Global Reinforcement Learning: Beyond Linear and Convex Rewards via Submodular Semi-gradient Methods

Riccardo De Santi [*] **Manish Prajapat** [*] **Andreas Krause**

Abstract

In classic Reinforcement Learning (RL), the agent maximizes an additive objective of the visited states, e.g., a value function. Unfortunately, objectives of this type cannot model many real-world applications such as experiment design, exploration, imitation learning, and risk-averse RL to name a few. This is due to the fact that additive objectives disregard interactions between states that are crucial for certain tasks. To tackle this problem, we introduce *Global* RL (GRL), where rewards are *globally* defined over trajectories instead of *locally* over states. Global rewards can capture *negative interactions* among states, e.g., in exploration, via submodularity, *positive interactions*, e.g., synergetic effects, via supermodularity, while mixed interactions via combinations of them. By exploiting ideas from submodular optimization, we propose a novel algorithmic scheme that converts any GRL problem to a sequence of classic RL problems and solves it efficiently with curvature-dependent approximation guarantees. We also provide hardness of approximation results and empirically demonstrate the effectiveness of our method on several GRL instances.

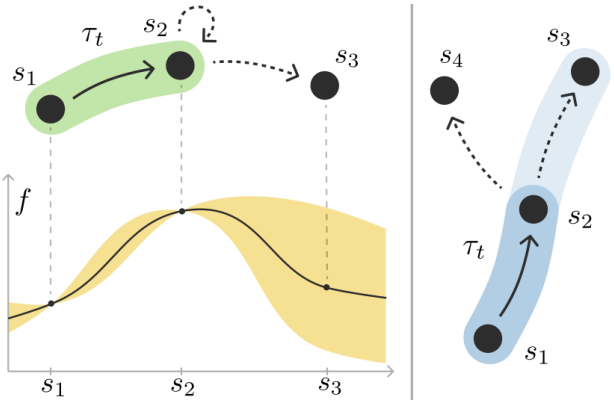


Figure 1: The agent has visited trajectory τ_t and must select the next state. On the left, the agent aims to estimate an unknown state function f: re-visiting s_2 leads to a negative interaction since the information gain has diminishing returns. On the right, the agent seeks a trajectory, i.e., ordered set of atoms, maximizing synergies seen as positive interactions among certain combinations of atoms e.g., adding s_3 to $\tau_t = \{s_1, s_2\}$ leads to a synergetic effect.

1. Introduction

Classic Reinforcement Learning (RL) (Puterman, 2014) represents the value of a trajectory as a sum of *local* rewards over its states(-actions). This fact allows us to exploit Bellman's optimality principle and therefore can find optimal policies using efficient algorithms inspired by dynamic programming (Bellman, 1966). Unfortunately, additive objectives cannot properly capture a multitude of real-world tasks including pure exploration (Hazan et al., 2019; Mutti et al., 2022b; Liu & Abbeel, 2021), function-estimation or experimental design (Mutny et al., 2023; Tarbouriech & Lazaric, 2019; Tarbouriech et al., 2020), imitation learning or distribution matching (Abbeel & Ng, 2004; Lee et al., 2019; Ghasemipour et al., 2020), risk-averse RL (Garcıa & Fernández, 2015; Mutti et al., 2022a), diverse skill discovery (Eysenbach et al., 2018; Campos et al., 2020; He et al., 2022), and constrained RL (Brantley et al., 2020; Qin et al., 2021). In these cases, the interactions between states play a fundamental role in determining the performance of a trajectory. As an example, consider the case of experiment design over a Markov chain (Mutny et al., 2023; Tarbouriech & Lazaric, 2019) in Fig. 1 (left), where an agent aims to minimize uncertainty about an unknown function of the states, by observing a noisy sample of it when visiting a state. Intuitively, the new information gained by observing the function value at a state depends on how often that state has been previously visited: the more it has been visited, the less new information is gained. This phenomenon of *negative interactions* between

[*]Equal contribution. All authors are from ETH Zurich and are affiliated with the ETH AI Center. Correspondence to: Riccardo De Santi <rdesanti@ethz.ch>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

states cannot be captured by additive objectives that simply sum over (fixed) local state rewards (Prajapat et al., 2023). Consider a second example (see Fig. 1, right) where an RL agent is used to design molecules (Thiede et al., 2022) by selecting a set of atoms, where each atom is represented as a state, via a trajectory of fixed length. In this case, *positive interactions* between states, e.g., synergies, can capture the positive effect of including within the trajectory certain combinations of atoms. But once again, additive objectives used in classic RL can only assign a fixed local reward to each atom thus disregarding the interactions among them.

To tackle this problem, we introduce *Global* RL (GRL), where an agent aims to compute a policy maximizing rewards that are *globally* defined over trajectories rather than *locally* over states. This makes it possible to capture non-additivity and interactions among states even in finite-sample processes. Then, we formally show how GRL can be interpreted as a specific combinatorial optimization problem (Section 3), and study its relation with Convex RL (Hazan et al., 2019; Zahavy et al., 2021) (Section 4), an alternative framework to deal with non-additive objectives. It is easy to see that Global RL is a hard problem in general, thus the rest of the work aims to answer the question:

When and how can we efficiently and approximately solve the Global RL problem?

Towards answering this question, we extend discrete semi-gradient methods from submodular optimization (Iyer et al., 2013) to design a meta-algorithm that converts a GRL problem into a sequence of classic RL planning problems. Then, we identify several structural properties of global rewards, leading to approximation guarentees for our algorithmic scheme. Among these, *submodular* (Lovász, 1983; Krause & Golovin, 2014; Krause & Guestrin, 2011) global rewards capture negative interactions between states, *supermodular* (Gallo & Simeone, 1989; Billionnet & Minoux, 1985) global rewards capture positive interactions, and monotone *suBmodular-suPermodular* (BP) (Bai & Bilmes, 2018; Ji et al., 2019) global rewards capture mixed interactions. We show that these reward structures model a wide range of applications that cannot be expressed via local rewards, including maximum entropy exploration (Hazan et al., 2019; Mutti et al., 2022b), informative path planning (Prajapat et al., 2023), experiment design (Mutny et al., 2023), and synergetic trajectory selection among others. Furthermore, we use these structures to study the computational hardness of approximation results and perform a thorough experimental evaluation of the proposed methods (Section 8) in the context of experimental design, optimization of design processes, and safe exploration. To sum up, in this work we present the following contributions:

- The notion of *Global MDP* and the *Global RL* (GRL) problem, which generalizes RL to non-additive objectives.
- A general algorithmic scheme to solve GRL by converting it to a sequence of classic MDPs via submodular semi-gradient methods (Section 6).
- Approximation guarantees for the proposed algorithms via the notion of *curvature* that explicitly connect the degree of non-additivity of a global reward with the approximation ratio (Section 7).
- A computational hardness result for GRL, thereby ruling out the possibility of achieving better approximation ratios (Section 7).
- An extensive experimental evaluation of the proposed algorithms on a wide range of applications (Section 8).

2. Preliminaries

We denote with $[N]$ a set of integers $\{1, \dots, N\}$. Let X be a set, $\Delta(X)$ is the probability simplex over X.

Controlled Markov Process (CMP). An episodic CMP (Puterman, 2014) is a tuple $\mathcal{M} := \langle \mathcal{S}, \mathcal{A}, P, \mu, H \rangle$, where $\mathcal{S}$ is a discrete state space, $\mathcal{A}$ is a discrete action space, $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition model, where $P(s'|s,a)$ is the probability of reaching state s' by taking action a in state s. Meanwhile, $\mu \in \Delta(\mathcal{S})$ is the initial state distribution, H is the horizon of an episode, and we define $\mathcal{T} = [H]$. In each episode, the agent observes an initial state $s_0 \sim \mu$, selects an action a_0, and transitions to $s_1 \sim P(\cdot|s_0, a_0)$. This interaction process is repeated until the episode ends.

Markov Decision Process (MDP). If a CMP $\mathcal{M} := \langle \mathcal{S}, \mathcal{A}, P, \mu, H \rangle$ is augmented with a scalar reward function $r : \mathcal{S} \to \mathbb{R}$ or $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ then we obtain the MDP $\mathcal{M}_r := \langle \mathcal{S}, \mathcal{A}, P, \mu, H, r \rangle$.

Policies. A *policy* encodes the behavior of an agent interacting with a CMP. A non-Markovian policy $\pi \in \Pi_{\text{NM}}$ is a function $\pi : \mathcal{H}_t \to \Delta(\mathcal{A})$, where $\mathcal{H}_t$ denotes the set of all histories, i.e., states visited in the past, up to length t. A Markovian non-stationary policy $\pi_t \in \Pi_{\text{M}}^{\text{NS}}$ is a function $\pi_t : \mathcal{S} \times \mathcal{T} \to \Delta(\mathcal{A})$, while a Markovian stationary policy $\pi \in \Pi_{\text{M}}^{\text{S}}$ is a function $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ independent of the time-step. One can notice that $\Pi_{\text{M}}^{\text{S}} \subseteq \Pi_{\text{M}}^{\text{NS}} \subseteq \Pi_{\text{NM}}$. Moreover, we denote with Π an arbitrary policy class.

Set Functions. We denote with the term *ground set* a set $\mathcal{V}$ inducing the *family* of subsets $f := \{0,1\}^{\mathcal{V}} = 2^{\mathcal{V}}$. Notice that a function $F : 2^{\mathcal{V}} \to \mathbb{R}$ takes subsets of the ground set $\mathcal{V}$ as input and outputs scalars, formally $F : X \mapsto r$ with $X \subseteq \mathcal{V}$ and $r \in \mathbb{R}$. Moreover, every set $X \subseteq \mathcal{V}$ can be represented as a binary vector $x \in \{0,1\}^{\mathcal{V}}$ with entries given by $x(i) = \mathbb{I}_{i \in X}$.

3. Global Reinforcement Learning (GRL)

In this section, we formulate the Global RL (GRL) problem and shed light on its connection with combinatorial

optimization, which will be essential to design efficient approximate algorithms. Towards this goal, we first introduce the concept of Global MDP (GMDP), which generalizes the notion of MDP to the case of general non-additive reward functions.

Definition 1 (Global Markov Decision Process)**.** *Consider a CMP* $\mathcal{M} := \langle \mathcal{S}, \mathcal{A}, P, \mu, H \rangle$ *and a global reward function* $F : 2^{\mathcal{S}\times\mathcal{T}} \to \mathbb{R}$ *mapping trajectories to scalar returns. We define a* Global Markov Decision Process *(GMDP) as the tuple* $\mathcal{M}_F := \langle \mathcal{S}, \mathcal{A}, P, \mu, H, F \rangle$.

We denote as *local* a reward function, e.g., $r : \mathcal{S} \to \mathbb{R}$, that assigns a scalar reward to each state(-action) and as *global* a reward function $F : 2^{\mathcal{S}\times\mathcal{T}} \mapsto \mathbb{R}$ that assigns a scalar reward to each trajectory $\tau := \{(s_t, t)_{t=0}^{H-1}\} \subseteq \mathcal{S} \times \mathcal{T}$. Given a GMDP, we can state the GRL problem as follows.

Global Reinforcement Learning

$$\max_{\pi\in\Pi} \; \mathcal{J}(\pi) := \mathop{\mathbb{E}}_{\tau\sim p_\pi} \big[F(\tau)\big] \tag{1}$$

Hereby, p_π is the distribution over trajectories induced by policy π, formally:

$$p_\pi(\tau) = \mu(s_0) \prod_{t=0}^{H-1} \pi_t(a_t|s_t) P(s_{t+1}|s_t, a_t)$$

Particularly, in this work we focus on the setting of known transition model P and global rewards F.

3.1. GRL as a Subset Selection Problem

Given a family $\mathcal{F} = 2^{\mathcal{V}}$ induced by a ground set $\mathcal{V}$, a representative problem in Combinatorial Optimization (CO) is the *subset selection problem* (Das & Kempe, 2011), where one aims to find an optimum of a set-function $F : \mathcal{F} \to \mathbb{R}$ while constrained to a sub-family $\mathcal{C} \subseteq \mathcal{F}$, as in Equation 2.

Subset Selection Problem

$$\max_{X\in\mathcal{C}} \; F(X) \tag{2}$$

Towards interpreting GRL as a CO problem, we define a path constraint denoted as *dynamics constraint*, and indicate it with $\mathcal{C}_\mathcal{M}$ (Blum et al., 2007). Given a CMP $\mathcal{M}$, $\mathcal{C}_\mathcal{M}$ intuitively represents the set of admissible trajectories according to the dynamics P. A formal construction of $\mathcal{C}_\mathcal{M}$ based on the time-extended CMP of $\mathcal{M}$ is presented in Appendix B.[1]

Given the notion of dynamics constraint, we can define the trajectory-optimization version of GRL (1) for *deterministic* GMDPs, i.e., fixed initial state and deterministic transitions, as the following subset selection problem:

[1]A time-extended CMP has state space $\mathcal{V} := \mathcal{S} \times \mathcal{T}$.

Global RL: Trajectory-Optimization

$$\max_{\tau\in\mathcal{C}_\mathcal{M}} \; F(\tau) \tag{3}$$

Notice that this problem formulation is sufficient to find optimal policies in deterministic GMDPs as in this case there exists an optimal deterministic policy (Prajapat et al., 2023), which can be interpreted as a trajectory. Moreover, this formulation can be straightforwardly extended to the general problem in Equation (1) by replacing $F(\tau)$ by its expectation according to p_π. Nonetheless, due to the natural analogy between trajectories and sets, rather than between policies and distributions over sets, we will first introduce novel concepts for the trajectory-optimization version of GRL (3), and then extend them to the general GRL problem (1).

Crucially, solving GRL (Equations (1) and (3)) is in general inapproximable, since even for a restricted class of global rewards, it is intractable up to any constant factors as shown by Prajapat et al. (2023, Theorem 1). Nonetheless, in Section 5, by leveraging the CO viewpoint introduced within this section, we identify structural properties of F that are both common in practice and lead to efficient approximate algorithms.

4. Relation with Convex RL

Before further discussing the GRL problem, we establish a connection with the area of *General Utilities RL* (GURL) (Zahavy et al., 2021; Geist et al., 2022; Zhang et al., 2020; Barakat et al., 2023), which offers an alternative way to tackle non-additive objectives. As stated in Equation (4),

$$\max_{d^\pi:\pi\in\Pi_\text{M}^\text{S}} f(d^\pi) \tag{4}$$

the GURL objective is to find a policy $\pi \in \Pi_\text{M}^\text{S}$ inducing an optimal state(-action) distribution w.r.t. a given functional $f : \Delta(\mathcal{S} \times \mathcal{A}) \to \mathbb{R}$. If f is convex (concave) in d^π, the problem in equation (4) is referred to as *Convex* (*Concave*) RL (CRL) (Geist et al., 2022; Zhang et al., 2020; Zahavy et al., 2021). In this case, the problem can be efficiently solved via standard constrained convex optimization schemes (Hazan et al., 2019), but unfortunately, it is characterized by a fundamental modelling limitation.

4.1. Fundamental Limitation of Convex RL

Recently, it has been shown (Mutti et al., 2022a; 2023; 2022b) that both theoretically and experimentally, an optimal policy w.r.t. the CRL objective (4) can perform arbitrarily poorly when released in an environment for a finite amount of interactions, which is unfortunately the case in most real-world applications. This phenomenon is due to the fact that CRL in Equation (4) optimizes asymptotic distributions rather than their empirical counterparts. To

Table 1: Applications of Global RL with (from top) submodular, supermodular, BP, and arbitrary global rewards.

