently with the theoretical prediction of Section 9.3. But with this method, each update requires $O(|S|^2)$ computation time. This is useful only if sample efficiency is the main concern.

7.2 The Bellman–Newton Operator

Thus, when estimating a Markov process online, in expectation, each new observation replaces the estimate $\hat{M}$ with $\hat{M} + \mathbb{E}[\delta M] = \hat{M}(1 + \frac{1}{t}) - \frac{1}{t}\hat{M}(\mathrm{Id} - \gamma P)\hat{M} + o(1/t)$ by (63). Interestingly, this expected update does not depend on the distribution ρ of sampled states s. This is because the $1/n_s$ factors behave asymptotically like $1/(t\rho_s)$, thus compensating the sampling probabilities ρ_s. The fluctuations between n_s and $t\rho_s$ are absorbed in the $o(1/t)$ terms. We gather this behavior in the following operator.

Definition 19 (Bellman–Newton operator). *We call* Bellman–Newton operator *with learning rate $\eta > 0$ the operator $M \mapsto M(1 + \eta) - \eta M(\mathrm{Id} - \gamma P)M$.*

The reason for the name is the following: With learning rate $\eta = 1$, this operator is $M \mapsto 2M - M(\mathrm{Id} - \gamma P)M$. Inverting a matrix A by iterating $M \leftarrow 2M - MAM$ is the *Newton method* for matrix inversion, going as far back as 1933 [PS91]. The Newton method has superexponential convergence, squaring the error (doubling precision) at each step. This property translates as follows in our context.

In terms of paths, the quadratic term in M realizes the path concatenation operation in Fig. 1. This is formalized as follows, and proved and further discussed in Appendix C. In contrast, forward and backward TD only increase the length of known paths by 1.

Proposition 20 (Bellman–Newton doubles the length of known paths). *Assumes that M represents exactly the successor states up to k steps, namely, $M = \sum_{i=0}^{k} \gamma^i P^i$ (as matrices or as operators). Then after one step of the Bellman–Newton operator with learning rate $\eta = 1$, M represents exactly the successor states up to $2k+1$ steps, namely, $2M - M(\mathrm{Id} - \gamma P)M = \sum_{i=0}^{2k+1} \gamma^i P^i$.*

Unfortunately, this method does not always converge. In particular, it is initialization-dependent. For instance, the initialization $M = 0$ is a fixed point. In general, the Bellman–Newton operator preserves the kernel and image of M, so there are many fixed points. Still, $M = (\mathrm{Id} - \gamma P)^{-1}$ is the only full-rank fixed point.

Convergence of the Newton method for matrix inversion is quite well understood [PS91] and works if the spectral radius of $\mathrm{Id} - AM$ is less than 1 at initialization. Otherwise, the method can diverge. For instance, $A = \mathrm{Id} - \gamma P$ for successor states, so initializing to $M = \mathrm{Id}$ converges.

Learning rates $\eta \ll 1$ improve convergence properties. In Section 9.3 we study convergence with infinitesimal learning rates, proving much faster asymptotic convergence than with simple TD on M. This is analogous to the faster convergence of second-order methods with respect to simple gradient descent. Even with $\eta \ll 1$, some initializations still diverge; however, if the initialization is of the form $M = (\mathrm{Id} - \gamma P_0)^{-1}$ for some stochastic or substochastic matrix P_0 (e.g., $P_0 = 0$, initializing M to Id) then the infinitesimal learning rate version converges.

Sampled Bellman–Newton update. Like the Bellman operator for TD on M, the Bellman–Newton operator lends itself to sampling the states at which the values are updated.

This works out as follows. Assume that S is discrete so that M is a matrix (we deal with the parametric case in the next section). Let as usual ρ be the probability distribution from which states are sampled, and let $\grave{\rho}$ be the matrix with diagonal entries ρ. Set $\tilde{m} := M\grave{\rho}^{-1}$ (this corresponds to the parameterization $\tilde{m}$ in (15)). Then the Bellman–Newton update is equivalent to $\tilde{m} \mapsto \tilde{m}(1+\varepsilon) - \eta\tilde{m}(\grave{\rho} - \gamma\grave{\rho}P)\tilde{m}$. In expectation, this update can be realized by sampling a state $s \sim \rho$ and a transition $s' \sim P(\mathrm{d}s'|s)$. Indeed, in that case we have $\mathbb{E}\mathbb{1}_s\mathbb{1}^\top_{s'} = \grave{\rho}P$ and $\mathbb{E}\mathbb{1}_s\mathbb{1}^\top_s = \grave{\rho}$, and therefore the update

$$\tilde{m}_{s_1 s_2} \leftarrow (1+\eta)\tilde{m}_{s_1 s_2} - \eta\, \tilde{m}_{s_1 s}\, \tilde{m}_{s s_2} + \eta\, \gamma\, \tilde{m}_{s_1 s}\, \tilde{m}_{s' s_2} \qquad \forall s_1, s_2 \qquad (65)$$

is equal to the Bellman–Newton update in expectation over (s, s'). [13]

This is still a full-matrix update: the value $\tilde{m}_{s_1 s_2}$ is updated for every s_1 and s_2, even if (s, s') is sampled. This is not scalable. It is possible to sample the states s_1 and s_2 from ρ as well: with this option, the expectation of the update is multiplied by $\dot{\rho}$ on the left and right.

7.3 Parametric Bellman–Newton Update

Perhaps surprisingly, the full-matrix tabular update of M lends itself well to a parametric version, by following the standard TD strategy of updating the parameter to bring M closer to its new value.

Theorem 21 (Bellman–Newton update with function approximation). *Maintain a parametric model of M via m_{θ_t} or $\tilde{m}_{\theta_t}$ as in Section 3.2, with θ_t the parameter at step t.*

Let $s \to s'$ be the transition in the Markov process observed at step t, with reward r_s. Define a target update of M by $M^{\mathrm{tar}} := M_{\theta_t} + \delta M$ with δM given by the online tabular estimate (61). Define the loss between M and M^{tar} via $J(\theta) := \frac{1}{2} \left\| M_\theta - M^{\mathrm{tar}} \right\|_\rho^2$ using the norm (1).

Then the gradient step on θ to reduce this loss is

$$\begin{aligned} -\partial_\theta J(\theta)_{|\theta=\theta_t} &= \tfrac{1}{t}\, \mathbb{E}_{s_1\sim\rho,\, s_2\sim\rho} \big[\gamma\, \partial_\theta m_{\theta_t}(s, s') + \gamma\, m_{\theta_t}(s_1, s)\, \partial_\theta m_{\theta_t}(s_1, s') \\ &+ (\gamma m_{\theta_t}(s', s_2) - m_{\theta_t}(s, s_2))\, (\partial_\theta m_{\theta_t}(s, s_2) + m_{\theta_t}(s_1, s)\, \partial_\theta m_{\theta_t}(s_1, s_2))\big] + o(1/t) \end{aligned} \tag{66}$$

for the model (16) using m_θ, and

$$\begin{aligned} -\partial_\theta J(\theta)_{|\theta=\theta_t} &= \tfrac{1}{t}\, \mathbb{E}_{s_1\sim\rho,\, s_2\sim\rho} \big[\tilde{m}_{\theta_t}(s_1, s)\, \partial_\theta \tilde{m}_{\theta_t}(s_1, s) \\ &+ \tilde{m}_{\theta_t}(s_1, s)(\gamma \tilde{m}_{\theta_t}(s', s_2) - \tilde{m}_{\theta_t}(s, s_2))\, \partial_\theta \tilde{m}_{\theta_t}(s_1, s_2)\big] + o(1/t) \end{aligned} \tag{67}$$

for the model (15) using $\tilde{m}_\theta$.

The update of V via M is discussed in Section 8.

Here the learning rate $1/t$ is inherited from the direct estimate of P via empirical averages, but can be replaced with any learning rate. As with TD, the update was derived from a tabular update, but makes sense in continuous state spaces. In particular, the parametric gradient does not involve the state counts n_s from (61): a cancellation occurs because $n_s \sim t\rho_s$ when $t \to \infty$.

Implementing this update requires sampling *two* additional states s_1 and s_2 from the dataset, in addition to the transition $s \to s'$. See the discussion after Theorem 6 for possible ways to sample these additional states. TD for

[13] This is not quite equivalent to the online update (61): using $n_s \approx t\rho_s$, the latter yields $\tilde{m}_{s_1 s_2} \leftarrow \tilde{m}_{s_1 s_2} + \eta\, \tilde{m}_{s_1 s}(\mathbb{1}_{s=s_2}/\rho_s - \tilde{m}_{s s_2} + \gamma\, \tilde{m}_{s' s_2}) + o(\eta)$ with $\eta = 1/t$. This difference disappears after taking expectations over s_1 and s_2 in addition to (s, s').

M required only one: this reflects the full matrix update (61), while TD only updates the s row of M when observing a new transition $s \to s'$ (Eq. 19).

For parametric Bellman–Newton, the model m_θ can be initialized to 0 while the model $\tilde{m}_\theta$ cannot. Indeed, setting m_θ to 0 corresponds to setting M to Id, a valid initialization for the Bellman–Newton operator, while setting $\tilde{m}_\theta$ to 0 corresponds to setting M to 0, an unwanted and unstable fixed point of the Bellman–Newton operator.

7.4 Discussion: strengths and weaknesses of second-order approaches

In a tabular setting, the full-matrix online update (61) of M (where a transition $s \to s'$ is sampled, but with the value $M_{s_1 s_2}$ updated for every state s_1 and s_2) converges much faster than TD to compute the value function, empirically. This is in line with the asymptotic convergence properties of the Bellman–Newton versus ordinary Bellman operator (Section 9.3).

However, this results in an $O(|S|^2)$ cost per time step, so it is only interesting if sample efficiency is the main issue. The alternative is to sample a few states s_1 and s_2 and only update $M_{s_1 s_2}$ for those states. But in practice, we have found that this introduces many instabilities and requires reducing the learning rate so much (typically η smaller than $1/|S|^2$) that the benefit of second-order Newton convergence is lost. The same phenomenon is observed for the parametric version of Theorem 21.

This sampling issue can be avoided if using a factorized representation $M = F^\top B$ as in Section 6. Namely, there exists an update of F and B that is compatible with sampling and that reproduces the Bellman–Newton update (Section F.2). This decouples the sampling of states s_1 and s_2, thus reducing variance and allowing for larger learning rates. However, this also exacerbates another issue of the Bellman–Newton update, namely, the existence of non-full-rank fixed points and the preservation of the kernel and image of M. The representation $M = F^\top B$ is usually not full-rank, and the Bellman–Newton update of Section F.2 preserves the kernels of F and B. As a consequence (at least for uniform ρ), this algorithm computes the inverse of $\mathrm{Id} - \gamma P$ in the subspace spanned by the initializations of F and B, but no features are learned. Currently, we have found no fully satisfactory second-order update beyond the full-matrix update (61).

8 Learning Value Functions and Policies via Successor States

There are many possible ways to use a model M of the successor state operator in policy and and value function learning. Choices include:

- Using policy gradient versus using Q-learning (greedy or Boltzmann policies, DDPG...).

  If the reward is a known goal state, we may directly use the optimal goal-dependent Q function of Section 5.1.

  For Q-learning with other types of rewards, the successor state operator can be defined on the Markov process over state-action pairs (as explained in Section 2). The Q function can be computed from this "successor state-action operator" in the same ways as the V function from the successor state operator. Thus, all methods described below to learn V can be extended to Q, and we do not discuss this option further here.

- Using the goal-dependent value function as in Section 5 (this leads to a goal-dependent policy for every goal state, simultaneously for all single-state reward functions), versus using the successor state operator of a single policy as in Section 4 (this works for dense rewards but with a single policy).

- Using the successor state operator directly in the policy gradient formula without a value function model, versus learning a model of V from successor states, then using this model normally in policy gradient.

- If learning a model of V from successor states, there are several options to do so. First, the FB representation of M directly yields a V function. Second, the V function may be learned from M in a supervised way based on $V = MR$. Third, M may be used only as one component of the value function ($V = MR + v_\varphi$ with v_φ learned via TD), or as an initialization. This is presumably better if M is approximate. Fourth, V may be learned via TD "preconditioned by M", based on the formula (64) for the true change of V when new transitions are observed (Theorem 18).

We now describe these options in greater detail. They have different bias-variance trade-offs, and the best option may differ based form case to case.

We recall the general form of the policy gradient estimator for a parametric policy π

$$\delta\pi := \mathbb{E}_{s\sim\rho}\mathbb{E}_{a\sim\pi(a|s)}\left[(\partial \ln \pi(a|s))\, \mathbb{E}_{s'\sim P(\mathrm{d}s'|s,a)}\left[r_{s,a} + \gamma V(s') - b(s)\right]\right] \quad (68)$$

where $\partial \ln \pi(a|s)$ is the derivative with respect to the policy parameters of the log-probability to select action a, where $r_{s,a}$ is the immediate reward received after action a, and where b is an arbitrary baseline function which reduces variance of the estimator (typically $b(s) = V(s)$ so that $r_{s,a} + \gamma V(s') - b(s)$ is centered, but we will see other choices below).

Learning goal-dependent policies. The simplest case is for learning policies to reach arbitrary target states, using the goal-dependent value function $v_\theta(s, g)$ of Section 5.2. Here g represents a variable goal, such as a target state, or a desired value for some function of states (Section 5.2).

This works with a goal-dependent policy $\pi(a|s, g)$ depending on goal g, and leads to the policy gradient update

$$\delta\pi = \mathbb{E}_{(s,g)\sim\rho_{SG},\, a\sim\pi(a|s,g),s'|P(\mathrm{d}s'|s,a)}(\partial \ln \pi(a|s, g))(\gamma v_\theta(s', g) - b(s, g)) \quad (69)$$

where v_θ is the goal-dependent value function model from Section 5.2, where b is an arbitrary baseline function (such as $b(s, g) = v_\theta(s, g)$), and where ρ_{SG} is the empirical distribution of state-goal pairs in the trajectories in the dataset (typically obtained by choosing a goal and following the associated policy for some time).

A few comments on this formula: First, with goal states, the reward $r_{s,a}$ in (68) is a Dirac mass, but it depends only on the previous state, not on a or s'; so by choosing the baseline b to include this Dirac, this term disappears in (69).

Second, in the formalism of Section 5.2, the value function is formally a measure over goals, $V_\theta(s, \mathrm{d}g) = v_\theta(s, \mathrm{d}g)\rho_G(\mathrm{d}g)$. Thus, the policy gradient update (68) is goal-dependent and is itself a measure over goals g. This measure can be integrated over all goals g; for each g we may choose the distribution $s \sim \rho_{SG}(\mathrm{d}s|g)$ of states given this goal. This is how we obtain the policy update (69) from (68). In the computation, the measures cancel out between $\rho_{SG}(\mathrm{d}s|g)$ and the $\rho_G(\mathrm{d}g)$ appearing in V_θ: this results in just v_θ in (69), and in the sampling of a pair (s, g) from $\rho_{SG(s,g)}$.

Learning V from M. Another option is to learn the value function V using M, then just use the value function via ordinary policy gradient. We now consider the case of a single (non-goal-dependent) policy to be learned, with an arbitrary reward function. There are several options again.

- The FB representation of Section 6 directly provides a representation of the value function as

$$V(s) \approx F(s)^\top B(R), \qquad B(R) := \mathbb{E}_{s\sim\rho}[r_s\, B(s)] \quad (70)$$

  where $B(R)$ is a "representation of the reward", which can be sampled by weighting the representation $B(s)$ of states by their reward. Thus $B(R)$ can be estimated online. Then the value of V can be plugged directly in the policy gradient formula (68).

  Since the FB representation will focus on low frequencies (long-range) features, it might be useful to used a "mixed" model for V, with $F(s)^\top B(R)$ as one component, and another component learned via ordinary TD; see (75) below.

- Another case is if the reward is located at a single known target state g. Then $V(s) = \tilde{m}(s, g)$ and the policy gradient (68) is equal to

$$\delta\pi = \mathbb{E}_{s\sim\rho}\mathbb{E}_{a\sim\pi(a|s)}\left[(\partial \ln \pi(a|s))\, \mathbb{E}_{s'\sim P(\mathrm{d}s'|s,a)}\left[\gamma\tilde{m}(s', g) - b(s)\right]\right] \tag{71}$$

(once more, the reward term $r_{s,a}$ does not depend on a in that case and can be absorbed in the baseline b). This assumes the model $\tilde{m}$ is used for M; the model m does not seem to lead to a usable formula in this case.

This is useful for sparse rewards: contrary to TD methods, M and V may be learned without ever seeing the reward, provided the target state is known. (By "known", we mean we know the features or input representation of the target state, as provided to the neural networks that learn M and V.) This also extends to linear combinations of a finite number of rewards at known states.

- For general (dense) rewards and without the FB representation, the simplest option is to learn a model of V based on $V = MR$. This becomes a supervised learning problem. No matrix product is necessary: we can perform a stochastic gradient descent of $\|V - MR\|^2_{L^2(\rho)}$ with respect to the parameters of V, just by sampling states, either with discrete or continuous states.

  With V parameterized as V_φ, and with M parameterized by the model $\tilde{m}_\theta$, we have

$$-\partial_\varphi \|V_\varphi - MR\|^2_{L^2(\rho)} = 2\mathbb{E}_{s\sim\rho,\, s_1\sim\rho}\left[\partial_\varphi V_\varphi(s_1)(r_s\, \tilde{m}(s_1, s) - V_\varphi(s_1))\right] \tag{72}$$

  where r_s is the reward obtained when visiting state s. As for other algorithms presented here, this requires sampling one or several additional states s_1 in addition to the state s currently visited.

  With M parameterized by the model m_θ instead, we have

$$\begin{aligned} &-\partial_\varphi \|V_\varphi - MR\|^2_{L^2(\rho)} = \\ &\quad 2\mathbb{E}_{s\sim\rho,\, s_1\sim\rho}\left[\, r_s\, \partial_\varphi V_\varphi(s) + (r_s\, m(s_1, s) - V_\varphi(s_1))\, \partial_\varphi V_\varphi(s_1)\right]. \end{aligned} \tag{73}$$

- Learning V via $V = MR$ assumes that the model of M is reasonably accurate: any error on M shows up on V. Another option is to just use MR as a component in the model of V, or as an initialization to V. For instance, V may be parameterized as

$$V := V_{\varphi_1} + V_{\varphi_2} \tag{74}$$

  where V_{φ_1} is trained to match MR using (72), and V_{φ_2} is learned via ordinary TD.

In the FB representation this would yield

$$V(s) = F(s)^\top B(R) + V_{\varphi_2}(s) \tag{75}$$

where $B(R)$ is estimated online as above, and φ_2 is estimated by ordinary TD.

This makes particular sense for the FB representation: in Appendix E we prove that the fb-FB algorithm minimizes a loss producing a truncated SVD of M, thus focussing on large eigenvalues of M (large eigenvalues of P, long-range dependencies in the environment). Thus $F(s)^\top B(R)$ will focus on large eigenvalues of P. The training of F and B is reward-independent ("unsupervised" reinforcement learning). Thus, ordinary TD on V_{φ_2} may be useful to catch short-range (high-frequency) behavior in the reward.

- Another option is to directly use samples from MR instead of V in the policy gradient update. This emphasizes M as a "credit assignment" for past actions.

  Abbreviate $sas' \sim \rho\pi P$ for the sampling of a state $s \sim \rho$, action $a \sim \pi(a|s)$, and next state $s' \sim P(\mathrm{d}s'|s,a)$. Starting with the policy gradient (68) with baseline $b = V$, substituting $V(s') = \mathbb{E}_{s_1\sim\rho}\tilde{m}(s', s_1)r_{s_1}$, and renaming variables so that all rewards are taken at the same point, we find

$$\delta\pi = \mathbb{E}_{\substack{sas'\sim\rho\pi P \\ s_1a_1s_1'\sim\rho\pi P}} \left[r_{s,a}\left(\partial \ln \pi(a|s) + (\gamma m_{s'1s} - m_{s_1s})\partial \ln \pi(a_1|s_1)\right)\right] \tag{76}$$

  where two independent transitions must be sampled from the dataset. In this expression, the model m serves as a credit assignment to increase the likelihood of those actions a_1 at other (past) states that are estimated to lead to a reward $r_{s,a}$ at the current state s. This is compatible with the view of M as a model of eligibility traces (Appendix D).

  However, this is probably a high-bias, high-variance option, requiring a good model of M.

- Finally, M may be used as a preconditioner for TD on V. Indeed, by Theorem 17, the "true" change of the value function upon observing a new transition $s \to s'$ with reward r_s is

$$\delta V_{s_1} = \tfrac{1}{n_s}(r_s + \gamma\hat{V}_{s'} - \hat{V}_s)\,\hat{M}_{s_1s} + o(1/n_s) \qquad \forall s_1 \tag{77}$$

  namely, the Bellman gap $r_s + \gamma\hat{V}_{s'} - \hat{V}_s$ is sent back to every "predecessor state" s_1 with coefficient $\hat{M}_{s_1s}$. (See Appendix D for M as an expected eligibility trace.)

  The resulting parametric update is obtained as follows.

Proposition 22 (TD preconditioned by M for the value function). *Let V_φ be a smooth parametric model of the value function. Define an update of V by setting $V^{\mathrm{tar}} := V_{\varphi_t} + \delta V$ with φ_t the parameter at step t, and δV given by (77), and taking the gradient of the loss $J^V(\varphi) := \frac{1}{2}\|V_\varphi - V^{\mathrm{tar}}\|^2_{L^2(\rho)}$. Assume $\hat{M}$ is equal to the model (16) using m_θ. Then this gradient is*

$$-\partial_\varphi J^V(\varphi)_{|\varphi=\varphi_t} = \tfrac{1}{t}\left(r_s + \gamma V_{\varphi_t}(s') - V_{\varphi_t}(s)\right)\left(\partial_\varphi V_{\varphi_t}(s) + \mathbb{E}_{s_1\sim\rho}[m_\theta(s_1, s)\,\partial_\varphi V_{\varphi_t}(s_1)]\right) + o(1/t) \quad (78)$$

where t is the total number of observations. For the model (15) using $\tilde{m}_\theta$, this gradient is

$$-\partial_\varphi J^V(\varphi)_{|\varphi=\varphi_t} = \tfrac{1}{t}\left(r_s + \gamma V_{\varphi_t}(s') - V_{\varphi_t}(s)\right)\mathbb{E}_{s_1\sim\rho}\,\tilde{m}_\theta(s_1, s)\,\partial_\varphi V_{\varphi_t}(s_1) + o(1/t) \quad (79)$$

The learning rate $1/t$ just results from the direct empirical averages used to estimate the process in Section 7.1, and may be replaced with any learning rate.

This involves sampling an additional state $s_1 \sim \rho$ and applying a TD update at that point, with weight depending on M. In the model of M using m_θ, this appears as a correction to ordinary TD; in the model of M using $\tilde{m}_\theta$, everything is included in $\tilde{m}$.

Notably, even if the model of M is wrong, , the true value function is still a fixed point of (78) and (79) in expectation over s' and r_s; it is the only fixed point provided $\hat{M}$ is invertible and $\rho > 0$. This is a theoretical advantage over all other estimates of V described above. However, the sampling of s_1 adds variance, and any negative eigenvalues in the estimate of M will produce divergence.

9 Small Learning Rates and the Continuous-Time Analysis

This section is a more informal discussion about intuitions coming from a continuous-time analysis when the learning rate is small. We will not present formal statements. For simplicity we restrict ourselves to the tabular, finite state case so that all objects are always well-defined without smoothness conditions, but in principle the analysis extends to any state space.

We also assume that states are sampled uniformly (ρ is uniform) so that the expected updates correspond to the Bellman operators. Introducing non-uniform ρ does not fundamentally change the results about the forward and backward Bellman operators (indeed, the eigenvalues of the matrix

$\grave{\rho}(\mathrm{Id} - \gamma P)$ have positive real part, just like those of $\mathrm{Id} - P$, for any positive ρ).

For the Bellman–Newton operator, full non-asymptotic convergence rates were provided in Theorem 16. Here, we provide a more intuitive asymptotic analysis that clarifies how the error decreases faster than with TD.

9.1 Continuous-Time Analysis of the Forward and Backward Bellman Operators

The forward Bellman operator on M with learning rate $\eta > 0$ is

$$M \leftarrow (1 - \eta)M + \eta(\mathrm{Id} + \gamma PM). \tag{80}$$

When η is small, after n iterations, the value of M approximates the value at time $t = n\eta$ of the solution of the matrix ordinary differential equation

$$\frac{\mathrm{d}M_t}{\mathrm{d}t} = \mathrm{Id} + \gamma PM_t - M_t = \mathrm{Id} - \Delta M_t \tag{81}$$

where $\Delta = \mathrm{Id} - \gamma P$ is the Laplacian associated with the Markov process. The solution to this equation is

$$M_t = \Delta^{-1} + e^{-t\Delta}(M_0 - M) \tag{82}$$

where Δ^{-1} is the true successor state matrix, M_0 is the initial value, and $e^{-t\Delta}$ is the exponential of the matrix $t\Delta$.

Likewise, the backward Bellman operator on M with learning rate $\eta > 0$ is

$$M \leftarrow (1 - \eta)M + \eta(\mathrm{Id} + \gamma MP). \tag{83}$$

When η is small, after n iterations, the value of M approximates the value at time $t = n\eta$ of the solution of the ordinary differential equation

$$\frac{\mathrm{d}M_t}{\mathrm{d}t} = \mathrm{Id} + \gamma M_t P - M_t = \mathrm{Id} - M_t \Delta \tag{84}$$

process. The solution to this equation is

$$M_t = \Delta^{-1} + (M_0 - M)e^{-t\Delta} \tag{85}$$

where M_0 is the initial value. Letting E_t be the error at time t:

$$E_t := M_t - \Delta^{-1} \tag{86}$$

then the errors at time t are $E_t = e^{-t\Delta}E_0$ and $E_t = E_0 e^{-t\Delta}$ for the forward and backward operators, respectively.

Thus the forward and backward equations converge at the same rate. Indeed, assume for simplicity that Δ is diagonalizable, with eigenvalues λ_i.

[14] By the spectral properties of stochastic matrices, the eigenvalues of Δ have positive real part: $\Re\lambda_i \geqslant 1-\gamma$. (The largest eigenvalue of Δ is $1-\gamma$, with multiplicity 1 if P is irreducible.) This implies that the errors tend to 0.

For a more precise analysis, let u_i and v_i be respectively the right and left eigenvectors of Δ, associated with eigenvalues λ_i.

Since the u_i's and the v_i's form bases, one can decompose the initial error E_0 as $E_0 = \sum_{i,j} \alpha_{ij} u_i v_j^\top$. Then one checks that the error at time t for the continuous-time forward Bellman operator is

$$E_t = \sum_{i,j} e^{-t\lambda_i} \alpha_{ij} u_i v_j^\top \tag{87}$$

for the forward operator, and

$$E_t = \sum_{i,j} e^{-t\lambda_j} \alpha_{ij} u_i v_j^\top \tag{88}$$

for the backward operator.

The eigenvalues are the same for the forward and backward operator. Each eigenvalue has multiplicity n (the number of states) over the state of matrices M, corresponding to all choices of j for a given i or conversely. Notably, the smallest eigenvalue of Δ is $1-\gamma$, corresponding to the direct of slowest convergence. This eigenvalue has multiplicity n when acting on M.