APPLICATION	SET FUNCTION $F(\tau)$	DETAILS
STATE ENTROPY EXPLORATION	$\frac{-1}{\lvert\tau\rvert}\sum_{s\in\mathcal{S}}\mathbb{I}_{(s,\cdot)\in\tau}\log\frac{\lvert\{t:(s,t)\in\tau\}\rvert}{\lvert\tau\rvert}$	
GOAL REACHABILITY	$\mathbb{I}\{\lvert\tau\cap S_g\rvert>0\}$	
COVERAGE	$\lvert\bigcup_{(s,t)\in\tau}D^s\rvert$	
BOUNDED CURVATURE COVERAGE	$\sum_{s\in\mathcal{S}}\mathbb{I}_{C(\tau,s)>0}\cdot[1-\alpha(C(\tau,s)-1)]$	$0\leq\alpha\leq 1$
INFORMATIVE PATH PLANNING	$g(\bigcup_{(s,t)\in\tau}D^s)$	$g(V)=\sum_{v\in V}\rho(v)$
D-OPTIMAL EXPERIMENTAL DESIGN	$I(y_\tau;f)=H(y_\tau)-H(y_\tau\lvert f)$	
NEIGHBOURS COVERAGE IN SPACE-TIME	$\sum_{v\in V}\max\{\alpha,\min\{\lvert S\cap S_v\rvert,1\}\}$	$0\leq\alpha\leq 1$
COVERAGE OF TIME-VARYING PROCESSES	$\lvert\bigcup_{v\in\tau}D^v\rvert$	
GOAL COMPLETION	$\mathbb{I}\{\tau\cap S_g=S_g\}$	$S_g\subseteq V$
AUTOMATIC TASK SELECTION	$\sum_{i=1}^{N}R_i(\tau)^\beta$	$R_i(\tau)=\sum_{s\in\tau}r_i(s),\beta>1$
SYNERGICAL TRAJECTORY SELECTION	$\sum_{i=1}^{K}\lvert\tau\cap S_i\rvert^\beta$	$S_i\subseteq V,\beta>1$
DIVERSE AND SYNERGICAL TRAJECTORY SELECTION	$\lvert\bigcup_{s\in\tau}D^s\rvert+\sum_{i=1}^{K}\lvert\tau\cap S_i\rvert^\beta$	$\beta\geq 1$
SAFE REWARD MAXIMIZATION	$R(\tau)+C\cdot\mathbb{I}\{\tau\cap S_u=0\}$	R ADDITIVE
SUBMODULAR + SAFETY	$Q(\tau)+C\cdot\mathbb{I}\{\tau\cap S_u=0\}$	Q SUBMODULAR

tackle this problem, Mutti et al. (2023) propose *Single Trial* Convex RL (ST-CRL) (Equation 5), which captures the finite-samples nature of the problem by optimizing the expected performance of an empirical distribution $d\in\Delta(\mathcal{S}\times\mathcal{A})$ induced by the interaction of a policy π with the environment for a finite number of steps.

$$\max_{\pi\in\Pi_{\text{NM}}}\ \mathop{\mathbb{E}}_{d\sim p^\pi}\big[f(d)\big] \tag{5}$$

Problem (5) does not suffer from the aforementioned issue; however, it is intractable, and developing algorithms that approximately solve Problem (5) is still an open problem (Mutti et al., 2022b). Notably, also GRL (1) overcomes the modelling limitation of Convex RL by directly optimizing a set function defined over trajectories of finite length. Moreover, interestingly, we show that any ST-CRL problem (5) can be rewritten as a Global RL problem (1).

Proposition 1 (Single Trial Convex RL $\subseteq$ Global RL)**.** *Given an instance $\mathcal{I}^+$ of ST-CRL it is possible to reduce it to an instance $\mathcal{I}_+$ of GRL* (1)*.*

The proof can be found in appendix C. Crucially, in ST-CRL (5) convexity is lost due to the empirical distributions constraints set, and alternative structural properties useful for optimization are not known yet. In contrast, GRL correctly captures the underlying combinatorial nature of the problem. This fact makes it possible to leverage structural assumptions for efficient approximate optimization common in a wide variety of real-world problems, including typical CRL applications, as shown in the next sections.

5. Exploiting Structure in Global RL

As previously mentioned, solving the GRL problem is computationally intractable even up to constant factor approximations (Prajapat et al., 2023; Chekuri & Pal, 2005). Nonetheless, in this section, we introduce two fundamental components of the global rewards set function F, namely *submodular* and *supermodular* rewards, which we leverage to approximately solve GRL efficiently. In the following, we show that these properties offer an intuitive characterization and can be used to model a variety of applications by decomposing global rewards into two fundamental components.

Definition 2 (Submodular rewards)**.** *A global reward function $Q:2^{\mathcal{V}}\to\mathbb{R}$ is called* submodular *if for every $\tau_A\subseteq\tau_B\subseteq\mathcal{V}$ and $v\in\mathcal{V}\backslash\tau_B$ it holds that $Q(\{v\}|\tau_A)\geq Q(\{v\}|\tau_B)$, where the marginal gain (discrete derivative) $Q(\{v\}|\tau_A)\coloneqq Q(\{v\}\cup\tau_A)-Q(\tau_A)$.*

Thus, submodularity naturally captures a diminishing return property, i.e., the marginal gain of adding a state v to a smaller trajectory τ_A is higher as compared to adding it's super set τ_B. This denotes *negative interaction* between states i.e. similar states are discouraged and thus maximizing such functions will encode diversity in the resulting trajectory.

Definition 3 (Supermodular rewards)**.** *A global reward function $G:2^{\mathcal{V}}\to\mathbb{R}$ is* supermodular *if for every $\tau_A\subseteq\tau_B\subseteq\mathcal{V}$ and $v\in\mathcal{V}\backslash\tau_B$ it holds that $G(\{v\}|\tau_A)\leq G(\{v\}|\tau_B)$.*

Therefore, contrary to submodular, maximizing supermodular rewards encodes complementarity in the resulting trajectory, i.e., having similar states complement each other and results in larger gains – *positive interaction*. Next, combining both we define another reward class called monotone suBmodular suPermodular (BP) rewards (Bai & Bilmes, 2018):

Definition 4 (BP rewards)**.** *A global reward function $F:2^{\mathcal{V}}\to\mathbb{R}$ admits a* BP *decomposition if there exists a submodular reward function Q and a supermodular reward function G both normalized ($Q(\emptyset)=G(\emptyset)=0$) and monotonic non-decreasing ($Q(\{v\}|\tau_A)\geq 0, G(\{v\}|\tau_A)\geq 0,\forall v\in$*

$\mathcal{V}, \tau_A \subseteq \mathcal{V}$ *such that* $F(\tau_A) = Q(\tau_A) + G(\tau_A)\ \forall \tau_A \subseteq \mathcal{V}$.

Thus, BP rewards enable us to capture applications involving both positive and negative interactions. Interestingly, any arbitrary global reward $F : 2^{\mathcal{V}} \to \mathbb{R}$ can be decomposed as the sum of submodular and supermodular rewards, i.e., $F(\tau_A) = Q(\tau_A) + G(\tau_A)$ as shown for set functions by Narasimhan & Bilmes (2012, Lemma 4). Moreover, under certain conditions, this decomposition can be computed in polynomial time (Iyer & Bilmes, 2012a, c.f. Lemma 3.2). Crucially, in Section 5 we present an algorithmic scheme that does not make any assumption (e.g., monotonicity) on the submodular or supermodular component, hence it is applicable to any GMDPs with arbitrary global rewards as long as the decomposition is available.

Examples of global rewards. In Table 1, we show a wide range of relevant applications that can be modelled with global rewards. Applications such as state entropy exploration, D-optimal experiment design, where the objective is maximized by visiting a diverse set of states, are captured using submodular rewards. In contrast, the supermodular component captures positive interactions among states, e.g., certain synergies among a specific set of states. Combining both components (BP rewards) becomes particularly relevant when addressing intricate processes that require both, e.g., diverse and synergical trajectory selection. Moreover, we can model constraints using arbitrary (e.g., non-monotone BP) global rewards and thus extend to applications involving policy learning under safety-critical conditions. Furthermore, since our rewards are defined on state time pairs, we can also model time-varying processes.

6. Semi-gradient Method for GRL

In this section, we present our novel algorithmic scheme to solve the GRL Problem. We begin by explaining the notion of subdifferentials, which are used to obtain semi-gradients of global rewards. Using the semi-gradients, we first present the algorithm for deterministic GMDPs (3) in order to build intuition, and then extend it to the general Problem (1).

Subdifferentials. Given a non-additive set function F, similar to convex functions, we can define the subdifferential $\partial_F(X)$ at a set $X \subseteq V$ as shown in Iyer & Bilmes (2015):

$$\partial_F(X) := \{x \in \mathbb{R}^n : F(Y) \geq F(X) + x(Y) - x(X), \forall Y \subseteq V\}$$

Here, $x(A) := \sum_{v \in A} x(v)$ is a modular function over V. Although the polyhedron $\partial_F(X)$ can be defined for any set function, it can be characterized efficiently for submodular functions. We denote a subgradient of F at X as $h_X \in \partial_F(X)$. For submodular F, the extreme points of the polyhedron can be characterized as follows:

First, we define a permutation $\sigma : [|V|] \to V$ that re-orders the elements of V such that the elements of X are assigned to the first $|X|$ positions of V (i.e., $\sigma(i) \in X \iff i \leq |X|$), whereas the remaining elements are assigned arbitrarily. We define the set $S_0^\sigma := \emptyset$ and $S_i^\sigma := \{\sigma(1), \ldots, \sigma(i)\}$. It is important to note that as a consequence, we have $S_{|X|} = X$. With this, we obtain the extreme point h_X^σ of the polyhedron $\partial_F(X)$ with entries $h_X^\sigma(\sigma(i)) = F(S_i^\sigma) - F(S_{i-1}^\sigma)$. Now, given any subgradient h_X, we define a modular function,

$$m_X^\sigma(Y) := F(X) + h_X(Y) - h_X(X),$$

which is defined $\forall Y \subseteq V$ and is a tight lower bound of F, i.e., $m_X(X) = F(X)$ with $m_X(Y) \leq F(Y), \forall Y \subseteq V$, see Proposition 2 for details. Notably, if we choose subgradient h_X to be the extreme point of $\partial_F(X)$, then the modular function simplifies to $m_X(Y) = h_X(Y)$, which is a marginal gain of F evaluated with respect to some permutation.

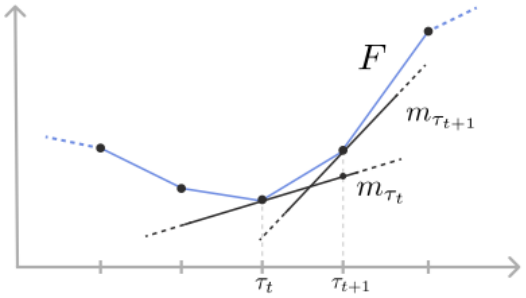


Figure 2: In GTO, at any step t, we construct m_{τ_t}, a tight modular lower bound about τ_t and optimize the resulting classic MDP, which results in an improved trajectory τ_{t+1}.

Analogously, for supermodular functions we define the following modular lower bound $\forall Y \subseteq V$ about the set X, as shown in Bai & Bilmes (2018); Iyer & Bilmes (2012b):

$$m_X(Y) := F(X) - \sum_{j \in X \setminus Y} F(j \mid X \backslash j) + \sum_{j \in Y \setminus X} F(j \mid \emptyset).$$

Then, $m_X(Y) \leq F(Y), \forall Y \subseteq V$ and $m_X(X) = F(X)$. Furthermore, by adding the tight modular lower bounds for submodular and supermodular functions, we obtain tight modular lower bounds for global rewards (Bai & Bilmes, 2018).

Global Trajectory Optimization (GTO). Naturally, two key questions emerge: How can we use these modular lower bounds to solve the GRL problem? And, how do we make sure the dynamics constraints are satisfied? Intuitively, a modular lower bound of the global reward function represents nothing but an additive reward. We recursively approximate the global rewards function about the current trajectory τ with additive rewards (as shown in Fig. 2), which results in a sequence of classic RL planning problems. These problems can be optimized efficiently through standard RL techniques while ensuring admissible trajectories, i.e., satisfying the dynamics constraints. In simpler terms,

Algorithm 1 Global Trajectory Optimization (GTO)

1: **Input:** Deterministic GMDP, τ_1
2: **for** $t = 1, 2, \ldots$ **do**
3: $\quad m_{\tau_t} \leftarrow$ Compute lower bound of F around τ_t
4: $\quad \tau_{t+1} \leftarrow \textsc{MdpSolver}(\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, P, s_0, H, m_{\tau_t} \rangle)$
5: **end for**
6: Return τ_t

our algorithmic scheme is a meta-algorithm that approximates GMDP with a sequence of local MDPs which then are solved with existing MDP solvers. The lower bounds being tight at the current trajectory ensures monotonic improvement across the iterations until convergence. Since F is defined on $\mathcal{V} \coloneqq \mathcal{S} \times T$, our modular lower bounds are defined on $\mathcal{V}$. To construct a modular lower bound tight at a trajectory τ, we define a permutation σ_τ induced by τ as follows,

$$\sigma_\tau = \{\tau; \mathcal{V} \backslash \tau\} \tag{6}$$

In the described permutation, the elements within the set $\mathcal{V}\backslash\tau$ or the set τ can be arbitrarily permuted among themselves, while still resulting in a valid tight modular lower bound (Proposition 2). This flexibility can also be exploited to incorporate prior knowledge for learning better policies. We refer the reader to Appendix F.2 for further details on modular lower bounds.

The resulting algorithm is summarized in Algorithm 1. We start with an arbitrary trajectory τ_1. The algorithm has two steps: *i)* computes a modular lower bound of the BP function as described above to obtain additive rewards (classic MDP) about the current trajectory τ_t (Line 3) and *ii)* solves the resulting classic MDP using existing MDPSOLVER tools, e.g., value iteration, which guarantees an optimal solution corresponding to the rewards m_{τ_t} (Line 4). On solving, the new trajectory will achieve a monotonic improvement in objective value (Proposition 3). The algorithm alternates between the two steps until convergence, i.e., the objective does not improve anymore.

Global Policy Optimization (GPO). For stochastic GMDPs, we solve the problem in policy space. For this, we generalize the concept of lower bound about a trajectory to distribution over trajectories generated by a policy. The corresponding modular lower bound about a policy, $m_\pi^E \in \mathbb{R}^V$ is defined as the expectation of the modular rewards about the trajectories induced by the policy π, i.e.,

$$m_\pi^E \coloneqq \mathop{\mathbb{E}}_{\tau \sim \pi}[m_\tau]. \tag{7}$$

This can be estimated, for instance, using Monte Carlo samples. Next, we define the linearization of the GRL objective $\mathcal{J}(\pi)$ about a policy π' and evaluating it (7) at a policy π as,

$$m_{\pi'}^E(\pi) \coloneqq \mathop{\mathbb{E}}_{\tau \sim \pi} \mathop{\mathbb{E}}_{\tau' \sim \pi'}[m_{\tau'}(\tau)]. \tag{8}$$

The algorithm steps for GPO largely remains the same; however, in the stochastic case, the MDPSOLVER solves $\max_\pi m_{\pi_{t-1}}^E(\pi)$ using standard RL techniques. Please see Appendix F for a detailed algorithm for stochastic GMDPs. While our algorithm can optimize arbitrary global rewards, in the next section we present approximation guarantees for monotone submodular, supermodular and BP rewards.

7. Approximation guarantees and Hardness

Due to the non-additivity of the reward function, performance guarantees cannot be based on the classic notion of value function (Puterman, 2014). Nonetheless, along the lines of (Tarbouriech & Lazaric, 2019; Mutti et al., 2023), we define the following notion of optimality.

Definition 5 (Non-additive Suboptimality Gap)**.** *Consider a policy $\pi \in \Pi$ interacting with a Global MDP $\mathcal{M}_F$. We define the non-additive suboptimality gap of π as:*

$$\mathcal{R}(\pi) \coloneqq \mathcal{J}^\star - \mathop{\mathbb{E}}_{\tau \sim p_\pi}\big[F(\tau)\big]$$

where $\mathcal{J}^\star \coloneqq \max_{\pi \in \Pi_{\mathrm{NM}}} \ \mathbb{E}_{\tau \sim p_\pi}\big[F(\tau)\big]$.

Since it is well-understood that in general $\Pi_{\mathrm{M}}^{\mathrm{NS}}$ is not sufficient to minimize the Non-additive Suboptimality Gap (Prajapat et al., 2023; Mutti et al., 2022b), in the following we give guarantees w.r.t. the reference class Π_{NM}.

7.1. How good is a modular approximation?

The algorithms proposed in Section 6 approximately solve the GRL problem (3 and 1) by optimizing a sequence of modular lower bounds of the global reward F. In this section we aim to derive first-iterate approximation guarantees (Iyer et al., 2013) to answer the question:

How well can a Global MDP be approximated by a single MDP with local rewards?

Intuitively, depends on how much the global reward is *non-additive*. For the case of submodular, supermodular, and BP rewards, this can be captured via the notions of *submodular* and *supermodular curvature* (Conforti & Cornuéjols, 1984).

Definition 6 (Submodular and Supermodular Curvature)**.** *Given a monotone set-function $F : 2^V \to \mathbb{R}$ we define the submodular and supermodular curvatures respectively as:*

$$k_F \coloneqq 1 - \min_{v \in V} \frac{F(v \mid V \backslash \{v\})}{F(v)} \in [0, 1] \quad (F \text{ submodular})$$

$$k^F \coloneqq 1 - \min_{v \in V} \frac{F(v)}{F(v \mid V \backslash \{v\})} \in [0, 1] \quad (F \text{ supermodular})$$

For a BP function $F = Q + G$ we can derive curvature-based guarantees by combining the curvatures of Q and G. Moreover, we say that a submodular (supermodular) function F

has bounded curvature if $k_F(k^F) < 1$. Otherwise, we say that F is fully-curved. We can now state the approximation guarantees achieved by GPO in a general stochastic GMDP w.r.t. the Non-additive Suboptimality Gap (Definition 5).