9.2 Mixing Forward and Backward TD Improves Convergence

Interestingly, if one mixes the forward and backward operators, then this eigenvalue analysis changes. The smallest eigenvalue is still the same, but its multiplicity decreases considerably, from n to 1. Indeed, assume that we perform alternatively one step of the forward and backward Bellman operators, each with learning rate η. When η is small, the dynamics tends to that of the continuous-time ordinary differential equation

$$\frac{\mathrm{d}M_t}{\mathrm{d}t} = \frac{1}{2}\left(\mathrm{Id} + \gamma P M_t - M_t\right) + \frac{1}{2}\left(\mathrm{Id} + \gamma M_t P - M_t\right) = \mathrm{Id} - \frac{1}{2}(\Delta M_t + M_t \Delta) \tag{89}$$

whose solution is

$$M_t = \Delta^{-1} + e^{-t\Delta/2}(M_0 - M)e^{-t\Delta/2} \tag{90}$$

where Δ^{-1} is the true successor state matrix. Thus, the error $E_t := M_t - \Delta^{-1}$ satisfies $E_t = e^{-t\Delta/2} E_0 e^{-t\Delta/2}$.

[14] This occurs for a dense subset of stochastic matrices P. If not, the analysis is more technical, with polynomials in front of the exponentials of the eigenvalues, but the conclusions are similar.

But now, with the same eigenvector decomposition as above, we find

$$E_t = \sum_{i,j} e^{-t(\lambda_i+\lambda_j)/2} u_i v_j^\top. \tag{91}$$

In particular, the error in the direction $u_i v_j^\top$ decreases fast if *at least* one of λ_i or λ_j has large real part. Notably, the slowest convergence now occurs ony if *both* i and j correspond to the smallest eigenvalue $1-\gamma$: this smallest eigenvalue now has multiplicity 1.

Thus, mixing the forward and backward Bellman operator does produce a positive effect on convergence speed, bringing the multiplicity of the worst eivengalue from n (the number of states) to 1, and generally picking the best eigenvalue in each direction of the error.

9.3 Continuous-Time Analysis of the Bellman–Newton Operator

Remember the Bellman–Newton operator $M \mapsto (1+\eta)M - \eta M(\mathrm{Id} - \gamma P)M$ (Definition 19) with learning rate η. When η is small, after n iterations of this operator, the value of M approximates the value at time $t = n\eta$ of the solution of the matrix ordinary differential equation

$$\frac{\mathrm{d}M_t}{\mathrm{d}t} = M_t - M_t \Delta M_t \tag{92}$$

where $\Delta = \mathrm{Id} - \gamma P$ as above. Obviously $M = \Delta^{-1}$ is a fixed point. However, as with the Bellman–Newton operator, there are other fixed points, such as $M = 0$: since the differential equation preserves the kernel and image of M_t, there is a (unique) fixed point for every choice of kernel and image, amounting to computing the inverse of Δ in the associated subspaces. Still, Δ^{-1} is the only full-rank fixed point.

The accelerated asymptotic convergence of the Bellman–Newton operator compared to TD on M becomes clear on this continuous-time version. Define the error

$$E_t := \mathrm{Id} - M_t \Delta \tag{93}$$

(beware this differs from the definition of E_t in the sections above). It evolves according to

$$\frac{\mathrm{d}E_t}{\mathrm{d}t} = -E_t + E_t^2. \tag{94}$$

This is generally convergent except for some initializations (more on this below).

When the error is small, the dynamics is $E_t' = -E_t + O(E_t^2) \approx -E_t$. The same holds for the error $M_t - \Delta^{-1} = -E_t \Delta^{-1}$. So, in the small error regime, the error E_t decreases at a constant exponential rate, *independently of the Markov process*. This contrasts with the forward Bellman equation, whose

convergence depends on the eigenvalues of $\mathrm{Id} - \gamma P$, and which will converge slowly if P has eigenvalues close to 1.

In this sense, the continuous-time Bellman–Newton dynamics is to the Bellman operator what continuous-time second-order gradient descent is to continuous-time gradient descent: it removes dependencies on the eigenvalues for convergence close to the solution.

Global initialization and convergence outside of the small-error regime is best understood by introducing a fictitious value of P associated with M_t. Since M_t converges to $(\mathrm{Id} - \gamma P)^{-1}$, let us introduce P_t such that $M_t = (\mathrm{Id} - \gamma P_t)^{-1}$, namely, $\gamma P_t := \mathrm{Id} - M_t^{-1}$, assuming M_t is invertible. On P_t, the evolution equation of M_t becomes

$$\frac{\mathrm{d}P_t}{\mathrm{d}t} = -P_t + P \tag{95}$$

which is affine, with solution $P_t = P + e^{-t}(P_0 - P)$. Thus, the solution for M_t is

$$M_t = (\mathrm{Id} - \gamma P + \gamma e^{-t}(P_0 - P))^{-1}. \tag{96}$$

Namely, on the variable P, the solution just follows a straight line from P_0 to P at a fixed exponential decay rate. P_t always converges; however, M_t may be undefined if $\mathrm{Id} - \gamma P_t$ is not invertible for some t. This depends on the initialization P_0 (therefore, on M_0).

For instance, if P_0 is equal to any (sub)stochastic matrix, then P_t is (sub)stochastic as well, and $\mathrm{Id} - \gamma P_t$ is always invertible, so that M_t converges. This happens for instance: if $P_0 = 0$, namely, $M_0 = \mathrm{Id}$; or if M_0 is initialized to the successor matrix of *any* Markov process.

More possible initializations appear if considering the dynamics of E_t. Assume E_t is diagonalizable (this is the case for random initializations). Then from (94), the eigenvectors of E_t stay the same over time, and each associated eigenvalue λ evolves according to $\lambda' = -\lambda + \lambda^2$. As long as $\lambda \neq 0$, this is equivalent to $(\lambda^{-1})' = \lambda^{-1} - 1$. So each eigenvalue λ^{-1} reaches $-\infty$, so that each eigenvalue λ reaches 0. The exception is when $\lambda^{-1} = 0$ at some point, in which case λ diverges. Since $(\lambda^{-1})' = \lambda^{-1} - 1$, this happens if and only if λ^{-1} is initially equal to some positive real value in the complex plane. So there is a half-line of eigenvalues of E_0 in the complex plane which will lead to divergence. [15]

References

[AWR+17] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin,

[15] This does not show that a pure random initialization converges with probability 1: indeed, a random *real* matrix will typically have some real eigenvalues, which will lie on the wrong half-line with some positive probability.

OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In *Advances in neural information processing systems*, pages 5048–5058, 2017.

[BB19] David Brandfonbrener and Joan Bruna. Geometric insights into the convergence of nonlinear td learning. *arXiv preprint arXiv:1905.12185*, 2019.

[BBQ+18] Diana Borsa, André Barreto, John Quan, Daniel Mankowitz, Rémi Munos, Hado van Hasselt, David Silver, and Tom Schaul. Universal successor features approximators. *arXiv preprint arXiv:1812.07626*, 2018.

[BDM+17] Andre Barreto, Will Dabney, Remi Munos, Jonathan J Hunt, Tom Schaul, Hado P van Hasselt, and David Silver. Successor features for transfer in reinforcement learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4055–4065. Curran Associates, Inc., 2017.

[Ber03] Marcel Berger. *A panoramic view of Riemannian geometry*. Springer, 2003.

[Ber12] Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control*, volume 2. Athena Scientific, 4th edition, 2012.

[BHB+20] André Barreto, Shaobo Hou, Diana Borsa, David Silver, and Doina Precup. Fast reinforcement learning with generalized policy updates. *Proceedings of the National Academy of Sciences*, 117(48):30079–30087, 2020.

[Bré99] Pierre Brémaud. *Markov chains: Gibbs fields, Monte Carlo simulation, and queues*, volume 31. 1999.

[Day93] Peter Dayan. Improving generalization for temporal difference learning: The successor representation. *Neural Computation*, 5(4):613–624, 1993.

[DSC96] Persi Diaconis and Laurent Saloff-Coste. Logarithmic Sobolev inequalities for finite Markov chains. *The Annals of Applied Probability*, 6(3):695–750, 1996.

[FLHI+18] Vincent François-Lavet, Peter Henderson, Riashat Islam, Marc G Bellemare, and Joelle Pineau. An introduction to deep reinforcement learning. *arXiv preprint arXiv:1811.12560*, 2018.

[GO19] Sam Greydanus and Chris Olah. The paths perspective on value learning. *Distill*, 2019. https://distill.pub/2019/paths-perspective-on-value-learning.

[GS97] Charles Miller Grinstead and James Laurie Snell. *Introduction to probability.* American Mathematical Soc., 1997.

[Gå53] Lars Gårding. On the asymptotic distribution of the eigenvalues and eigenfunctions of elliptic differential operators. *Math. Scand.*, (1), 1953.

[Hai06] Martin Hairer. Ergodic properties of Markov processes. Lecture notes, 2006.

[Hai10] Martin Hairer. Convergence of Markov processes. Lecture notes, 2010.

[JKSY20] Chi Jin, Akshay Krishnamurthy, Max Simchowitz, and Tiancheng Yu. Reward-free exploration for reinforcement learning. *ArXiv*, abs/2002.02794, 2020.

[KS60] J. G. Kemeny and J. L. Snell. *Finite Markov Chains.* Van Nostrand, New York, 1960.

[KSGG16] Tejas D Kulkarni, Ardavan Saeedi, Simanta Gautam, and Samuel J Gershman. Deep successor reinforcement learning. *arXiv preprint arXiv:1606.02396*, 2016.

[LTL17] Lucas Lehnert, Stefanie Tellex, and Michael L Littman. Advantages and limitations of using successor features for transfer in reinforcement learning. *arXiv preprint arXiv:1708.00102*, 2017.

[MBB19] Marlos C. Machado, Marc G. Bellemare, and Michael Bowling. Count-based exploration with the successor representation, 2019.

[MRG+18] Marlos C. Machado, Clemens Rosenbaum, Xiaoxiao Guo, Miao Liu, Gerald Tesauro, and Murray Campbell. Eigenoption discovery through the deep successor representation. In *International Conference on Learning Representations*, 2018.

[MWB18] Chen Ma, Junfeng Wen, and Yoshua Bengio. Universal successor representations for transfer reinforcement learning. *arXiv preprint arXiv:1804.03758*, 2018.

[Oll18] Yann Ollivier. Approximate temporal difference learning is a gradient descent for reversible policies, 2018.

[Par05] Kalyanapuram R Parthasarathy. *Probability measures on metric spaces*, volume 352. American Mathematical Soc., 2005.

[PG17] Lerrel Pinto and Abhinav Gupta. Learning to push by grasping: Using multiple tasks for effective learning. *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2161–2168, 2017.

[PS91] Victor Pan and Robert Schreiber. An improved newton iteration for the generalized inverse of a matrix, with applications. *SIAM Journal on Scientific and Statistical Computing*, 12(5):1109–1130, 1991.

[Put14] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[PW19] Ashwin Pananjady and Martin J. Wainwright. Value function estimation in Markov reward processes: Instance-dependent ℓ_∞-bounds for policy evaluation. 2019.

[SB18] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018. 2nd edition.

[SBG17] Kimberly L Stachenfeld, Matthew M Botvinick, and Samuel J Gershman. The hippocampus as a predictive map. *Nature neuroscience*, 20(11):1643, 2017.

[SHGS15] Tom Schaul, Daniel Horgan, Karol Gregor, and David Silver. Universal value function approximators. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1312–1320, Lille, France, 07–09 Jul 2015. PMLR.

[SJK+19] Wen Sun, Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, and John Langford. Model-based RL in contextual decision processes: Pac bounds and exponential improvements over model-free approaches. In *COLT*, 2019.

[SMD+11] 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. The 10th International Conference on Au- tonomous Agents and Multiagent Systems-Volume 2, pages 761–768, 2011.

[Tsi94] John N Tsitsiklis. Asynchronous stochastic approximation and q-learning. *Machine learning*, 16(3):185–202, 1994.

[TVR97] John N. Tsitsiklis and Benjamin Van Roy. An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control*, 42(5):674–690, 1997.

[vHMH+20] Hado van Hasselt, Sephora Madjiheurem, Matteo Hessel, David Silver, André Barreto, and Diana Borsa. Expected eligibility traces. *arXiv preprint arXiv:2007.01839*, 2020.

[Wik] Wikipedia. Weyl law.

[WOS+03] Tsachy Weissman, Erik Ordentlich, Gadiel Seroussi, Sergio Verdu, and Marcelo J Weinberger. Inequalities for the L1 deviation of the empirical distribution. 2003.

[XZZ17] Jinchao Xu, Hongxuan Zhang, and Ludmil Zikatanov. On the weyl's law for discretized elliptic operators. *arXiv preprint arXiv:1705.07803*, 2017.

[ZSBB17] J. Zhang, J. T. Springenberg, J. Boedecker, and W. Burgard. Deep reinforcement learning with successor features for navigation across similar environments. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2371–2378, 2017.

A Further Variants and Properties of TD for Successor States

A.1 Using a Target Network

In parametric TD, it is possible to get closer to an exact application of the Bellman operator, by performing several gradient steps to bring the model M_θ closer to the Bellman operator $\mathrm{Id} + \gamma P M_{\theta^{\mathrm{tar}}}$ for a fixed previous value of the parameter θ^{tar}, and only update $\theta^{\mathrm{tar}} \leftarrow \theta$ once in a while. The formulas are as follows.

Theorem 23 (Parametric TD for M with a target network). *Keep the setting of Theorem 6, but set the target M^{tar} to $M^{\mathrm{tar}} := \mathrm{Id} + \gamma P M_{\theta^{\mathrm{tar}}}$ for some value θ^{tar} of the parameter. Then the gradient step to bring M_θ closer to M^{tar} is*

$$-\partial_\theta J(\theta) = \mathbb{E}_{s\sim\rho,\, s'\sim P(s,\mathrm{d}s'),\, s_2\sim\rho}\big[\gamma\, \partial_\theta m_\theta(s,s') + \partial_\theta m_\theta(s,s_2)\,(\gamma m_{\theta^{\mathrm{tar}}}(s',s_2) - m_\theta(s,s_2))\big] \quad (97)$$

for the model (16) using m_θ, and

$$-\partial_\theta J(\theta) = \mathbb{E}_{s\sim\rho,\, s'\sim P(s,\mathrm{d}s'),\, s_2\sim\rho}\big[\partial_\theta \tilde{m}_\theta(s,s) + \partial_\theta \tilde{m}_\theta(s,s_2)\,(\gamma \tilde{m}_{\theta^{\mathrm{tar}}}(s',s_2) - \tilde{m}_\theta(s,s_2))\big] \quad (98)$$

for the model (15) using $\tilde{m}_\theta$.

A.2 TD on M with Multi-Step Returns

A multistep, horizon-h version of TD on M can be defined by iterating the Bellman equation, which yields $M = \mathrm{Id} + \gamma P + \cdots + \gamma^{h-1} P^{h-1} + \gamma^h P^h M$. This requires being able to observe h consecutive transitions from the process. The corresponding parametric update is as follows.

Theorem 24 (Multi-step TD for successor states with function approximation). *Maintain a parametric model of M as in Section 3.2 via $M_{\theta_t}(s_1, \mathrm{d}s_2) = \delta_{s_1}(\mathrm{d}s_2) + m_{\theta_t}(s_1,s_2)\rho(\mathrm{d}s_2)$, with θ_t the value of the parameter at step t, and with m_θ some smooth family of functions over pairs of states.*

For $h \geqslant 1$, define a target update of M via the horizon-h Bellman equation, $M^{\mathrm{tar}} := \mathrm{Id} + \gamma P + \cdots + \gamma^{h-1} P^{h-1} + \gamma^h P^h M_{\theta_t}$. Define the loss between M and M^{tar} via $J(\theta) := \frac{1}{2}\left\|M_\theta - M^{\mathrm{tar}}\right\|_\rho^2$ using the norm (1).

Then the gradient step on θ to reduce this loss is

$$-\partial_\theta J(\theta)_{|\theta=\theta_t} = \mathbb{E}_{s_0\sim\rho,\, s_1\sim P(s_0,\mathrm{d}s_1),\, \ldots,\, s_h\sim P(s_{h-1},\mathrm{d}s_h),\, s_{\mathrm{tar}}\sim\rho} \Big[\gamma\, \partial_\theta m_{\theta_t}(s_0,s_1) + \gamma^2\, \partial_\theta m_{\theta_t}(s_0,s_2) + \cdots + \gamma^h\, \partial_\theta m_{\theta_t}(s_0,s_h) + \partial_\theta m_{\theta_t}(s_0,s_{\mathrm{tar}})\,(\gamma^h\, m_{\theta_t}(s_h,s_{\mathrm{tar}}) - m_{\theta_t}(s_0,s_{\mathrm{tar}}))\Big]. \quad (99)$$

For the model (15) using $\tilde{m}_\theta$, this update is

$$-\partial_\theta J(\theta)_{|\theta=\theta_t} = \mathbb{E}_{s_0\sim\rho,\, s_1\sim P(s_0,\mathrm{d}s_1),\,\ldots,\, s_h\sim P(s_{h-1},\mathrm{d}s_h),\, s_{\mathrm{tar}}\sim\rho} \Big[\partial_\theta \tilde{m}_{\theta_t}(s_0,s_0) + \gamma\,\partial_\theta \tilde{m}_{\theta_t}(s_0,s_1) + \cdots + \gamma^{h-1}\,\partial_\theta \tilde{m}_{\theta_t}(s_0,s_{h-1}) + \partial_\theta \tilde{m}_{\theta_t}(s_0,s_{\mathrm{tar}})\left(\gamma^h\,\tilde{m}_{\theta_t}(s_h,s_{\mathrm{tar}}) - \tilde{m}_{\theta_t}(s_0,s_{\mathrm{tar}})\right)\Big]. \quad (100)$$

A.3 Tabular TD on MR Is Tabular TD on V

In the tabular case, if the reward is deterministic, learning V via ordinary TD is equivalent to learning V via the matrix product $V = MR$ with M learned via tabular TD, as follows.

Theorem 25. *Consider a Markov reward process with deterministic reward R. Initialize an estimate $\hat{V}$ of V to 0 and an estimate $\hat{M}$ of M to 0. Each time a transition $s \to s'$ with reward $r_s = R_s$ is observed, update $\hat{V}$ via ordinary TD and $\hat{M}$ via TD for successor states, with learning rate η, namely*

$$\hat{V}_s \leftarrow \hat{V}_s + \eta\left(r_s + \gamma\hat{V}_{s'} - \hat{V}_s\right), \quad (101)$$

$$\hat{M}_{ss_2} \leftarrow \hat{M}_{ss_2} + \eta\left(\mathbb{1}_{s=s_2} + \gamma\hat{M}_{s's_2} - \hat{M}_{ss_2}\right) \qquad \forall s_2. \quad (102)$$

Then at every time step, $\hat{V} = \hat{M}R$.

Proof. By induction on the time step. This is true at time 0 thanks to the initialization. If $\hat{V} = \hat{M}R$ at one time step, then the update of $\hat{M}R$ at the next time step is

$$(\hat{M}R)_s = \sum_{s_2} \hat{M}_{ss_2}R_{s_2} \quad (103)$$

$$\leftarrow \sum_{s_2}\left(\hat{M}_{ss_2}R_{s_2} + \eta\left(\mathbb{1}_{s=s_2} + \gamma\hat{M}_{s's_2} - \hat{M}_{ss_2}\right)R_{s_2}\right) \quad (104)$$

$$= (\hat{M}R)_s + \eta\left(R_s + \gamma(\hat{M}R)_{s'} - (\hat{M}R)_s\right) \quad (105)$$

which is the same update as $\hat{V}_s$. The values at the other states are not updated. Therefore, if $\hat{V} = \hat{M}R$ before the update, this still holds after the update. □

A.4 The Parametric Update for Backward TD

We now state the analogue of Theorem 6 for backward TD; this provides the associated parametric update.

Theorem 26 (Backward TD for successor states with function approximation). *Maintain a parametric model of M as in Section 3.2 via $M_{\theta_t}(s_1,\mathrm{d}s_2) = \delta_{s_1}(\mathrm{d}s_2) + m_{\theta_t}(s_1,s_2)\rho(\mathrm{d}s_2)$, with θ_t the value of*

the parameter at step t, and with m_θ some smooth family of functions over pairs of states.

Define a target update of M via the Bellman equation, $M^{\text{tar}} := \mathrm{Id} + \gamma M_{\theta_t} P$. Define the loss between M and M^{tar} via $J(\theta) := \frac{1}{2} \|M_\theta - M^{\text{tar}}\|_\rho^2$ using the norm (1).

Then the gradient step on θ to reduce this loss is

$$\begin{aligned} -\partial_\theta J(\theta)_{|\theta=\theta_t} = \mathbb{E}_{s\sim\rho,\, s'\sim P(s,\mathrm{d}s'),\, s_1\sim\rho} \big[& \gamma\, \partial_\theta m_{\theta_t}(s, s') \\ & + m_{\theta_t}(s_1, s)\, (\gamma\, \partial_\theta m_{\theta_t}(s_1, s') - \partial_\theta m_{\theta_t}(s_1, s)) \big] . \end{aligned} \tag{106}$$

For the model variant in Eq. 15, $M_{\theta_t}(s_1, \mathrm{d}s_2) = \tilde{m}_{\theta_t}(s_1, s_2)\rho(\mathrm{d}s_2)$, the gradient step on θ to reduce the loss $J(\theta)$ is

$$\begin{aligned} -\partial_\theta J(\theta)_{|\theta=\theta_t} = \mathbb{E}_{s\sim\rho,\, s'\sim P(s,\mathrm{d}s'),\, s_1\sim\rho} \big[& \partial_\theta \tilde{m}_{\theta_t}(s, s) \\ & + \tilde{m}_{\theta_t}(s_1, s)\, (\gamma\, \partial_\theta \tilde{m}_{\theta_t}(s_1, s') - \partial_\theta \tilde{m}_{\theta_t}(s_1, s)) \big] . \end{aligned} \tag{107}$$

A.5 Having Targets on Features of the State

Learning M is particularly suitable when the reward is located at a single known goal state g: then, the value function $V(s)$ is proportional to $\tilde{m}(s, g)$. For how to exploit M with dense rewards, we refer to Section 8.

Another scenario is to have a target value for *some* features of the state, not necessarily the whole state itself: namely, the reward is nonzero when some known feature $\varphi(s)$ of state s is equal to some known goal g. In that case, it is convenient to learn a smaller object than M, from which the value function can be read directly. This is also useful if the reward is known to depend only on $\varphi(s)$.

Definition 27. *Let $\varphi\colon S \to \mathbb{R}^k$ be any measurable map. The* successor feature operator *M^φ is defined as follows: for each state s_1, $M^\varphi(s_1, \mathrm{d}g)$ is a measure on $\mathbb{R}^k$ equal to the pushforward of $M(s_1, \mathrm{d}s_2)$ by the map $s_2 \mapsto g = \varphi(s_2)$.*

This operator is different from successor representations: here we keep track of the whole future *distribution* of values of φ, not just the expected future value of φ.

M^φ can be used to compute the value function of any reward that depends only on $\varphi(s)$.

Proposition 28. *Assume that the reward function at state s is equal to $R(\varphi(s))$, namely, it depends only on φ. Let τ be any probability distribution on features in $\mathbb{R}^k$. Assume that M^φ is parameterized as $M^\varphi(s, \mathrm{d}g) = m^\varphi(s, g)\tau(\mathrm{d}g)$. Then the value function of a state s for this reward is*

$$V(s) = \mathbb{E}_{g\sim\tau}[m^\varphi(s, g)R(g)]. \tag{108}$$

In particular, if the reward is nonzero exactly when the feature $\varphi(s)$ is equal to some target value g, then the value function is proportional to $m^\varphi(s,g)$.

This is useful only if an algorithm to learn m^φ is available. Forward TD can be defined on M^φ, based on the following Bellman equation.

Proposition 29. *M^φ satisfies the Bellman equation $M^\varphi(s,\mathrm{d}g) = \delta_{\varphi(s)}(\mathrm{d}g) + \gamma\mathbb{E}_{s'\sim P(s,\mathrm{d}s')}M^\varphi(s',\mathrm{d}g)$.*

Theorem 30. *Let τ be any probability distribution on features in $\mathbb{R}^k$. Assume that M^φ is parameterized as $M^\varphi_\theta(s,\mathrm{d}g) = m^\varphi_\theta(s,g)\tau(\mathrm{d}g)$ for some parametric family of functions $m^\varphi_\theta(s,g)$ with parameter θ.*

Let θ_0 be some value of the parameter, and define a target operator M^{tar} by the Bellman equation: $M^{\mathrm{tar}} := \delta_{\varphi(s)}(\mathrm{d}g) + \gamma\mathbb{E}_{s'\sim P(s,\mathrm{d}s')}M^\varphi_{\theta_0}(s',\mathrm{d}g)$. Define the loss between M^φ and M^{tar} via $J(\theta) := \mathbb{E}_{s\sim\rho,\, g\sim\tau}((M^\varphi_\theta(s,\mathrm{d}g) - M^{\mathrm{tar}}(s,\mathrm{d}g))/\tau(\mathrm{d}g))^2$.

Then the gradient step to bring M^φ closer to M^{tar} in this norm is

$$-\partial_\theta J(\theta) = \mathbb{E}_{s\sim\rho,\, s'\sim P(s,\mathrm{d}s'),\, g\sim\tau}\big[\partial_\theta m^\varphi_\theta(s,\varphi(s)) + \partial_\theta m^\varphi_\theta(s,g)\left(\gamma m^\varphi_{\theta_0}(s',g) - m^\varphi_\theta(s,g)\right)\big]. \quad (109)$$

Once more, the term $\partial_\theta m^\varphi_\theta(s,\varphi(s))$ makes every transition informative: when visiting state s, we increase the probability to reach the goal $\varphi(s)$.

A.6 Taking γ Close to 1: Relative TD

For γ close to 1, it is known that the value function behaves like a large constant plus an informative signal, $V(s) = \frac{c}{1-\gamma} + V^{\mathrm{rel}}(s)$. A similar phenomenon occurs with M. The large constant affects learning in practice, especially for Bellman–Newton which has terms scaling like M^2.

V^{rel} can be learned directly via *relative TD*, adapted from *relative value iteration* [Ber12, §5.3.1], [Put14, §6.6], just by removing the value of V at a reference state from the Bellman equation. Namely, with reference state s_{rel}, the relative TD update upon observing a transition $s \to s'$ with reward r_s is

$$\delta V^{\mathrm{rel}}_s = r_s + \gamma V^{\mathrm{rel}}_{s'} - V^{\mathrm{rel}}_s - \gamma V^{\mathrm{rel}}_{s_{\mathrm{rel}}}. \quad (110)$$

This makes it possible to use a γ very close to 1, or even $\gamma = 1$ if the Markov process is ergodic or "unichain".

Relative TD can be transposed to M directly. The relative Bellman equation above rewrites as $V^{\mathrm{rel}} = R + \gamma(P - \mathbb{1}\mathbb{1}^\top_{s_{\mathrm{rel}}})V^{\mathrm{rel}}$. Therefore, the solution is given by

$$V^{\mathrm{rel}} = (\mathrm{Id} - \gamma P + \gamma\mathbb{1}\mathbb{1}^\top_{s_{\mathrm{rel}}})^{-1}R. \quad (111)$$

Thus we can set $M^{\mathrm{rel}} := (\mathrm{Id} - \gamma P + \gamma\mathbb{1}\mathbb{1}^\top_{s_{\mathrm{rel}}})^{-1}$.

More generally, working with a distribution of reference states rather than a single reference state, we will set

$$M^{\mathrm{rel}} := (\mathrm{Id} - \gamma P + \gamma \mathbb{1}\rho_{\mathrm{rel}}^{\top})^{-1} \tag{112}$$

where ρ_{rel} is the probability vector for reference states. When $\gamma = 1$ and $\rho_{\mathrm{rel}} = \rho$ is the invariant distribution of the Markov process, this is exactly the fundamental matrix of the Markov process [KS60].

The effect of relative TD is just to replace the operator P with $P - \mathbb{1}\rho_{\mathrm{rel}}^{\top}$ everywhere. In practice, in the various formulas, for every term involving the second state s' of a transition $s \to s'$, a corresponding term is added with s_{rel} instead of s' and with the opposite sign. Thus, the update (20) for parametric TD for M becomes

$$\begin{aligned}\mathbb{E}_{s\sim\rho,\, s'\sim P(s,\mathrm{d}s'),\, s_2\sim\rho,\, s_{\mathrm{rel}}\sim\rho_{\mathrm{rel}}} \big[&\gamma\, \partial_\theta m_{\theta_t}(s, s') - \gamma\, \partial_\theta m_{\theta_t}(s, s_{\mathrm{rel}}) \\ &+ \partial_\theta m_{\theta_t}(s, s_2)\, (\gamma m_{\theta_t}(s', s_2) - \gamma m_{\theta_t}(s_{\mathrm{rel}}, s_2) - m_{\theta_t}(s, s_2))\big]\,. \end{aligned} \tag{113}$$

The update for parametric backward TD becomes

$$\begin{aligned}\mathbb{E}_{s\sim\rho,\, s'\sim P(s,\mathrm{d}s'),\, s_1\sim\rho,\, s_{\mathrm{rel}}\sim\rho_{\mathrm{rel}}} \big[&\gamma\, \partial_\theta m_\theta(s, s') - \gamma\, \partial_\theta m_\theta(s, s_{\mathrm{rel}}) \\ &+ m_\theta(s_1, s)\, (\gamma\, \partial_\theta m_\theta(s_1, s') - \gamma\, \partial_\theta m_\theta(s_1, s_{\mathrm{rel}}) - \partial_\theta m_\theta(s_1, s))\big]\,. \end{aligned} \tag{114}$$

The parametric update (78) of V via M becomes

$$\begin{aligned}\mathbb{E}_{s\sim\rho,\, s'\sim P(s,\mathrm{d}s'),\, s_1\sim\rho,\, s_{\mathrm{rel}}\sim\rho_{\mathrm{rel}}} \big[&(r_s + \gamma V_{\varphi_t}(s') - \gamma V_{\varphi_t}(s_{\mathrm{rel}}) - V_{\varphi_t}(s)) \\ &\times (\partial_\varphi V_{\varphi_t}(s) + m_{\theta_t}(s_1, s)\, \partial_\varphi V_{\varphi_t}(s_1))\big]\,. \end{aligned} \tag{115}$$

Finally, the parametric Bellman–Newton update (66) for M becomes

$$\begin{aligned}\mathbb{E}_{s\sim\rho,\, s'\sim P(s,\mathrm{d}s'),\, s_1\sim\rho,\, s_2\sim\rho,\, s_{\mathrm{rel}}\sim\rho_{\mathrm{rel}}} \big[&\gamma\, \partial_\theta m_{\theta_t}(s, s') - \gamma\, \partial_\theta m_{\theta_t}(s, s_{\mathrm{rel}}) \\ &+ \gamma\, m_{\theta_t}(s_1, s)\, \partial_\theta m_{\theta_t}(s_1, s') - \gamma\, m_{\theta_t}(s_1, s)\, \partial_\theta m_{\theta_t}(s_1, s_{\mathrm{rel}}) \\ &+ (\gamma m_{\theta_t}(s', s_2) - \gamma m_{\theta_t}(s_{\mathrm{rel}}, s_2) - m_{\theta_t}(s, s_2))\, (\partial_\theta m_{\theta_t}(s, s_2) + m_{\theta_t}(s_1, s)\, \partial_\theta m_{\theta_t}(s_1, s_2))\big]\,. \end{aligned} \tag{116}$$

B Proofs for Sections 3, 4, 5, 7, 8, and Appendix A

In this text we consider two parametric models of M, (15) and (16), given by $\tilde{m}_\theta$ and m_θ respectively. In most proofs, we only cover the more complex model m_θ; the proofs with $\tilde{m}_\theta$ are similar but simpler.

B.1 Proofs for Sections 3 and 4: TD for M

Proof of Theorem 2. By the definition of M in (9), for any measurable set $A \subset \mathcal{S}$, for any $s \in \mathcal{S}$, $M(s, A)$ is defined as

$$M(s, A) = \sum_{n \geqslant 0} \gamma^n P^n(s, A). \tag{117}$$

Since each $P^n(s,\cdot)$ is a probability distribution, $P^n(s,A) \leqslant 1$ so that this sum of non-negative terms is bounded by $\frac{1}{1-\gamma}$, and therefore the sum converges. $M(s,\cdot)$ is a positive measure as a convergent sum of positive measures (σ-additivity for $M(s,\cdot)$ follows from the dominated convergence theorem). Its total mass is $M(s,\mathcal{S}) = \sum_{n\geqslant 0}\gamma^n P(s,\mathcal{S}) = \sum_{n\geqslant 0}\gamma^n = \frac{1}{1-\gamma}$.

As a positive measure with finite mass, $M(s,\cdot)$ acts on bounded measurable functions, just like P, via $(Mf)(s) = \int f(s')M(s,\mathrm{d}s')$. Since M has mass $\frac{1}{1-\gamma}$ for any s, this integral is bounded by $\frac{1}{1-\gamma}\sup f$, so that $\sup Mf \leqslant \frac{1}{1-\gamma}\sup f$ for any function $f \in B(\mathcal{S})$. Thus, M is well-defined as an operator from $B(\mathcal{S})$ to $B(\mathcal{S})$.

As an operator, one has $\gamma PM = \gamma P\sum_{n\geqslant 0}\gamma^n P^n = \sum_{n\geqslant 1}\gamma^n P^n$. Therefore, $(\mathrm{Id}-\gamma P)M = M - \gamma PM = \sum_{n\geqslant 0}\gamma^n P^n - \sum_{n\geqslant 1}\gamma^n P^n = \gamma^0 P^0 = \mathrm{Id}$ (the sums converge absolutely by the same boundedness argument as before, thus justifying the infinite sum manipulations). This proves that M is a right inverse of $\mathrm{Id}-\gamma P$ as operators. The computation is identical for the left inverse; therefore, M and $\mathrm{Id}-\gamma P$ are inverses as operators on $B(\mathcal{S})$.

Finally, let R be any (bounded, measurable) reward function. Since $(\mathrm{Id}-\gamma P)M = \mathrm{Id}$, one has $(\mathrm{Id}-\gamma P)MR = R$ namely $MR = R + \gamma PMR$. This proves that $V = MR$ satisfies the Bellman equation $V = R+\gamma PV$, and so MR is the value function of the Markov reward process.

Proof of Theorems 3 and 9. An operator M' satisfies the left Bellman equation $M' = \mathrm{Id}+\gamma PM'$ if and only if $M' - \gamma PM' = \mathrm{Id}$, or $(\mathrm{Id}-\gamma P)M' = \mathrm{Id}$, namely, M' is a right inverse of $\mathrm{Id}-\gamma P$. By Theorem 2, $\mathrm{Id}-\gamma P$ is invertible and its inverse is M. Therefore, the only right inverse of $\mathrm{Id}-\gamma P$ is M.

The proof is identical for the backward Bellman equation, with left inverses instead of right inverses.

Proof of Propositions 4 and 10. By definition of the operator P, for any function f we have $\|Pf\|_\infty = \sup_s \int f(s')P(s,\mathrm{d}s') \leqslant \sup_{s'} f(s') = \|f\|_\infty$, so that P is 1-contracting. Therefore, for any bounded operator M and function f, one has $\|PMf\|_\infty \leqslant \|Mf\|_\infty \leqslant \|M\|_{\mathrm{op}}\|f\|_\infty$, so that $\|PM\|_{\mathrm{op}} \leqslant \|M\|_{\mathrm{op}}$ for any M. Therefore, given two operators M and M', one has $\|(\mathrm{Id}+\gamma PM) - (\mathrm{Id}+\gamma PM')\|_{\mathrm{op}} = \gamma\|P(M-M')\|_{\mathrm{op}} \leqslant \gamma\|M-M'\|_{\mathrm{op}}$.

For the backward Bellman operator, $M \mapsto \mathrm{Id}+\gamma MP$, the proof is similar, using that for any bounded operator M and function f, one has $\|MPf\|_\infty \leqslant \|M\|_{\mathrm{op}}\|Pf\|_\infty \leqslant \|M\|_{\mathrm{op}}\|f\|_\infty$, so that $\|MP\|_{\mathrm{op}} \leqslant \|M\|_{\mathrm{op}}$ for any M.

Proof of Theorem 6. In this proof, we freely go back and forth between M or M^{tar} as measure-valued functions, and M or M^{tar} as operators on bounded functions. Notably, the operator Id corresponds to the measure $\delta_{s_1}(\mathrm{d}s_2)$.

We start with the statement for the first model, $M_{\theta_t}(s_1, \mathrm{d}s_2) = \delta_{s_1}(\mathrm{d}s_2) + m_{\theta_t}(s_1, s_2)\rho(\mathrm{d}s_2)$.

By definition of $M^{\mathrm{tar}} = \mathrm{Id} + \gamma P M_{\theta_t}$, and by definition of the action of the operator P, we have

$$M^{\mathrm{tar}}(s, \mathrm{d}s_2) = \delta_s(\mathrm{d}s_2) + \gamma \int_{s'} P(s, \mathrm{d}s') M_{\theta_t}(s', \mathrm{d}s_2) \tag{118}$$

$$= \delta_s(\mathrm{d}s_2) + \gamma \int_{s'} P(s, \mathrm{d}s') \delta_{s'}(\mathrm{d}s_2) + \gamma \int_{s'} P(s, \mathrm{d}s') m_{\theta_t}(s', s_2)\rho(\mathrm{d}s_2) \tag{119}$$

$$= \delta_s(\mathrm{d}s_2) + \gamma P(s, \mathrm{d}s_2) + \gamma\, \mathbb{E}_{s' \sim P(s, \mathrm{d}s')}[m_{\theta_t}(s', s_2)\rho(\mathrm{d}s_2)] \tag{120}$$

by the definition of the Dirac measure $\delta_{s'}$. Therefore,

$$\begin{aligned} M^{\mathrm{tar}}(s, \mathrm{d}s_2) - M_\theta(s, \mathrm{d}s_2) &= M^{\mathrm{tar}}(s, \mathrm{d}s_2) - \delta_s(\mathrm{d}s_2) - m_\theta(s, s_2)\rho(\mathrm{d}s_2) \\ &= \gamma P(s, \mathrm{d}s_2) + \gamma\, \mathbb{E}_{s' \sim P(s, \mathrm{d}s')}[m_{\theta_t}(s', s_2)\rho(\mathrm{d}s_2)] - m_\theta(s, s_2)\rho(\mathrm{d}s_2) \end{aligned} \tag{121}$$

By definition of $J(\theta)$ and of the norm $\|\cdot\|_\rho$, we have

$$J(\theta) = \frac{1}{2} \iint j_\theta(s, s_2)^2 \, \rho(\mathrm{d}s)\rho(\mathrm{d}s_2) \tag{122}$$

where $j_\theta(s, s_2) := (M^{\mathrm{tar}}(s, \mathrm{d}s_2) - M_\theta(s, \mathrm{d}s_2))/\rho(\mathrm{d}s_2)$ (assuming this density exists). [16] Consequently,

$$\partial_\theta J(\theta) = \iint j_\theta(s, s_2)\, \partial_\theta j_\theta(s, s_2)\rho(\mathrm{d}s)\rho(\mathrm{d}s_2) \tag{123}$$

assuming j_θ is smooth enough so that the derivative makes sense and commutes with the integral. From the definition of j_θ and from (121) we have

$$j_\theta(s, s_2) = \gamma \frac{P(s, \mathrm{d}s_2)}{\rho(\mathrm{d}s_2)} + \gamma\, \mathbb{E}_{s' \sim P(s, \mathrm{d}s')}[m_{\theta_t}(s', s_2)] - m_\theta(s, s_2) \tag{124}$$

and

$$\partial_\theta j_\theta(s, s_2) = -\partial_\theta m_\theta(s, s_2) \tag{125}$$

(and consequently, j_θ is smooth if m_θ is smooth). Therefore,

$$-\partial_\theta J(\theta) = \iint \partial_\theta m_\theta(s, s_2) \left(\gamma \frac{P(s, \mathrm{d}s_2)}{\rho(\mathrm{d}s_2)} + \gamma\, \mathbb{E}_{s' \sim P(s, \mathrm{d}s')}[m_{\theta_t}(s', s_2)] - m_\theta(s, s_2) \right) \rho(\mathrm{d}s)\rho(\mathrm{d}s_2) \tag{126}$$

[16] This proof involves $P(s, \mathrm{d}s_2)/\rho(\mathrm{d}s_2)$, but this quantity only appears as $(P(s, \mathrm{d}s_2)/\rho(\mathrm{d}s_2))\rho(\mathrm{d}s_2)$ in the final result (126). Therefore, the argument extends by continuity to the case when $P(s, \cdot)$ is not absolutely continuous with respect to ρ: in that case the norm $J(\theta)$ is infinite but its gradient $\partial_\theta J(\theta)$ is still well-defined by continuity.

The first term $\iint \partial_\theta m_\theta(s, s_2)\gamma \frac{P(s,\mathrm{d}s_2)}{\rho(\mathrm{d}s_2)}\rho(\mathrm{d}s)\rho(\mathrm{d}s_2)$ rewrites as $\gamma \iint \partial_\theta m_\theta(s, s_2)P(s, \mathrm{d}s_2)\rho(\mathrm{d}s)$ namely $\gamma \mathbb{E}_{s\sim\rho}\mathbb{E}_{s_2\sim P(s,\mathrm{d}s_2)}\partial_\theta m_\theta(s, s_2)$. Renaming s_2 to s' in this term ends the proof.

Let us now turn to the model $M_{\theta_t}(s_1, \mathrm{d}s_2) = \tilde{m}_{\theta_t}(s_1, s_2)\rho(\mathrm{d}s_2)$. Here, there is a hidden mathematical subtlety with continuous states. Indeed, in that case, M_{θ_t} is absolutely continuous with respect to ρ, while M^{tar} is not, due to the Id term, as discussed in Section 3.2. (With the other model, the Id terms cancel between M_{θ_t} and M^{tar}.) This makes the norm $J(\theta) = \frac{1}{2}\left\|M_\theta - M^{\mathrm{tar}}\right\|_\rho^2$ infinite (see its definition in (1)). However, the *gradient* of this norm is actually still well-defined. There are at least two ways to handle this rigorously, which lead to the same result: either do the computation in the finite case and observe that the resulting gradient still makes sense in the continuous case (which can be obtained by a limiting argument), or observe that the loss $J(\theta)$ is equal to $\frac{1}{2}\left\|M_\theta\right\|_\rho^2 - \langle M_\theta, M^{\mathrm{tar}}\rangle_\rho + \frac{1}{2}\left\|M^{\mathrm{tar}}\right\|_\rho^2$ and has the same minima and the same gradients as the loss $J'(\theta) = \frac{1}{2}\left\|M_\theta\right\|_\rho^2 - \langle M_\theta, M^{\mathrm{tar}}\rangle_\rho$ for a given M^{tar}. Namely, J and J' differ by a constant in the finite case, and by an "infinite constant" in the continuous case. We will work with the loss J', which is finite even in the continuous case.

Here $\langle M_1, M_2\rangle_\rho = \int_{s,s_2} \frac{M_1(s,\mathrm{d}s_2)}{\rho(\mathrm{d}s_2)}\frac{M_2(s,\mathrm{d}s_2)}{\rho(\mathrm{d}s_2)}\rho(\mathrm{d}s)\rho(\mathrm{d}s_2)$ is the dot product associated with the norm (1). Since the integrand can be rewritten as $\frac{M_1(s,\mathrm{d}s_2)}{\rho(\mathrm{d}s_2)}\rho(\mathrm{d}s)M_2(s, \mathrm{d}s_2)$, it is well-defined as soon as at least one of M_1 or M_2 is absolutely continuous with respect to ρ. Namely,

$$\langle M_1, M_2\rangle_\rho = \int_{s,s_2} \frac{M_1(s, \mathrm{d}s_2)}{\rho(\mathrm{d}s_2)}\rho(\mathrm{d}s)\, M_2(s, \mathrm{d}s_2). \tag{127}$$

Let us compute $J'(\theta) = \frac{1}{2}\left\|M_\theta\right\|_\rho^2 - \langle M_\theta, M^{\mathrm{tar}}\rangle_\rho$. By definition of $M^{\mathrm{tar}} = \mathrm{Id} + \gamma P M_{\theta_t}$, and by definition of the action of the operator P, we have

$$\begin{aligned} M^{\mathrm{tar}}(s, \mathrm{d}s_2) &= \delta_s(\mathrm{d}s_2) + \gamma \int_{s'} P(s, \mathrm{d}s')M_{\theta_t}(s', \mathrm{d}s_2) && (128)\\ &= \delta_s(\mathrm{d}s_2) + \gamma\, \mathbb{E}_{s'\sim P(s,\mathrm{d}s')}[\tilde{m}_{\theta_t}(s', s_2)\rho(\mathrm{d}s_2)] && (129)\end{aligned}$$

by definition of the model $M_{\theta_t}(s_1, \mathrm{d}s_2) = \tilde{m}_{\theta_t}(s_1, s_2)\rho(\mathrm{d}s_2)$. Therefore, by (127),

$$\begin{aligned} &\langle M_\theta, M^{\mathrm{tar}}\rangle_\rho = \int_{s,s_2} \tilde{m}_\theta(s, s_2)\, \rho(\mathrm{d}s)\, M^{\mathrm{tar}}(s, \mathrm{d}s_2)\\ &= \int_s \tilde{m}_\theta(s, s)\, \rho(\mathrm{d}s) + \gamma \int_{s,\, s',\, s_2} \tilde{m}_\theta(s, s_2)\, \tilde{m}_{\theta_t}(s', s_2)\, \rho(\mathrm{d}s)\, P(s, \mathrm{d}s')\, \rho(\mathrm{d}s_2) \end{aligned} \tag{130}$$

thanks to (129). Next, since $M_\theta(s, \mathrm{d}s_2) = \tilde{m}_\theta(s, s_2)\rho(\mathrm{d}s_2)$, the definition of

the norm (1) yields

$$\frac{1}{2}\left\|M_\theta\right\|_\rho^2 = \frac{1}{2}\int_{s,s_2} \tilde{m}_\theta(s,s_2)^2\,\rho(\mathrm{d}s)\,\rho(\mathrm{d}s_2). \tag{131}$$

Collecting, and rewriting the integrals as expectations, we find

$$\begin{aligned} J'(\theta) = \mathbb{E}_{s\sim\rho,\,s_2\sim\rho}\left[\frac{1}{2}\tilde{m}_\theta(s,s_2)^2 - \tilde{m}_\theta(s,s)\right] \\ - \gamma\mathbb{E}_{s\sim\rho,\,s'\sim P(s,\mathrm{d}s'),\,s_2\sim\rho}[\tilde{m}_\theta(s,s_2)\,\tilde{m}_{\theta_t}(s',s_2)] \end{aligned} \tag{132}$$

hence

$$\begin{aligned} \partial_\theta J'(\theta) = \mathbb{E}_{s\sim\rho,\,s_2\sim\rho}\left[\partial\tilde{m}_\theta(s,s_2)\,\tilde{m}_\theta(s,s_2) - \partial\tilde{m}_\theta(s,s)\right] \\ - \gamma\mathbb{E}_{s\sim\rho,\,s'\sim P(s,\mathrm{d}s'),\,s_2\sim\rho}[\partial\tilde{m}_\theta(s,s_2)\,\tilde{m}_{\theta_t}(s',s_2)] \end{aligned} \tag{133}$$

which is the expression given in Theorem 6 for $\theta = \theta_t$. This ends the proof.

B.2 Proofs for Appendix A: Further properties of TD for M

Proof of Theorem 23. The proof is identical to that of Theorem 6, but with θ^{tar} instead of θ^t and no substitution $\theta = \theta_t$ in the last step.

Proof of Theorem 24. Exactly as in Theorem 6, setting $j_\theta(s,s') := (M^{\mathrm{tar}}(s,\mathrm{d}s') - M_\theta(s,\mathrm{d}s'))/\rho(\mathrm{d}s')$, we have

$$\partial_\theta J(\theta) = \iint j_\theta(s,s')\,\partial_\theta j_\theta(s,s')\rho(\mathrm{d}s)\rho(\mathrm{d}s') \tag{134}$$

$$= \iint \partial_\theta j_\theta(s,s')\rho(\mathrm{d}s)(j_\theta(s,s')\rho(\mathrm{d}s')) \tag{135}$$

$$= \iint \partial_\theta j_\theta(s,s')\rho(\mathrm{d}s)(M^{\mathrm{tar}}(s,\mathrm{d}s') - M_\theta(s,\mathrm{d}s')) \tag{136}$$

and since M^{tar} depends on θ_t but not on θ,

$$\partial_\theta j_\theta(s,s') = -\partial_\theta\left(\frac{M_\theta(s,\mathrm{d}s')}{\rho(\mathrm{d}s')}\right) = -\partial_\theta m_\theta(s,s'). \tag{137}$$

From the definition of M^{tar} we have

$$M^{\mathrm{tar}}(s,\mathrm{d}s') = \delta_s(\mathrm{d}s') + \sum_{i=1}^{h-1}\gamma^i P^i(s,\mathrm{d}s') + \gamma^h(P^h M_{\theta_t})(s,\mathrm{d}s') \tag{138}$$

and since $M_{\theta_t}(s,\mathrm{d}s') = \delta_s(\mathrm{d}s') + m_{\theta_t}(s,s')\rho(\mathrm{d}s')$ we have $(P^h M_{\theta_t})(s,\mathrm{d}s') = P^h(s,\mathrm{d}s') + \int P^h(s,\mathrm{d}s'')m_{\theta_t}(s'',s')\rho(\mathrm{d}s')$ so the above rewrites as

$$M^{\mathrm{tar}}(s,\mathrm{d}s') = \delta_s(\mathrm{d}s') + \sum_{i=1}^{h-1}\gamma^i P^i(s,\mathrm{d}s') + \gamma^h P^h(s,\mathrm{d}s') + \gamma^h\int P^h(s,\mathrm{d}s'')m_{\theta_t}(s'',s')\rho(\mathrm{d}s') \tag{139}$$

and so

$$M^{\mathrm{tar}}(s,\mathrm{d}s') - M_{\theta_t}(s,\mathrm{d}s') = \\ - m_{\theta_t}(s,s')\rho(\mathrm{d}s') + \sum_{i=1}^{h} \gamma^i P^i(s,\mathrm{d}s') + \gamma^h \int P^h(s,\mathrm{d}s'') m_{\theta_t}(s'',s')\rho(\mathrm{d}s'). \tag{140}$$

Let us plug this into (136) for $\theta = \theta_t$, and study each contribution in turn. The term $-m_{\theta_t}(s,s')\rho(\mathrm{d}s')$ produces a contribution

$$-\iint \partial_\theta j_{\theta_t}(s,s')\rho(\mathrm{d}s) m_{\theta_t}(s,s')\rho(\mathrm{d}s') = \mathbb{E}_{s\sim\rho,\, s'\sim\rho}\, m_{\theta_t}(s,s')\, \partial_\theta m_{\theta_t}(s,s') \tag{141}$$

by (137). Each term $\gamma^i P^i$ produces a contribution

$$\gamma^i \iint \partial_\theta j_\theta(s,s')\rho(\mathrm{d}s) P^i(s,\mathrm{d}s') \tag{142}$$

which by definition of P^i, can be rewritten as

$$\gamma^i\, \mathbb{E}_{s_0\sim\rho,\, s_1\sim P(s_0,\mathrm{d}s_1),\, \ldots,\, s_i\sim P(s_{i-1},\mathrm{d}s_i)}\, \partial_\theta j_\theta(s_0,s_i). \tag{143}$$

For the same reason, the term $\gamma^h P^h m_{\theta_t}$ produces a contribution

$$\gamma^h\, \mathbb{E}_{s_0\sim\rho,\, s_1\sim P(s_0,\mathrm{d}s_1),\, \ldots,\, s_h\sim P(s_{i-h},\mathrm{d}s_h),\, s'\sim\rho}[m_{\theta_t}(s_h,s')\, \partial_\theta j_\theta(s_0,s')]. \tag{144}$$

Collecting all terms and using (137) to replace $\partial_\theta j$ with $-\partial_\theta m$ leads to the expression in the theorem.

For the case of the model (15) using $\tilde{m}_\theta$, proceed as for Theorem 6 and use the loss $J'(\theta) = \frac{1}{2}\|M_\theta\|_\rho^2 - \langle M_\theta, M^{\mathrm{tar}}\rangle_\rho$, which has the same minima as the loss J but makes sense in a more general setting. In this case we have

$$M^{\mathrm{tar}}(s,\mathrm{d}s') = \delta_s(\mathrm{d}s') + \sum_{i=1}^{h-1} \gamma^i P^i(s,\mathrm{d}s') + \gamma^h \int P^h(s,\mathrm{d}s'')\tilde{m}_{\theta_t}(s'',s')\rho(\mathrm{d}s') \tag{145}$$

The dot product $\langle M_\theta, M^{\mathrm{tar}}\rangle_\rho$ is given by (127). Expand the value of M^{tar} into (127), and proceed as above.