Theorem 7.1 (Approximation Guarantees GPO)**.** *Let* $\mathcal{J}^\star := \max_{\pi \in \Pi_{\text{NM}}} \mathbb{E}_{\tau \sim p_\pi}\left[F(\tau)\right]$ *and* π_1 *the policy resulting from one iteration of* GPO *on a GMDP* $\mathcal{M}_F$*. Then* GPO *guarantees that for*

i) Monotone submodular reward function, F

$$\mathcal{R}(\pi_1) \leq k_F \mathcal{J}^\star,$$

ii) Monotone supermodular reward function, F

$$\mathcal{R}(\pi_1) \leq \frac{2k^F - (k^F)^2}{1 - k^F} \mathcal{J}^\star,$$

iii) BP reward function, $F = Q + G$

$$\mathcal{R}(\pi_1) \leq \alpha \mathcal{J}^\star,\ \alpha = \begin{cases} \frac{2k^G - (k^G)^2}{1 - k^G} & \text{if } k_F \leq k^G \\ \frac{1 - (1 - k_Q)(1 - k^G)}{1 - k^G} & \text{otherwise} \end{cases}$$

Interestingly, the notion of Non-additive Suboptimality Gap (Definition 5) can be tailored to the trajectory-optimization version of GRL (3), and similar guarantees for GTO can be derived for deterministic GMDPs. The analysis for this case and the proof of Theorem 7.1 are in Appendices C and D.

Discussion. Theorem 7.1 relates the suboptimality gap achieved by Markovian policies with the degree of non-additivity of the reward function as captured by curvature.[2] Notably, the submodular functions enjoy the best approximation ratio, while high supermodular curvature seems to substantially affect the solution quality. Moreover, notice that since GPO optimizes a sequence of lower bounds, the quality of the returned solutions can be better than the ones stated in Theorem 7.1, as shown in the experiments (Section 8). Ultimately, notice that these results extend classic submodular function maximization results to the case of constrained optimization under dynamics constraints, as presented in Section 3.

7.2. Hardness of Global RL

To conclude this section, we present a hardness result for the trajectory-optimization GRL Problem (3) in deterministic environments. The following result extends known results for BP-function maximization (Bai & Bilmes, 2018) to the case of dynamics-constrained optimization.

[2]Computable in linear time w.r.t. $|\mathcal{S} \times \mathcal{T}|$.

Theorem 7.2 (Hardness of GRL, trajectory-optimization (3))**.** *For all* $0 \leq \beta \leq 1$*, there exists an instance of a BP global reward* $F = Q + G$ *with* $k^G = \beta$ *such that no poly-time algorithm can achieve an approximation factor better than* $1 - k^G + \epsilon\ \forall \epsilon > 0$ *w.r.t. the Non-additive Suboptimality Gap in deterministic GMDPs, unless P = NP.*

The proof of Theorem 7.2 can be found in Appendix E. Crucially, theorem 7.2 shows that the supermodular curvature plays a fundamental role in determining a lower bound on the approximability of the problem.

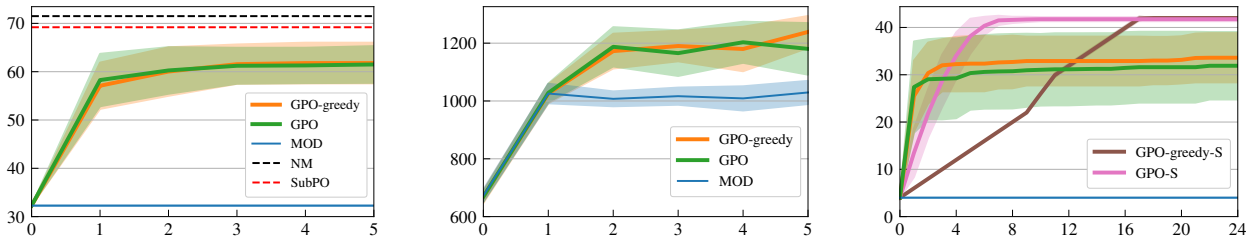
8. Experiments

We present an experimental analysis of GTO and GPO on three tasks: *i)* D-optimal experimental design, *ii)* diverse and synergical trajectory selection, and *iii)* safe state coverage. These environments have deterministic (*i,iii*) and stochastic (*ii*) transitions, consider global rewards with bounded curvature (*i*) as well as fully-curved (*ii,iii*), and cover submodular (*i*), BP (*ii*), and arbitrary global rewards (*iii*).

We compare the performances of the policy π obtained via GTO (Algorithm 1) and GPO (Algorithm 2) with the performance of an optimal policy w.r.t. the additive objective $\mathcal{J}_m(\pi) = \mathbb{E}_{\tau \sim p_\pi}\left[\sum_{v \in \tau} F(v)\right]$, which disregards interactions between states within the same trajectory, as it is the case in classic RL. Moreover, wherever possible[3], we compare the performance of π with the optimal non-Markovian policy w.r.t. F, and with SubPO (Prajapat et al., 2023), a policy gradient method for optimizing under submodular rewards. We do not explicitly compare with other RL algorithms, since, SubPO is a representative baseline for Reinforce-type algorithms with variance reduction techniques catering to non-additive returns. Notably, for efficient performance, we run SubPO with policy parameterized with a neural network and thus it loses its approximation guarantees in contrast to GPO. Moreover, we consider two variants of our algorithm GPO-S and GPO-greedy which build lower bounds based state-dependent and greedy permutation respectively (see Appendix F.2 for details). We run all experiments over a grid environment with 400 states, discrete actions {left, right, up, down, stay}, and plot 95% confidence intervals over 20 runs.

Bayesian D-Optimal experimental design. In this task, the agent aims to optimally select a trajectory over sampling points to estimate an unknown function f represented via Gaussian Processes (Rasmussen et al., 2006), a problem known as optimal experimental design (Mutny et al., 2023). We aim to maximize the mutual information $I(y_\tau; f) = H(y_\tau) - H(y_\tau | f)$ between the observations y_τ, and the

[3]For computational reasons, we can solve it in deterministic environments with monotonic non-decreasing global rewards.



(a) Experiment design (b) Diverse synergies (c) Safe state coverage

Figure 3: We compare GTO and GPO with the optimal policy for the modularized objective F_m (MOD). We observe that MOD performs sub-optimally as its objective cannot capture interactions between states. The alternative versions of the algorithm tested are presented in Section F.2. (Y-axis: $\mathcal{J}(\pi)$, X-axis: iterations)

unknown function f evaluated at locations in τ, which is a submodular global reward. In Fig. 3a, we observe that GTO performs significantly better than the optimal policy for the modularized objective, competitively with SubPO, and nearly optimally w.r.t. to the optimal non-Markovian policy.

Diverse synergies. In this task, we maximize the BP rewards $F(\tau) = |\bigcup_{s\in\tau} D^s| + \sum_{i=1}^{K} |\tau \cap S_i|^\beta$ with $S_i \subseteq \mathcal{V} \coloneqq \mathcal{S} \times T$, in a stochastic environment, where with 0.1 probability the agent transitions to a neighbouring state uniformly at random. This objective induces policies maximizing coverage over state space while seeking specific complementarity between states within the trajectory resembling, for instance, synergetic effects between atoms in a molecular design process. As hinted by Theorems 7.1 and 7.2, maximizing the fully-curved supermodular component of F may perform arbitrarily poorly. Nonetheless, in Fig. 3b we observe that GPO performs significantly better than the modular optimal policy w.r.t. F_m. Moreover, in Appendix G, we show that in the deterministic setting, where we can compute the optimal non-Markovian policy, GTO performs optimally.

Safe state coverage. In Fig. 3c, we consider a non-monotone BP function encoding the task of safe state space coverage defined as $|\bigcup_{s\in\tau} D^s| + C \cdot \mathbb{I}\{\tau \cap S_u = 0\}$ with D^s being a disk covering neighbouring states (Prajapat et al., 2022). An optimal policy w.r.t. this objective is highly explorative while avoiding unsafe areas. Since this objective is invariant w.r.t. the time-dimension, we can leverage the lower bounds introduced in Appendix F.2. We observe that the policy returned by GTO performs significantly better than the optimal policy w.r.t. to the modularized objective F_m. This is due to the fact that the task embeds a notion of diversity that cannot be captured by local reward functions.

8.1. Experimental Insights and Observations

In the following, we aim to present several claims emerging from the experiments presented in Section 8 and in Appendix G.

Practical versus theoretical performances. In the case of bounded-curvature reward functions, the theory presented in Section 7 gives first-iteration performance guarantees for GTO and GPO. Nonetheless, in practice these algorithms can significantly improve the solution after the first iteration thus potentially reach significantly better performances than the ones captured via such curvature-based guarantees. The experimental results show that this fact is particularly true for submodular global reward functions where the objective value increases significantly over multiple iterations.

Beyond bounded-curvature functions. While the curvature assumption is needed to build theoretical guarantees, the proposed algorithms seem to show outstanding performances even with fully-curved reward functions.

Non-monotonic improvements in GPO. As pointed out in Section 6, in the stochastic setting GPO does not guarantee a monotonic performance improvement, but we still have a curvature-based performance guarantee as shown in Section 7. In the experiments above, we show that in practice GPO shows promising performances even in settings with stochastic dynamics. Clearly, concentrating the estimates in GPO requires sampling multiple trajectories thus increasing the computational complexity of the algorithm.

Greedy submodular lower bounds. Observing the performances of greedy lower bounds (indicated with $-greedy$ in Fig. 3 and in Appendix G), we can notice that they outperform the normal counterpart in certain instances. In particular, a further analysis reported in Fig. 5 within Appendix G, shows that this is the case when the horizon is neither too large nor too small, but just enough to properly maximize a certain submodular reward via a specific trajectory (or policy), as shown in Figure 5 (right). Meanwhile, when the horizon is arbitrarily small, the two lower bounds seem to perform equally well. This is due to the fact that in this latter case random actions will also perform well as there is less chance of overlapping states and therefore better planning strategies cannot show performance improvements.

9. Related Work

Convex RL. Convex RL (CRL) (Hazan et al., 2019; Zahavy et al., 2021) is a recent framework that extends RL to non-additive rewards by optimizing convex functionals of state(-action) distributions. It can capture a wide range of applications including exploration (Hazan et al., 2019), experimental design (Mutný et al., 2023), imitation learning (Lee et al., 2019), and risk-averse RL (Garcıa & Fernández, 2015; Mutti et al., 2022a). Interestingly, also common CRL algorithms reduce the problem to solving a sequence of MDPs. Moreover, notice that several CRL objectives, e.g., entropy maximization and experimental design, can be cast as global rewards as shown in Table 1. Furthermore, while (mixtures of) Markovian stationary policies are sufficient to optimize CRL objectives, Global RL leverages Markovian non-stationary policies, a fact that manifests the need of more general policy classes to optimize non-additive rewards in finite-samples settings, as shown by Mutti et al. (2022b). Further, similar to some Convex RL works (Tarbouriech & Lazaric, 2019; Zahavy et al., 2021), we believe it is possible, and an interesting direction of future work, to extend GPO to the case of unknown dynamics, via an optimistic estimate of P.

RL with Non-Markovian Rewards. The concepts of global rewards and non-additive rewards presented within this work have a strong link with the notion of non-Markovian rewards, namely history-dependent reward functions. Non-Markovian rewards in RL has been studied from the lenses of logic and formal language theory (Gaon & Brafman, 2020; De Giacomo & Vardi, 2013; Brafman et al., 2018). Currently, the analysis within these works seems orthogonal to ours; however, exploring connections between these two approaches is an interesting research direction.

Submodularity in Decision Making. Submodular functions have been used to model problems in active machine learning (Krause & Golovin, 2014; Bilmes, 2022), bandits (Yue & Guestrin, 2011; Chen et al., 2017; Gabillon et al., 2013), planning (Chekuri & Pal, 2005; Wang et al., 2020) and RL (Prajapat et al., 2023). The most related to us is from Prajapat et al. (2023) which introduces Submodular MDPs and proposes a policy gradient (PG) based method for it. Beyond the larger generality of the function classes we consider, the main difference is that we transport an algorithmic scheme from submodular optimization tailored for BP function maximization to MDPs, rather than using a function-agnostic optimization scheme like PG.

BP function maximization. Historically, submodular and supermodular function maximization have been treated with significantly different approaches (Krause & Golovin, 2014; Iwata, 2008). Recently, novel optimization schemes based on discrete subgradients (Iyer et al., 2013; Iyer & Bilmes, 2015) made it possible to treat submodular and supermodular function maximization in a unified manner, directly optimize suBmodular-suPermodular (BP) functions (Bai & Bilmes, 2018), as well as optimize arbitrary set functions with a known decomposition into a submodular and a supermodular component. In Section 7 we extend hardness results for BP function maximization to GRL by shedding light on the effect of the transition dynamics underlying the Markov process. This removes the possibility of *teleporting* from any element of the set to any other element, which we capture in Section 3 via the notion of dynamics constraint (see Appendix B).

10. Conclusion

We introduce a novel problem formulation denoted *Global* RL (GRL), for sequential decision-making under non-additive objectives. Then, leveraging tools from submodular optimization, we provide a novel algorithmic scheme that converts the GRL problem to a sequence of standard RL planning problems. We identify structural properties that render the problem efficiently and approximately solvable for a wide variety of applications according to curvature-based approximation guarantees. Moreover, we derive a hardness result for a representative class of GRL and extensively showcase the empirical performances of the proposed algorithms to solve RL problems that cannot be captured via classic additive (or local) rewards.

Acknowledgement

This publication was made possible by the ETH AI Center doctoral fellowship to Riccardo De Santi and Manish Prajapat. We would like to thank Mohammad Reza Karimi and Mojmír Mutný for the insightful discussions.

The project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program grant agreement No 815943 and the Swiss National Science Foundation under NCCR Automation grant agreement 51NF40 180545.

Impact Statement

This paper presents work whose goal is to advance the field of Reinforcement Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Abbeel, P. and Ng, A. Y. Apprenticeship learning via inverse reinforcement learning. In *International Conference on Machine learning*, 2004.

Agarwal, A., Jiang, N., Kakade, S. M., and Sun, W. Reinforcement learning: Theory and algorithms. *CS Dept.,*

UW Seattle, Seattle, WA, USA, Tech. Rep, 32:96, 2019.

Bai, W. and Bilmes, J. Greed is still good: Maximizing monotone Submodular+Supermodular (BP) functions. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 304–313. PMLR, 10–15 Jul 2018.

Barakat, A., Fatkhullin, I., and He, N. Reinforcement learning with general utilities: Simpler variance reduction and large state-action space. *arXiv preprint arXiv:2306.01854*, 2023.

Bellman, R. Dynamic programming. *Science*, 153(3731): 34–37, 1966.

Billionnet, A. and Minoux, M. Maximizing a supermodular pseudoboolean function: A polynomial algorithm for supermodular cubic functions. *Discrete Applied Mathematics*, 12(1):1–11, 1985.

Bilmes, J. Submodularity in machine learning and artificial intelligence. *arXiv preprint arXiv:2202.00132*, 2022.

Blum, A., Chawla, S., Karger, D. R., Lane, T., Meyerson, A., and Minkoff, M. Approximation algorithms for orienteering and discounted-reward tsp. *SIAM Journal on Computing*, 37(2):653–670, 2007.

Brafman, R., De Giacomo, G., and Patrizi, F. Ltlf/ldlf non-markovian rewards. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Brantley, K., Dudik, M., Lykouris, T., Miryoosefi, S., Simchowitz, M., Slivkins, A., and Sun, W. Constrained episodic reinforcement learning in concave-convex and knapsack settings. In *Advances in Neural Information Processing Systems*, 2020.

Campos, V., Trott, A., Xiong, C., Socher, R., Giró-i Nieto, X., and Torres, J. Explore, discover and learn: Unsupervised discovery of state-covering skills. In *International Conference on Machine Learning*, 2020.

Chekuri, C. and Pal, M. A recursive greedy algorithm for walks in directed graphs. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*, pp. 245–253, 2005. doi: 10.1109/SFCS.2005.9.

Chen, L., Krause, A., and Karbasi, A. Interactive submodular bandit. In *NIPS*, 2017.

Conforti, M. and Cornuéjols, G. Submodular set functions, matroids and the greedy algorithm: Tight worst-case bounds and some generalizations of the rado-edmonds theorem. *Discrete Applied Mathematics*, 7(3):251–274, 1984. ISSN 0166-218X.