Proof of Theorem 26. As in the proof of Theorems 6 and 24, set $j_\theta(s,s') := (M^{\mathrm{tar}}(s,\mathrm{d}s') - M_\theta(s,\mathrm{d}s'))/\rho(\mathrm{d}s')$. Then

$$\partial_\theta J(\theta) = \iint j_\theta(s,s')\, \partial_\theta j_\theta(s,s')\rho(\mathrm{d}s)\rho(\mathrm{d}s') \tag{146}$$

$$= \iint \partial_\theta j_\theta(s,s')\rho(\mathrm{d}s)(j_\theta(s,s')\rho(\mathrm{d}s')) \tag{147}$$

$$= \iint \partial_\theta j_\theta(s,s')\rho(\mathrm{d}s)(M^{\mathrm{tar}}(s,\mathrm{d}s') - M_\theta(s,\mathrm{d}s')) \tag{148}$$

and since M^{tar} depends on θ_t but not on θ,

$$\partial_\theta j_\theta(s, s') = -\partial_\theta \left(\frac{M_\theta(s, \mathrm{d}s')}{\rho(\mathrm{d}s')} \right) = -\partial_\theta m_\theta(s, s'). \tag{149}$$

From the definition of M^{tar} and the composition of operators, we have

$$M^{\text{tar}}(s, \mathrm{d}s') = \delta_s(\mathrm{d}s') + \gamma \int M_{\theta_t}(s, \mathrm{d}s'') P(s'', \mathrm{d}s') \tag{150}$$

$$= \delta_s(\mathrm{d}s') + \gamma P(s, \mathrm{d}s') + \gamma \int m_{\theta_t}(s, s'') \rho(\mathrm{d}s'') P(s'', \mathrm{d}s') \tag{151}$$

thanks to the parameterization $M_{\theta_t}(s, \mathrm{d}s'') = \delta_s(\mathrm{d}s'') + m_{\theta_t}(s, s'')\rho(\mathrm{d}s'')$. Thus

$$M^{\text{tar}}(s, \mathrm{d}s') - M_{\theta_t}(s, \mathrm{d}s') = \gamma P(s, \mathrm{d}s') + \gamma \mathbb{E}_{s'' \sim \rho}[m_{\theta_t}(s, s'') P(s'', \mathrm{d}s')] - m_{\theta_t}(s, s')\rho(\mathrm{d}s'). \tag{152}$$

and plugging this into (148) at $\theta = \theta_t$, substituting $-\partial_\theta m_\theta$ for $\partial_\theta j$ as per (149), and rewriting the integrals as expectations under ρ and P, we find

$$\begin{aligned} -\partial_\theta J(\theta)_{|\theta=\theta_t} &= \gamma \mathbb{E}_{s \sim \rho,\, s' \sim P(s, \mathrm{d}s')} \partial_\theta m_{\theta_t}(s, s') \\ &+ \gamma \mathbb{E}_{s \sim \rho,\, s'' \sim \rho,\, s' \sim P(s'', \mathrm{d}s')}[m_{\theta_t}(s, s'')\, \partial_\theta m_{\theta_t}(s, s')] - \mathbb{E}_{s \sim \rho,\, s' \sim \rho}[m_{\theta_t}(s, s')\, \partial_\theta m_{\theta_t}(s, s')] \end{aligned} \tag{153}$$

which yields the expression in the theorem after renaming variables. The proof for $\tilde{m}$ is similar, using the loss J' instead of J as in the proof of Theorem 6.

Proof of Propositions 28 and 29. The pushforward by φ of a measure μ is the unique measure μ^φ such that, for any function f, one has $\int f(g)\mu^\varphi(\mathrm{d}g) = \int f(\varphi(s))\mu(\mathrm{d}s)$.

For Proposition 28, assume that the reward function at a state s is equal to $R(\varphi(s))$. By definition of the successor state operator M, the corresponding value function satisfies $V(s) = \int_{s'} R(\varphi(s'))M(s, \mathrm{d}s')$. By definition of the pushforward measure, the latter is equal to $\int_g R(g)M^\varphi(s, \mathrm{d}g)$. If $M^\varphi(s, \mathrm{d}g)$ is equal to $m^\varphi(s, g)\tau(\mathrm{d}g)$ for some probability distribution τ, this rewrites as $\mathbb{E}_{g \sim \tau} m^\varphi(s, g)R(g)$. This proves Proposition 28.

For Proposition 29, just start with the Bellman equation for M: $M(s, \mathrm{d}s_2) = \delta_s(\mathrm{d}s_2) + \gamma \mathbb{E}_{s' \sim P(\mathrm{d}s'|s)} M(s', \mathrm{d}s_2)$. Then take the pushforward by φ on both sides, using that the pushforward of measures is linear. Finally, use that the pushforward of the Dirac mass at s is the Dirac mass at $\varphi(s)$. This provides the Bellman equation for M^φ.

Proof of Theorem 30. The proof is entirely analogous to the proof of Theorem 6 for the model $\tilde{m}$.

B.3 Proofs for Section 5: Goal-Dependent Methods

Proof of Theorem 12. The proof is very similar to that of Theorem 6 and is omitted. Theorem 12 can also be obtained as a particular case of Theorem 13 applied to the state-action process.

Proof of Theorem 13. Proceed similarly to Theorem 6. Define a norm on $V(s, \mathrm{d}g)$ similarly to (1), as the L^2 norm of its density with respect to ρ_G:

$$\|V(s,\mathrm{d}g)\|^2_{\rho_{SG},\rho} := \mathbb{E}_{(s,g)\sim\rho_{SG}}\left[\frac{V(s,\mathrm{d}g)}{\rho_G(\mathrm{d}g)}^2\right]. \tag{154}$$

Let $v_\theta(s,g)$ be any smooth parametric model, and set $V_\theta(s,\mathrm{d}g) := v_\theta(s,g)\rho_G(\mathrm{d}g)$.

Let θ_0 be some value of the parameter θ, and define a target update V^{tar} via the Bellman equation (41):

$$V^{\mathrm{tar}}(s,\mathrm{d}g) := \alpha(s,g)\,\delta_{\varphi(s)}(\mathrm{d}g) + \gamma\mathbb{E}_{s'\sim P(\mathrm{d}s'|s,g)}V_{\theta_0}(s',\mathrm{d}g). \tag{155}$$

For any parameter θ, define the loss

$$J(\theta) := \frac{1}{2}\left\|V_\theta - V^{\mathrm{tar}}\right\|^2_{\rho_{SG},\rho}. \tag{156}$$

Then, as in Theorem 6 one finds

$$\begin{aligned}
-\partial_\theta J(\theta) &= \iint_{s,g}\rho_{SG}(\mathrm{d}s,\mathrm{d}g)\,\frac{\partial_\theta V_\theta(s,\mathrm{d}g)}{\rho_G(\mathrm{d}g)}\,\frac{V^{\mathrm{tar}}(s,\mathrm{d}g)-V_\theta(s,\mathrm{d}g)}{\rho_G(\mathrm{d}g)}\\
&= \iint_{s,g}\rho_{SG}(\mathrm{d}s,\mathrm{d}g)\,\partial_\theta v_\theta(s,g)\left(\frac{\alpha(s,g)\,\delta_{\varphi(s)}(\mathrm{d}g)}{\rho_G(\mathrm{d}g)}\right.\\
&\qquad\left.+\,\gamma\mathbb{E}_{s'\sim P(\mathrm{d}s'|s,g)}v_{\theta_0}(s',\mathrm{d}g) - v_\theta(s,\mathrm{d}g)\right)
\end{aligned} \tag{157}$$

The second part of this equation matches the Bellman gap part of the TD update in the statement of the theorem, with $\theta_0 = \theta$. (This also provides the TD update with an arbitrary target network defined by θ_0.)

For the first part with the Dirac term, remember that $\alpha(s,g) = \rho_S(\mathrm{d}g)\rho_G(\mathrm{d}g)/\rho_{SG}(\mathrm{d}s,\mathrm{d}g)$. Thus,

$$\begin{aligned}
&\iint_{s,g}\rho_{SG}(\mathrm{d}s,\mathrm{d}g)\,\partial_\theta v_\theta(s,g)\,\frac{\alpha(s,g)\,\delta_{\varphi(s)}(\mathrm{d}g)}{\rho_G(\mathrm{d}g)}\\
&= \int_s\rho_S(\mathrm{d}s)\int_g\partial_\theta v_\theta(s,g)\,\delta_{\varphi(s)}(\mathrm{d}g) = \int_s\rho_S(\mathrm{d}s)\,\partial_\theta v_\theta(s,\varphi(s)) = \mathbb{E}_{s\sim\rho_S}\,\partial_\theta v_\theta(s,\varphi(s))
\end{aligned} \tag{158}$$

as needed. This proves that the TD update is as announced in the statement.

Obviously, $\alpha = 1$ when s and g are independent.

For the statement about $\varphi = \mathrm{Id}$, note that the Bellman equation only depends on the value of α on pairs (s, g) such that $\varphi(s) = g$. Therefore, if the statement holds for some function α, then it also holds for any other function α' such that $\alpha'(s, g) = \alpha(s, g)$ when $\varphi(s) = g$, because this will define the same V^{tar}. With $\varphi = \mathrm{Id}$, this means that the statement holds for any other function α' with $\alpha'(g, g) = \alpha(g, g)$. Define $\alpha'(s, g) := \alpha(g, g)$. Then $\alpha'(g, g) = \alpha(g, g)$, and α' only depends on g. This completes the proof.

Proof of Theorem 14. Assume the action space A is countable. Let $\mathcal{Q}$ be the set of measurable functions from $S \times A$ to the set of measures on S.

For Q_1 and Q_2 in $\mathcal{Q}$, we write $Q_1 \leqslant Q_2$ if $Q_1(s, a, X) \leqslant Q_2(s, a, X)$ for any state-action (s, a) and measurable set $X \subset S$. The Bellman operator of Definition 11 acts on $\mathcal{Q}$ and is obviously monotonous: if $Q_1 \leqslant Q_2$ then $TQ_1 \leqslant TQ_2$.

Since the zero measure $\mathbf{0} \in \mathcal{Q}$ is the smallest measure, we have $T\mathbf{0} \geqslant \mathbf{0}$. Since T is monotonous, by induction we have $T^{t+1}\mathbf{0} \geqslant T^t\mathbf{0}$ for any $t \geqslant 0$. Thus, the $(T^t\mathbf{0})_{t \geqslant 0}$ form an increasing sequence of measures. Therefore, for every state-action (s, a) and measurable set X, the sequence $(T^t\mathbf{0})(s, a, X)$ is increasing, and thus converges to a limit. We denote this limit by $Q^*(s, a, X)$. We have to prove that $Q^* \in \mathcal{Q}$, namely, that for each (s, a), $Q^*(s, a, \cdot)$ is a measure. The only non-trivial point is σ-additivity.

Denote $Q_t := T^t\mathbf{0}$. If (X_i) is a countable collection of disjoint measurable sets, we have

$$\begin{aligned} Q^*(s, a, \cup_i X_i) = \lim_{t \to \infty} Q_t(s, a, \cup_i X_i) = \lim_{t \to \infty} \sum_i Q_t(s, a, X_i) \\ = \sum_i \lim_{t \to \infty} Q_t(s, a, X_i) = \sum_i Q^*(s, a, X_i) \end{aligned} \tag{159}$$

where the limit commutes with the sum thanks to the monotone convergence theorem, using that Q_t is non-decreasing. Therefore, Q^* is a measure.

Let us prove that $TQ^* = Q^*$. We have

$$TQ^*(s, a, \cdot) = \delta_s + \gamma \mathbb{E}_{s' \sim P(s'|s,a)} \sup_{a'} Q^*(s', a', \cdot) \tag{160}$$

by definition. For any s', denote $\tilde{Q}_t(s', \cdot) := \sup_{a'} Q_t(s', a', \cdot)$ where the supremum is as measures over S. Since Q_t is non-decreasing, so is $\tilde{Q}_t$.

For any state s', we have

$$\sup_{a'} Q^*(s', a', \cdot) = \sup_{a'} \sup_t Q_t(s', a', \cdot) = \sup_t \sup_{a'} Q_t(s', a', \cdot) = \sup_t \tilde{Q}_t(s', \cdot) \tag{161}$$

since supremums commute. Now, since $\tilde{Q}_t$ is non-decreasing, thanks to the monotone convergence theorem, the supremum commutes with integration

over $s' \sim P(s'|s,a)$ (which does not depend on t), namely,

$$\begin{aligned}\mathbb{E}_{s'\sim P(s'|s,a)} \sup_{a'} Q^*(s',a',\cdot) &= \mathbb{E}_{s'\sim P(s'|s,a)} \sup_t \tilde{Q}_t(s',\cdot) \\ &= \sup_t \mathbb{E}_{s'\sim P(s'|s,a)} \tilde{Q}_t(s',\cdot) = \sup_t \mathbb{E}_{s'\sim P(s'|s,a)} \sup_{a'} Q_t(s',a',\cdot) \quad (162)\end{aligned}$$

and so $TQ^* = \sup_t TQ_t$. Now, since $Q^t = T^t\mathbf{0}$, we have $TQ^t = T^{t+1}\mathbf{0}$, so that $\sup_{t\geqslant 0} TQ^t = \sup_{t\geqslant 1} T^t\mathbf{0} = Q^*$. So Q^* is a fixed point of T.

Let us prove that Q^* is the smallest such fixed point. Let Q' such that $TQ' = Q'$. Since $\mathbf{0} \leqslant Q'$ and T is monotonous, we have $T\mathbf{0} \leqslant TQ' = Q'$. By induction, $T^t\mathbf{0} \leqslant Q'$ for any $t \geqslant 0$. Therefore, $\sup_t T^t\mathbf{0} \leqslant Q'$, i.e., $Q^* \leqslant Q'$.

The statement for finite state spaces reduces to the classical uniqueness property of the usual Q function, separately for each goal state.

B.4 Examples of MDPs with Infinite Mass for Q^*

Here are two simple examples of MDPs with finite action space, for which the mass of the goal-dependent Q-function $Q^*(s,a,s_2)$ is infinite. The first has discrete states, the second, continuous ones.

Take for S an infinite rooted dyadic tree, namely, $S = \{\varnothing, 0, 1, 00, 01, \ldots\}$ the set of binary strings of finite length $k \geqslant 0$. Consider the two actions "add a 0 at the end" and "add a 1 at the end". Then, for every state s, $Q^*(s,a,\cdot)$ is a measure that gives mass γ^k to all states s_2 that are extensions of s by a length-k string that starts with a. Thus, its mass is $1 + \sum_{k\geqslant 1} \gamma^k 2^{k-1}$. This is infinite as soon as $\gamma \geqslant 1/2$. This extends to any number of actions by considering higher-degree trees.

A similar example with continuous states is obtained as follows. Let $S = [0;1) \times [0;1)$. Let $C = \{\varnothing, 0, 1, 00, 01, \ldots\}$ the dyadic tree above. For each string $w \in X$, consider the set $B_w \subset S$ defined as follows: B_w is made of those points $(x,y) \in S$ such that the binary expansion of x starts with w, and $y \in [1 - 1/2^k; 1 - 1/2^{k+1})$ where k is the length of w. Graphically, this creates a tree-like partition of the square S, where the empty string corresponds to the bottom half, the strings $w = 0$ and $w = 1$ correspond to two sets on the left and right above the bottom hald, etc. Define the following MDP with two actions 0 and 1: with action 0, every state $s \in B_w$ goes to a uniform random state in B_{w0}, and with action 1, every state $s \in B_w$ goes to a uniform random state in B_{w1}. The goal-dependent Q-function Q^* is similar to the dyadic tree above, but is continuous. Its mass is infinite for the same reasons.

B.5 Proofs for Sections 7 and 8: Second-Order Methods

Proof of Theorem 17. Define $\hat{M} := (\mathrm{Id} - \gamma\hat{P})^{-1}$ where $\hat{P}$ is updated by (58). The update (58) can be rewritten as $\hat{P} \leftarrow \hat{P} + (1/n_s)\mathbb{1}_s(\mathbb{1}_{s'}^\top - \mathbb{1}_s^\top\hat{P})$.

This is a rank-one update of $\hat{P}$. The update of $\mathrm{Id} - \gamma\hat{P}$ is $-\gamma$ times the update of $\hat{P}$, and is still rank-one: it is equal to $uv^\top$ with $u := -(\gamma/n_s)\mathbb{1}_s$ and $v^\top := (\mathbb{1}_{s'}^\top - \mathbb{1}_s^\top\hat{P})$. The Sherman–Morrison formula gives the update of the inverse of a matrix after a rank-one update. By this formula, the update of $\hat{M} = (\mathrm{Id} - \gamma\hat{P})^{-1}$ is

$$\hat{M} \leftarrow \hat{M} - \frac{\hat{M}uv^\top\hat{M}}{1 + v^\top\hat{M}u} = \hat{M} + \frac{1}{n_s}\frac{\hat{M}\mathbb{1}_s(\gamma\mathbb{1}_{s'}^\top - \gamma\mathbb{1}_s^\top\hat{P})\hat{M}}{1 - \frac{1}{n_s}(\gamma\mathbb{1}_{s'}^\top - \gamma\mathbb{1}_s^\top\hat{P})\hat{M}\mathbb{1}_s} \tag{163}$$

Now, since $\hat{M} = (\mathrm{Id} - \gamma\hat{P})^{-1}$, we have $\gamma\hat{P}\hat{M} = \hat{M} - \mathrm{Id}$. Therefore, the terms $(\gamma\mathbb{1}_{s'}^\top - \gamma\mathbb{1}_s^\top\hat{P})\hat{M}$ are equal to $\gamma\mathbb{1}_{s'}^\top\hat{M} - \mathbb{1}_s^\top\hat{M} + \mathbb{1}_s^\top$, and the update is

$$\hat{M} \leftarrow \hat{M} + \frac{1}{n_s}\frac{\hat{M}\mathbb{1}_s(\gamma\mathbb{1}_{s'}^\top\hat{M} - \mathbb{1}_s^\top\hat{M} + \mathbb{1}_s^\top)}{1 - \frac{1}{n_s}(\gamma\mathbb{1}_{s'}^\top\hat{M} - \mathbb{1}_s^\top\hat{M} + \mathbb{1}_s^\top)\mathbb{1}_s} \tag{164}$$

$$= \hat{M} + \frac{1}{n_s}\frac{\hat{M}\mathbb{1}_s(\gamma\mathbb{1}_{s'}^\top\hat{M} - \mathbb{1}_s^\top\hat{M} + \mathbb{1}_s^\top)}{1 - \frac{1}{n_s}(\gamma\hat{M}_{s's} - \hat{M}_{ss} + 1)} \tag{165}$$

which is the exact update of $\hat{M}$. This provides the update (61).

The value function $\hat{V}$ of the estimated process is $(Id - \gamma\hat{P})^{-1}\hat{R} = \hat{M}\hat{R}$. When $\hat{M} \leftarrow \hat{M} + \delta M$ and $\hat{R} \leftarrow \hat{R} + \delta R$ one has $\hat{V} \leftarrow \hat{V} + \hat{M}\,\delta R + \delta M\hat{R} + \delta M\,\delta R$. From (58) we have $\delta R = \frac{1}{n_s}(r_s - \hat{R}_s)\mathbb{1}_s = \frac{1}{n_s}(r_s\mathbb{1}_s - \mathbb{1}_s\mathbb{1}_s^\top\hat{R})$. Plugging in the value of δM from (165), keeping only first-order terms in $1/n_s$, and using $\mathbb{1}_s^\top\hat{M}\hat{R} = \mathbb{1}_s^\top\hat{V} = \hat{V}_s$, provides the update of $\hat{V}$ in (62).

Proof of Theorem 18. First, note that the expectation in the statement is averaged over the next step, but conditional to all quantities $\hat{M}$, $\hat{V}$, etc., computed in the previous steps. In this proof, we will just write $\mathbb{E}$ for short.

Since the the denominator in (165) is $1 + o(1/n_s)$, the update (165) of $\hat{M}$ is $\hat{M} \leftarrow \hat{M} + \delta M$ with

$$\delta M = \frac{1}{n_s}\hat{M}\mathbb{1}_s(\gamma\mathbb{1}_{s'}^\top\hat{M} - \mathbb{1}_s^\top\hat{M} + \mathbb{1}_s^\top) + o(1/n_s). \tag{166}$$

We want to compute the expectation of this update when s is sampled from ρ and s' from $P_{ss'}$. This yields

$$\mathbb{E}[\delta M] = \sum_{s,s'}\rho_s P_{ss'}\frac{1}{n_s}\hat{M}\mathbb{1}_s(\gamma\mathbb{1}_{s'}^\top\hat{M} - \mathbb{1}_s^\top\hat{M} + \mathbb{1}_s^\top) + o(1/n_s) \tag{167}$$

$$= \frac{1}{t}\sum_{s,s'}P_{ss'}\hat{M}\mathbb{1}_s(\gamma\mathbb{1}_{s'}^\top\hat{M} - \mathbb{1}_s^\top\hat{M} + \mathbb{1}_s^\top) + o(1/t) \tag{168}$$

where the last equality holds because $n_s = t\rho_s + o(t)$ by the law of large numbers (since s is sampled from ρ). Now, we have $\sum_{s,s'}P_{ss'}\mathbb{1}_s\mathbb{1}_{s'}^\top = P$ and $\sum_{s,s'}P_{ss'}\mathbb{1}_s\mathbb{1}_s^\top = \sum_s \mathbb{1}_s\mathbb{1}_s^\top = \mathrm{Id}$. Thus,

$$\mathbb{E}[\delta M] = \frac{1}{t}\hat{M}(\gamma P\hat{M} - \hat{M} + \mathrm{Id}) + o(1/t) \tag{169}$$

as needed.

To compute the update of $\hat{V} = \hat{M}\hat{R}$, let us first compute the update of $\hat{R}$. By (58), the latter is $\hat{R} \leftarrow \hat{R} + \delta R$ with

$$\delta R = \frac{1}{n_s}(r_s - \hat{R}_s)\mathbb{1}_s = \frac{1}{t\rho_s}(r_s - \hat{R}_s)\mathbb{1}_s + o(1/t). \tag{170}$$

Now, the update of $\hat{V} = \hat{M}\hat{R}$ is $\delta V = \delta M\,\hat{R} + \hat{M}\,\delta R + \delta M\,\delta R$. The last term $\delta M\,\delta R$ is $O(1/t^2)$, so we can drop it. We find

$$\mathbb{E}[\delta V] = \mathbb{E}[\delta M\,\hat{R}] + \mathbb{E}[\hat{M}\,\delta R] + o(1/t) \tag{171}$$

$$= \mathbb{E}[\delta M]\hat{R} + \hat{M}\mathbb{E}[\delta R] + o(1/t) \tag{172}$$

since the expectations are averaged over the next step but conditional on the previous steps, which comprises the previous values of $\hat{R}$ and $\hat{M}$. Next,

$$\mathbb{E}[\delta M]\hat{R} = \frac{1}{t}\hat{M}(\gamma P\hat{M} - \hat{M} + \mathrm{Id})\hat{R} + o(1/t) \tag{173}$$

$$= \frac{1}{t}\hat{M}(\gamma P\hat{V} - \hat{V} + \hat{R}) + o(1/t) \tag{174}$$

since $\hat{V} = \hat{M}\hat{R}$. Next,

$$\hat{M}\mathbb{E}[\delta R] = \hat{M}\sum_s \rho_s \frac{1}{t\rho_s}(\mathbb{E}[r_s] - \hat{R}_s)\mathbb{1}_s + o(1/t) \tag{175}$$

$$= \frac{1}{t}\hat{M}(R - \hat{R}) + o(1/t) \tag{176}$$

since $\sum_s \mathbb{E}[r_s]\mathbb{1}_s = R$ and $\sum_s \hat{R}_s\mathbb{1}_s = \hat{R}$. Summing, we find $\mathbb{E}[\delta V] = \frac{1}{t}\hat{M}(\gamma P\hat{V} - \hat{V} + \hat{R} + R - \hat{R}) + o(1/t)$ as needed.

Proof of Theorem 21. First, note that we expressed this theorem for a single transition $s \to s'$, while we expressed the similar theorem for TD using the Bellman operator $M^{\mathrm{tar}} = \mathrm{Id} + \gamma PM$, which is the sum of the single-transition update for all values of s.

This is because the single-transition update is more informative in this case, especially given the exact update of M in Theorem 17. (At first, we worked at the operator level, and found a parametric expression which was the same in expectation over transitions $s \to s'$, but did not correspond to a single-transition update, had a larger variance, and performed much worse in practice.)

However, the single-transition updates (58) and (61) only make sense in a discrete-state setting. Thus, to best preserve the information from observing a single transition, we state and derive Theorem 21 in a discrete-state setting. The resulting parametric update makes sense for continuous states.

(In Appendix H we rigorously derive this same update for continuous states, in expectation over $s \to s'$, as we did for TD. The analogue of the Bellman operator $\delta M = \mathrm{Id} + \gamma PM - M$ for implicit process updating is the Newton–Bellman operator $\delta M = M - M(\mathrm{Id} - \gamma P)M$ of Definition 19.)

Thus, let us work in a discrete setting, using matrix notation. Let us consider δM and δV given by (61)–(62). For simplicity we omit the $o(1/n_s)$ terms in (61)–(62): they are absorbed in the $o(1/t)$ of the final statement of the theorem, because $n_s \sim \rho(s)t$ by the law of large numbers. (Indeed, $\rho(s)$ is *defined* as the probability to sample a transition from s in our data model.)

With $M^{\mathrm{tar}} := M_{\theta_t} + \delta M$ and from the definition (1) of $\|\cdot\|_\rho$, we obtain

$$\left\|M_\theta - M^{\mathrm{tar}}\right\|_\rho^2 = \mathbb{E}_{s_1\sim\rho,\, s_2\sim\rho} \frac{(M_\theta - M^{\mathrm{tar}})^2_{s_1 s_2}}{\rho(s_2)^2} \tag{177}$$

$$= \sum_{s_1,s_2} \frac{\rho(s_1)}{\rho(s_2)} (M_\theta - M^{\mathrm{tar}})^2_{s_1 s_2} \tag{178}$$

so that the gradient step on the loss is

$$-\partial_\theta J(\theta) = -\partial_\theta \left(\tfrac{1}{2} \left\|M_\theta - M^{\mathrm{tar}}\right\|_\rho^2 \right) \tag{179}$$

$$= \sum_{s_1,s_2} \frac{\rho(s_1)}{\rho(s_2)} (M^{\mathrm{tar}} - M_\theta)_{s_1 s_2} \, \partial_\theta (M_\theta)_{s_1 s_2} \tag{180}$$

and we compute the gradient step at $\theta = \theta_t$; since $M^{\mathrm{tar}} - M_{\theta_t} = \delta M$, we get

$$-\partial_\theta J(\theta)_{\theta=\theta_t} = \sum_{s_1,s_2} \frac{\rho(s_1)}{\rho(s_2)} (\delta M)_{s_1 s_2} \, \partial_\theta (M_\theta)_{s_1 s_2}. \tag{181}$$

Now, remember that the parameterization is $(M_\theta)_{s_1 s_2} = \mathbb{1}_{s_1=s_2} + m_\theta(s_1, s_2)\rho(s_2)$. We obtain

$$\partial_\theta (M_\theta)_{s_1 s_2} = \partial_\theta m_\theta(s_1, s_2)\rho(s_2) \tag{182}$$

and from the expression (61) for δM, up to $O(1/n_s^2)$ terms, we have

$$-\partial_\theta J(\theta)_{\theta=\theta_t} = \sum_{s_1,s_2} \frac{\rho(s_1)}{\rho(s_2)} \frac{1}{n_s} (M_{\theta_t})_{s_1 s} \left(\mathbb{1}_{s_2=s} + \gamma (M_{\theta_t})_{s' s_2} - (M_{\theta_t})_{s s_2}\right) \partial_\theta m_{\theta_t}(s_1, s_2)\rho(s_2) \tag{183}$$

so that $\rho(s_2)$ cancels out. Now let us expand $(M_{\theta_t})_{s_1 s_2} = \mathbb{1}_{s_1=s_2} + m_{\theta_t}(s_1, s_2)\rho(s_2)$ into this expression. We have

$$\mathbb{1}_{s_2=s} + \gamma (M_{\theta_t})_{s' s_2} - (M_{\theta_t})_{s s_2} = \mathbb{1}_{s'=s_2} + \gamma m_{\theta_t}(s', s_2)\rho(s_2) - m_{\theta_t}(s, s_2)\rho(s_2) \tag{184}$$

and after tediously collecting all terms, we arrive at

$$
\begin{aligned}
-\partial_\theta J(\theta)_{\theta=\theta_t} = &\sum_{s_1,s_2} \frac{\rho(s_1)}{n_s} \mathbb{1}_{s_1=s}\, \mathbb{1}_{s'=s_2}\, \partial_\theta m_{\theta_t}(s_1,s_2) \\
&+ \sum_{s_1,s_2} \frac{\rho(s_1)}{n_s} m_{\theta_t}(s_1,s)\rho(s)\, \mathbb{1}_{s'=s_2}\, \partial_\theta m_{\theta_t}(s_1,s_2) \\
&+ \sum_{s_1,s_2} \frac{\rho(s_1)}{n_s} \mathbb{1}_{s_1=s} \left(\gamma m_{\theta_t}(s',s_2) - m_{\theta_t}(s,s_2)\right) \rho(s_2)\, \partial_\theta m_{\theta_t}(s_1,s_2) \\
&+ \sum_{s_1,s_2} \frac{\rho(s_1)}{n_s} m_{\theta_t}(s_1,s)\, \rho(s) \left(\gamma m_{\theta_t}(s',s_2) - m_{\theta_t}(s,s_2)\right) \rho(s_2)\, \partial_\theta m_{\theta_t}(s_1,s_2).
\end{aligned}
\tag{185}
$$

The first term rewrites

$$
\sum_{s_1,s_2} \frac{\rho(s_1)}{n_s} \mathbb{1}_{s_1=s}\, \mathbb{1}_{s'=s_2}\, \partial_\theta m_{\theta_t}(s_1,s_2) = \frac{\rho(s)}{n_s} \partial_\theta m_{\theta_t}(s,s'). \tag{186}
$$

The second one rewrites

$$
\sum_{s_1,s_2} \frac{\rho(s_1)}{n_s} m_{\theta_t}(s_1,s)\rho(s)\, \mathbb{1}_{s'=s_2}\, \partial_\theta m_{\theta_t}(s_1,s_2) = \frac{\rho(s)}{n_s}\, \mathbb{E}_{s_1\sim\rho}[m_{\theta_t}(s_1,s)\, \partial_\theta m_{\theta_t}(s_1,s')]. \tag{187}
$$

Similarly, the third term in (185) rewrites as

$$
\frac{\rho(s)}{n_s}\, \mathbb{E}_{s_2\sim\rho} \left(\gamma m_{\theta_t}(s',s_2) - m_{\theta_t}(s,s_2)\right) \partial_\theta m_{\theta_t}(s,s_2) \tag{188}
$$

and the fourth as

$$
\frac{\rho(s)}{n_s}\, \mathbb{E}_{s_1\sim\rho,\, s_2\sim\rho} \left(\gamma m_{\theta_t}(s',s_2) - m_{\theta_t}(s,s_2)\right) \partial_\theta m_{\theta_t}(s_1,s_2). \tag{189}
$$

Now, by definition of ρ in our data model, $\rho(s)$ is the frequency with which a transition starting at s is sampled. Therefore, by the law of large numbers, $n_s \sim t\rho(s)$ when $t \to \infty$. Therefore,

$$
\frac{\rho(s)}{n_s} = 1/t + o(1/t) \tag{190}
$$

when $t \to \infty$. (This is the advantage of defining all norms with respect to ρ; anyway, in a general setting, ρ is the only available measure on $\mathcal{S}$ to define norms with.)