Das, A. and Kempe, D. Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection. *arXiv preprint arXiv:1102.3975*, 2011.

De Giacomo, G. and Vardi, M. Y. Linear temporal logic and linear dynamic logic on finite traces. In *IJCAI'13 Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pp. 854–860. Association for Computing Machinery, 2013.

Eysenbach, B., Gupta, A., Ibarz, J., and Levine, S. Diversity is all you need: Learning skills without a reward function. In *International Conference on Learning Representations*, 2018.

Gabillon, V., Kveton, B., Wen, Z., Eriksson, B., and Muthukrishnan, S. Adaptive submodular maximization in bandit setting. *Advances in Neural Information Processing Systems*, 26, 2013.

Gallo, G. and Simeone, B. On the supermodular knapsack problem. *Mathematical Programming*, 45:295–309, 1989.

Gaon, M. and Brafman, R. Reinforcement learning with non-markovian rewards. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 3980–3987, 2020.

Garcıa, J. and Fernández, F. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(1):1437–1480, 2015.

Geist, M., Pérolat, J., Laurière, M., Elie, R., Perrin, S., Bachem, O., Munos, R., and Pietquin, O. Concave utility reinforcement learning: The mean-field game viewpoint. In *International Conference on Autonomous Agents and Multiagent Systems*, 2022.

Ghasemipour, S. K. S., Zemel, R., and Gu, S. A divergence minimization perspective on imitation learning methods. In *Conference on Robot Learning*, 2020.

Hazan, E., Kakade, S., Singh, K., and Van Soest, A. Provably efficient maximum entropy exploration. In *International Conference on Machine Learning*, 2019.

He, S., Jiang, Y., Zhang, H., Shao, J., and Ji, X. Wasserstein unsupervised reinforcement learning. In *AAAI Conference on Artificial Intelligence*, 2022.

Iwata, S. Submodular function minimization. *Mathematical Programming*, 112:45–64, 2008.

Iyer, R. and Bilmes, J. Algorithms for approximate minimization of the difference between submodular functions, with applications. *arXiv preprint arXiv:1207.0560*, 2012a.

Iyer, R. and Bilmes, J. Polyhedral aspects of submodularity, convexity and concavity. *arXiv preprint arXiv:1506.07329*, 2015.

Iyer, R. and Bilmes, J. A. Submodular-bregman and the lovász-bregman divergences with applications. *Advances in Neural Information Processing Systems*, 25, 2012b.

Iyer, R., Jegelka, S., and Bilmes, J. Fast semidifferential-based submodular function optimization. In *International Conference on Machine Learning*, pp. 855–863. PMLR, 2013.

Ji, S., Xu, D., Li, M., Wang, Y., and Zhang, D. Stochastic greedy algorithm is still good: maximizing submodular+ supermodular functions. In *World Congress on Global Optimization*, pp. 488–497. Springer, 2019.

Krause, A. and Golovin, D. Submodular function maximization. *Tractability*, 3:71–104, 2014.

Krause, A. and Guestrin, C. Submodularity and its applications in optimized information gathering. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(4): 1–20, 2011.

Lee, L., Eysenbach, B., Parisotto, E., Xing, E., Levine, S., and Salakhutdinov, R. Efficient exploration via state marginal matching. *arXiv preprint arXiv:1906.05274*, 2019.

Liu, H. and Abbeel, P. Behavior from the void: Unsupervised active pre-training. In *Advances in Neural Information Processing Systems*, 2021.

Lovász, L. Submodular functions and convexity. *Mathematical Programming The State of the Art: Bonn 1982*, pp. 235–257, 1983.

Mutný, M., Janik, T., and Krause, A. Active exploration via experiment design in markov chains. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2023.

Mutny, M., Janik, T., and Krause, A. Active exploration via experiment design in markov chains. In *International Conference on Artificial Intelligence and Statistics*, pp. 7349–7374. PMLR, 2023.

Mutti, M., De Santi, R., De Bartolomeis, P., and Restelli, M. Challenging common assumptions in convex reinforcement learning. *Advances in Neural Information Processing Systems*, 35:4489–4502, 2022a.

Mutti, M., De Santi, R., and Restelli, M. The importance of non-markovianity in maximum state entropy exploration. In *International Conference on Machine Learning*, 2022b.

Mutti, M., De Santi, R., De Bartolomeis, P., and Restelli, M. Convex reinforcement learning in finite trials. *Journal of Machine Learning Research*, 24(250):1–42, 2023.

Narasimhan, M. and Bilmes, J. A. A submodular-supermodular procedure with applications to discriminative structure learning. *arXiv preprint arXiv:1207.1404*, 2012.

Prajapat, M., Turchetta, M., Zeilinger, M., and Krause, A. Near-optimal multi-agent learning for safe coverage control. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 14998–15012. Curran Associates, Inc., 2022.

Prajapat, M., Mutnỳ, M., Zeilinger, M. N., and Krause, A. Submodular reinforcement learning. *arXiv preprint arXiv:2307.13372*, 2023.

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

Qin, Z., Chen, Y., and Fan, C. Density constrained reinforcement learning. In *International Conference on Machine Learning*, 2021.

Rasmussen, C. E., Williams, C. K., et al. *Gaussian processes for machine learning*, volume 1. Springer, 2006.

Tarbouriech, J. and Lazaric, A. Active exploration in markov decision processes. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 974–982. PMLR, 2019.

Tarbouriech, J., Shekhar, S., Pirotta, M., Ghavamzadeh, M., and Lazaric, A. Active model estimation in markov decision processes. In *Conference on Uncertainty in Artificial Intelligence*, pp. 1019–1028. PMLR, 2020.

Thiede, L. A., Krenn, M., Nigam, A., and Aspuru-Guzik, A. Curiosity in exploring chemical spaces: intrinsic rewards for molecular reinforcement learning. *Machine Learning: Science and Technology*, 3(3):035008, 2022.

Wang, R., Zhang, H., Chaplot, D. S., Garagić, D., and Salakhutdinov, R. Planning with submodular objective functions. *arXiv preprint arXiv:2010.11863*, 2020.

Yue, Y. and Guestrin, C. Linear submodular bandits and their application to diversified retrieval. *Advances in Neural Information Processing Systems*, 24, 2011.

Zahavy, T., O'Donoghue, B., Desjardins, G., and Singh, S. Reward is enough for convex mdps. *Advances in Neural Information Processing Systems*, 34:25746–25759, 2021.

Zhang, J., Koppel, A., Bedi, A. S., Szepesvari, C., and Wang, M. Variational policy gradient method for reinforcement learning with general utilities. *Advances in Neural Information Processing Systems*, 33:4572–4583, 2020.

A. List of symbols

General Mathematical Objects

$[N]$	$\triangleq$	Set of integers $\{1, \dots, N\}$
$\Delta(X)$	$\triangleq$	Probability simplex over X

Controlled Markov Process

$\mathcal{M}$	$\triangleq$	Controlled Markov Process (CMP), $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, P, \mu, H \rangle$
$\mathcal{S}$	$\triangleq$	State space
$\mathcal{A}$	$\triangleq$	Action space
P	$\triangleq$	Transition model, $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$
μ	$\triangleq$	Initial state distribution, $\mu \in \Delta(\mathcal{S})$
H	$\triangleq$	Horizon
$\mathcal{T}$	$\triangleq$	Time steps set, $\mathcal{T} = [H]$
s_0	$\triangleq$	Initial state $s_0 \sim \mu$
s_t	$\triangleq$	State at time step t
τ	$\triangleq$	Trajectory $\tau = \{s_0, \dots, s_{H-1}\}$
$p_\pi(\tau)$	$\triangleq$	Probability of trajectory τ given a fixed $\mathcal{M}$ and π, see equation (3)

MDP and GMDP

$\mathcal{M}_r$	$\triangleq$	Markov Decision Process (MDP), $\mathcal{M}_r := \langle \mathcal{S}, \mathcal{A}, P, \mu, H, r \rangle$
r	$\triangleq$	Scalar reward function $r : \mathcal{S} \to \mathbb{R}$ or $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$
$\mathcal{M}_F$	$\triangleq$	Global Markov Decision Process, $\mathcal{M}_F := \langle \mathcal{S}, \mathcal{A}, P, \mu, H, F \rangle$
F	$\triangleq$	Global reward function $F : \Gamma_H \to \mathbb{R}$
$\mathcal{C}_\mathcal{M}$	$\triangleq$	CMP constraint, see definition 8

Policies

$\Pi_{\mathrm{M}}^{\mathrm{S}}$	$\triangleq$	Class of Markovian stationary policies
$\Pi_{\mathrm{M}}^{\mathrm{NS}}$	$\triangleq$	Class of Markovian non-stationary policies
Π_{NM}	$\triangleq$	Class of non-Markovian policies
$\mathcal{H}_t$	$\triangleq$	History space until step t
π	$\triangleq$	Markovian policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ or non-Markovian policy $\pi : \mathcal{H}_t \to \Delta(\mathcal{A})$
π_t	$\triangleq$	Markovian Non-stationary policy $\pi_t : \mathcal{S} \times \mathcal{T} \to \Delta(\mathcal{A})$

Combinatorial Structure and Submodular Optimization

V	$\triangleq$	Ground set of elements e.g., $V = [N]$
f	$\triangleq$	Family of subsets induced by a ground set e.g., $f := \{0,1\}^V = 2^V$
F	$\triangleq$	Pseudo-Boolean set-function $F : 2^V \to \mathbb{R}$
k_F	$\triangleq$	Submodular curvature, see Definition 6
k^F	$\triangleq$	Supermodular curvature, see Definition 6

Algorithms: GTO and GPO

$\mathcal{R}(\pi)$	$\triangleq$	Non-additive suboptimality gap of policy π, see equations (5, 9)

B. Dynamics constraints

Towards interpreting GRL as a CO problem, we introduce a type of constraint set that we call *dynamics constraint*, and a trajectory-optimization version of GRL. But first, we need to introduce the following auxiliary notion.

Definition 7 (Time-extended CMP)**.** *Given a CMP $\mathcal{M} \coloneqq \langle \mathcal{S}, \mathcal{A}, P, H, \mu \rangle$, we call time-extended CMP, the new CMP $\mathcal{M}_l$ defined by $\mathcal{M}_l \coloneqq \langle V \coloneqq \mathcal{S} \times [l], \mathcal{A}, P_l, H, \mu \rangle$, which is a l-layered CMP where P_l is defined as:*

$$P_l((s', t') \mid (s, t), a) = \begin{cases} P(s' \mid s, a) & \text{if } t' = t + 1 \\ 0 & \text{otherwise} \end{cases}$$

Now we can define the dynamics constraint $\mathcal{C}_{\mathcal{M}}$ as follows.

Definition 8 (Dynamics constraint $\mathcal{C}_{\mathcal{M}}$)**.** *Given a CMP $\mathcal{M} \coloneqq \langle \mathcal{S}, \mathcal{A}, P, H, \mu \rangle$, we consider the time-extended CMP $\mathcal{M}_H \coloneqq \langle V \coloneqq \mathcal{S} \times \mathcal{T}, \mathcal{A}, P_H, H, \mu \rangle$ and define $\mathcal{C}_{\mathcal{M}}$ as:*

$$\mathcal{C}_{\mathcal{M}} \coloneqq \{\tau \subseteq V \mid \tau \text{ is a path in } \mathcal{G}_P\}$$

where $\mathcal{G}_P = (V, E)$ is the graph induced by P_H where $e = ((s, t), (s', t')) \in E \iff \exists a \in \mathcal{A}$ s.t. $P_H((s', t') \mid (s, t), a)) > 0$.

Notice we can interpret a dynamics constraint $\mathcal{C}_{\mathcal{M}}$ as the set of admissible trajectories of a CMP $\mathcal{M}$.

C. Proofs

C.1. Proofs for Section 4

Proposition 1 (Single Trial Convex RL $\subseteq$ Global RL)**.** *Given an instance $\mathcal{I}^+$ of ST-CRL it is possible to reduce it to an instance $\mathcal{I}_+$ of GRL* (1)*.*

Proof. For the sake of clarity, without loss of generality, we consider an empirical distribution defined over $\mathcal{S} \times \mathcal{T}$. Notice that this is indeed general since in the case the original instance does not take into account the time dimension, e.g., optimizes over distributions $d \in \Delta(\mathcal{S})$, then it is sufficient to define a functional f' marginalizing the input distribution, e.g., $d' \in \Delta(\mathcal{S} \times \mathcal{T})$ over the time dimension. Moreover, by considering trajectories composed of state-action pairs rather than only states, it is trivial to extend the following proof to the case of distributions over $\mathcal{S} \times \mathcal{A} \times \mathcal{T}$. Given an instance of the Single Trial General Utilities RL, namely:

$$\max_{\pi \in \Pi} \left(\mathop{\mathbb{E}}_{d \sim p^\pi} \big[f(d) \big] \right) \tag{9}$$

we show how to build an equivalent instance of the GRL problem, namely:

$$\max_{\pi \in \Pi} \mathop{\mathbb{E}}_{\tau \sim p_\pi} \big[F(\tau) \big] \tag{10}$$

such that the two problems are equivalent. In particular, we prove a stronger result than equivalence on the set of maximizers. We show that the objective functions are equal, formally:

$$\mathop{\mathbb{E}}_{d \sim p^\pi} \big[f(d) \big] = \mathop{\mathbb{E}}_{\tau \sim p_\pi} \big[F(\tau) \big] \quad \forall \pi \in \Pi$$

We build the instance of the GRL problem as follows. First, we realize that every empirical distribution $d \sim p^\pi$ must be induced by a trajectory τ following policy π due to the definition of p^π. Hence, for the sake of clarity, we express a distribution of this type as d_τ. As a consequence, for every empirical distribution d_τ we can define the trajectory τ inducing that distribution. This is trivial since d_τ is fundamentally an indicator function over $\mathcal{S} \times \mathcal{A} \times \mathcal{T}$ such that $d_\tau(s, t) = \mathbf{1}_{\tau(t) = (s)}$, where $\tau(t)$ represent the state visited within the trajectory τ at time step t. Since there exists a bijection between the space of trajectories τ and the space of empirical distributions d_τ, we can express the probability of sampling a certain trajectory τ, namely p_τ as follows:

$$p_\pi(\tau) \coloneqq p^\pi(d_\tau) \tag{11}$$

For the sake of clarity, we prove Equation (11) explicitly by leveraging the definitions of $p_\pi(\tau)$ and $p^\pi(d_\tau)$. In particular, given a policy π, we want to prove that for any trajectory τ it holds that $p_\pi(\tau) = p^\pi(d_\tau)$. Consider a trajectory $\tau =$

$\{((s_0,0),a_0),((s_1,1),a_1),\dots,((s_t,t),a_t),\dots,((s_{H-1},H-1),a_{H-1})\}$ and the empirical distribution $d_\tau \in \Delta(\mathcal{V}) = \Delta(\mathcal{S}\times\mathcal{T})$. Consider the function $d_\tau^t = (s,t) \in \mathcal{V} : d_\tau > 0$, which returns the state-time pair (s,t) visited by trajectory τ in time-step t. Now we can express $p_\pi(\tau)$ and $p^\pi(d_\tau)$ as follows:

$$p_\pi(\tau) = \mu((s_0,0)) \prod_{t=0}^{H-1} P((s_{t+1},t+1)|(s_t,t),a_t)\pi(a_t|(s_t,t)) \tag{12}$$

$$p^\pi(d_\tau) = \mu(d_\tau^0) \prod_{t=0}^{H-1} P(d_\tau^{t+1}|d_\tau^t,a_t)\pi(a_t|d_\tau^t) \tag{13}$$

Crucially, it is immediate to notice that, due to the definition of $d_\tau^t \forall t \in \mathcal{T}$ we have that:

$$\begin{aligned}\mu((s_0,0)) &= \mu(d_\tau^0)\\ P((s_{t+1},t+1)|(s_t,t),a_t) &= P(d_\tau^{t+1}|d_\tau^t,a_t)\\ \pi(a_t|(s_t,t)) &= \pi(a_t|d_\tau^t)\end{aligned}$$

Ultimately, we define the GRL global reward F as:

$$F(\tau) = f(d_\tau)$$

Hence, by construction of the GRL instance, we have that $\forall \pi \in \Pi$:

$$\begin{aligned}\mathop{\mathbb{E}}_{d\sim p^\pi}\big[f(d)\big] &= \sum_d f(d)p^\pi(d)\\ &= \sum_{d_\tau} f(d_\tau)p^\pi(d_\tau)\\ &= \sum_{d_\tau} f(d_\tau)p_\pi(\tau)\\ &= \sum_\tau F(\tau)p_\pi(\tau)\\ &= \mathop{\mathbb{E}}_{\tau\sim p_\pi}\big[F(\tau)\big]\end{aligned}$$