Thus, when $t \to \infty$,

$$
\begin{aligned}
-\partial_\theta J(\theta)_{\theta=\theta_t} = \frac{1}{t} &\left(\partial_\theta m_{\theta_t}(s,s') + \mathbb{E}_{s_1\sim\rho}[m_{\theta_t}(s_1,s)\, \partial_\theta m_{\theta_t}(s_1,s')]\right. \\
&+\mathbb{E}_{s_2\sim\rho} \left(\gamma m_{\theta_t}(s',s_2) - m_{\theta_t}(s,s_2)\right) \partial_\theta m_{\theta_t}(s,s_2) \\
&\left.+\mathbb{E}_{s_1\sim\rho,\, s_2\sim\rho} \left(\gamma m_{\theta_t}(s',s_2) - m_{\theta_t}(s,s_2)\right) \partial_\theta m_{\theta_t}(s_1,s_2)\right) + o(1/t)
\end{aligned}
\tag{191}
$$

which is the expression (66) given in Theorem 21.

This ends the proof of Theorem 21 for discrete states, which is the only setting in which the single-transition update δM makes sense. Yet the expressions obtained also make sense for continuous states. Appendix H contains a rigorous derivation for continuous states, in expectation over $s \to s'$, as we did for parametric TD.

Proof of Proposition 22. Given a parametric model V_φ with parameter φ, at each step t, define a target $V^{\text{tar}} := V_{\varphi_t} + \delta V$ with δV given by (62). As for Theorem 21, the update (62) is defined via a single transition $s \to s'$ and only makes sense in a discrete space, as does V^{tar}. So we work with a parametric model on a discrete space and observe that the resulting update is well-defined in continuous spaces. (As with Theorem 21, continuous spaces can be treated rigorously by considering the expectation over $s \to s'$, see Appendix H.)

The loss function on φ is $J^V(\varphi) := \frac{1}{2} \left\| V_\varphi - V^{\text{tar}} \right\|^2_{L^2(\rho)}$. Then

$$-\partial_\varphi J^V(\varphi) = \sum_{s_1} \rho(s_1) \left(V_{\varphi_t}(s_1) + \delta V_{s_1} - V_\varphi(s_1) \right) \partial_\varphi V_\varphi(s_1) \tag{192}$$

and so

$$-\partial_\varphi J^V(\varphi)_{|\varphi=\varphi_t} = \sum_{s_1} \rho(s_1)\, \delta V_{s_1}\, \partial_\varphi V_{\varphi_t}(s_1). \tag{193}$$

Plugging in the expression (62) for δV (with $\hat{V}_s = V_{\varphi_t}(s)$ and $\hat{M}_{s_1 s_2} = M_{\theta_t}(s_1, s_2)$) yields, again omitting the $o(1/n_s)$ terms,

$$-\partial_\varphi J^V(\varphi)_{|\varphi=\varphi_t} = \left(r_s + \gamma V_{\varphi_t}(s') - V_{\varphi_t}(s) \right) \sum_{s_1} \frac{\rho(s_1)}{n_s} M_{\theta_t}(s_1, s)\, \partial_\varphi V_{\varphi_t}(s_1) \tag{194}$$

and plugging in the parametric model $M_{\theta_t}(s_1, s) = \mathbb{1}_{s_1=s} + m_{\theta_t}(s_1, s)\rho(s)$ yields

$$-\partial_\varphi J^V(\varphi)_{|\varphi=\varphi_t} = \left(r_s + \gamma V_{\varphi_t}(s') - V_{\varphi_t}(s) \right) \sum_{s_1} \frac{\rho(s_1)}{n_s} (\mathbb{1}_{s_1=s} + m_{\theta_t}(s_1, s)\rho(s))\, \partial_\varphi V_{\varphi_t}(s_1) \tag{195}$$

$$= \left(r_s + \gamma V_{\varphi_t}(s') - V_{\varphi_t}(s) \right) \left(\frac{\rho(s)}{n_s}\, \partial_\varphi V_{\varphi_t}(s) + \frac{\rho(s)}{n_s} \mathbb{E}_{s_1 \sim \rho}[m_{\theta_t}(s_1, s) \partial_\varphi V_{\varphi_t}(s_1)] \right) \tag{196}$$

and as above, $\rho(s)/n_s = 1/t + o(1/t)$ when $t \to \infty$, so this yields the expression (78) in the theorem.

Proof of Theorem 16. This convergence analysis is partially inspired by [PW19]. The main differences are the data model and the metrics computed.

We assume that ρ is an invariant probability measure of P, and that the reward is bounded by $R_{\max}$ with probability 1. We define the empirical distribution of states $\hat{\rho}_t$ as: $(\hat{\rho}_t)_s = \frac{n_s}{t}$, with n_s the number of visits to state s up to time t. We also consider $\hat{P}_t$ and $\hat{R}_t$ as defined in (58).

The initialization of $\hat{P}$ and $\hat{R}$ does not matter, as it is erased the first time a state is visited. To fix ideas, we initialize $\hat{P}$ and $\hat{R}$ to 0; this helps if $\rho = 0$ for some states. (In particular, $\hat{P}$ may be substochastic: $0 \leqslant \sum_j \hat{P}_{ij} \leqslant 1$ for all i.)

We define $\widehat{\rho P}_t$ as the empirical distribution of transitions: $(\widehat{\rho P}_t)_{s_1 s_2} := \frac{n_{s_1 s_2}}{t}$ where $n_{s_1 s_2}$ is the number of observations of a transition (s_1, s_2) up to time t. We have $(\hat{P}_t)_{s_1 s_2} = \frac{n_{s_1 s_2}}{n_{s1}}$ if $n_{s1} > 0$, or 0 if $n_{s1} = 0$. Hence $\left(\widehat{\rho P}_t\right)_{s_1 s_2} = \hat{\rho}_{s_1} (\hat{P}_t)_{s_1 s_2}$.

The proof strategy is to bound the errors $\|\hat{M} - M\|_{\rho,\mathrm{TV}}$ and $\|\hat{V} - V\|_\rho$ by errors on $\widehat{\rho P}$ and $\hat{R}$. The error on $\widehat{\rho P}$ can then be controlled by concentration inequalities on empirical distributions, and the error on $\hat{R}$ can be bounded via the Hoeffding inequality.

The successor state operator estimate $\hat{M}$ is $(\mathrm{Id} - \gamma \hat{P})^{-1}$. By the Bellman equation for M and $\hat{M}$,

$$\hat{M} - M = \gamma \hat{P} \hat{M} - \gamma P M \tag{197}$$

$$= \gamma P (\hat{M} - M) + \gamma (\hat{P} - P) \hat{M} \tag{198}$$

and therefore

$$(\mathrm{Id} - \gamma P)(\hat{M} - M) = \gamma (\hat{P} - P) \hat{M} \tag{199}$$

and thus

$$\hat{M} - M = \gamma M (\hat{P} - P) \hat{M} \tag{200}$$

by definition of M.

Therefore,

$$\|\hat{M} - M\|_{\rho,\mathrm{TV}} = \gamma \|M (\hat{P} - P) \hat{M}\|_{\rho,\mathrm{TV}} \tag{201}$$

$$= \frac{\gamma}{2} \sum_{i,j} \rho_i \left| \sum_{k,l} M_{ik} (\hat{P} - P)_{kl} \hat{M}_{lj} \right| \tag{202}$$

$$\leqslant \frac{\gamma}{2} \sum_{i,j,k,l} \rho_i M_{ik} |\hat{P} - P|_{kl} \hat{M}_{lj}. \tag{203}$$

We know that $(1 - \gamma) M$ is a stochastic matrix, and ρ is an invariant probability measure. Therefore, $\sum_i \rho_i M_{ik} = \frac{1}{1-\gamma} \rho_k$. Moreover, if $\hat{P}$ is substochastic, $\sum_j \hat{M}_{lj} \leqslant \frac{1}{1-\gamma}$ (with equality if P is stochastic). Therefore,

$$\|\hat{M} - M\|_{\rho,\mathrm{TV}} \leqslant \frac{\gamma}{(1-\gamma)^2} \|\hat{P} - P\|_{\rho,\mathrm{TV}}. \tag{204}$$

We define (ρP) as the matrix $\mathrm{Diag}(\rho)P$. We now bound the error $\|\hat{P} - P\|_{\rho,\mathrm{TV}}$ by the error $\|\widehat{\rho P} - (\rho P)\|_{\mathrm{TV}}$, in order to use standard concentration inequalities on empirical distributions:

$$\begin{aligned}
\|\hat{P} - P\|_{\rho,\mathrm{TV}} &= \frac{1}{2}\|\mathrm{Diag}(\rho)\hat{P} - (\rho P)\|_1 && (205)\\
&\leqslant \|\widehat{\rho P} - (\rho P)\|_{\mathrm{TV}} + \frac{1}{2}\|\mathrm{Diag}(\hat{\rho} - \rho)\hat{P}\|_1 && (206)\\
&\leqslant \|\widehat{\rho P} - \rho P\|_{\mathrm{TV}} + \|\hat{\rho} - \rho\|_{\mathrm{TV}} && (207)\\
&\leqslant \|\widehat{\rho P} - \rho P\|_{\mathrm{TV}} + \frac{1}{2}\sum_i |\sum_j \hat{\rho}_i \hat{P}_{ij} - \rho_i P_{ij}| && (208)\\
&\leqslant \|\widehat{\rho P} - \rho P\|_{\mathrm{TV}} + \frac{1}{2}\sum_{i,j} |\hat{\rho}_i \hat{P}_{ij} - \rho_i P_{ij}| && (209)\\
&\leqslant 2\|\widehat{\rho P} - \rho P\|_{\mathrm{TV}} && (210)
\end{aligned}$$

Therefore,

$$\|\hat{M} - M\|_{\rho,\mathrm{TV}} \leqslant \frac{2\gamma}{(1-\gamma)^2}\|\widehat{\rho P} - \rho P\|_{\mathrm{TV}}. \tag{211}$$

We now consider the error on $\hat{V}$. We have:

$$\begin{aligned}
\|\hat{V} - V\|_\rho &= \|\hat{M}\hat{R} - MR\|_\rho && (212)\\
&\leqslant \|(\hat{M} - M)\hat{R}\|_\rho + \|M(\hat{R} - R)\|_\rho && (213)\\
&\leqslant 2R_{\max}\|\hat{M} - M\|_{\rho,\mathrm{TV}} + \frac{1}{1-\gamma}\|\hat{R} - R\|_\rho && (214)\\
&\leqslant \frac{4R_{\max}}{(1-\gamma)^2}\|\widehat{\rho P} - \rho P\|_{\mathrm{TV}} + \frac{1}{1-\gamma}\|\widehat{R} - R\|_{\hat{\rho}} && (215)
\end{aligned}$$

We now bound $\|\widehat{\rho P} - \rho P\|_{\mathrm{TV}}$ and $\|\widehat{R} - R\|_{\hat{\rho}}$.

We can bound the error $\|\widehat{R} - R\|_{\hat{\rho}}$ with the Hoeffding inequality. $\widehat{R}_s$ is the average of n_s independent samples of expectation R_s. Since the reward is bounded by R_{max} with probability 1, we can use Hoeffding's inequality. For any s with $n_s > 0$, we have:

$$\mathbb{P}(|\widehat{R} - R|_s > u) \leqslant 2\exp\left(-\frac{n_s u^2}{2R_{\max}^2}\right) \tag{216}$$

Hence, for any s with $n_s > 0$, we have with probability $1 - \frac{\delta}{S}$:

$$|\widehat{R_t} - R|_s \leqslant R_{\max}\sqrt{\frac{2}{n_s}\log\frac{2S}{\delta}} \tag{217}$$

and since $\hat{\rho}_s = n_s/t$, this rewrites as

$$\hat{\rho}_s|\widehat{R_t} - R|_s \leqslant \frac{R_{\max}}{t}\sqrt{2n_s\log\frac{2S}{\delta}}. \tag{218}$$

For states with $n_s = 0$, $\hat{\rho}_s = 0$ and the inequality still holds. If for all s, $\mathbb{P}(|\widehat{R_t} - R|_s \geqslant \varepsilon_s) \leqslant \frac{\delta}{S}$, then

$$\mathbb{P}(\|\hat{R} - R\|_{\hat{\rho}} \geqslant \sum_s \varepsilon_s) \leqslant \sum_s \mathbb{P}(\hat{\rho}_s |\widehat{R_t} - R|_s \geqslant \varepsilon_s) \tag{219}$$

$$\leqslant \sum_s \frac{\delta}{S} = \delta \tag{220}$$

Thus, with probability $1 - \delta$,

$$\|\widehat{R} - R\|_{\hat{\rho}} \leqslant \frac{R_{\max}}{t} \sqrt{2 \log \frac{2S}{\delta}} \sum_s \sqrt{n_s} \tag{221}$$

$$\leqslant \frac{R_{\max}}{t} \sqrt{2 \log \frac{2S}{\delta}} \sqrt{\sum_s n_s} \sqrt{S} \tag{222}$$

$$\leqslant \frac{R_{\max}}{\sqrt{t}} \sqrt{2S \log \frac{2S}{\delta}} \tag{223}$$

since $\sum_s n_s = t$.

We now bound $\|\widehat{\rho P} - \rho P\|_{\mathrm{TV}}$. $\widehat{\rho P}$ is the empirical distribution over all possible transitions. The set of all possible transitions is of size S^2. However, if $(\rho P)_{s_1 s_2} = 0$, then with probability 1, $\widehat{\rho P}_{s_1 s_2} = 0$. Therefore, if E is the number of edges of the MDP ((s, s') is an edge if $P_{ss'} > 0$), $\|\widehat{\rho P} - \rho P\|_{\mathrm{TV}}$ can be bounded by an inequality on the total variation error of the empirical measure on a set of size E. We use Theorem 2.2 from [WOS+03][17], and have with with probability $1 - \delta$:

$$\|\widehat{\rho P}_t - \rho P\|_{\mathrm{TV}} \leqslant \frac{1}{2\sqrt{t}} \sqrt{2E \log \frac{2}{\delta}} \tag{224}$$

By plugging equation (224) into (211), with probability $1 - \delta$,

$$\|\hat{M} - M\|_{\rho,\mathrm{TV}} \leqslant \frac{\gamma}{(1-\gamma)^2 \sqrt{t}} \sqrt{2E \log \frac{2}{\delta}} \tag{225}$$

Finally, by plugging (223) and (224) into (215), with probability $1 - \delta$, we obtain

$$\|\hat{V} - V\|_{\rho} \leqslant \frac{2R_{\max}}{(1-\gamma)^2} \frac{1}{\sqrt{t}} \sqrt{2E \log \frac{4}{\delta}} + \frac{1}{1-\gamma} \frac{R_{\max}}{\sqrt{t}} \sqrt{2S \log \frac{4S}{\delta}} \tag{226}$$

$$\leqslant \frac{3R_{\max}}{(1-\gamma)^2} \sqrt{\frac{2E}{t} \log \frac{4S}{\delta}} \tag{227}$$

which ends the proof.

[17] We use the trivial bound $\varphi(\pi) \geqslant 2$ with the notation of the original paper.

C The Bellman–Newton Operator and Path Composition

In Section 4.3, we explained the link between learning successor states and counting paths in a Markov process. Here, we formalize that link, by studying how updating M via the Bellman equation (or the backward Bellman equation) updates the paths represented in M. We will prove that after t steps, the estimate of M via Bellman–Newton exactly contains all paths up to length $2^t - 1$ with their correct probabilities in the Markov process, while forward and backward TD exactly contain all paths up to length t.

Thus for each algorithm (forward TD, backward TD, and Bellman–Newton), we consider the exact (deterministic, non-sampled) update: we set $M_0 = \mathrm{Id}$ and then define at step $t+1$ the update M_{t+1} as the target update given by the corresponding fixed point equation. For forward TD, the operator update is defined as:

$$M_{t+1}^{\mathrm{TD}} = \mathrm{Id} + \gamma P M_t^{\mathrm{TD}}. \tag{228}$$

For backward TD, the operator update is defined as:

$$M_{t+1}^{\mathrm{BTD}} = \mathrm{Id} + \gamma M_t^{\mathrm{BTD}} P. \tag{229}$$

The Bellman–Newton update (Definition 19) with learning rate 1 is

$$M_{t+1}^{\mathrm{BN}} = 2M_t^{\mathrm{BN}} - M_t^{\mathrm{BN}}(\mathrm{Id} - \gamma P) M_t^{\mathrm{BN}}. \tag{230}$$

In expectation over the transition $s \to s'$, the expected exact online update is $\delta M = \frac{1}{t}(M - M^2 + \gamma MPM)$ (Eq. 63). Here for the corresponding deterministic operator we use a learning rate 1 instead of $1/t$, which yields $\delta M = M - M(\mathrm{Id} - \gamma P)M$, hence the update (230). The successor state operator $M = (\mathrm{Id} - \gamma P)^{-1}$ is a fixed point of this update. [18] This corresponds to the *Newton method* $M \leftarrow 2M - MAM$ for inverting a matrix A [PS91].

We now relate the forward TD, backward TD, and Bellman–Newton updates to path composition. For each algorithm, we prove by induction that at step t, there exists an integer n_t such that $(M_t)_{ss'}$ is equal to the number of paths from s to s' with length at most n_t, weighted by their probability and discounted by their length, namely,

$$(M_t)_{ss'} = \sum_{p \text{ path from } s \text{ to } s', |p| \leqslant n_t} \gamma^{|p|} \mathbb{P}(p) = \sum_{k=0}^{n_t} \gamma^k \sum_{s=s_0,\dots,s_{k-1},s_k=s'} P_{s_0 s_1} \cdots P_{s_{n-1} s_{n_t}} \tag{231}$$

[18] It is not the only fixed point; for instance, $M = 0$ is another. But it is the only full-rank fixed point.

where as in Section 2, $|p|$ denotes the length of a path p and $\mathbb{P}(p) = P_{s_0 s_1} \cdots P_{s_{n-1} s_n}$ its probability in the Markov process. Equivalently,

$$M_t = \sum_{0 \leqslant k \leqslant n_t} \gamma^k P^k. \tag{232}$$

The three algorithms will differ by the value of n_t.

For $t = 0$, $M_0 = \mathrm{Id}$, and the induction hypothesis is satisfied.

If the end point of a path p_1 corresponds to the starting point of a path p_2, we denote $p_1 \cdot p_2$ the concatenation of the two paths.

For forward TD, we have $M_{t+1}^{\mathrm{TD}} = \mathrm{Id} + \gamma P M_t^{\mathrm{TD}}$. By induction, if $M_t^{\mathrm{TD}} = \sum_{0 \leqslant k \leqslant n_t^{\mathrm{TD}}} \gamma^k P^k$, then we find $M_{t+1}^{\mathrm{TD}} = \mathrm{Id} + \gamma P \sum_{0 \leqslant k \leqslant n_t^{\mathrm{TD}}} \gamma^k P^k = \sum_{0 \leqslant k \leqslant n_t^{\mathrm{TD}}+1} \gamma^k P^k$. Equivalently, looking at paths we have

$$\begin{aligned}
(M_{t+1}^{\mathrm{TD}})_{ss'} &= \delta_{s=s'} + \gamma (P M_t^{\mathrm{TD}})_{ss'} && (233)\\
&= \delta_{s=s'} + \gamma \sum_{s''} P_{ss''} \sum_{p \text{ path from } s'' \text{ to } s', |p| \leqslant n_t^{\mathrm{TD}}} \gamma^{|p|} \mathbb{P}(p) && (234)\\
&= \delta_{s=s'} + \sum_{s''} \sum_{p \text{ path from } s'' \text{ to } s', |p| \leqslant n_t^{\mathrm{TD}}} \gamma^{|p|+1} \mathbb{P}((s, s'') \cdot p) && (235)\\
&= \delta_{s=s'} + \sum_{p \text{ path from } s \text{ to } s', 1 \leqslant |p| \leqslant n_t^{\mathrm{TD}}} \gamma^{|p|} \mathbb{P}(p) && (236)\\
&= \sum_{p \text{ path from } s \text{ to } s', |p| \leqslant n_t^{\mathrm{TD}}+1} \gamma^{|p|} \mathbb{P}(p) && (237)
\end{aligned}$$

Thus the induction hypothesis is satisfied with $n_{t+1}^{\mathrm{TD}} = n_t^{\mathrm{TD}} + 1$. By induction, $n_t^{\mathrm{TD}} = t$: at step t, M_t^{TD} is the weighted sum of paths of length at most t. M_{t+1}^{TD} is obtained from M_t^{TD} by adding a transition to the left to every known path (and re-adding the length-0 paths via the Id term).

Likewise, with backward TD we have

$$\begin{aligned}
(M_{t+1}^{\mathrm{BTD}})_{ss'} &= \delta_{s=s'} + \gamma (M_t^{\mathrm{BTD}} P)_{ss'} && (238)\\
&= \delta_{s=s'} + \sum_{s''} \sum_{p \text{ path from } s'' \text{ to } s', |p| \leqslant n_t^{\mathrm{BTD}}} \gamma^{|p|+1} \mathbb{P}(p \cdot (s'', s')) && (239)\\
&= \sum_{p \text{ path from } s \text{ to } s', |p| \leqslant n_t^{\mathrm{BTD}}+1} \gamma^{|p|} \mathbb{P}(p) && (240)
\end{aligned}$$

Contrary to forward TD, M_{t+1}^{BTD} is obtained from M_t^{BTD} by adding a transition to the right to every known path. This still leads to $n_t^{\mathrm{BTD}} = t$.

We now consider the Bellman–Newton operator update. We have

$$M_{t+1}^{\mathrm{BN}} = 2M_t^{\mathrm{BN}} - M_t^{\mathrm{BN}} (\mathrm{Id} - \gamma P) M_t^{\mathrm{BN}}. \tag{241}$$

Let us first compute $(\mathrm{Id} - \gamma P) M_t^{\mathrm{BN}}$. By the induction hypothesis and by the same reasoning as for forward TD, we have

$$((\mathrm{Id} - \gamma P) M_t^{\mathrm{BN}})_{ss'} = M_t^{\mathrm{BN}} - \gamma P M_t^{\mathrm{BN}} \tag{242}$$

$$= \sum_{p \text{ path from } s \text{ to } s', |p| \leqslant n_t^{\mathrm{BN}}} \gamma^{|p|} \mathbb{P}(p) - \sum_{p \text{ path from } s \text{ to } s', 1 \leqslant |p| \leqslant n_t^{\mathrm{BN}}+1} \gamma^{|p|} \mathbb{P}(p) \tag{243}$$

$$= \delta_{s=s'} - \gamma^{n_t^{\mathrm{BN}}+1} \left(P^{n_t^{\mathrm{BN}}+1} \right)_{ss'}. \tag{244}$$

Therefore,

$$M_{t+1}^{\mathrm{BN}} = 2M_t^{\mathrm{BN}} - M_t^{\mathrm{BN}} (\mathrm{Id} - \gamma P) M_t^{\mathrm{BN}} \tag{245}$$

$$= 2M_t^{\mathrm{BN}} - M_t^{\mathrm{BN}} (\mathrm{Id} - \gamma^{n_t^{\mathrm{BN}}+1} P^{n_t^{\mathrm{BN}}+1}) \tag{246}$$

$$= M_t^{\mathrm{BN}} + \gamma^{n_t^{\mathrm{BN}}+1} M_t^{\mathrm{BN}} P^{n_t^{\mathrm{BN}}+1} \tag{247}$$

$$= \sum_{p \text{ path from } s \text{ to } s', |p| \leqslant n_t^{\mathrm{BN}}} \gamma^{|p|} \mathbb{P}(p) + \sum_{p \text{ path from } s \text{ to } s', n_t^{\mathrm{BN}}+1 \leqslant |p| \leqslant 2n_t^{\mathrm{BN}}+1} \gamma^{|p|} \mathbb{P}(p) \tag{248}$$

$$= \sum_{p \text{ path from } s \text{ to } s', |p| \leqslant 2n_t^{\mathrm{BN}}+1} \gamma^{|p|} \mathbb{P}(p) \tag{249}$$

Therefore, $n_{t+1}^{\mathrm{BN}} = 2n_t^{\mathrm{BN}} + 1$. At every step the Bellman–Newton operator update is doubling the maximal length of all known paths.