Intuitively, it is likely possible to show also the other direction of the inclusion, i.e., GRL $\subseteq$ Single Trial General Utilities RL, and therefore that the two problem classes are equivalent. Although interesting, this is not necessary in order to prove the theorem stated. □

C.2. Proofs for Section 7

Theorem 7.1 (Approximation Guarantees GPO)**.** *Let* $\mathcal{J}^\star \coloneqq \max_{\pi\in\Pi_{\text{NM}}} \mathbb{E}_{\tau\sim p_\pi}\big[F(\tau)\big]$ *and* π_1 *the policy resulting from one iteration of* GPO *on a GMDP* $\mathcal{M}_F$*. Then* GPO *guarantees that for*

i) Monotone submodular reward function, F

$$\mathcal{R}(\pi_1) \le k_F \mathcal{J}^\star,$$

ii) Monotone supermodular reward function, F

$$\mathcal{R}(\pi_1) \le \frac{2k^F - (k^F)^2}{1-k^F}\mathcal{J}^\star,$$

iii) BP reward function, $F = Q + G$

$$\mathcal{R}(\pi_1) \le \alpha\mathcal{J}^\star,\ \alpha = \begin{cases}\frac{2k^G-(k^G)^2}{1-k^G} & \text{if } k_F \le k^G\\ \frac{1-(1-k_Q)(1-k^G)}{1-k^G} & \text{otherwise}\end{cases}$$

Proof. The proof is composed of three sequential parts proving each one of the three cases independently. We first show that $\mathcal{J}(\pi_1) \geq (1 - k_F)\mathcal{J}(\pi^*)$ where $\mathcal{J}(\pi) = \mathbb{E}_{\tau\sim\pi}\left[F(\tau)\right]$. Once this is proved, the theorem statement can be trivially derived by using the definition of $\mathcal{V}(\pi)$.

$$
\begin{aligned}
\mathcal{J}(\pi_1) &\overset{(1)}{\geq} m^{\sigma,E}_{\pi_0}(\pi_1) && \text{(lower bound)}\\
&\overset{(2)}{\geq} m^{\sigma,E}_{\pi_0}(\pi^*) && (\pi_1 \text{ maximizer of } m^{\sigma,E}_{\pi_0})\\
&\overset{(3)}{=} \mathop{\mathbb{E}}_{\tau^*\sim\pi^*}\left[\mathop{\mathbb{E}}_{\tau_0\sim\pi_0}\left[m^{\sigma}_{\tau_0}(\tau^*)\right]\right] && (\text{def. } m^{\sigma,E}_{\pi_0}(\pi^*))\\
&= \mathop{\mathbb{E}}_{\tau^*\sim\pi^*}\left[\mathop{\mathbb{E}}_{\tau_0\sim\pi_0}\left[\sum_{x\in\tau^*} m^{\sigma}_{\tau_0}(I_{\tau_0}(x))\right]\right]\\
&= \mathop{\mathbb{E}}_{\tau^*\sim\pi^*}\left[\mathop{\mathbb{E}}_{\tau_0\sim\pi_0}\left[\sum_{x\in\tau^*} F\big(S^{\sigma(\tau_0)}_{I_{\tau_0}(x)}\big) - F\big(S^{\sigma(\tau_0)}_{I_{\tau_0}(x)-1}\big)\right]\right]\\
&= \mathop{\mathbb{E}}_{\tau^*\sim\pi^*}\left[\mathop{\mathbb{E}}_{\tau_0\sim\pi_0}\left[\sum_{x\in\tau^*} F(\{\sigma(\tau_0)[1],\dots,\sigma(\tau_0)[I_{\tau_0}(x)]\}) - F(\{\sigma(\tau_0)[1],\dots,\sigma(\tau_0)[I_{\tau_0}(x)-1]\})\right]\right]\\
&= \mathop{\mathbb{E}}_{\tau^*\sim\pi^*}\left[\mathop{\mathbb{E}}_{\tau_0\sim\pi_0}\left[\sum_{x\in\tau^*} F(\sigma(\tau_0)[I_{\tau_0}(x)] \mid \{\sigma(\tau_0)[1],\dots,\sigma(\tau_0)[I_{\tau_0}(x)-1]\})\right]\right]\\
&\geq (1-k_F)\mathop{\mathbb{E}}_{\tau^*\sim\pi^*}\left[\mathop{\mathbb{E}}_{\tau_0\sim\pi_0}\left[\sum_{x\in\tau^*} F(\sigma(\tau_0)[I_{\tau_0}(x)])\right]\right]\\
&\overset{(4)}{=} (1-k_F)\mathop{\mathbb{E}}_{\tau^*\sim\pi^*}\left[\mathop{\mathbb{E}}_{\tau_0\sim\pi_0}\left[\sum_{x\in\tau^*} F(x)\right]\right]\\
&\geq (1-k_F)\mathop{\mathbb{E}}_{\tau^*\sim\pi^*}\left[\mathop{\mathbb{E}}_{\tau_0\sim\pi_0}\left[F(\tau^*)\right]\right]\\
&\overset{(5)}{\geq} (1-k_F)\mathop{\mathbb{E}}_{\tau^*\sim\pi^*}\left[F(\tau^*)\right]\\
&= (1-k_F)\mathcal{J}(\pi^*)
\end{aligned}
$$

We define $I_\tau(x)$ as the function returning the index of element x in trajectory τ. This proof is heavily based on the proof of theorem D.1. Analogously, in step (1) we use the fact that $m^{\sigma,E}_{\pi_0}(\pi_1)$ is by construction a lower bound of $\mathcal{J}(\pi_1)$, in step (2) we notice that π_1 is a maximizer of $m^{\sigma,E}_{\pi_0}(\cdot)$ due to the local optimization step in GPO, while in step (3) we leverage the definition of $m^{\sigma,E}_{\pi_0}(\pi^*)$. In step (4) we leverage the notion of submodular curvature (Definition 6), and in step (5) we exploit submodularity of F (Definition 2).

We now prove the second statement. This proof is heavily based on the proof of theorem D.2. We first show that $\mathcal{J}(\pi_1) \geq \frac{(k^F)^2-3k^F+1}{1-k^F}\mathcal{J}(\pi^*)$ where $\mathcal{J}(\pi) = \mathbb{E}_{\tau\sim\pi}\left[F(\tau)\right]$. Once this is proved, the theorem statement can be trivially derived by using the definition of $\mathcal{V}(\pi)$.

$$
\begin{aligned}
\mathcal{J}(\pi_1) &\overset{(1)}{\geq} m^{E}_{\pi_0}(\pi_1)\\
&\overset{(2)}{\geq} m^{E}_{\pi_0}(\pi^*)\\
&\overset{(3)}{=} \mathop{\mathbb{E}}_{\tau^*\sim\pi^*}\left[\mathop{\mathbb{E}}_{\tau_0\sim\pi_0}\left[m_{\tau_0}(\tau^*)\right]\right]\\
&\overset{(4)}{\geq} \mathop{\mathbb{E}}_{\tau^*\sim\pi^*}\left[\mathop{\mathbb{E}}_{\tau_0\sim\pi_0}\left[(1-k^F)F(\tau^*) - \frac{k^F}{1-k^F}\sum_{j\in\tau_0\setminus\tau^*} F(j)\right]\right]\\
&\overset{(5)}{\geq} (1-k^F)\mathop{\mathbb{E}}_{\tau^*\sim\pi^*}\left[\mathop{\mathbb{E}}_{\tau_0\sim\pi_0}\left[F(\tau^*)\right]\right] - \frac{k^F}{1-k^F}\mathop{\mathbb{E}}_{\tau^*\sim\pi^*}\left[\mathop{\mathbb{E}}_{\tau_0\sim\pi_0}\left[\sum_{j\in\tau_0\setminus\tau^*} F(j)\right]\right]
\end{aligned}
\tag{14}
$$

In step (1) we use the fact that $m^E_{\pi_0}(\pi_1)$ is by construction a lower bound of $\mathcal{J}(\pi_1)$, in step (2) we notice that π_1 is a maximizer of $m^E_{\pi_0}(\cdot)$ due to the local optimization step in GPO, while in step (3) we leverage the definition of $m^E_{\pi_0}(\pi^*)$. Meanwhile, in step (4), we trivially follow steps (1) to (1) of the proof of theorem D.2. Now, we lower bound the second term of equation 14 (without the preceding constant component) as follows:

$$\begin{aligned}
-\mathop{\mathbb{E}}_{\tau^*\sim\pi^*}\left[\mathop{\mathbb{E}}_{\tau_0\sim\pi_0}\left[\sum_{j\in\tau_0\setminus\tau^*}F(j)\right]\right] &\geq -\mathop{\mathbb{E}}_{\tau^*\sim\pi^*}\left[\mathop{\mathbb{E}}_{\tau_0\sim\pi_0}\left[F(\tau_0)\right]\right] \\
&= -\mathop{\mathbb{E}}_{\tau_0\sim\pi_0}\left[F(\tau_0)\right] \\
&= -\mathcal{J}(\pi_0) \\
&\geq -J(\pi^*)
\end{aligned} \tag{15}$$

By plugging the result in eq. 15 into equation 14, we obtain:

$$\begin{aligned}
\mathcal{J}(\pi_1) &\geq (1-k^F)\mathcal{J}(\pi^*) - \frac{k^F}{1-k^F}\mathcal{J}(\pi^*) \\
&= \frac{(k^F)^2 - 3k^F + 1}{1-k^F}\mathcal{J}(\pi^*)
\end{aligned}$$

where we have employed the definition of $\mathcal{J}(\pi^*)$. Notice that a critical aspect of this proof is that we cannot compare $F(\tau^*)$ and $F(\tau)$ as we did for the proofs in the deterministic setting, since in the stochastic setting the notion of optimality is defined over the policy space and therefore we can only compare

$$\mathcal{J}(\pi^*) = \mathop{\mathbb{E}}_{\tau^*\sim\pi^*}\left[F(\tau^*)\right] \geq \mathop{\mathbb{E}}_{\tau\sim\pi}\left[F(\tau)\right] = \mathcal{J}(\pi) \quad \forall \pi \in \Pi$$

We now prove the third statement. First, notice that:

$$\begin{aligned}
\mathcal{J}(\pi) &= \mathop{\mathbb{E}}_{\tau\sim\pi}\left[F(\tau)\right] \\
&= \mathop{\mathbb{E}}_{\tau\sim\pi}\left[Q(\tau) + G(\tau)\right] \\
&= \mathop{\mathbb{E}}_{\tau\sim\pi}\left[Q(\tau)\right] + \mathop{\mathbb{E}}_{\tau\sim\pi}\left[G(\tau)\right] \\
&= \mathcal{J}_Q(\pi) + \mathcal{J}_G(\pi)
\end{aligned} \tag{16}$$

where we define:

$$\mathcal{J}_Q \coloneqq \mathop{\mathbb{E}}_{\tau\sim\pi}\left[Q(\tau)\right] \qquad \mathcal{J}_G \coloneqq \mathop{\mathbb{E}}_{\tau\sim\pi}\left[G(\tau)\right]$$

and:

$$\pi^* \in \arg\max_\pi \mathcal{J}(\pi) \quad \pi^*_Q \in \arg\max_\pi \mathcal{J}_Q(\pi) \quad \pi^*_G \in \arg\max_\pi \mathcal{J}_G(\pi)$$

Given these quantities, we can proceed with the core of the proof. We first show that $\mathcal{J}(\pi_1) \geq \alpha \cdot \mathcal{J}(\pi^*)$ for the value of α stated within the theorem. From this, we deduce that $\mathcal{J}(\hat{\pi}) \geq \alpha \cdot \mathcal{J}(\pi^*)$ since further policy updates are performed only if

they don't worsen the policy performance according to $\mathcal{J}$.

$$\begin{aligned}
\mathcal{J}(\pi_1) &\overset{(1)}{=} \mathcal{J}_Q(\pi_1) + \mathcal{J}_G(\pi_1) \\
&\overset{(2)}{\geq} m^{\sigma,E}_{\pi_0}(\pi_1) + m^{E}_{\pi_0}(\pi_1) \\
&\overset{(3)}{\geq} m^{\sigma,E}_{\pi_0}(\pi^*) + m^{E}_{\pi_0}(\pi^*) \\
&\overset{(4)}{=} \mathop{\mathbb{E}}_{\tau^*\sim\pi^*}\left[\mathop{\mathbb{E}}_{\tau_0\sim\pi_0}\left[m^{\sigma}_{\tau_0}(\tau^*)\right]\right] + \mathop{\mathbb{E}}_{\tau^*\sim\pi^*}\left[\mathop{\mathbb{E}}_{\tau_0\sim\pi_0}\left[m_{\tau_0}(\tau^*)\right]\right] \\
&= \mathop{\mathbb{E}}_{\tau^*\sim\pi^*}\left[\mathop{\mathbb{E}}_{\tau_0\sim\pi_0}\left[m^{\sigma}_{\tau_0}(\tau^*) + m_{\tau_0}(\tau^*)\right]\right] \\
&\overset{(5)}{\geq} \mathop{\mathbb{E}}_{\tau^*\sim\pi^*}\left[\mathop{\mathbb{E}}_{\tau_0\sim\pi_0}\left[(1-k_Q)Q(\tau^*) + m_{\tau_0}(\tau^*)\right]\right] \\
&= (1-k_Q)\mathcal{J}_Q(\pi^*) + \mathop{\mathbb{E}}_{\tau^*\sim\pi^*}\left[\mathop{\mathbb{E}}_{\tau_0\sim\pi_0}\left[m_{\tau_0}(\tau^*)\right]\right]
\end{aligned} \tag{17}$$

In step (1) we use equation (16), in step (2) we use the fact that $m^{\sigma,E}_{\pi_0}(\pi_1)$ and $m^{E}_{\pi_0}(\pi_1)$ are by construction respectively lower bounds of $\mathcal{J}_Q(\pi_1)$ and $\mathcal{J}_G(\pi_1)$. In step (3), we notice that π_1 is a maximizer of the sum of marginal lower bounds, namely $m^{\sigma,E}_{\pi_0}(\pi^*) + m^{E}_{\pi_0}(\pi^*)$ due to the local optimization step in GPO. Meanwhile in step (4) we leverage the definition of $m^{\sigma,E}_{\pi_0}(\pi^*)$ and $m^{E}_{\pi_0}(\pi^*)$, and step (5) can be trivially derived by following steps (2) to (4) used to prove theorem D.1. Now we can lower bound the second term of equation 17 as:

$$\begin{aligned}
\mathop{\mathbb{E}}_{\tau^*\sim\pi^*}\left[\mathop{\mathbb{E}}_{\tau_0\sim\pi_0}\left[m_{\tau_0}(\tau^*)\right]\right] &\overset{(1)}{\geq} \mathop{\mathbb{E}}_{\tau^*\sim\pi^*}\left[\mathop{\mathbb{E}}_{\tau_0\sim\pi_0}\left[(1-k^G)G(\tau^*) - \frac{k^G}{1-k^G}\sum_{j\in\tau_0\setminus\tau^*} G(j)\right]\right] \\
&= (1-k^G)\mathcal{J}_G(\pi^*) - \frac{k^G}{1-k^G}\mathop{\mathbb{E}}_{\tau^*\sim\pi^*}\left[\mathop{\mathbb{E}}_{\tau_0\sim\pi_0}\left[\sum_{j\in\tau_0\setminus\tau^*} G(j)\right]\right]
\end{aligned} \tag{18}$$

where step (1) can be trivially derived by following steps (2) to (4) used to prove theorem D.2. Now we can lower bound the second term of equation 18 (without the preceding constant component) as follows:

$$\begin{aligned}
-\mathop{\mathbb{E}}_{\tau^*\sim\pi^*}\left[\mathop{\mathbb{E}}_{\tau_0\sim\pi_0}\left[\sum_{j\in\tau_0\setminus\tau^*} G(j)\right]\right] &\overset{(1)}{\geq} -\mathop{\mathbb{E}}_{\tau^*\sim\pi^*}\left[\mathop{\mathbb{E}}_{\tau_0\sim\pi_0}\left[G(\tau_0)\right]\right] \\
&= -\mathcal{J}_Q(\pi_0) \\
&\geq -\mathcal{J}_Q(\pi^*_G) \\
&\overset{(2)}{\geq} -\mathcal{J}(\pi^*)
\end{aligned} \tag{19}$$

using in step (1) the fact that $(\tau_0\setminus\tau^*) \subset \tau_0$ and supermodularity of G, while the step (2) is due to the following trivial chain of inequalities:

$$\begin{aligned}
\mathcal{J}(\pi^*) &= \mathop{\mathbb{E}}_{\tau^*\sim\pi^*}\left[F(\tau^*)\right] \\
&\geq \mathop{\mathbb{E}}_{\tau^*_G\sim\pi^*_G}\left[F(\tau^*_G)\right] \\
&= \mathop{\mathbb{E}}_{\tau^*_G\sim\pi^*_G}\left[Q(\tau^*_G)\right] + \mathop{\mathbb{E}}_{\tau^*_G\sim\pi^*_G}\left[G(\tau^*_G)\right] \\
&\geq \mathcal{J}_G(\pi^*_G) \qquad \text{(Q non-negative)}
\end{aligned}$$