The efficiency of the operator Bellman–Newton update can also be explained from properties of the Newton method. Indeed, the Bellman–Newton update in (230) corresponds to the Newton update $M \leftarrow 2M - MAM$ for inverting the matrix A, applied to $A = \mathrm{Id} - \gamma P$ [PS91]. With this method, the error $\mathrm{Id} - AM$ gets squared at each iteration: $\mathrm{Id} - AM \leftarrow (\mathrm{Id} - AM)^2$ [PS91]. Here at each step, if M_t exactly contains all paths up to length n_t, then the error $\mathrm{Id} - AM_t$ contains all paths of length $n_t + 1$, namely, if $M_t = \sum_{k \leqslant n_t} \gamma^k P^k$ then $\mathrm{Id} - AM_t = \gamma^{n_t+1} P^{n_t+1}$. Thus squaring the error corresponds to doubling $n_t + 1$.

D Successor States, Eligibility Traces, and the Backward Process

In this section, we relate the update equation (78) for the value function using M, to the algorithm TD(λ) and eligibility traces. We also prove the statement that backward TD is forward TD on the time-reversed process (Theorem 33).

More precisely, we prove (Theorem 31) that the expectation of the TD(1) update (expectation over the eligibility traces given the current state) is

the update (78) of the value function using the successor state operator. Thus, updating V via (78) using a learned model m_θ of M is equivalent to estimating the true M via a model, while eligibility traces are an unbiased Monte Carlo estimator of the true M. This suggests the possibility of using mixed estimates, such as eligibility traces over a few past steps, and a model m_θ for the older past.

Eligibility traces require access to an arbitrarily long trajectory $(s_t)_{t\in\mathbb{Z}}$ (which, for convenience, we index with both positive and negative integers, with s_0 the state at the current time). Thus, contrary to the rest of this text, we assume that the Markov process is ergodic and that the data are coming from a stationary random trajectory of the process. In this case, the sampling measure ρ is the stationary distribution, and the law of any sequence of consecutive observations $(s_t, \dots, s_{t+n})$ from the trajectory is $\rho(\mathrm{d}s_t)P(s_t, \mathrm{d}s_{t+1}) \cdots P(s_{t+n-1}, \mathrm{d}s_{t+n})$.

We also assume that for every s, $P(s, \mathrm{d}s')$ is absolutely continuous with respect to $\rho(\mathrm{d}s')$. This is not necessary but leads to nicer expressions. In that case, $M(s, \mathrm{d}s') = \delta_s(\mathrm{d}s') + m(s, s')\rho(\mathrm{d}s')$ for some function m.

In the tabular setting, TD(λ) maintains a vector e_t over states; e_t is updated by

$$e_t(\tilde{s}) = \mathbb{1}_{s_t} + \gamma\lambda e_{t-1}(\tilde{s}) \qquad \forall \tilde{s} \tag{250}$$

$$\delta V(\tilde{s}) = e_t(\tilde{s})(r_t + \gamma V(s_{t+1}) - V(s_t)) \qquad \forall \tilde{s}. \tag{251}$$

This can be generalized to continuous environments and to a parametric model V_φ of V, by formally defining e as the discounted empirical measure of the past:

$$e_t(\mathrm{d}\tilde{s}) := \sum_{n\geqslant 0} (\gamma\lambda)^n \delta_{s_{t-n}}(\mathrm{d}\tilde{s}) = \delta_{s_t}(\mathrm{d}\tilde{s}) + \gamma\lambda e_{t-1}(\mathrm{d}\tilde{s}) \tag{252}$$

corresponding to the parametric update of V_φ by

$$\delta\varphi = (r_t + \gamma V_\varphi(s_{t+1}) - V_\varphi(s_t)) \int_{\tilde{s}} \partial_\varphi V_\varphi(\tilde{s})\, e_t(\mathrm{d}\tilde{s}) \tag{253}$$

$$= (r_t + \gamma V_\varphi(s_{t+1}) - V_\varphi(s_t)) \sum_{n\geqslant 0} (\gamma\lambda)^n \partial_\varphi V_\varphi(s_{t-n}). \tag{254}$$

We have the following statement:

Theorem 31. *Let ρ be the invariant measure of the Markov process, and $M_{\gamma\lambda} := (\mathrm{Id} - \gamma\lambda P)^{-1}$ the successor state operator with discount factor $\gamma\lambda$. Let $m_{\gamma\lambda}$ be the density of $(M_{\gamma\lambda} - \mathrm{Id})$ with respect to ρ: $M_{\gamma\lambda}(\tilde{s}, \mathrm{d}s_2) = \delta_{\tilde{s}}(\mathrm{d}s_2) + m_{\gamma\lambda}(\tilde{s}, s_2)\rho(\mathrm{d}s_2)$.*

Then, the expected eligibility trace $e_t(\mathrm{d}\tilde{s})$ knowing $s_t = s$ is:

$$\mathbb{E}\left[e_t(\mathrm{d}\tilde{s}) | s_t = s\right] = \delta_s(\mathrm{d}\tilde{s}) + m_{\gamma\lambda}(\tilde{s}, s)\rho(\mathrm{d}\tilde{s}) = \frac{M_{\gamma\lambda}(\tilde{s}, \mathrm{d}s)\rho(\mathrm{d}\tilde{s})}{\rho(\mathrm{d}s)} \tag{255}$$

Moreover, the expectation of the parametric TD(λ) update (254) when a transition (s, s') is observed is equal to the update (78) of V using $m_{\gamma\lambda}$:

$$\mathbb{E}[\delta\varphi|(s_t, s_{t+1}) = (s, s')] = (r_t + \gamma V_\varphi(s') - V_\varphi(s)) \left(\partial_\varphi V_\varphi(s) + \mathbb{E}_{\tilde{s}\sim\rho}\left[m_{\gamma\lambda}(\tilde{s}, s)\, \partial_\varphi V_\varphi(\tilde{s})\right]\right) \tag{256}$$

with ρ-probability 1 over s_t.

The proof of this theorem involves the time-reversal of the Markov process; indeed, eligibility traces are a Monte Carlo estimate of the discounted measure of *predecessor* states.

Define the *backward process* $P_{\text{back}}(s', \mathrm{d}s)$ by reversing time: it is the law of s given s' in a transition $s \to s'$. More precisely, let (s, s') be a random pair of states distributed according to $\rho(\mathrm{d}s)P(s, \mathrm{d}s')$, and define $P_{\text{back}}(s', \mathrm{d}s)$ to be the conditional distribution of s given s' under this distribution. (This exists by the general theory of conditional distributions [Par05, Thm. 8.1], and is well-defined up to a set of ρ-measure 0.) Since ρ is the invariant measure of the process, the law of both s and s' is ρ, and one has

$$\rho(\mathrm{d}s)P(s, \mathrm{d}s') = \rho(\mathrm{d}s')P_{\text{back}}(s', \mathrm{d}s) \tag{257}$$

by definition of conditional probabilities.

Then, given s_t, the distribution of s_{t-n} follows the backward process from s_t. Namely, the law of any sequence of observations $(s_{t-n}, \dots, s_t)$ from the stationary distribution of the process satisfies

$$\rho(\mathrm{d}s_{t-n})P(s_{t-n}, \mathrm{d}s_{t-n+1}) \cdots P(s_{t-1}, \mathrm{d}s_t) = \rho(\mathrm{d}s_t)P_{\text{back}}(s_t, \mathrm{d}s_{t-1}) \cdots P_{\text{back}}(s_{t-n+1}, \mathrm{d}s_{t-n}). \tag{258}$$

Lemma 32. *Let m be the density of M, namely, $M(s, \mathrm{d}s') = \delta_s(\mathrm{d}s') + m(s, s')\rho(\mathrm{d}s')$. (This exists under the assumption above that P is absolutely continuous with respect to ρ.)*

Let $M^{\text{back}} := (\mathrm{Id} - \gamma P_{\text{back}})^{-1}$ be the successor state operator of the backward process, and let m^{back} be the associated density, $M^{\text{back}}(s, \mathrm{d}s') = \delta_s(\mathrm{d}s') + m^{\text{back}}(s, s')\rho(\mathrm{d}s')$. Then $\rho(\mathrm{d}s')M^{\text{back}}(s', \mathrm{d}s) = \rho(\mathrm{d}s)M(s, \mathrm{d}s')$ and

$$m^{\text{back}}(s', s) = m(s, s') \tag{259}$$

for ρ-almost every (s, s').

Proof. By induction from the definition of the backward process, we have $\rho(\mathrm{d}s')P^n_{\text{back}}(s', \mathrm{d}s) = \rho(\mathrm{d}s)P^n(s, \mathrm{d}s')$. Then by definition of M^{back},

$$\rho(\mathrm{d}s')M^{\text{backward}}(s', \mathrm{d}s) = \rho(\mathrm{d}s') \sum_{n\geqslant 0} \gamma^n P^n_{\text{backward}}(s', \mathrm{d}s) = \sum_{n\geqslant 0} \gamma^n \rho(\mathrm{d}s)P^n(s, \mathrm{d}s') \tag{260}$$

$$= \rho(\mathrm{d}s)M(s, \mathrm{d}s') \tag{261}$$

Since $M(s, \mathrm{d}s') = \delta_s(\mathrm{d}s') + m(s, s')\rho(\mathrm{d}s')$, and likewise for M^{back}, this implies

$$\rho(\mathrm{d}s')m^{\text{back}}(s', s)\rho(\mathrm{d}s) = \rho(\mathrm{d}s)m(s, s')\rho(\mathrm{d}s') \tag{262}$$

as needed. □

Proof of Theorem 31. By definition of eligibility traces, one has $e_t(\mathrm{d}\tilde{s}) = \sum_{n\geqslant 0}(\gamma\lambda)^n \delta_{s_{t-n}}(\mathrm{d}\tilde{s})$. Therefore, the expectation of e_t over the past of s_t knowing s_t is:

$$\mathbb{E}[e_t(\mathrm{d}\tilde{s})|s_t = s] = \mathbb{E}\left[\sum_{n\geqslant 0}(\gamma\lambda)^n \delta_{s_{t-n}}(\mathrm{d}\tilde{s})|s_t = s\right] \tag{263}$$

$$= \sum_{n\geqslant 0}(\gamma\lambda)^n P^n_{\text{back}}(s, \mathrm{d}\tilde{s}) \tag{264}$$

$$= M^{\text{back}}_{\gamma\lambda}(s, \mathrm{d}\tilde{s}) \tag{265}$$

where $M^{\text{back}}_{\gamma\lambda} := (\mathrm{Id} - \gamma\lambda P_{\text{back}})^{-1}$ is the successor state operator of the backward process. By Lemma 32, this is

$$\mathbb{E}[e_t(\mathrm{d}\tilde{s})|s_t = s] = \delta_s(\mathrm{d}\tilde{s}) + m(\tilde{s}, s)\rho(\mathrm{d}\tilde{s}) \tag{266}$$

as needed.

Therefore, the expectation of the update (254) of V with TD(λ) is:

$$\mathbb{E}\left[\delta\varphi|s_t = s, s_{t+1} = s'\right] = (r_t + \gamma V_\varphi(s_{t+1}) - V_\varphi(s_t)) \int_{\tilde{s}} \partial_\varphi V_\varphi(\tilde{s})\, \mathbb{E}[e_t(\mathrm{d}\tilde{s})|s_t = s, s_{t+1} = s'] \tag{267}$$

$$= (r_t + \gamma V_\varphi(s_{t+1}) - V_\varphi(s_t)) \int_{\tilde{s}} \partial_\varphi V_\varphi(\tilde{s})(\delta_s(\mathrm{d}\tilde{s}) + m(\tilde{s}, s)\rho(\mathrm{d}\tilde{s})) \tag{268}$$

$$= (r_t + \gamma V_\varphi(s') - V_\varphi(s))\left(\partial_\varphi V_\varphi(s) + \mathbb{E}_{\tilde{s}\sim\rho}\left[\partial_\varphi V_\varphi(\tilde{s}) m_{\gamma\lambda}(\tilde{s}, s)\right]\right) \tag{269}$$

□

Finally, the backward process provides a simple proof that backward TD is forward TD on the backward process. Remember that the forward and backward successor state operators are linked by $\rho(\mathrm{d}s_1)M(s_1, \mathrm{d}s_2) = \rho(\mathrm{d}s_2)M^{\text{back}}(s_2, \mathrm{d}s_1)$.

Theorem 33 (Backward TD is forward TD on the backward process). *Let M and M^{back} be measure-valued functions such that M^{back} is the time-reverse of M, namely $\rho(\mathrm{d}s_1)M(s_1, \mathrm{d}s_2) = \rho(\mathrm{d}s_2)M^{\text{back}}(s_2, \mathrm{d}s_1)$. Then the backward TD update*

$$M \leftarrow \mathrm{Id} + \gamma MP \tag{270}$$

is equivalent (ρ-almost everywhere) to

$$M^{\text{back}} \leftarrow \mathrm{Id} + \gamma P_{\text{back}} M^{\text{back}}. \tag{271}$$

Proof. Let $D_\rho(\mathrm{d}s_1, \mathrm{d}s_2)$ be the diagonal measure with marginal ρ, namely, $D_\rho(\mathrm{d}s_1, \mathrm{d}s_2) = \rho(\mathrm{d}s_1)\delta_{s_1}(\mathrm{d}s_2) = \rho(\mathrm{d}s_2)\delta_{s_2}(\mathrm{d}s_1)$. Remember that the operator Id corresponds to the process $\delta_{s_1}(\mathrm{d}s_2)$. By multiplying the backward TD update by $\rho(\mathrm{d}s_1)$ one gets

$$\rho(\mathrm{d}s_1)M(s_1, \mathrm{d}s_2) \leftarrow D_\rho(\mathrm{d}s_1, \mathrm{d}s_2) + \gamma\rho(\mathrm{d}s_1)(MP)(s_1, \mathrm{d}s_2) \tag{272}$$

$$= D_\rho(\mathrm{d}s_1, \mathrm{d}s_2) + \gamma \int_{s'} \rho(\mathrm{d}s_1)M(s_1, \mathrm{d}s')P(s', \mathrm{d}s_2) \tag{273}$$

$$= D_\rho(\mathrm{d}s_1, \mathrm{d}s_2) + \gamma \int_{s'} M^{\mathrm{back}}(s', \mathrm{d}s_1)\rho(\mathrm{d}s')P(s', \mathrm{d}s_2) \tag{274}$$

$$= D_\rho(\mathrm{d}s_1, \mathrm{d}s_2) + \gamma \int_{s'} M^{\mathrm{back}}(s', \mathrm{d}s_1)\rho(\mathrm{d}s_2)P_{\mathrm{back}}(s_2, \mathrm{d}s') \tag{275}$$

$$= D_\rho(\mathrm{d}s_1, \mathrm{d}s_2) + \gamma\rho(\mathrm{d}s_2)(P_{\mathrm{back}}M^{\mathrm{back}})(s_2, \mathrm{d}s_1) \tag{276}$$

and since $\rho(\mathrm{d}s_1)M(s_1, \mathrm{d}s_2) = \rho(\mathrm{d}s_2)M^{\mathrm{back}}(s_2, \mathrm{d}s_1)$, this rewrites as

$$\rho(\mathrm{d}s_2)M^{\mathrm{back}}(s_2, \mathrm{d}s_1) \leftarrow \rho(\mathrm{d}s_2)\delta_{s_2}(\mathrm{d}s_1) + \gamma\rho(\mathrm{d}s_2)(P_{\mathrm{back}}M^{\mathrm{back}})(s_2, \mathrm{d}s_1) \tag{277}$$

namely (ρ-almost everywhere),

$$M^{\mathrm{back}}(s_2, \mathrm{d}s_1) \leftarrow \delta_{s_2}(\mathrm{d}s_1) + \gamma(P_{\mathrm{back}}M^{\mathrm{back}})(s_2, \mathrm{d}s_1) \tag{278}$$

which is forward TD on M^{back} for the time-reversed process. □

E Fixed Points for the FB Representation of M

Here we state precisely, and prove, the fixed points properties for the four variants of successor state learning in the FB representation (Section 6), in the tabular and in the overparameterized case. The "tabular" case for F and B means that the state space is finite and the values of $F(s)$ and $B(s)$ are stored explicitly for every state s.

We fully describe the fixed points of the four algorithms ff-FB, bb-FB, fb-FB, and bf-FB, which have quite different properties.

We state these properties for the tabular case; by a simple argument the fixed points are the same for *overparameterized* F and B. [19]

In this section, we abuse notation by considering F and B both as functions from the state space to $\mathbb{R}^r$ (as in Section 6), and as $r \times \#S$-matrices. The model $M(s_1, \mathrm{d}s_2) = F(s_1)^\top B(s_2)\rho(\mathrm{d}s_2)$ rewrites as $M = F^\top B\,\mathrm{diag}(\rho)$ or

[19] Namely, parameterizations F_{θ_F} and B_{θ_B} such that any function F can be realized for some θ_F, and moreover the map $\partial_{\theta_F} F_{\theta_F}$ is surjective for any θ_F, and likewise for B. In short, any F and B can be realized, and any small *change* of F or B can be realized by a small change in θ_F and θ_B.

$\tilde{m} = F^\top B$, viewing everything as matrices with entries indexed by the state space.

We also assume that $\rho_s > 0$ for every state s: every state is sampled with nonzero probability.

By direct identification in Proposition 15, in the tabular case we find the following expressions for the updates of F and B.

Proposition 34 (Tabular FB updates). *Assume the state space is finite and let F and B be two $r \times \#S$-matrices. Let the parameter θ_F of F be the matrix F itself and likewise for B.*

Abbreviate $\grave{\rho}$ for the diagonal matrix with entries ρ_s for each state s.

Then the updates $\delta\theta_F$ and $\delta\theta_B$ of Proposition 15 for the FB representation of M are equal to

$$\delta F = B\grave{\rho} - \Sigma_B F \Delta^\top \grave{\rho}, \qquad \delta B = F\grave{\rho} - F\grave{\rho}\Delta F^\top B\grave{\rho} \tag{279}$$

for forward TD on F and B respectively, and to

$$\delta F = B\grave{\rho} - B(\grave{\rho}\Delta)^\top B^\top F\grave{\rho}, \qquad \delta B = F\grave{\rho} - \Sigma_F B\grave{\rho}\Delta \tag{280}$$

for backward TD on F and B respectively. Here Δ is the matrix $\mathrm{Id} - \gamma P$, $\Sigma_B = B\grave{\rho}B^\top$, and $\Sigma_F = F\grave{\rho}F^\top$.

Proposition 35 (The fixed points of fb-FB approximate M in $L^2(\rho)$ norm). *The fixed points of the tabular fb-FB algorithm are the local extrema of the error*

$$\ell(F, B) := \mathbb{E}_{s_1 \sim \rho,\, s_2 \sim \rho} \left(F^\top(s_1) B(s_2) - \tilde{m}(s_1, s_2) \right)^2 \tag{281}$$

where $\tilde{m}(s_1, s_2) := M(s_1, \mathrm{d}s_2)/\rho(\mathrm{d}s_2)$ is the value of $\tilde{m}$ for the true successor state operator M. [20]

In that case, $F^\top B\grave{\rho}$ is a truncated singular value decomposition of the operator M acting on the space of functions over S equipped with the $L^2(\rho)$ norm.

Proposition 36 (Fixed points of ff-FB correspond to eigenspaces of M). *The set of approximations $F^\top B\grave{\rho}$ of M that appear as a fixed point of the tabular ff-FB algorithm is exactly the set of operators such that there exists an $L^2(\rho)$-orthogonal decomposition $\mathbb{R}^{\#S} = E \oplus E'$ of functions over the state space such that E is stable by M (namely, $ME \subset E$), E has dimension at most r, and $F^\top B\grave{\rho}$ is equal to M on E and to 0 on E'.*

[20] This is the Hilbert-Schmidt norm of the difference between M and its approximation $F^\top B\rho$, as operators on the space of functions over S equipped with the $L^2(\rho)$ norm (Appendix I).

Proposition 37 (Fixed points of bb-FB). *The set of approximations $F^\top B\grave{\rho}$ of M that appear as a fixed point of the tabular bb-FB algorithm is exactly the set of operators such that there exists an $L^2(\rho)$-orthogonal decomposition $\mathbb{R}^{\#S} = E \oplus E'$ of functions over the state space such that E' is stable by M (namely, $ME' \subset E'$), E has dimension at most r, and $F^\top B\grave{\rho}$ is the projection of M onto E, namely, $F^\top B\grave{\rho} = \Pi_E M$ with Π_E the $L^2(\rho)$-orthogonal projector onto E.*

Remark 38 (Fixed points of bb-FB correspond to eigen-probability densities of M). Stability of E' by M is equivalent to stability of E by $\grave{\rho}^{-1}M^\top\grave{\rho}$. This corresponds to the Markov operator acting on probability densities: if the state at time t has probability distribution $f\rho$ for some vector f, then the state at time $t+1$ has probability distribution $(\grave{\rho}^{-1}P^\top\grave{\rho}f)\rho$.

Thus, in bb-FB, the space E is a stable space of probability densities for P and M.

In contrast, the bf-FB algorithm can stabilize on any subspace of features. For instance, in rank 1, set $F^\top$ to any vector, then set $B^\top = \alpha F^\top$ where $\alpha = 1/(F\grave{\rho}(\mathrm{Id} - \gamma P)F^\top)$ (assuming this is nonzero). In fact, fixed points of bf-FB just compute a weak inverse of $\grave{\rho}(\mathrm{Id} - \gamma P)$ in an arbitrary r-dimensional subspace.

Proposition 39 (Fixed points of bf-FB). *The set of approximations $F^\top B\grave{\rho}$ of M that appear as a fixed point of the tabular bf-FB algorithm is exactly the set of operators such that there exists a subspace E of $L^2(\rho)$ of dimension at most r such that $F^\top B\grave{\rho}$ has image E and kernel $E^\perp$, and $F^\top B\grave{\rho}$ is the inverse of $\Pi(\mathrm{Id} - \gamma P)\Pi$ as operators from E to E, where Π is the $L^2(\rho)$-orthogonal projector on E.*

Moreover, if ρ is an invariant probability distribution of the Markov process, then every subspace E of $L^2(\rho)$ of dimension at most r provides such a fixed point $F^\top B\grave{\rho}$.

Proof of Proposition 35. Viewing $\tilde{m}$, F and B as matrices, the loss is

$$\ell(F, B) = \sum_{ij} \rho(i)\rho(j) \left(\sum_k F_{ki} B_{kj} - \tilde{m}_{ij} \right)^2 \tag{282}$$

so that

$$\frac{\partial \ell(F, B)}{\partial F_{ki}} = 2 \sum_j \rho(i)\rho(j) B_{kj} \left(\sum_{k'} F_{k'i} B_{k'j} - \tilde{m}_{ij} \right) \tag{283}$$

which is the ki entry of the matrix $2B\grave{\rho}(B^\top F - \tilde{m}^\top)\grave{\rho}$.

Now, F is a local extremum of this loss if and only if this derivative is 0 for every ki, namely, if and only if the matrix $B\grave{\rho}(B^\top F - \tilde{m}^\top)\grave{\rho}$ is 0. Now, by definition of $\tilde{m}$ we have $M = \tilde{m}\grave{\rho}$, namely, $\tilde{m} = \Delta^{-1}\grave{\rho}^{-1}$. So $B\grave{\rho}(B^\top F - \tilde{m}^\top)\grave{\rho} = 0$ is equivalent to $B\grave{\rho}B^\top F\grave{\rho} - B(\Delta^{-1})^\top\grave{\rho} = 0$. Since $\grave{\rho}$ and

Δ are invertible, by multiplying by $\grave{\rho}^{-1}\Delta^\top\grave{\rho}$ on the right, this is equivalent to $B\grave{\rho}B^\top F\Delta^\top\grave{\rho} - B\grave{\rho} = 0$. This is equivalent to $\delta F = 0$ in (279), namely, to F being a fixed point of forward TD.

A similar computation with B proves that $\partial\ell(F,B)/\partial B = 0$ if and only if $\delta B = 0$ in (280), namely, if and only if B is a fixed point of *backward* TD. Therefore, F and B are a local optimum of ℓ if and only if they are a fixed point of the fb-FB algorithm.

Let us turn to the statement about singular value decompositions. Generally speaking, we know that the matrices of a given rank which are local extrema of the matrix norm of the difference with $\tilde{m}$ are truncated singular value decompositions of $\tilde{m}$; however, here these matrices are parameterized as $F^\top B$, and a priori this parameterization might change the local extrema, so we give a full proof.

By Lemma 46, the matrix $F^\top B\grave{\rho}$ is a truncated SVD of M if and only if $F^\top B\grave{\rho}$ and M coincide on $(\operatorname{Ker} F^\top B\grave{\rho})^\perp$ and $M(\operatorname{Ker} F^\top B\grave{\rho})\perp \operatorname{Im} F^\top B\grave{\rho}$. Here all orthogonality relations are defined with respect to the $L^2(\rho)$ inner product, namely, $\langle x, y\rangle = x^\top\grave{\rho}y$.

If F is a fixed point of (279), then $0 = B\grave{\rho} - \Sigma_B F\Delta^\top\grave{\rho}$. Since $\grave{\rho}$ is invertible and since $\Sigma_B = B\grave{\rho}B^\top$, this rewrites as $B(\operatorname{Id} - \grave{\rho}B^\top F\Delta^\top) = 0$. Taking transposes, this is $(\operatorname{Id} - \Delta F^\top B\grave{\rho})B^\top = 0$. By definition, M is the inverse of Δ; multiplying by M, we find $(M - F^\top B\grave{\rho})B^\top = 0$. This implies that M and $F^\top B\grave{\rho}$ coincide on the image of $B^\top$. A fortiori, they coincide on the image of $B^\top F\grave{\rho}$, which is included in the image of $B^\top$.

In general, for an operator A on a Euclidean space, $\operatorname{Im} A = (\operatorname{Ker} A^*)^\perp$ with A^* the adjoint of A. Here, with the inner product from $L^2(\rho)$, the adjoint of A is $\grave{\rho}^{-1}A^\top\grave{\rho}$ (Appendix I). So the adjoint of $B^\top F\grave{\rho}$ is $F^\top B\grave{\rho}$. Therefore, $\operatorname{Im} B^\top F\grave{\rho}$ is $(\operatorname{Ker} F^\top B\grave{\rho})^\perp$. So M and $F^\top B\grave{\rho}$ coincide on $(\operatorname{Ker} F^\top B\grave{\rho})^\perp$. This was the first point to be proved.

Next, if B is a fixed point of (280), then $0 = F\grave{\rho} - F\grave{\rho}F^\top B\grave{\rho}\Delta$. Multiplying on the right by $M = \Delta^{-1}$ this is equivalent to $F\grave{\rho}(M - F^\top B\grave{\rho}) = 0$. This states that the image of $M - F^\top B\grave{\rho}$ is ρ-orthogonal to the image of $F^\top$. So for any x, $(M - F^\top B\grave{\rho})x \perp \operatorname{Im} F^\top$. Take $x \in \operatorname{Ker} F^\top B\grave{\rho}$. Then $Mx \perp \operatorname{Im} F^\top$. Since $\operatorname{Im} F^\top B\grave{\rho} \subset \operatorname{Im} F^\top$, we have $Mx \perp \operatorname{Im} F^\top B\grave{\rho}$ as well. Therefore, the image of $\operatorname{Ker} F^\top B\grave{\rho}$ by M is orthogonal to the image of $F^\top B\grave{\rho}$. This was the second point to be proved. □

Proof of Proposition 36. In this proof, we denote

$$f := F\grave{\rho}^{1/2}, \qquad b := B\grave{\rho}^{1/2}, \qquad D := \grave{\rho}^{1/2}\Delta\grave{\rho}^{-1/2}, \tag{284}$$

using that ρ is invertible. Then the fixed point equations $\delta F = 0$ and $\delta B = 0$ for the forward TD updates (279) rewrite as

$$0 = b - bb^\top fD^\top, \qquad 0 = f - fDf^\top b. \tag{285}$$

This change of variables cancels the ρ factors and maps $L^2(\rho)$-orthogonality to usual orthogonality.

($\Rightarrow$). Assume that $F^\top B\grave{\rho}$ is a fixed point of ff-FB, so that the fixed point equations above are satisfied.

The first fixed point equation yields $Df^\top bb^\top = b^\top$. Let b' be the Moore-Penrose pseudoinverse of $b^\top$ (equal to $(bb^\top)^{-1}b$ if invertible). By the general properties of the Moore-Penrose pseudoinverse, $b^\top b'$ is the orthogonal projector onto $\operatorname{Im} b^\top$, and $bb^\top b' = b$. Thus, multiplying $Df^\top bb^\top = b^\top$ by b' on the right, we find $Df^\top b = \Pi$ where Π is the orthogonal projector onto $\operatorname{Im} b^\top$. This rewrites as $f^\top b = D^{-1}\Pi$, so that $f^\top b$ is equal to D^{-1} on $\operatorname{Im}\Pi$ and to 0 on its orthogonal.

The second fixed point equation reads $fDf^\top b = f$. Since $Df^\top b = \Pi$ this means that $f\Pi = f$, or $f^\top = \Pi f^\top$. Consequently, $\operatorname{Im} f^\top \subset \operatorname{Im}\Pi$, and a fortiori $\operatorname{Im} f^\top b \subset \operatorname{Im}\Pi$. Thus, $\operatorname{Im} D^{-1}\Pi \subset \operatorname{Im}\Pi$, namely, $\operatorname{Im}\Pi$ is stable by D^{-1}.

Note that $\operatorname{Im}\Pi = \operatorname{Im} b^\top$, so its dimension is at most the rank of b which is at most r.

Unwinding the change of variables with $\grave{\rho}^{1/2}$, we see that $\Pi_E := \grave{\rho}^{-1/2}\Pi\grave{\rho}^{1/2}$ is an $L^2(\rho)$-orthogonal projector, whose image $E := \operatorname{Im}\Pi_E$ is stable by Δ^{-1}, and such that $F^\top B\rho$ is equal to $\Delta^{-1}\Pi_E$. Thus $F^\top B\rho$ is equal to Δ^{-1} on E and to 0 on its $L^2(\rho)$-orthogonal.

($\Leftarrow$). Let E be a stable subspace of M, of dimension at most r, such that $F^\top B\rho$ is equal to M on E and to 0 on the $L^2(\rho)$-orthogonal E' of E.

Let Π_E be the $L^2(\rho)$-orthogonal projector onto E. Since E is stable by M, we have $M\Pi_E = \Pi_E M\Pi_E$. Moreover, the condition that $F^\top B\rho$ is equal to M on E and to 0 on E' is equivalent to saying that $F^\top B\rho = M\Pi_E$.

Define $H = \grave{\rho}^{1/2}E$ and $H' = \grave{\rho}^{1/2}E'$, so that H and H' are orthogonal in the usual sense. Note that H' is stable by $\grave{\rho}^{1/2}M\grave{\rho}^{-1/2} = D^{-1}$. The property $M\Pi_E = \Pi_E M\Pi_E$ rewrites as $D^{-1}\Pi = \Pi D^{-1}\Pi$ with Π the orthogonal projector onto H. Moreover, $F^\top B\rho = M\Pi_E$ rewrites as $f^\top b = D^{-1}\Pi$.

Let b be any matrix such that $\operatorname{Im} b^\top = H$ (e.g., made of a basis of H padded with 0's up to dimension r). Let b' be its Moore–Penrose pseudoinverse. Define $f := b'(D^{-1})^\top$. Then $bb^\top fD^\top = bb^\top b' = b$ so that the first fixed point equation $0 = b - bb^\top fD^\top$ is satisfied.

Since $D^{-1}\Pi = \Pi D^{-1}\Pi$, we have $\Pi(D^{-1})^\top = \Pi(D^{-1})^\top\Pi$, thus $b'\Pi(D^{-1})^\top = b'\Pi(D^{-1})^\top\Pi$. As above, $\Pi = b^\top b'$. Therefore, $b'b^\top b'(D^{-1})^\top = b'b^\top b'(D^{-1})^\top\Pi$. Now, the Moore–Penrose pseudoinverse of $b^\top$ satisfies $b'b^\top b' = b'$. Thus $b'(D^{-1})^\top = b'(D^{-1})^\top\Pi$, namely, $f = f\Pi$. Since $f^\top b = D^{-1}\Pi$ this rewrites as $f = fDf^\top b$, namely, the second fixed point equation is satisfied.

This proves that $F^\top B\rho$ satisfies the fixed point equations. Moreover, given E, many such fixed points exist: a fixed point can be built using any matrix B which spans E, then defining F from B. □

Proof of Proposition 37. Denoting

$$f := F\grave{\rho}^{1/2}, \qquad b := B\grave{\rho}^{1/2}, \qquad D := (\grave{\rho}^{1/2}\Delta\grave{\rho}^{-1/2})^\top, \tag{286}$$

the fixed point equations $\delta F = 0$ and $\delta B = 0$ for the backward TD updates (280) rewrite as

$$0 = f - f f^\top b D^\top, \qquad 0 = b - b D b^\top f. \tag{287}$$

These are the same equations as (285) with f and b swapped. Therefore, the same proof yields the following. Let Π be the orthogonal projector on $\operatorname{Im} f^\top$, we obtain that $b^\top f$ is equal to $D^{-1}\Pi$, and that $\operatorname{Im}\Pi$ is stable by D^{-1}.

Equivalently, $f^\top b$ is equal to $\Pi(D^{-1})^\top$ and $\operatorname{Ker}\Pi$ is stable by $(D^{-1})^\top$.

Set $\Pi_E := \grave{\rho}^{-1/2}\Pi\grave{\rho}^{1/2}$. Then $\operatorname{Ker}\Pi_E$ is stable by $\grave{\rho}^{-1/2}(D^{-1})^\top\grave{\rho}^{1/2}$. Moreover, Π_E is an $L^2(\rho)$-orthogonal projector.

Here $(D^{-1})^\top = \grave{\rho}^{1/2} M \grave{\rho}^{-1/2}$. Therefore, $\operatorname{Ker}\Pi_E$ is stable by M. Moreover, the relationship $f^\top b = \Pi(D^{-1})^\top$ rewrites as $F^\top B\rho = \Pi_E M$. □

Lemma 40. *Let ρ be an invariant probability distribution of P. Then for any vector f,*

$$f^\top\grave{\rho}(\mathrm{Id} - \gamma P)f = (1-\gamma)\mathbb{E}_{s\sim\rho}[f(s)^2] + \frac{\gamma}{2}\mathbb{E}_{s\sim\rho,\, s'\sim P(s,\mathrm{d}s')}[(f(s) - f(s'))^2] \tag{288}$$

and in particular, this is positive for any nonzero f.

Proof. The proof is left as an exercise. The second term is known as the Dirichlet form of a Markov chain [DSC96], and plays an important role in the convergence analysis of TD in some situations [Oll18]. □

Proof of Proposition 39. Denoting again

$$f := F\grave{\rho}^{1/2}, \qquad b := B\grave{\rho}^{1/2}, \qquad D := \grave{\rho}^{1/2}\Delta\grave{\rho}^{-1/2}, \tag{289}$$

then the fixed point equations $\delta F = 0$ and $\delta B = 0$ in (279)–(280) for backward TD for F and forward TD for B rewrite as

$$0 = f - f D f^\top b, \qquad 0 = b - b D^\top b^\top f. \tag{290}$$

Moreover, if ρ is an invariant probability distribution of the Markov process, then Lemma 40 implies

$$x^\top D x > 0 \tag{291}$$

for any nonzero vector x.

We will work on f, b, and D; the statements on $F^\top B\grave{\rho}$ follow by unwinding the change of variables.

($\Leftarrow$). Assume that X is an operator with image H and kernel $H^\perp$, such that X and $\Pi D\Pi$ are inverses as operators from H to H, with Π the orthogonal projector onto H. Let O be any isometry from H to $\mathbb{R}^r$. Set $f = O\Pi$ and $b = OX$, so that $f^\top b = \Pi X = X$. Note that $\operatorname{Im} f^\top = \operatorname{Im} b^\top = H$. Moreover, $f\Pi = f$, and $b\Pi = b$ because $X\Pi = X$. So $f^\top b$ and $\Pi D\Pi$

are inverses as operators on H. Therefore, for any $x, y \in H$, we have $x^\top(\Pi D\Pi)(f^\top b)y = x^\top y$ and $x^\top(f^\top b)(\Pi D\Pi)y = x^\top y$. Since x and y lie in H and $\operatorname{Im} f^\top = H$, we can write $x = f^\top z$ and $y = \Pi z'$, with z and z' not necessarily in H. Then $x^\top(\Pi D\Pi)(f^\top b)y = z^\top f\Pi D\Pi f^\top b\Pi z' = x^\top y = z^\top f\Pi z'$ for any z and z' in the whole space. Since $f\Pi = f$ and $b\Pi = b$ this rewrites as $z^\top fDf^\top bz' = z^\top fz'$ for any z and z' in the whole space. Therefore, $fDf^\top b = f$, namely, the first fixed point equation is satisfied. The second fixed point equation $b^\top = f^\top bDb^\top$ is similar, starting with $x^\top(f^\top b)(\Pi D\Pi)y = x^\top y$ and substituting $x = \Pi z$, $y = b^\top z'$.

($\Rightarrow$). Assume that the two fixed point equations are satisfied. Since $f = fDf^\top b$ we have $\operatorname{Ker} b \subset \operatorname{Ker} f$. Using the other equation proves that $\operatorname{Ker} f \subset \operatorname{Ker} b$, thus f and b have the same kernel. Therefore $f^\top$ and $b^\top$ have the same image. Let H be this image, and let Π be the orthogonal projector onto H.

The second fixed point equation is $b^\top = f^\top bDb^\top$. Thus $H = \operatorname{Im} b^\top = \operatorname{Im} f^\top bDb^\top \subset \operatorname{Im} f^\top b \subset \operatorname{Im} f^\top = H$. Therefore the image of $f^\top b$ is H. Likewise, the first equation $f = fDf^\top b$ implies that the kernel of $f^\top b$ is $H^\perp$.

Let us prove that $f^\top b$ and $\Pi D\Pi$ are inverses as operators from H to H. This is equivalent to proving that for any $x, y \in H$, we have $x^\top(\Pi D\Pi)(f^\top b)y = x^\top y$ and $x^\top(f^\top b)(\Pi D\Pi)y = x^\top y$. Since $\operatorname{Im} f^\top = H$, we can write $x = f^\top z$. Hence $x^\top(\Pi D\Pi)(f^\top b)y = z^\top f\Pi D\Pi f^\top by$. Since $\operatorname{Im} f^\top = H$ we have $\Pi f^\top = f^\top$ and $f\Pi = \Pi$, so $z^\top f\Pi D\Pi f^\top by = z^\top fDf^\top by$, which is $z^\top fy = x^\top y$ by the first fixed point equation. Therefore, we have $x^\top(\Pi D\Pi)(f^\top b)y = x^\top y$. For the other equality, since $\operatorname{Im} b^\top = H$, we can write $y = b^\top z$. Hence $x^\top(f^\top b)(\Pi D\Pi)y = x^\top f^\top b\Pi D\Pi b^\top z$. Again, $\Pi b^\top = b^\top$ and $b\Pi = b$, so $x^\top f^\top b\Pi D\Pi b^\top z = x^\top f^\top bDb^\top z$. By the second fixed point equation, this is $x^\top b^\top z = x^\top y$. This proves the claim.

Finally, let us turn to the statement about realizing any subspace E this way. Let E be an arbitrary subspace of $\mathbb{R}^{\#S}$, of dimension r. Let $H := \hat{\rho}^{1/2}E$. Let Π be the *rectangular* orthogonal projector onto H (namely, its range is H only; its transpose $\Pi^\top$ is the inclusion map from H to $\mathbb{R}^{\#S}$), and let A be any invertible linear map from H to $\mathbb{R}^r$. Set $f := A\Pi$.

First, we claim that the square matrix $fDf^\top$ is invertible. Indeed, assume there exists a vector $x \in \mathbb{R}^r$ such that $fDf^\top x = 0$. Then $x^\top fDf^\top x = 0$. By (291) this implies $f^\top x = 0$, or $\Pi^\top A^\top x = 0$. Since $A^\top x \in H$ we have $\Pi^\top A^\top x = A^\top x$, so $A^\top x = 0$. But since A is invertible this implies $x = 0$. Therefore, $fDf^\top$ is invertible.

Then we set $b := (fDf^\top)^{-1}f$. Let us check that the fixed point equations are satisfied. Obviously, $f = fDf^\top b$, so the first fixed point equation holds. For the second one, we have

$$bD^\top b^\top f = (fDf^\top)^{-1}fD^\top f^\top(fD^\top f^\top)^{-1}f = (fDf^\top)^{-1}f = b. \tag{292}$$

Therefore, the second equation holds as well, so that f and b are a fixed point of the bf-FB algorithm. □

F The FB Representation and Bellman–Newton

F.1 The FB Representation Coincides With Bellman–Newton for Symmetric P

Here we prove that the tabular FB updates (all four versions) coincide with the Bellman–Newton update in the small-learning-rate (continuous-time) limit, on-policy, with suitable initializations, and provided that the transition matrix P of the Markov process is symmetric.

On a finite space, if P is symmetric then the uniform measure is an invariant distribution of the process. Therefore, being on-policy means that ρ is uniform.

Theorem 41 (The FB update is Bellman–Newton for symmetric P). *Assume that the state space S is finite, and that the transition matrix P is symmetric.*

Let ρ be the uniform distribution on S, which is invariant under the Markov process. Let $\grave{\rho} = \frac{1}{\#S}\,\mathrm{Id}$ be the diagonal matrix with entries ρ.

Let F_0 and B_0 be two $r \times \#S$-matrices Consider the continuous-time equation

$$\frac{\mathrm{d}F_t}{\mathrm{d}t} = \delta F, \qquad \frac{\mathrm{d}B_t}{\mathrm{d}t} = \delta B \tag{293}$$

initialized at F_0 and B_0, where δF and δB are the tabular FB updates of Proposition 34, computed at F_t and B_t. Any of the four variants ff-FB, fb-FB, bf-FB, bb-FB of Proposition 34 may be used.

Assume that $F_0 = B_0$. For the ff-FB, fb-FB, and bb-FB variants, furthermore assume that Δ commutes with $F_0^\top B_0$ (e.g., initialize to $F_0 = B_0 = \mathrm{Id}$).

Let $M_t := F_t^\top B_t \grave{\rho}$ be the resulting estimate of the successor state matrix. Then M_t evolves according to the Bellman–Newton update

$$\frac{\mathrm{d}M_t}{\mathrm{d}t} = \eta M_t - \eta M_t(\mathrm{Id} - \gamma P)M_t \tag{294}$$

with learning rate $\eta = 2/\#S$.

This bears some discussion with respect to feature learning. As discussed elsewhere, the Bellman–Newton update does not learn features (the kernel and image of M_t are preserved), and neither does the bf-FB variant in the case of uniform ρ. All other variants (ff-FB, bf-FB, bb-FB) learn features by changing the kernel of F or B, and have fixed points corresponding to various eigenspaces of Δ (Appendix E). Thus, how is it possible that these FB updates coincide with Bellman–Newton? This is where the assumption $[\Delta, F_0^\top B_0] = 0$ comes in: this commutation occurs, broadly speaking, if $F_0^\top B_0]$ is already aligned with the eigenspaces of Δ. In that case, the FB updates will coincide with Bellman–Newton and enjoy its improved asymptotic convergence. If not, they will avoid the shortcoming of Bellman–Newton and learn features, stabilizing to such an alignment.

Proof. We abbreviate F_t' for $\mathrm{d}F_t/\,\mathrm{d}t$ and likewise for all other quantities.

According to Proposition 34, the forward-TD equations for F and B are

$$F_t' = B_t\dot\rho - B_t\dot\rho B_t^\top F_t\Delta^\top\dot\rho, \qquad B_t' = F_t\dot\rho - F_t\dot\rho\Delta F_t^\top B_t\dot\rho \tag{295}$$

and the backward-TD equations are

$$F_t' = B_t\dot\rho - B_t(\dot\rho\Delta)^\top B_t^\top F_t\dot\rho, \qquad B_t' = F_t\dot\rho - F_t\dot\rho F_t^\top B_t\dot\rho\Delta \tag{296}$$

Here we have $\dot\rho = \frac{1}{\#S}\,\mathrm{Id}$. Moreover, since P is symmetric, we have $\Delta = \Delta^\top$.

Let us start with the bf-FB variant (backward-TD on F and forward-TD on B). In that case, the evolution equations are symmetric between F and B, because $\Delta = \Delta^\top$. Therefore, if $F = B$ at startup then $F = B$ at all times. Thus, we have $M_t = F_t^\top F_t\dot\rho$. Since $\dot\rho$ is proportional to Id, it commutes with all other terms. Thus, using $F_t = B_t$ and $\Delta = \Delta^\top$, we find

$$\begin{aligned} M_t' &= (F_t')^\top F_t\dot\rho + F_t^\top F_t'\dot\rho && (297)\\ &= 2F_t^\top F_t\dot\rho^2 - 2F_t^\top F_t\Delta F_t^\top F_t\dot\rho^3 && (298)\\ &= 2M_t\dot\rho - 2M_t\Delta M_t\dot\rho && (299)\\ &= \frac{2}{\#S}(M_t - M_t\Delta M_t) && (300) \end{aligned}$$

as $\dot\rho = \frac{1}{\#S}\,\mathrm{Id}$. This is the Bellman–Newton update.

In the other cases there is one more argument, after which the computation is similar. At startup, by assumption we have $F = B$ and Δ commutes with $F^\top B$. Assume that, at some particular time t, we have $F_t = B_t$ and Δ commutes with $F_t^\top B_t$. Then, since $\Delta = \Delta^\top$ and $\dot\rho$ commutes with everything, all the updates of F and B at that time t amount to just

$$F_t' = F_t\dot\rho - F_tF_t^\top F_t\Delta\dot\rho^2. \tag{301}$$

Therefore, at that time t, the derivative of the commutator between Δ and $F_t^\top B_t$ is

$$\begin{aligned} [\Delta, F_t^\top B_t]' &= [\Delta, (F_t^\top F_t)'] && (302)\\ &= [\Delta, 2F_t^\top F_t\dot\rho - F_t^\top F_tF_t^\top F_t\Delta\dot\rho^2 - \Delta F_t^\top F_tF_t^\top F_t\dot\rho^2] && (303)\\ &= 0 && (304) \end{aligned}$$

since Δ commutes with $F_t^\top F_t$ and $\dot\rho$ commutes with everything.

Thus, if at some time t one has $F_t = B_t$ and Δ commutes with $F_t^\top B_t$, then F_t and B_t have the same derivative at time t, and the derivative of the commutator of Δ and $F_t^\top B_t$ is 0. Therefore, if these conditions hold at startup, they hold at all times t. In that case, the evolution equations become identical to the bf-FB case, and the conclusion holds as above. □

F.2 The BN-FB update

Here we introduce Bellman–Newton FB (BN-FB), a variant of the FB updates that coincides with Bellman–Newton in the tabular case for arbitrary P, not only symmetric P. It is compatible with sampling, without the sampling issues of the Bellman–Newton update, and admits a parametric version.

However, it still keeps the main shortcoming of the Bellman–Newton update, namely, that the kernel and image of the estimate of M are fixed (no features are learned).

In the tabular case, let F and B be two $r \times \#S$-matrices, and define the updates

$$\delta F^\top := F^\top - F^\top B \grave{\rho} \Delta F^\top, \qquad \delta B = B - B \grave{\rho} \Delta F^\top B \tag{305}$$

where as usual $\grave{\rho}$ is the diagonal matrix with entries ρ, and $\Delta = \mathrm{Id} - \gamma P$.

The updates with learning rate η lead to a Bellman–Newton udpate on the model $M = F^\top B \grave{\rho}$:

$$\begin{aligned} F \leftarrow F + \eta\, \delta F, \qquad B \leftarrow B + \eta\, \delta B \\ \Rightarrow \qquad M \leftarrow (1+\eta)M - \eta M \Delta M + O(\eta^2) \end{aligned} \tag{306}$$

at first order in η. In particular, the continuous-time dynamics will be an exact Bellman–Newton update.

The parametric version is obtained as before, by approximating these ideal updates in ρ-norm, and by sampling Δ using $\grave{\rho}\Delta = \mathbb{E}_{s\sim\rho,\, s'\sim P(\mathrm{d}s'|s)}(\mathbb{1}_s \mathbb{1}_s^\top - \gamma \mathbb{1}_s \mathbb{1}_{s'}^\top)$. Letting θ_F and θ_B be the parameters of F and B respectively, this yields

$$\mathbb{E}_{s_1\sim\rho}\, (\partial_{\theta_F} F(s_1)^\top)(F(s_1) - D^\top F(s_1)) \tag{307}$$

for the update of the parameters of F, and

$$\mathbb{E}_{s_1\sim\rho}\, (\partial_{\theta_B} B(s_1)^\top)(B(s_1) - DB(s_1)) \tag{308}$$

for the parameters of B. Here D is an $r \times r$ matrix (even for continuous states) given by

$$D := \mathbb{E}_{s\sim\rho,\, s'\sim P(s,\mathrm{d}s')}\, B(s)(\gamma F(s') - F(s))^\top. \tag{309}$$

It is possible to use a single sampled transition $s \to s'$ for D (this option requires no storage of D since the updates simplify), or to estimate D online using a moving average over a number of past transitions $s \to s'$. This second option reduces variance but introduces some bias due to old values in the moving average.

G Sampling Simplified States for s_1 and s_2

This section addresses two potential shortcomings of the parametric TD and Bellman–Newton algorithms for M.

- The parametric updates for TD and for Bellman–Newton involve sampling additional states s_1 and s_2 unrelated to the states $s \to s'$ currently visited (and actually use *every* state s_1 and s_2 for the tabular case). A simple option is to sample s_1 and s_2 at random among a dataset of past visited states. But if actual states and transitions are few, or complicated to come by, or if it is inconvenient to store many states (pure online setting), sampling additional states according to the data distribution ρ may be a limitation.

  We show that s_1 and s_2 can be sampled according to any "simple" distribution ρ_{simple}. This could help learning by making it possible to use many samples of s_1 and s_2 for each observed transition $s \to s'$, thus bringing the parametric updates closer to the tabular updates (which use every s_1 and s_2).

- Defining m_θ as a density with respect to the unknown distribution ρ may pose numerical problems: In regions where M is not small but ρ is small, this attributes a large value to m_θ, which may perturb learning.

Here, we show that using simplified states $s_1, s_2 \sim \rho_{\text{simple}}$ in the parametric updates, and defining m_θ with respect to an arbitrary reference measure ρ_{ref} on $\mathcal{S}$, just amounts to using different learning rates on different parts of the state, and different norms to define the parametric updates. Thus, these simplified algorithms still make sense; however, proper factors must be included, given in (310)–(312) below.

We consider three different measures on states: the main "data" measure $\rho(\mathrm{d}s)$ from which we obtain observations $s \to s'$, and which we do not control; a "simple", user-chosen probability measure ρ_{simple} from which we can cheaply sample states, real or synthetic (such as a uniform distribution, or a Gaussian with the same mean and variance as real states, or real states with added Gaussian noise); and a user-chosen reference measure ρ_{ref} used to parameterize M via $M(s_1, \mathrm{d}s_2) = \delta_{s_1}(\mathrm{d}s_2) + m_\theta(s_1, s_2)\, \rho_{\text{ref}}(\mathrm{d}s_2)$. The measure ρ_{ref} is not necessarily of mass 1, and may for instance be the Lebesgue measure.

We assume that the ratio $\rho_{\text{simple}}/\rho_{\text{ref}}$ is known; this is the case for instance if we take $\rho_{\text{ref}} := \rho_{\text{simple}}$, or if ρ_{ref} is the Lebesgue measure and ρ_{simple} is Gaussian. The simplest case is to use an arbitrary ρ_{simple} and set $\rho_{\text{ref}} = \rho_{\text{simple}}$: in that case all expressions are the same as before, but they correspond to different learning rates at different states depending on ρ_{simple} (since ρ_{simple} controls how we sample states), and to learning the density m_θ of M with respect to ρ_{simple} not ρ.

The parametric TD update for M becomes

$$\mathbb{E}_{s\sim\rho,\, s'\sim P(s,\mathrm{d}s'),\, s_2\sim\rho_{\mathrm{simple}}}\Bigg[\gamma\,\partial_\theta m_{\theta_t}(s,s')\,\frac{\rho_{\mathrm{simple}}(\mathrm{d}s')}{\rho_{\mathrm{ref}}(\mathrm{d}s')} + \partial_\theta m_{\theta_t}(s,s_2)\,(\gamma m_{\theta_t}(s',s_2) - m_{\theta_t}(s,s_2))\Bigg]. \quad (310)$$

The parametric update (78) for V becomes

$$(r_s + \gamma V_{\varphi_t}(s') - V_{\varphi_t}(s))\left(\partial_\varphi V_{\varphi_t}(s)\,\frac{\rho_{\mathrm{simple}}(\mathrm{d}s)}{\rho_{\mathrm{ref}}(\mathrm{d}s)} + \mathbb{E}_{s_1\sim\rho_{\mathrm{simple}}}[m_{\theta_t}(s_1,s)\,\partial_\varphi V_{\varphi_t}(s_1)]\right). \quad (311)$$

The parametric Bellman–Newton update (66) for M becomes

$$\begin{aligned}\mathbb{E}_{s_1\sim\rho_{\mathrm{simple}},\, s_2\sim\rho_{\mathrm{simple}}}\Bigg[&\gamma\,\partial_\theta m_{\theta_t}(s,s')\,\frac{\rho_{\mathrm{simple}}(\mathrm{d}s)}{\rho_{\mathrm{ref}}(\mathrm{d}s)}\frac{\rho_{\mathrm{simple}}(\mathrm{d}s')}{\rho_{\mathrm{ref}}(\mathrm{d}s')}\\ &+\gamma\, m_{\theta_t}(s_1,s)\,\partial_\theta m_{\theta_t}(s_1,s')\,\frac{\rho_{\mathrm{simple}}(\mathrm{d}s')}{\rho_{\mathrm{ref}}(\mathrm{d}s')}\\ &+(\gamma m_{\theta_t}(s',s_2) - m_{\theta_t}(s,s_2))\left(\partial_\theta m_{\theta_t}(s,s_2)\,\frac{\rho_{\mathrm{simple}}(\mathrm{d}s)}{\rho_{\mathrm{ref}}(\mathrm{d}s)} + m_{\theta_t}(s_1,s)\,\partial_\theta m_{\theta_t}(s_1,s_2)\right)\Bigg].\end{aligned} \quad (312)$$

We now justify each of these updates in turn, by deriving them in the same way as above, but using different norms and learning rates.