By plugging equation 19 into equation 18 and then equation 18 into equation 17 we obtain:

$$\begin{aligned}\mathcal{J}(\pi_1) &\geq (1-k_Q)\mathcal{J}_Q(\pi^*) + (1-k^G)\mathcal{J}_G(\pi^*) - \frac{k^G}{1-k^G}\mathcal{J}(\pi^*) \\ &= (1-k_Q)\mathcal{J}_Q(\pi^*) + (1-k^G)\mathcal{J}_G(\pi^*) - \frac{k^G}{1-k^G}\Big[\mathcal{J}_Q(\pi^*) + \mathcal{J}_G(\pi^*)\Big] \\ &= \frac{(1-k_Q)(1-k^G)-k^G}{1-k^G}\mathcal{J}_Q(\pi^*) + \frac{(1-k^G)^2-k^G}{1-k^G}\mathcal{J}_Q(\pi^*) \end{aligned} \tag{20}$$

and by trivially lower bounding equation 20 depending on the value of k_Q and k^G leads to the theorem statement. □

C.3. Auxiliary Lemmas

Lemma C.1 (Supermodular telescoping bound)**.** *Consider a supermodular function F defined over the ground set V. For every set $S \subseteq V$ we have*

$$\sum_{j\in S} F(j) \geq (1-k^F)F(S)$$

Proof. First, we order the elements of the set $S = \{j_1, \ldots, j_n\}$, then we have

$$\begin{aligned}\sum_{j\in S} F(j) &\geq (1-k^F)\big[F(j_1) + F(j_2 \mid j_1) + \ldots + F(j_n \mid j_1, \ldots, l_{n-1})\big] && \text{curvature inequality} \\ &= (1-k^F)F(S) && \text{telescoping sum}\end{aligned}$$

□

D. Analysis Deterministic Case via Trajectory Optimization

Analogously to the analysis presented in section 7, it is possible to study the approximation guarantees achieved by algorithm GTO for the trajectory-optimization version of the problem for deterministic GMDPs as presented in equation (3).

For the sake of completeness, in the following we specialize the concept of Non-additive Suboptimality Gap (Definition 5) in order to capture the sub-optimality gap achieved by a given trajectory in a deterministic GMDP rather than a policy in a possibly stochastic GMDP.

Definition 9 (Non-additive Suboptimality Gap: Trajectory-Optimization, Deterministic GMDP Version)**.** *Consider a trajectory $\tau \in \mathcal{C}_{\mathcal{M}}$, where $\mathcal{C}_{\mathcal{M}}$ denotes the CMP constraint i.e., the set of admissible trajectories, for a given Global MDP $\mathcal{M}_F$. We define the non-additive suboptimality gap $\mathcal{R}(\tau)$ of a trajectory τ as:*

$$\mathcal{R}(\tau) \coloneqq F^* - F(\tau) \tag{21}$$

where $F^* \coloneqq \max_{\tau \in \mathcal{C}_{\mathcal{M}}} \quad F(\tau)$ and we implicitly consider an arbitrary initial starting state s_0.

Given this definition we can state the following guarantees.

Theorem D.1 (Approximation Guarantee GTO, F submodular, Deterministic Case)**.** *By running for one iteration algorithm* GTO *on a GMDP $\mathcal{M}_F$ with F submodular, we obtain a trajectory τ_1 such that:*

$$\mathcal{R}(\tau_1) \leq k_F F^*$$

where $F^* \coloneqq \max_{\tau \in \mathcal{C}_{\mathcal{M}}} \quad F(\tau)$.

Proof. We first show that $F(\tau_1) \geq (1 - k_F) F(\tau^*)$ where $\tau^* \in \arg\max_{\tau \in \mathcal{C}_{\mathcal{M}}} \quad F(\tau)$. Once this is proved, the theorem statement can be trivially derived by using the definition of $\mathcal{R}(\tau)$.

$$\begin{aligned}
F(\tau_1) &\overset{(1)}{\geq} m^{\sigma}_{\tau_0}(\tau_1) \\
&\overset{(2)}{\geq} m^{\sigma}_{\tau_0}(\tau^*) \\
&= \sum_{x \in \tau^*} m^{\sigma}_{\tau_0}(I_{\tau_0}(x)) \\
&= \sum_{x \in \tau^*} F\big(S^{\sigma(\tau_0)}_{I_{\tau_0}(x)}\big) - F\big(S^{\sigma(\tau_0)}_{I_{\tau_0}(x)-1}\big) \\
&= \sum_{x \in \tau^*} F(\{\sigma(\tau_0)[1], \dots, \sigma(\tau_0)[I_{\tau_0}(x)]\}) - F(\{\sigma(\tau_0)[1], \dots, \sigma(\tau_0)[I_{\tau_0}(x) - 1]\}) \\
&= \sum_{x \in \tau^*} F(\sigma(\tau_0)[I_{\tau_0}(x)] \mid \{\sigma(\tau_0)[1], \dots, \sigma(\tau_0)[I_{\tau_0}(x) - 1]\}) \\
&\overset{(3)}{\geq} (1 - k_F) \sum_{x \in \tau^*} F(\sigma(\tau_0)[I_{\tau_0}(x)]) \\
&= (1 - k_F) \sum_{x \in \tau^*} F(x) \\
&\overset{(4)}{\geq} (1 - k_F) F(\tau^*)
\end{aligned}$$

□

We define $I_\tau(x)$ as the function returning the index of element x in trajectory τ. In step (1) we use the fact that $m^{\sigma}_{\tau_0}(\tau_1)$ is by construction a lower bound of $F(\tau_1)$, in step (2) we notice that τ_1 is a maximizer of $m^{\sigma}_{\tau_0}(\cdot)$ due to the local optimization step in GTO, in step (3) we leverage the notion of submodular curvature (Definition 6), and in step (4) we exploit submodularity of F (Definition 2).

Theorem D.2 (Approximation Guarantee GTO, F supermodular, Deterministic Case)**.** *By running for one iteration algorithm* GTO *on a GMDP $\mathcal{M}_F$ with F supermodular, we obtain a trajectory τ_1 such that:*

$$\mathcal{R}(\tau_1) \leq \frac{2k^F - (k^F)^2}{1 - k^F} F^*$$

where $F^* \coloneqq \max_{\tau \in \mathcal{C}_\mathcal{M}} \quad F(\tau)$.

Proof. We first show that $F(\tau_1) \geq \frac{(k^F)^2 - 3k^F + 1}{1 - k^F} F(\tau^*)$ where $\tau^* \in \arg\max_{\tau \in \mathcal{C}_\mathcal{M}} \quad F(\tau)$. Once this is proved, the theorem statement can be trivially derived by using the definition of $\mathcal{R}(\tau)$.

$$\begin{aligned}
F(\tau_1) &\overset{(1)}{\geq} m_{\tau_0}(\tau_1) \\
&\overset{(2)}{\geq} m_{\tau_0}(\tau^*) \\
&= F(\tau_0) - \sum_{j \in \tau_0 \backslash \tau^*} F(j \mid \tau_0 \backslash j) + \sum_{j \in \tau^* \backslash \tau_0} F(j) \\
&= F(\tau_0) - \sum_{j \in B} F(j \mid \tau_0 \backslash j) + \sum_{j \in C} F(j) \\
&\geq \sum_{j \in A \cup B} F(j) - \sum_{j \in B} F(j \mid \tau_0 \backslash j) + \sum_{j \in C} F(j) \\
&= \sum_{j \in A \cup C} F(j) - \sum_{j \in B} F(j \mid \tau_0 \backslash j) + \sum_{j \in B} F(j) \\
&\overset{(3)}{\geq} \sum_{j \in A \cup C} F(j) - \frac{1}{1 - k^F} \sum_{j \in B} F(j) + \sum_{j \in B} F(j) \\
&\geq \sum_{j \in A \cup C} F(j) - \frac{k^F}{1 - k^F} \sum_{j \in B} F(j) \\
&\overset{(4)}{\geq} (1 - k^F) F(\tau^*) - \frac{k^F}{1 - k^F} \sum_{j \in B} F(j) \\
&\overset{(5)}{\geq} (1 - k^F) F(\tau^*) - \frac{k^F}{1 - k^F} F(\tau^*) \\
&= \frac{(k^F)^2 - 3k^F + 1}{1 - k^F} F(\tau^*)
\end{aligned}$$

where where we have defined $A \coloneqq \tau_0 \cap \tau^*, B \coloneqq \tau_0 \backslash \tau^*, C \coloneqq \tau^* \backslash \tau_0$. In step (1) we use the fact that $m^\sigma_{\tau_0}(\tau_1)$ is by construction a lower bound of $F(\tau_1)$, in step (2) we notice that τ_1 is a maximizer of $m_{\tau_0}(\cdot)$ due to the local optimization step in GTO, in step (3) we leverage the notion of supermodular curvature (Definition 6), and in step (4) we use Lemma C.1. Meanwhile, step (5) is due to the following chain of inequalities:

$$\begin{aligned}
\sum_{j \in B} F(j) &\leq F(B) && \text{(supermodularity)} \\
&\leq F(\tau_0) && \text{(monotonicity)} \\
&\leq F(\tau^*) && \text{(optimality of } \tau^*\text{)}
\end{aligned}$$

□

Theorem D.3 (Approximation Guarantee GTO, F BP, Deterministic Case)**.** *By running for one iteration algorithm* GTO *on a GMDP $\mathcal{M}_F$ with $F = Q + G$ BP, we obtain a trajectory τ_1 such that:*

$$\mathcal{R}(\tau_1) \leq \alpha F^*$$

with

$$\alpha = \begin{cases} \frac{2k^G - (k^G)^2}{1-k^G} & \text{if } k_F \leq k^G \\ \frac{1-(1-k_Q)(1-k^G)}{1-k^G} & \text{otherwise} \end{cases}$$

where k_Q is the submodular curvature of Q, k^G is the supermodular curvature of G, and $F^* \coloneqq \max_{\tau \in \mathcal{C}_\mathcal{M}} \quad F(\tau)$.

Proof. We define $I_\tau(x)$ as the function returning the index of element x in trajectory τ. We first show that $F(\tau_1) \geq (1-\alpha)F(\tau^*)$ where $\tau^* \in \arg\max_{\tau \in \mathcal{C}_\mathcal{M}} \quad F(\tau)$. Once this is proved, the theorem statement can be trivially derived by using the definition of $\mathcal{R}(\tau)$.

$$\begin{aligned} F(\tau_1) &= Q(\tau_1) + G(\tau_1) \\ &\overset{(1)}{\geq} m^\sigma_{\tau_0}(\tau_1) + m_{\tau_0}(\tau_1) \\ &\overset{(2)}{\geq} m^\sigma_{\tau_0}(\tau^*) + m_{\tau_0}(\tau^*) \\ &\overset{(3)}{\geq} (1-k_Q)Q(\tau^*) + m_{\tau_0}(\tau^*) \end{aligned}$$

In step (1) we use the fact that $m^\sigma_{\tau_0}(\tau_1)$ and $m_{\tau_0}(\tau_1)$ are by construction respectively lower bounds of $Q(\tau_1)$ and $G(\tau_1)$. In step (2), we notice that τ_1 is a maximizer of the sum of marginal lower bounds, namely $m^\sigma_{\tau_0}(\cdot) + m_{\tau_0}(\cdot)$ due to the local optimization step in GTO. Meanwhile step (3) can be trivially derived by following steps (1) to (4) used to prove theorem D.1. Now, we define $A \coloneqq \tau_0 \cap \tau^*, B \coloneqq \tau_0 \backslash \tau^*, C \coloneqq \tau^* \backslash \tau_0$ and lower bound the second term as follows:

$$\begin{aligned} m_{\tau_0}(\tau^*) &\overset{(1)}{\geq} (1-k^G)G(\tau^*) - \frac{k^G}{1-k^G}\sum_{j \in B} G(j) \\ &\overset{(2)}{\geq} (1-k^G)G(\tau^*) - \frac{k^G}{1-k^G}G(\tau^*_G) \\ &\overset{(3)}{\geq} (1-k^G)G(\tau^*) - \frac{k^G}{1-k^G}F(\tau^*_G) \end{aligned} \tag{22}$$

where in step (1) we have trivially followed steps (2) to (4) used to prove theorem D.2, while in step (2) we have used the following chain of inequalities:

$$\sum_{j \in B} G(j) \leq G(B) \leq G(\tau_0) \leq G(\tau^*_G)$$

and in step (3) we have used the fact that:

$$F(\tau^*) \geq F(\tau^*_G) = Q(\tau^*_G) + G(\tau^*_G) \geq G(\tau^*_G)$$

since Q is non-negative. By plugging equation 22 into the initial chain of inequalities, we obtain:

$$\begin{aligned} F(\tau_1) &\geq (1-k_Q)Q(\tau^*) + (1-k^G)G(\tau^*) - \frac{k^G}{1-k^G}F(\tau^*) \\ &= (1-k_Q)Q(\tau^*) + (1-k^G)G(\tau^*) - \frac{k^G}{1-k^G}\left[Q(\tau^*) + G(\tau^*)\right] \\ &= \frac{(1-k_Q)(1-k^G) - k^G}{1-k^G}Q(\tau^*) + \frac{(1-k^G)^2 - k^G}{1-k^G}G(\tau^*) \end{aligned}$$

from which we can straightforwardly derive the statement by lower bounding the coefficients depending on the values of k_Q, k^G and using the definition of F. □

E. Computational Hardness

Theorem 7.2 (Hardness of GRL, trajectory-optimization (3))**.** *For all $0 \leq \beta \leq 1$, there exists an instance of a BP global reward $F = Q + G$ with $k^G = \beta$ such that no poly-time algorithm can achieve an approximation factor better than $1 - k^G + \epsilon \; \forall \epsilon > 0$ w.r.t. the Non-additive Suboptimality Gap in deterministic GMDPs, unless P = NP.*

Proof. The proof is based on a reduction from a problem with a known hardness result, namely (Bai & Bilmes, 2018, Theorem 4.1), which gives the same approximation ratio as in the lemma, but for the cardinality constrained case. We refer with $P1$ to the problem of BP maximization with a cardinality constraint, while with $P2$ to the problem of BP maximization under a CMP constraint. The reduction works as follows. First, we define a poly-time reduction from any instance of $P1$ to a specific instance of $P2$. In particular, given an instance of $P1$ with a function F, a ground set $\mathcal{V}$ and a cardinality constraint k, we define an instance of $P2$ with CMP constraint $\mathcal{C}_{\mathcal{M}}$ induced by the fully-connected CMP $\mathcal{M} := \langle \mathcal{V}, \mathcal{A}, P, H = k \rangle$ as explained in definition 8. The time-extended CMP $\mathcal{M}_H$ will be a k-layered CMP of which the state space is a k-fold cartesian product of the original ground set $\mathcal{V}$. We define the objective function of $P2$ as $F' : D \coloneqq \mathcal{V} \times [k] \to \mathbb{R}$ and $F(S_d) = F(\Pi S_d)$ where $S_d \subseteq D$ and $\Pi : \mathcal{V} \times [k] \to \mathcal{V}$ is a projector map that drops the time-coordinate of its input, e.g. $\Pi(\{(s,t),(s',t')\}) = \{s,s'\}$. For the sake of notational simplicity we write ΠS instead of $\Pi(S)$. Notice that the instance of $P2$ can be computed in poly-time w.r.t. the cardinality of the original ground set $\mathcal{V}$, which represents the complexity of the initial instance of $P1$. In order to show that the instance we have built for $P2$ is a valid one, it is left to show that $F' \in BP$. We start by noticing that

$$F'(S_d) = F(\Pi S_d) = Q(\Pi S_d) + G(\Pi S_d) = Q'(S_d) + G'(S_d)$$

where we have used the fact that $F \in BP$ and have defined $Q' := Q\Pi$ and $G' := G\Pi$. Since Q and G are non-negative then also Q' and G' must be non-negative, and since Q, G, Π are monotone then also Q' and G' are monotone. Moreover, once can easily check that Q' is submodular. Next, we show that G' is supermodular and that it preserves the supermodular curvature of G, i.e., $k^{G'} = k^G$. Consider the sets $A_d \subseteq B_d \subseteq D$ and the element $d \notin B_d$. In order to prove that G' is supermodular we must show that $G'(d \mid A_d) \leq G'(d \mid B_d)$. We define $S := \Pi S_d$ and write:

$$\begin{aligned}
G'(d \mid B_d) &= G'(B_d \cup d) - G'(B_d) \\
&= G(\Pi(B_d \cup d)) - G(\Pi B_d) \\
&= G(\Pi B_d \cup \Pi d) - G(\Pi B_d) \\
&= G(\Pi d \mid \Pi B_d) \\
&\geq G(\Pi d \mid \Pi A_d) && (A_d \subseteq B_d \implies \Pi A_d \subseteq \Pi B_d) \\
&= G(\Pi A_d \cup \Pi d) - G(\Pi A_d) \\
&= G(\Pi(A_d \cup d)) - G(\Pi A_d) \\
&= G'(A_d \cup d) - G'(A_d) \\
&= G'(d \mid A_d)
\end{aligned}$$

which proves that G' is supermodular. As for its curvature, we have:

$$\begin{aligned}
k^{G'} &= 1 - \min_{S_d \subseteq D, d \notin S_d} \frac{G'(d)}{G'(d \mid S_d)} \\
&= 1 - \min_{S_d \subseteq D, d \notin S_d} \frac{G(\Pi S_d)}{G(\Pi S_d \cup \Pi d) - G(\Pi S_d)} \\
&= 1 - \min_{S \subseteq \mathcal{V}, v \notin S} \frac{G(v)}{G(v \mid S)} \\
&= k^G
\end{aligned}$$