On the other hand, for various reasons this does not work for backward TD (even if ρ is the invariant distribution from the Markov process). Reversing time changes the parameterization of M: instead of $\mathrm{Id} + m(s_1,s_2)\rho_{\mathrm{ref}}(\mathrm{d}s_2)$ with a user-chosen factor on s_2, one gets a user-chosen factor on s_1.

Given three measures ρ_1, ρ_2, and ρ_{ref} (not necessarily of mass 1), and two measure-valued functions $M_1(s,\mathrm{d}s')$ and $M_2(s,\mathrm{d}s')$ on $\mathcal{S}$, we define the norm

$$\|M_1 - M_2\|^2_{\rho_1,\rho_2,\rho_{\mathrm{ref}}} := \iint (m_1(s,s') - m_2(s,s'))^2\,\rho_1(\mathrm{d}s)\,\rho_2(\mathrm{d}s') \quad (313)$$

where $m_1(s,s') := M_1(s,\mathrm{d}s')/\rho_{\mathrm{ref}}(\mathrm{d}s')$ is the density of M_1 with respect to ρ_{ref} (if it exists; if not, the norm is infinite), and likewise for M_2. This generalizes the norm (1).

Theorem 42 (TD for successor states with function approximation and simple sample states). *Maintain a parametric model of M via $M_{\theta_t}(s_1,\mathrm{d}s_2) = \delta_{s_1}(\mathrm{d}s_2) + m_{\theta_t}(s_1,s_2)\rho_{\mathrm{ref}}(\mathrm{d}s_2)$, with θ_t the value of the parameter at step t, and with m_θ some smooth family of functions over pairs of states.*

Define a target update of M via the Bellman equation, $M^{\mathrm{tar}} := \mathrm{Id} + \gamma P M_{\theta_t}$. Define the loss between M and M^{tar} via $J(\theta) := \frac{1}{2}\left\|M_\theta - M^{\mathrm{tar}}\right\|^2_{\rho,\rho_{\mathrm{simple}},\rho_{\mathrm{ref}}}$ using the norm (313).

Then the update (310) is equal to the gradient step $-\partial_\theta J(\theta)_{|\theta=\theta_t}$.

For the updates of V and M, we will assume that we learn the implicit Markov process $\hat{P}$ and $\hat{R}$ with state-dependent learning rates inversely proportional to ρ_{ref}. (The standard case has $1/n_s$ for the learning rates; since $n_s \sim t\rho_s$, this produces learning rates inversely proportional to ρ.)

Namely, let $\eta_t \to 0$ be an overall learning rate schedule. Upon observing a transition $s \to s'$ with reward r_s, learn $\hat{P}$ and $\hat{R}$ via

$$\hat{P}_{ss_2} \leftarrow \hat{P}_{ss_2} + \frac{\eta_t}{\rho_{\text{ref}}(s)}(\mathbb{1}_{s_2=s'} - \hat{P}_{ss_2}) \qquad \forall s_2 \tag{314}$$

$$\hat{R}_s \leftarrow \hat{R}_s + \frac{\eta_t}{\rho_{\text{ref}}(s)}(r_s - \hat{R}_s). \tag{315}$$

Thus, different states learn at different speeds, but this still converges to the true values when $t \to \infty$.

Theorem 43 (Value function update via successor states with simple sample states). *Consider the empirical estimates $\hat{P}$ and $\hat{R}$ of a finite Markov reward process. Let $s \to s'$ be the transition in the Markov process observed at step t, with reward r_s. Let δV be the update of the value function $(\mathrm{Id} - \gamma\hat{P})^{-1}\hat{R}$ of the estimated process when $\hat{P}$ and $\hat{R}$ are updated by (314)–(315).*

Given a parametric model V_φ of the value function, define a target update of V via $V^{\text{tar}} := V_{\varphi_t} + \delta V$ with φ_t the parameter at step t. Define the loss between V and V^{tar} via $J^V(\varphi) := \frac{1}{2}\left\|V_\varphi - V^{\text{tar}}\right\|^2_{L^2(\rho_{\text{simple}})}$. Assume $\hat{M} = (\mathrm{Id} - \gamma\hat{P})^{-1}$ is given by (16).

Then, up to $O(\eta_t^2)$, the gradient step $-\partial_\varphi J^V(\varphi)_{\varphi=\varphi_t}$ is η_t times (311).

Theorem 44 (Successor states via online inversion, with function approximation and simple sample states). *Maintain a parametric model of M via $M_{\theta_t}(s_1, \mathrm{d}s_2) = \delta_{s_1}(\mathrm{d}s_2) + m_{\theta_t}(s_1, s_2)\rho_{\text{ref}}(\mathrm{d}s_2)$, with θ_t the value of the parameter at step t, and with m_θ some smooth family of functions over pairs of states.*

Let $s \to s'$ be the transition in the Markov process observed at step t, with reward r_s. Let δM be the update of $(\mathrm{Id} - \hat{P})^{-1}$ corresponding to the update (314) of $\hat{P}$.

Define a target update of M by $M^{\text{tar}} := M_{\theta_t} + \delta M$. Define the loss between M and M^{tar} via $J(\theta) := \frac{1}{2}\left\|M_\theta - M^{\text{tar}}\right\|_{\rho_{\text{simple}},\rho_{\text{simple}},\rho_{\text{ref}}}$ using the norm (313).

Then, up to $O(\eta_t^2)$ the gradient step $-\partial_\theta J(\theta)_{|\theta=\theta_t}$ is η_t times (312).

The proofs of these theorems are identical to their analogues with a single measure ρ, up to tracking where ρ_{simple} and ρ_{ref} appear instead of ρ at suitable places; they are omitted.

H Formal Approach to Theorem 21 for Continuous Environments

Contrary to TD on M, for Theorem 21, we have defined the update for a single transition $s \to s'$. The resulting parametric update makes sense in any state space. But strictly speaking, this restricts the statement of Theorem 21 to discrete spaces, since it is defined via the tabular update (61) which is defined only in discrete spaces.

For TD on M in general spaces (Theorem 6), we directly defined the Bellman operator on any space; the Bellman operator does not depend on a single transition $s \to s'$, but it updates all states s at once.

It is possible to proceed analogously for Theorem 21: this provides a rigorous proof of Theorem 21 for general state spaces, in expectation over the transition $s \to s'$.

We first have to define the analogue of the Bellman operator. We do this by going back to discrete states and averaging the updates δM and δV over transitions $s \to s'$. Averaging (61) and (62) yields (omitting again the $o(1/n_s) = o(1/t)$ terms)

$$\mathbb{E}_{s\sim\rho,\, s'\sim P_{ss'}}\, \delta M_{s_1 s_2} = \sum_s \sum_{s'} \frac{\rho_s}{n_s} P_{ss'} \hat{M}_{s_1 s} (\mathbb{1}_{s_2=s} + \gamma \hat{M}_{s' s_2} - M_{s s_2}) \tag{316}$$

and

$$\mathbb{E}_{s\sim\rho,\, s'\sim P_{ss'}}\, \delta V_{s_1} = \sum_s \sum_{s'} \frac{\rho_s}{n_s} P_{ss'} (r_s + \gamma \hat{V}_{s'} - \hat{V}_s) \hat{M}_{s_1 s}. \tag{317}$$

Once more, since $n_s \sim t\rho_s$ when $s \to \infty$, we have $\frac{\rho_s}{n_s} = 1/t + o(1/t)$. Thus, up to $o(1/t)$ terms, the above rewrite as

$$\mathbb{E}_{s\sim\rho,\, s'\sim P_{ss'}}\, \delta M_{s_1 s_2} = \tfrac{1}{t} \sum_s \sum_{s'} P_{ss'} \hat{M}_{s_1 s} (\mathbb{1}_{s_2=s} + \gamma \hat{M}_{s' s_2} - M_{s s_2}) \tag{318}$$

$$\mathbb{E}_{s\sim\rho,\, s'\sim P_{ss'}}\, \delta V_{s_1} = \tfrac{1}{t} \sum_s \sum_{s'} P_{ss'} (r_s + \gamma \hat{V}_{s'} - \hat{V}_s) \hat{M}_{s_1 s}. \tag{319}$$

Since $\sum_{s'} P_{ss'} = 1$ and $\sum_{s,s'} \hat{M}_{s_1 s} P_{ss'} \hat{M}_{s' s_2} = (\hat{M} P \hat{M})_{s_1 s_2}$ and likewise $\sum_s \hat{M}_{s_1 s} \hat{M}_{s s_2} = (\hat{M}^2)_{s_1 s_2}$, the update of M rewrites as

$$\mathbb{E}_{s\sim\rho,\, s'\sim P_{ss'}}\, \delta M = \tfrac{1}{t} (\hat{M} + \gamma \hat{M} P \hat{M} - \hat{M}^2). \tag{320}$$

Likewise, for the update of $\hat{V}$, since $\mathbb{E} r_s = R_s$ and $\sum_{s'} P_{ss'} (\gamma \hat{V}_{s'} - \hat{V}_s) = (\gamma P \hat{V} - \hat{V})_s$, we have

$$\mathbb{E}_{s\sim\rho,\, s'\sim P_{ss'}}\, \delta V = \tfrac{1}{t} \hat{M} (R + \gamma P \hat{V} - \hat{V}). \tag{321}$$

Thus, in the continuous case, we can define target updates at step t by

$$M^{\text{tar}} := M_{\theta_t} + \frac{1}{t} (M_{\theta_t} - M_{\theta_t} (\mathrm{Id} - \gamma P) M_{\theta_t}) \tag{322}$$

(well-defined for continuous states as an operator on functions) and

$$V^{\mathrm{tar}} := V_{\varphi_t} + \frac{1}{t} M_{\theta_t}(R + \gamma P V_{\varphi_t} - V_{\varphi_t}) \tag{323}$$

and define, as before, the losses

$$J(\theta) := \frac{1}{2} \left\| M_\theta - M^{\mathrm{tar}} \right\|_\rho^2 \tag{324}$$

and

$$J^V(\varphi) := \frac{1}{2} \left\| V_\varphi - V^{\mathrm{tar}} \right\|_{L^2(\rho)}^2 . \tag{325}$$

From now on we only work with M, as the computation for V is similar but simpler.

As in the proof of Theorem 6, by definition of $J(\theta)$ and of the norm $\|\cdot\|_\rho$, we have

$$J(\theta) = \frac{1}{2} \iint j_\theta(s_1, s_2)^2 \, \rho(\mathrm{d}s_1)\rho(\mathrm{d}s_2) \tag{326}$$

and

$$\partial_\theta J(\theta) = \iint j_\theta(s_1, s_2) \, \partial_\theta j_\theta(s_1, s_2) \rho(\mathrm{d}s_1)\rho(\mathrm{d}s_2) \tag{327}$$

where

$$j_\theta(s_1, s_2) := (M^{\mathrm{tar}}(s_1, \mathrm{d}s_2) - M_\theta(s_1, \mathrm{d}s_2))/\rho(\mathrm{d}s_2) \tag{328}$$

Since M^{tar} does not depend on θ (it depends on θ_t, but we optimize with respect to θ for fixed M^{tar}), we have

$$\partial_\theta j_\theta(s_1, s_2) = \partial_\theta \left(-\frac{M_\theta(s_1, \mathrm{d}s_2)}{\rho(\mathrm{d}s_2)} \right) = -\partial_\theta m_\theta(s_1, s_2) \tag{329}$$

in the parameteriation $M_\theta(s_1, \mathrm{d}s_2) = \mathrm{Id} + m_\theta(s_1, s_2)\rho(\mathrm{d}s_2)$.

Consequently, by (327), (328) and (329), at $\theta = \theta_t$ we have

$$-\partial_\theta J(\theta)_{|\theta=\theta_t} = \iint \left(\frac{M^{\mathrm{tar}}(s_1, \mathrm{d}s_2) - M_{\theta_t}(s_1, \mathrm{d}s_2)}{\rho(\mathrm{d}s_2)} \right) \partial_\theta m_{\theta_t}(s_1, s_2) \rho(\mathrm{d}s_1)\rho(\mathrm{d}s_2) \tag{330}$$

$$= \iint \left(M^{\mathrm{tar}}(s_1, \mathrm{d}s_2) - M_{\theta_t}(s_1, \mathrm{d}s_2) \right) \partial_\theta m_{\theta_t}(s_1, s_2) \rho(\mathrm{d}s_1) \tag{331}$$

Define $A_\theta := M_\theta - \mathrm{Id}$. By definition of the parameterization $M_\theta(s_1, \mathrm{d}s_2) = \mathrm{Id} + m_\theta(s_1, s_2)\rho(\mathrm{d}s_2)$, we have

$$A_\theta(s_1, \mathrm{d}s_2) = m_\theta(s_1, s_2)\rho(\mathrm{d}s_2). \tag{332}$$

By a direct but tedious substitution of $M_{\theta_t} = \mathrm{Id} + A_{\theta_t}$ in the expression (322) for M^{tar}, we find

$$M^{\mathrm{tar}} - M_{\theta_t} = \tfrac{1}{t}(\gamma P + \gamma A_{\theta_t} P + \gamma P A_{\theta_t} - A_{\theta_t} + A_{\theta_t} \gamma P A_{\theta_t} - A_{\theta_t}^2) \tag{333}$$

as operators, with the product of operator denoting composition. (For instance, for a function f, the operator PA acts by $(PAf)(s) = \int P(s, \mathrm{d}s')A(s', \mathrm{d}s_2)f(s_2)$.)

Let us study the contributions of all the terms of $M^{\mathrm{tar}} - M_{\theta_t}$ to the gradient step (331). The γP term provides a contribution

$$\iint \gamma P(s_1, \mathrm{d}s_2)\, \partial_\theta m_{\theta_t}(s_1, s_2)\rho(\mathrm{d}s_1) = \gamma \mathbb{E}_{s\sim\rho,\, s'\sim P(s,\mathrm{d}s')}\, \partial_\theta m_{\theta_t}(s, s'). \quad (334)$$

Next, by (332) we have

$$(A_{\theta_t}P)(s_1, \mathrm{d}s_2) = \int A_{\theta_t}(s_1, \mathrm{d}s)P(s, \mathrm{d}s_2) \quad (335)$$

$$= \int m_{\theta_t}(s_1, s)\rho(\mathrm{d}s)P(s, \mathrm{d}s_2) \quad (336)$$

and therefore, the $\gamma A_{\theta_t}P$ term provides a contribution

$$\begin{aligned}\iint \gamma(A_{\theta_t}P)(s_1, \mathrm{d}s_2)\, \partial_\theta m_{\theta_t}(s_1, s_2)\rho(\mathrm{d}s_1) \\ = \gamma \iiint m_{\theta_t}(s_1, s)\rho(\mathrm{d}s)P(s, \mathrm{d}s_2)\, \partial_\theta m_{\theta_t}(s_1, s_2)\rho(\mathrm{d}s_1) \\ = \gamma \mathbb{E}_{s_1\sim\rho,\, s\sim\rho,\, s'\sim P(s,\mathrm{d}s')}[m_{\theta_t}(s_1, s)\, \partial_\theta m_{\theta_t}(s_1, s')]. \quad (337)\end{aligned}$$

Next, by (332) we have

$$(PA_{\theta_t})(s, \mathrm{d}s_2) = \int P(s, \mathrm{d}s')A_{\theta_t}(s', \mathrm{d}s_2) = \mathbb{E}_{s'\sim P(s,\mathrm{d}s')}\, m_{\theta_t}(s', s_2)\rho(\mathrm{d}s_2) \quad (338)$$

and therefore, the γPA_{θ_t} term provides a contribution

$$\iint \gamma(PA_{\theta_t})(s_1, \mathrm{d}s_2)\, \partial_\theta m_{\theta_t}(s_1, s_2)\rho(\mathrm{d}s_1) = \gamma \mathbb{E}_{s\sim\rho,\, s'\sim P(s,\mathrm{d}s'),\, s_2\sim\rho}\, m_{\theta_t}(s', s_2)\, \partial_\theta m_{\theta_t}(s, s_2) \quad (339)$$

and by (332), the term $-A_{\theta_t}$ provides a contribution

$$-\iint A_{\theta_t}(s_1, \mathrm{d}s_2)\, \partial_\theta m_{\theta_t}(s_1, s_2)\rho(\mathrm{d}s_1) = -\mathbb{E}_{s\sim\rho,\, s_2\sim\rho}\, m_{\theta_t}(s, s_2)\, \partial_\theta m_{\theta_t}(s, s_2). \quad (340)$$

Next, the term $-A_{\theta_t}^2$ provides a contribution

$$\begin{aligned}-\iint (A_{\theta_t}^2)(s_1, \mathrm{d}s_2)\, \partial_\theta m_{\theta_t}(s_1, s_2)\rho(\mathrm{d}s_1) \\ = -\iiint A_{\theta_t}(s_1, \mathrm{d}s)A_{\theta_t}(s, \mathrm{d}s_2)\, \partial_\theta m_{\theta_t}(s_1, s_2)\rho(\mathrm{d}s_1) \\ = -\iiint m_{\theta_t}(s_1, s)\rho(\mathrm{d}s)m_{\theta_t}(s, s_2)\rho(\mathrm{d}s_2)\, \partial_\theta m_{\theta_t}(s_1, s_2)\rho(\mathrm{d}s_1) \\ = -\mathbb{E}_{s\sim\rho,\, s_1\sim\rho,\, s_2\sim\rho}\, m_{\theta_t}(s_1, s)m_{\theta_t}(s, s_2)\, \partial_\theta m_{\theta_t}(s_1, s_2). \quad (341)\end{aligned}$$

For the final tem $\gamma A_{\theta_t} P A_{\theta_t}$, observe that

$$(A_{\theta_t} P A_{\theta_t})(s_1, \mathrm{d}s_2) = \iint A_{\theta_t}(s_1, \mathrm{d}s) P(s, \mathrm{d}s') A_{\theta_t(s', \mathrm{d}s_2)} \tag{342}$$

$$= \iint m_{\theta_t}(s_1, s) \rho(\mathrm{d}s) P(s, \mathrm{d}s') m_{\theta_t(s', s_2)} \rho(\mathrm{d}s_2) \tag{343}$$

$$= \mathbb{E}_{s \sim \rho,\, s' \sim P(s, \mathrm{d}s')}\, m_{\theta_t}(s_1, s) m_{\theta_t}(s', s_2) \rho(\mathrm{d}s_2) \tag{344}$$

and therefore, the contribution of the term $\gamma A_{\theta_t} P A_{\theta_t}$ is

$$\gamma \iint (A_{\theta_t} P A_{\theta_t})(s_1, \mathrm{d}s_2)\, \partial_\theta m_{\theta_t}(s_1, s_2) \rho(\mathrm{d}s_1)$$
$$= \gamma \mathbb{E}_{s_1 \sim \rho,\, s_2 \sim \rho,\, s \sim \rho,\, s' \sim P(s, \mathrm{d}s')}\, m_{\theta_t}(s_1, s) m_{\theta_t(s', s_2)}\, \partial_\theta m_{\theta_t}(s_1, s_2). \tag{345}$$

Collecting everything, we find

$$\begin{aligned} -\partial_\theta J(\theta)_{|\theta=\theta_t} = \mathbb{E}_{s_1 \sim \rho,\, s_2 \sim \rho,\, s \sim \rho,\, s' \sim P(s, \mathrm{d}s')} \big[& \gamma \partial_\theta m_{\theta_t}(s, s') + \gamma m_{\theta_t}(s_1, s)\, \partial m_{\theta_t}(s_1, s') \\ & + \gamma m_{\theta_t}(s', s_2)\, \partial m_{\theta_t}(s, s_2) - m_{\theta_t}(s, s_2)\, \partial m_{\theta_t}(s, s_2) \\ - m_{\theta_t}(s_1, s) m_{\theta_t}(s, s_2)\, \partial_\theta m_{\theta_t}(s_1, s_2) & + \gamma m_{\theta_t}(s_1, s) m_{\theta_t}(s', s_2)\, \partial_\theta m_{\theta_t}(s_1, s_2) \big]. \end{aligned} \tag{346}$$

This is the expectation over $s \sim \rho$, $s' \sim P(s, \mathrm{d}s')$, of the update (66). This formally proves Theorem 21 for general state spaces, in expectation over (s, s').

I Background on Singular Value Decompositions

In the text, we often work with the space of functions over S equipped with the $L^2(\rho)$ norm. Since $\rho \neq \mathrm{Id}$, we include here a reminder on how the usual notions of Euclidean vector spaces play out in non-orthonormal bases. We also include details on what constitutes a "truncated singular value decomposition".

A Euclidean vector space E is a finite-dimensional vector space equipped with a dot product; the dot product is given by some symmetric, positive definite matrix q in some basis, namely, $\langle x, y \rangle_E = x^\top q y$ for any vectors x and y.

If $A \colon E_1 \to E_2$ is a linear map between two Euclidean spaces, its adjoint A^* is the map from E_2 to E_1 such that $\langle y, Ax \rangle_{E_2} = \langle A^* y, x \rangle_{E_1}$ for any vectors $x \in E_1$ and $y \in E_2$. Expressed in bases of E_1 and E_2, its matrix is $A^* = q_1^{-1} A^\top q_2$, or just $A^\top$ if the bases are orthonormal.

Such a map A is orthogonal if $AA^* = \mathrm{Id}_{E_2}$ and $A^* A = \mathrm{Id}_{E_1}$.

The Hilbert–Schmidt norm for an operator M on a Euclidean vector space is $\mathrm{Tr}(M^* M)$ where M^* is the adjoint of M. In an orthonormal basis this is $\mathrm{Tr}(M^\top M)$ viewing M as a matrix, but in a non-orthonormal basis this is $\mathrm{Tr}(q^{-1} M^\top q M)$ where q is the matrix defining the norm in the basis.

A *singular value decomposition* of such a map A is a triplet of linear maps $U\colon \mathbb{R}^{\dim(E_2)} \to E_2$, $D\colon \mathbb{R}^{\dim(E_1)} \to \mathbb{R}^{\dim(E_2)}$ and $V\colon \mathbb{R}^{\dim(E_1)} \to E_1$ such that $A = UDV^*$, U and V are orthogonal, and D is rectangular diagonal. Equivalently, a singular value decomposition can be written as $Ax = \sum_i u_i d_i \langle v_i, x\rangle_{E_1}$ where each $d_i > 0$, the u_i's make an orthonormal family in E_2, and the v_i's make an orthonormal family in E_1 (or equivalently, an orthonormal family of linear forms on E_1 by identifying v_i with the map $x \mapsto \langle v_i, x\rangle_{E_1}$).

Definition 45 (Truncated SVD). *A linear map B is a* truncated singular value decomposition *of a linear map $A\colon E_1 \to E_2$ if there is a singular value decomposition $A = UDV^*$ of A and a rectangular diagonal matrix D' such that D' is obtained from D by replacing some elements with 0, and $B = UD'V^*$.*

Lemma 46. *A linear map $B\colon E_1 \to E_2$ is a truncated singular value decomposition of $A\colon E_1 \to E_2$ if and only if A and B are equal on $(\operatorname{Ker} B)^\perp$ and the image of $\operatorname{Ker} B$ by A is orthogonal to the image of B.*

Proof. ($\Leftarrow$) Define $E_1' = \operatorname{Ker} B$ and $E_1'' = (\operatorname{Ker} B)^\perp$ so that $E_1 = E_1' \oplus E_1''$. Let A' and A'' be the restrictions of A to E_1' and E_1'' respectively, so that $A = A' + A''$. Define B' and B'' likewise.

Since E_1' is $\operatorname{Ker} B$, we have $B' = 0$ so $B = B''$.

By assumption, A and B are equal on E_1''. Therefore, $A'' = B''$, so $B = A''$.

By assumption, the image of E_1' by A is orthogonal to the image of B. The former is $\operatorname{Im} A'$ while the latter is $\operatorname{Im} A''$. Therefore, $\operatorname{Im} A' \perp \operatorname{Im} A''$.

Consider singular value decompositions of A' and A'' as $A' = \sum_i u_i' d_i' v_i'$ and $A'' = \sum_j u_j'' d_j'' v_j''$, where the d_i' are positive real numbers, the u_i' are an orthonormal basis of $\operatorname{Im} A'$, the v_i' are an orthonormal set of linear forms on E_1', and likewise for A''. (Any zero singular values have been dropped in this decomposition.)

Since $\operatorname{Im} A' \perp \operatorname{Im} A''$, the u_i''s are orthogonal to the u_j'''s. Likewise, since the decomposition $E_1 = E_1' \oplus E_1''$ is orthogonal, the v_i''s are orthogonal to the v_j'''s as linear forms on E_1.

It follows that $\sum_i u_i' d_i' v_i' + \sum_j u_j'' d_j'' v_j''$ is a singular value decomposition of A (with the zero singular values omitted). Since $B = A''$, $\sum_j u_j'' d_j'' v_j''$ is a singular value decomposition of B, so that B is a truncated SVD of A.

($\Rightarrow$) Let $A = UDV^*$ and $B = UD'V^*$ as in Definition 45. Up to swapping rows and columns, we can assume that the nonzero entries of D and D' are located in the first rows. Let k be the number of nonzero entries in D'. Then $\operatorname{Ker} D'$ is spanned by the last $\dim(E_1) - k$ basis vectors in $\mathbb{R}^{\dim(E_1)}$, and $(\operatorname{Ker} D')^\perp$ is spanned by the first k basis vectors. Thus, by construction, D and D' coincide on $(\operatorname{Ker} D')^\perp$. Moreover, $\operatorname{Im} D'$ is spanned by the first

k basis vectors, and $D(\operatorname{Ker} D')$ is spanned by the last $\dim(E_1) - k$ basis vectors, so $\operatorname{Im} D'$ and $D(\operatorname{Ker} D')$ are orthogonal.

Since $A = UDV^*$ and $B = UD'V^*$, and since U is invertible, A and B are equal on $(\operatorname{Ker} B)^\perp$ if and only if DV^* and $D'V^*$ are equal on $(\operatorname{Ker} B)^\perp$. Since V^* is invertible, this happens if and only if D and D' are equal on $V^*((\operatorname{Ker} B)^\perp)$. Since V^* is orthogonal, the latter is $(V^*(\operatorname{Ker} B))^\perp$.

Since U and V are orthogonal, hence invertible, one has $\operatorname{Ker} B = \operatorname{Ker}(UD'V^*) = \operatorname{Ker}(D'V^*) = V(\operatorname{Ker} D')$. Hence $V^*(\operatorname{Ker} B) = \operatorname{Ker} D'$. Thus, we need D and D' to be orthogonal on $\operatorname{Ker} D'$, which we have established above.

Next, let us prove that $A(\operatorname{Ker} B) \perp \operatorname{Im} B$, namely, that $UDV^*(\operatorname{Ker} B) \perp \operatorname{Im}(UD'V^*)$. Since U is orthogonal, this is equivalent to $DV^*(\operatorname{Ker} B) \perp \operatorname{Im}(D'V^*)$. We have seen that $V^*(\operatorname{Ker} B) = \operatorname{Ker} D'$; moreover $\operatorname{Im}(D'V^*) \subset \operatorname{Im}(D')$, so it is enough to prove that $D(\operatorname{Ker} D') \perp \operatorname{Im} D'$, which we have established above. This proves the first part of the equivalence. $\square$