For the sake of contradiction, we now suppose that there exists a poly-time algorithm that can solve $P2$ by computing a set $\hat{S}$ such that for every function $F' \in BP$ and $\epsilon > 0$ we have:

$$F'(\hat{S}_d) > (1 - k^{G'} + \epsilon) F'(S^*_d) \tag{23}$$

where S_d^* is an optimizer of F'. We claim that eq. 23 implies that $F(\hat{S}_d) > (1 - k^G + \epsilon)F(S^*)$, where S^* is an optimizer of F. Which would be a contradiction with the aforementioned hardness result and would imply the result stated in the lemma. In order to prove that

$$F'(\hat{S}_d) > (1 - k^{G'} + \epsilon)F'(S_d^*) \implies F(\hat{S}_d) > (1 - k^G + \epsilon)F(S^*)$$

we notice that $F'(S_d) = F(\Pi S_d)$ by definition, and therefore it is left to prove that $F'(S_d^*) = F(S^*)$. By def. of F' we have that $F'(S_d^*) = F(\Pi S_d^*)$, hence it suffices to show that $\Pi S_d^* = S^*$. By contradiction, we suppose that $S^* \neq \Pi S_d^*$. By def. of S^* this would imply that $F(S^*) > F(\Pi S_d^*)$. But notice that $\forall S \subset \mathcal{V}$, the pre-image of S along Π is always a non-empty subset of D, namely $\Pi^{-1}(S)$ with $\Pi^{-1} : 2^V \to 2^D$. Therefore we can pick $\bar{S}_d \in \Pi^{-1}(S)$ and we would obtain that:

$$F'(\bar{S}_d) = F(\Pi \bar{S}_d) = F(S^*) > F(\Pi S_d^*) = F'(S_d^*)$$

which is a contradiction since S_d^* is a maximizer of F' by definition. This fact, together with the fact that $k^{G'} = k^G$ proves our claim. Ultimately, notice that once an optimal solution for $P2$ is computed, an optimal solution for $P1$ can be computed in poly-time. □

F. Algorithm

In this section, we first present two propositions, especially for the algorithm GTO that ensure the build modular functions are tight lower bound of the global reward function, and guarantee monotonic improvement. Then we present the GPO algorithm and finally discuss some efficient ways to build modular lower bounds.

Proposition 2 (Tight modular lower bound)**.** *Let $F : 2^V \to \mathbb{R}$ be a submodular function. For any set $X \subseteq V$, define permutation $\sigma : [|V|] \to V$ such that $S^\sigma_{|X|} = X$, where $S^\sigma_i = \{\sigma(1), \sigma(2), \dots, \sigma(i)\}$ and $S^\sigma_0 = \emptyset$. Define a modular function about the set X, $m^\sigma_X = \sum_{v \in X} m^\sigma_X(v)$ with entries for element $i \in [|V|]$ given by $m^\sigma_X(\sigma(i)) \coloneqq F(S^\sigma_i) - F(S^\sigma_{i-1})$. Then m^σ_X is a tight modular lower bound of the submodular function F, i.e., $m_X(X) = F(X)$ with $m_X(Y) \leq F(Y), \forall Y \subseteq V$.*

Proof. As per definition, $m^\sigma_X(X) = \sum_{v \in X} m^\sigma_X(v) = \sum_{i \in [|X|]} F(S^\sigma_i) - F(S^\sigma_{i-1}) = \sum_{i \in [|X|]} F(S^\sigma_i | S^\sigma_{i-1}) = F(X)$. The last equality follows since $S^\sigma_{|X|} = X$. Hence the modular function is tight at X. Next, we prove that it is lower bound, i.e., $\forall Y : m^\sigma_X(Y) \leq F(Y)$.

Let $Y = \{i_1, \dots i_k\}$, wlog, s.t, $i_j <^\sigma i_{j+1}$, i.e., the elements are arranged in Y as per permutation σ.

$$\begin{aligned} F(Y) &= \sum_{j=1}^{k} F(i_j | i_1, \dots i_{j-1}) && (\{i_1, \dots i_{j-1}\} \subseteq \{\sigma(1), \dots \sigma(j-1)\}) \\ &\geq \sum_{j=1}^{k} F(i_j | \sigma(1), \dots \sigma(j-1)) = m^\sigma(Y) \end{aligned}$$

□

Proposition 3 (Monotonic Improvement)**.** GTO *monotonically improves the objective function, i.e., at any iteration t, it holds that $F(\tau_{t+1}) \geq F(\tau_t)$.*

Proof. Let m_{τ_t} be a modular lower bound of a global reward function, F about the trajectory τ_t. Then,

$$\begin{aligned} F(\tau_t) &= \sum_{v \in \tau_t} m_{\tau_t}(v) \\ &\overset{(1)}{\leq} \sum_{v \in \tau_{t+1}} m_{\tau_t}(v) \\ &\overset{(2)}{\leq} F(\tau_{t+1}) \end{aligned}$$

In the above proof, (1) follows since τ_{t+1} is the optimal policy for modular rewards m_{τ_t} obtained by MDPSOLVER, e.g., value iteration. Step (2) follows since m_{τ_t} is a lower bound of the global reward function F. □

F.1. Policy optimization for stochastic GMDPs

In this section, we extend the approach defined for deterministic GMDPs to stochastic GMDPs. The core idea stays the same, i.e., we recursively approximate the stochastic GMDPs with stochastic linearized MDP and solve it with standard MDP tools. However, we need a mechanism to convert the stochastic GMDPs to linearized MDPs. What should we linearise it about?

We solve the problem in policy space and linearize the GMDP around the current policy. However, we can compute the lower bounds only around the sets. Policy can thus interpreted as distribution over the trajectories and we define the modular rewards around a policy π as:

$$m^E_\pi \coloneqq \mathop{\mathbb{E}}_{\tau \sim \pi}[m_\tau] \tag{24}$$

where $m^E_\pi \in \mathbb{R}^{\mathcal{V}}$ defines a modular reward around a policy π and $m_\tau \in \mathbb{R}^{\mathcal{V}}$ are reward computed around a trajectory τ. Furthermore we define the linearization of the GRL objective $J(\pi)$ around the policy π', by evaluating this Equation (24) for a policy π as,

$$m^E_{\pi'}(\pi) \coloneqq \mathop{\mathbb{E}}_{\tau \sim \pi} \mathop{\mathbb{E}}_{\tau' \sim \pi'}[m_{\tau'}(\tau)]$$

Algorithm 2 Global Policy Optimization (GPO)

1: **Initialize** GMDP, $\pi_1 \leftarrow random$, $t \leftarrow 0$,
2: **do**
3: $\quad t \leftarrow t + 1$
4: $\quad$ Estimate $m^E_{\pi_t} \coloneqq \mathbb{E}_{\tau \sim \pi_t}[m_\tau]$
5: $\quad \pi_{t+1} \leftarrow \textsc{MdpSolver}(\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, P, s_0, m^E_{\pi_t}, H \rangle)$
6: **while** $J(\pi_{t+1}) > J(\pi_t)$
7: Return π_t

Global Policy Optimization (GPO). The outline of the steps is given in Algorithm 2. The algorithm starts with an arbitrary policy π_1, which may be represented using e.g., a value function. In each iteration, we estimate a modular lower bound of the global reward function about the current policy π (Line 4) using Monte Carlo samples. In particular, we sample trajectories utilizing the current policy π_t in the GMDP and compute modular lower bound w.r.t. each of the sampled trajectories. On averaging across all the modular lower bounds, we define a reward for each state-time pair which forms a classic MDP with modular rewards. Given the modular rewards, we can deploy any MDPSOLVER, e.g., value iteration, policy iteration or linear program, etc that solves $\arg\max_\pi m^E_{\pi_t}(\pi)$ and results in the optimal policy for the linearized MDP $\mathcal{M}$ in Line 5. The algorithm continues and we linearize the GMDP about the improved stochastic policy and resolve it for a better policy. In every iteration, we compare the objective with the last policy empirically using samples. In case the objective doesn't improve anymore, we terminate in Line 7 with the best policy π_t.

F.2. Alternative modular lower bounds for submodular rewards

Computational complexity of computing lower bounds (LB's). The computation of LB is $\mathcal{O}(|\mathcal{S} \times \mathcal{T}|)$. Notably, solving a finite horizon MDP (Agarwal et al., 2019, c.f. Chapter 1.2), for example, using value iteration has per-iteration complexity of $\mathcal{O}(|\mathcal{S} \times \mathcal{T}|^2|\mathcal{A}|)$, and thus computing the lower bounds is a non-dominant operation that does not affect the computational complexity of the overall algorithm. The challenge of scaling to a large state space and horizon exists in finite horizon MDP solvers as well and is not exclusive to our approach. However, in order to make the computation of LB more efficient in practice, one can incorporate approaches tailored to the problem such as GPO-S, where the lower bound remains the same for a fixed state across multiple time steps and is empirically faster to compute. We next elaborate more on this.

Only state dependent lower bounds. In many applications, submodular rewards are naturally defined on the state space $\mathcal{S}$ rather than joint state time space $\mathcal{V}$. For instance, consider a submodular function $F' : 2^{\mathcal{S} \to \mathbb{R}}, F'(S) \coloneqq |\bigcup_{s \in S} D^s|$. We can build a submodular function $F : 2^{\mathcal{S} \times \mathcal{T}} \to \mathbb{R}$ using an operator $A : 2^{\mathcal{S} \times \mathcal{T}} \to 2^{\mathcal{S}}$ that drops the time indices and define $F(\tau) \coloneqq F'(A(\tau))$ (Prajapat et al., 2023, Section 2).

For such functions, we can build a modular lower bound about current trajectory $\tau = \{(s_i, i)\}_{i=0}^{H-1}$ with a permutation $\sigma = \{\tau, \mathcal{V} \backslash \tau\}$ as:

$$m^\sigma(s,t) \coloneqq \begin{cases} F(S^\sigma_i) - F(S^\sigma_{i-1}) & (s,t) \in \tau \\ \left(F\left(S^\sigma_i\right) - F\left(S^\sigma_{i-1}\right)\right)/H & (s,t) \in \tau_r \\ 0 & (s,t) \in \tau_v \end{cases} \tag{25}$$

where $S^\sigma_i = \{\sigma(1), \sigma(2), ..., \sigma(i)\}$, $(s,t) = \sigma(i)$, $\tau_r = \{(s,t) \in \mathcal{V} | (s, \cdot) \notin \tau\}$ is the set of state-time pair where the state is not yet visited by τ and $\tau_v = \{(s,t) \in \mathcal{V} | (s, \cdot) \in \tau, (s,t) \notin \tau\}$ is the set of the state-time pair where the state is visited but at a different time.

Note that these bounds are computationally more efficient to compute, especially in the case of large horizons. Essentially we compute marginal gain for each state not yet visited and assign this as a reward to that state for any future visit.

Greedy σ-permutation for lower bounds. In general, we can pick any permutation $\sigma_\tau = \{\tau, \mathcal{V} \backslash \tau\}$ randomly as long as first H elements are τ. This results in a valid modular lower bound and hence our theoretical results hold. However empirical performance may vary based on the permutation (c.f. Fig. 5). Here we present a strategy to build these lower bounds greedily. For simplicity, we present it for the state-dependent case explained above.

Define a permutation $\sigma = \{\tau, \tau_r, \tau_v\}$, where $\tau_r \coloneqq \{(s,t) \in \mathcal{V} | (s,\cdot) \notin \tau\}$ is the set of state-time pair where the state is not visited earlier and $\tau_v \coloneqq \{(s,t) \in \mathcal{V} | (s,\cdot) \in \tau, (s,t) \notin \tau\}$ is the set of state-time pair where the state is visited but at a different time. Ignoring the time dimension, we order the states in the set τ_r as follows,

$$\sigma \coloneqq \arg\max_{s} F'(\{s\} \cup S_i) - F'(S_i),$$

where $S_i^\sigma = \{\sigma(1), \sigma(2), \ldots, \sigma(i)\}$. In this permutation, we randomly shuffle the last τ_v states, which were visited but at different times. Using this permutation, we define the modular rewards, $m^\sigma(s,t)$, as given in Equation (25).

Note that in Equation (25), we divide modular reward by H for $(s,t) \in \tau_r$ which is a careful choice that ensures $m^\sigma(s,t)$ is a valid lower bound and provides equal weightage (reward) across all times in the modularized MDP for the exploring unvisited state. In particular, for a LB, $m^\sigma(s,t)$ to be valid requires $F(V) \leq \sum_{(s,t)\in V} m^\sigma(s,t)$ for all sets $V \in \mathcal{V}$. Consider a particular case of permutation $\sigma = \{\tau, V, ...\}$ where $V = \{(s',2)(s',6), ..., (s',H-1)\}$ is a set containing the same state s' for all times and for simplicity let $F(\tau \cup (s',2)) - F(\tau) = F((s',2))$. In this case, if we do not divide by H, a valid LB is $m^\sigma(s',2) = F((s',\cdot))$ and $m^\sigma(s,t) = 0, \forall t \neq 2$, i.e., only visiting s' at horizon $t = 2$ may not be possible due to MDP constraints and visiting it at any other time is not encouraged by the rewards. Hence, we use the normalized equal reward across all times to enhance exploring the unvisited state at any time.

G. Experiments Details

Computation of Non-Markovian policy. We compute the optimal non-Markovian policy by solving a linear program (LP). The LP is defined with optimization variables, $\pi(a|s)$, cost as objective (1), and constraints that $\pi(a|s)$ is a probability simplex. The cost is defined as an expectation over all the trajectories. Computing all the possible trajectories is computationally exponential in the horizon. Thus, we can compare against it only for deterministic environments with small horizons.

All experiments within Section 8 are run on a squared grid with $|\mathcal{S}| = 400$ and action space $\mathcal{A} = \{$left, up, down, right, stay$\}$. Each experiment is conducted over 20 runs and the empirical standard deviation is shown. Here, we first report a table summarizing the configuration shared by most experiments and subsequently we will list the deviations from this configuration for a subset of the experiments.

Variable	Value
env.cov_module	Matern
env.alpha	0.1
env.beta	2
env.stochasticity_degree	0 (deterministic), 0.1 (stochastic)
env.unsafety_penalty	500
env.n_traj_samples	1 (deterministic), 20 (stochastic)

Table 3: Base experimental configuration.

Bayesian D-Optimal Experimental Design. We have run the experiments with horizon $H = 10$ for 6 iterations of GTO.

Diverse Synergies. We have run the experiments with horizon $H = 8$ for 6 iterations of GPO.

Safe State Coverage. We have run the experiments with horizon $H = 20$ for 25 iterations of GTO.

In the following, we extend Section 8 to showcase the performances of GTO and GPO on a wider variety of global reward functions capturing more real-world applications and representative enough to later discuss important insights.

In the following experiments, we consider a squared grid with $|\mathcal{S}| = 100$, with action space $\mathcal{A} = \{left, up, down, right, stay\}$, horizon $H = 10$, and show the performance of running GTO and GPO for 15 iterations. Each experiment is conducted over 20 runs and the empirical standard deviation is shown in the following plots. Moreover, the trajectories illustrated in figure 5 are generated using $H = 31$ and 35 iterations of GTO. The plots for the Safe State Coverage experiment in Figure 10 have been created with $H = 20$, 25 iterations, and stochasticity degree of 0.05.

States Coverage. In Fig. 4, we consider the state-coverage submodular global reward function $F(\tau) \coloneqq |\bigcup_{s\in\tau} D^s|$, with D^s being a disk of size 2×2 containing the agent's current state, and its right, up, and right-up neighbouring states (Prajapat et al., 2022). Notice that this global reward is fully-curved. A policy maximizing the objective $\mathcal{J}(\pi)$ induced by F will try to explore the state space to maximize coverage according to the application-specific definition of the set D^s, which in practice depends on the sensors with which the agent is equipped.

Bounded Curvature Coverage. The notion of coverage can often be captured via a bounded-curvature submodular global reward, which we denote as *bounded curvature coverage*. It can be expressed as $F(\tau) = \sum_{s\in\mathcal{S}} \phi(\tau, s)$ where $\phi(\tau, s) = \mathbb{I}_{C(\tau,s)>0} \cdot [1 - \alpha(C(\tau, s) - 1)]$ and $C(\tau, s) \coloneqq |\{t \in [H] : (s, t) \in \tau\}|$. Similar to a classic coverage function presented above, this function value is increased by 1 once a state is visited for the first time, while it increases by an arbitrary value α when a state is visited again. Interestingly, one can prove that the submodular curvature of F has value $k_F = 1 - \alpha$. In the following plot, we consider a significantly curved instance, where $\alpha = 0.9$.

D-Optimal Experimental Design. Here we consider the optimal experimental design setting as introduced in Section 8. To ensure robustness in our findings, we conduct 20 experiments across 4 different environments.

Synergical Trajectory Selection. As previously mentioned, in the context of scientific discovery applications, it could be particularly relevant to model positive interactions or synergies among state within a certain trajectory. As an illustrative example, consider states representing atoms and trajectories encoding molecules. Certain combinations e.g., pairs, triplets

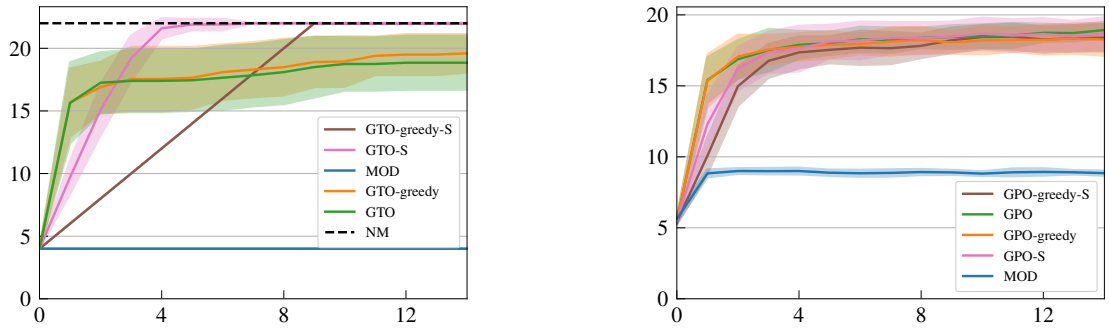


Figure 4: States Coverage: (left) values of $F(\tau)$ in deterministic GMDP setting where τ is the trajectory computed by GTO at each iteration (x-axis), which matches the optimal non-Markovian policy. (right) values of $\mathcal{J}(\pi)$ in stochastic GMDP setting, where π is the policy computed by GPO at each iteration (x-axis).

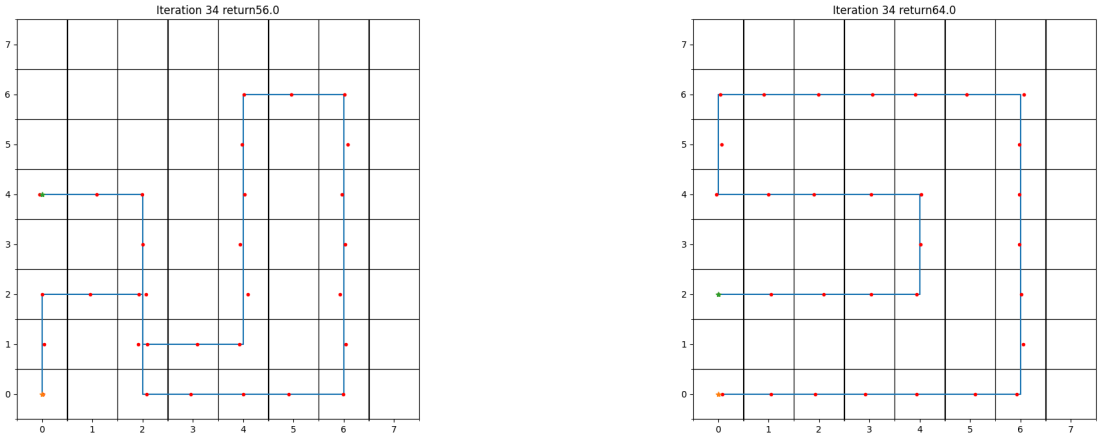


Figure 5: State Coverage, $H = 31, 35$ iterations: (left) trajectory τ_1 induced by output policy of GTO using GTO-S lower bounds achieves $F(\tau_1) = 56$, (right) trajectory τ_2 induced by output policy of GTO using GTO-greedy-S lower bounds achieves $F(\tau_2) = 64$. GTO-greedy-S outperforms GTO-S in those instances where the horizon is just enough to reach optimality.

etc., of states i.e., atom, can have a synergistic effect that can be captured via supermodular global reward functions. In figure 8, we consider the supermodular global reward function defined as $F(\tau) \coloneqq \sum_{i=1}^{K} |\tau \cap S_i|^{\beta}$ with $S_i \subseteq V \coloneqq \mathcal{S} \times T$ indicating a synergy, which we see as a subset of V capturing complementary among its elements.

Diverse and Synergical Trajectory Selection. Interestingly, the notions of exploration mentioned above and encoded through submodularity can be mixed with notions of complementary among states within the same trajectory. This leads to BP objectives such as $F(\tau) = |\bigcup_{s \in \tau} D^s| + \sum_{i=1}^{K} |\tau \cap S_i|^{\beta}$ with $S_i \subseteq V \coloneqq \mathcal{S} \times T$. This objective induces policies maximizing state space covering while seeking complementarity between states within the trajectory. The performances of GTO and GPO on this global reward are illustrated in figure 9. We believe that objectives of this type can be particularly relevant in the context of computational chemistry, where often a scientist wishes to discovery chemical compounds that show a certain diversity and complementarity among its elements at the same time.

Safe States Coverage. Here we consider the notion of safe state coverage as introduced in Section 8. As illustrated in figure 11, an optimal policy w.r.t. to this objective is highly explorative while avoiding unsafe areas. Notice that the concept of safety is captured via a penalty term which can be arbitrary calibrated w.r.t. the maximum value of the submodular component. Nonetheless, in order to guarantee the satisfiability of a safety constraint one would have to express the global reward as a Lagrangian and compute the optimal Lagrangian multiplier by an outer optimization scheme. This procedure,

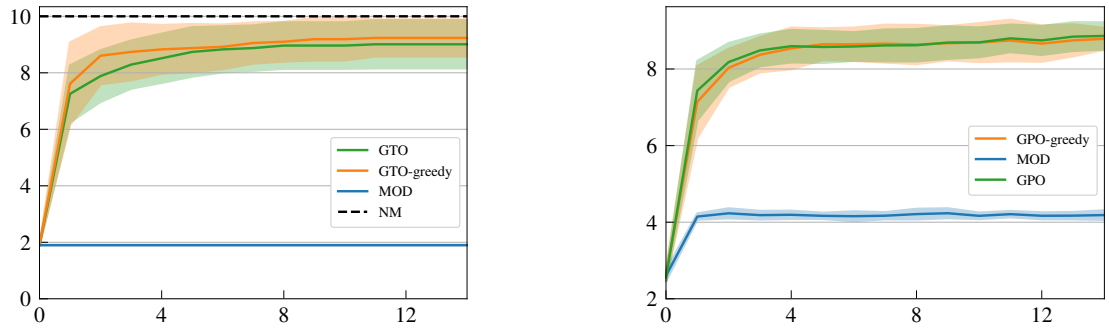


Figure 6: Bounded Curvature Coverage: (left) values of $F(\tau)$ in deterministic GMDP setting where τ is the trajectory computed by GTO at each iteration (x-axis), (right) values of $\mathcal{J}(\pi)$ in stochastic GMDP setting, where π is the policy computed by GPO at each iteration (x-axis). The left plot shows that in practice τ is nearly-optimal w.r.t. the optimal non-Markovian policy.

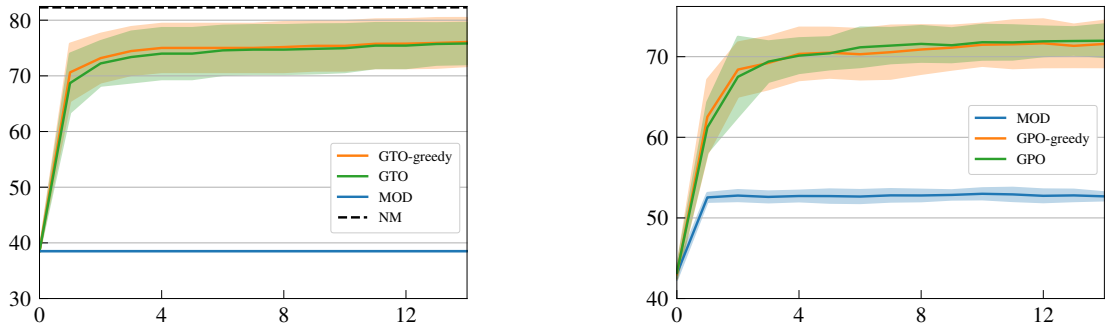


Figure 7: D-Optimal Experimental Design: (left) values of $F(\tau)$ in deterministic GMDP setting where τ is the trajectory computed by GTO at each iteration (x-axis), (right) values of $\mathcal{J}(\pi)$ in stochastic GMDP setting, where π is the policy computed by GPO at each iteration (x-axis). GTO and GPO perform nearly optimally in both cases.

which we leave as future work, seems particularly viable and may lead to high probability guarantees on safety satisfiability.

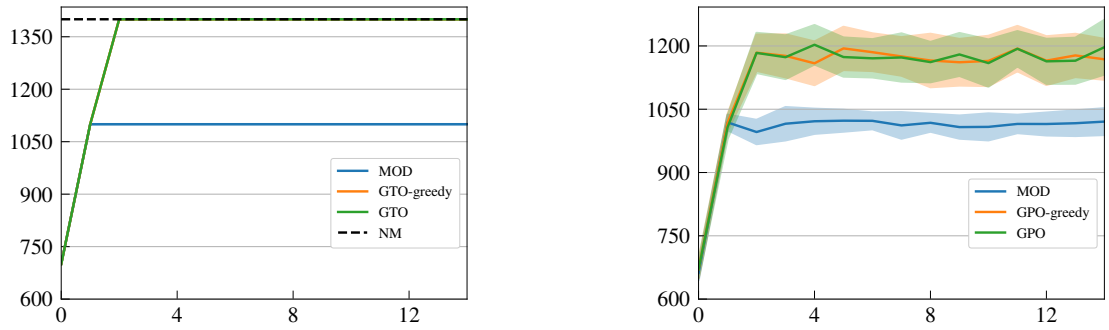


Figure 8: Synergical Trajectory Selection: (left) values of $F(\tau)$ in deterministic GMDP setting where τ is the trajectory computed by GTO at each iteration (x-axis), (right) values of $\mathcal{J}(\pi)$ in stochastic GMDP setting, where π is the policy computed by GPO at each iteration (x-axis). In (left) τ matches the optimal non-Markovian policy.

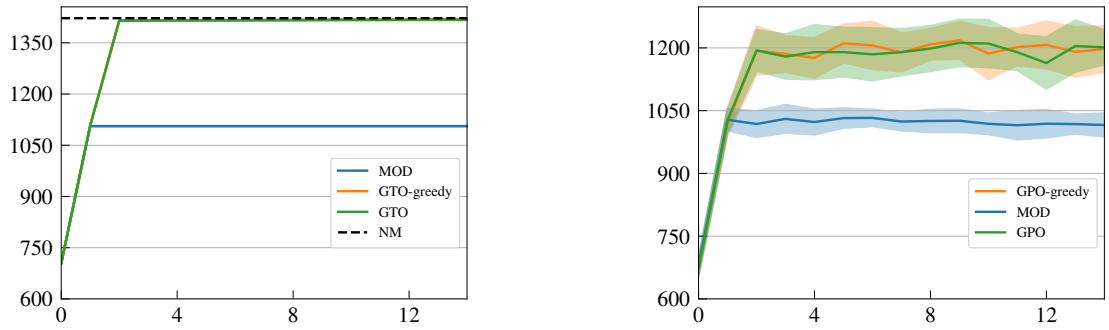


Figure 9: Diverse and Synergical Trajectory Selection: (left) values of $F(\tau)$ in deterministic GMDP setting where τ is the trajectory computed by GTO at each iteration (x-axis), (right) values of $\mathcal{J}(\pi)$ in stochastic GMDP setting, where π is the policy computed by GPO at each iteration (x-axis). From (left) we can deduce that τ can properly trade-off diversity and complementary and match the optimal non-Markovian policy.

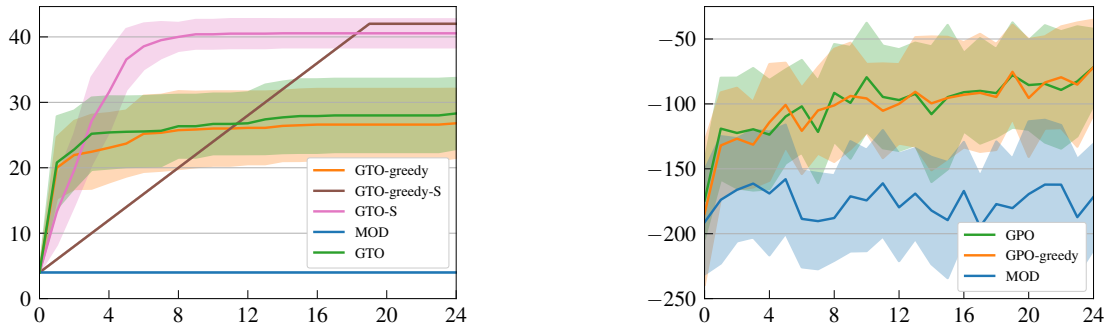


Figure 10: Safe States Coverage: (left) values of $F(\tau)$ in deterministic GMDP setting where τ is the trajectory computed by GTO at each iteration (x-axis), (right) values of $\mathcal{J}(\pi)$ in stochastic GMDP setting, where π is the policy computed by GPO at each iteration (x-axis). Negative values in (right) are due to high unsafety penalty and unavoidable possibility of visiting unsafe states, see figure 10.

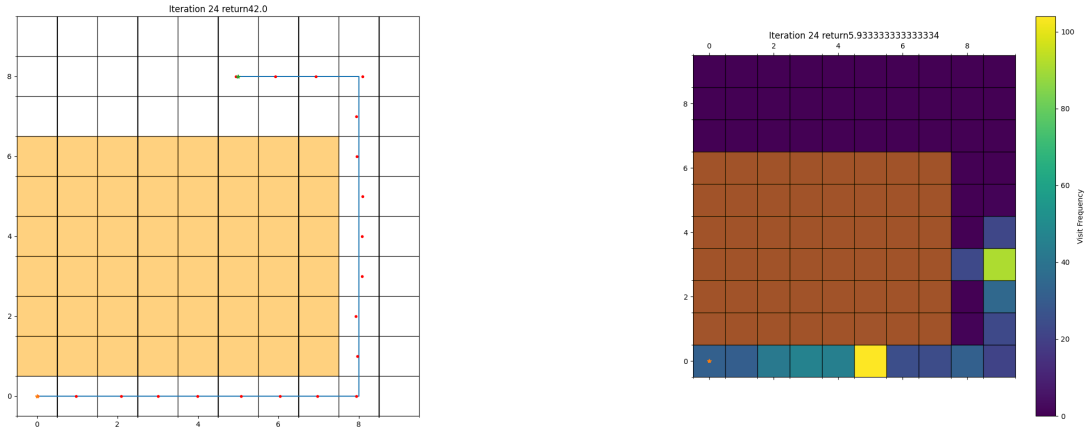


Figure 11: Safe States Coverage $H = 20, 25$ iterations: (left) trajectory τ computed by GTO at last iteration, (right) empirical distribution (blue = low probability, yellow = high probability) over the state space induced by the policy π computed by GPO at each iteration (x-axis). The initial state is the bottom left state of the grid.