Follow the STARs:

Dynamic ω-Regular Shielding of Learned Probabilistic Policies

Ashwani Anand
Max Planck Institute for Software Systems, Kaiserslautern, Germany

Satya Prakash Nayak
Max Planck Institute for Software Systems, Kaiserslautern, Germany

Ritam Raha
Max Planck Institute for Software Systems, Kaiserslautern, Germany

Anne-Kathrin Schmuck
Max Planck Institute for Software Systems, Kaiserslautern, Germany

Abstract

This paper presents a novel dynamic post-shielding framework that enforces the full class of ω-regular correctness properties over pre-computed probabilistic policies. This constitutes a paradigm shift from the predominant setting of safety-shielding – i.e., ensuring that nothing bad ever happens – to a shielding process that additionally enforces liveness – i.e., ensures that something good eventually happens. At the core, our method uses *Strategy-Template-based Adaptive Runtime Shields (STARs)*, which leverage permissive strategy templates to enable post-shielding with minimal interference. As its main feature, STARs introduce a mechanism to *dynamically control interference*, allowing a tunable enforcement parameter to balance formal obligations and task-specific behavior *at runtime*. This allows to trigger more aggressive enforcement when needed while allowing for optimized policy choices otherwise. In addition, STARs support runtime adaptation to changing specifications or actuator failures, making them especially suited for cyber-physical applications. We evaluate STARs on a mobile robot benchmark to demonstrate their controllable interference when enforcing (incrementally updated) ω-regular correctness properties over learned probabilistic policies.

1 Introduction

Adhering to formal correctness while simultaneously optimizing performance is a core challenge in the design of autonomous cyber-physical systems (CPS) [16, 51]. This has led to a rich body of work integrating logical specifications into traditional controller synthesis via multi-objective formulations [13, 14, 25, 47], or into policy synthesis via reinforcement learning (RL) through automata-based reward shaping [12, 21, 22, 28, 34, 40, 56]. While these approaches can produce policies that satisfy complex goals while adhering to formal specifications, they embed the specification into the synthesis procedures – requiring re-synthesis whenever formal objectives, environment conditions, or reward structures are changing.

To overcome these limitations, the concept of *shielding* was introduced: a runtime enforcement mechanism that monitors and, if necessary, overrides the actions proposed by a controller or agent to ensure adherence to a formal specification. Shields treat existing policies as a black box and ensure correctness in a *minimally interfering* manner i.e., they intervene if *and only if* the systems executions will (surely) violate the formal specification (in the future). The concept of shielding traces back to the foundational works on runtime monitoring of program executions in computer science [23, 35], and formal supervision of feedback control software in engineering [45]. More recently, shielding frameworks for learned policies, especially for autonomous CPS, have been proposed (surveyed e.g. in [24, 30, 40, 55]).

Shielding can be applied at different stages of the control pipeline. In *pre-shielding*, the shield is active during policy computation – e.g., during RL training – ensuring the policy avoids 'bad' actions even in the learning stage. However, this tight integration with learning comes at the cost of modularity and flexibility: any change to the system or specification requires recomputing the shielded policy. In contrast, *post-shielding* is applied at deployment - after a nominal policy has been computed. This separation between shielding and training often allows recomputing the shield only when environment conditions or specifications change. The challenge in post-shielding is to still ensure correctness and minimal interference – with only black-box runtime access to the nominal policy. As a result, existing shielding frameworks have primarily focused on safety, where synthesizing maximally permissive shields (i.e., 'inherently' minimally interfering) is tractable [2, 31, 46]. Post-shielding for richer specifications, particularly those involving *liveness*, has remained largely unaddressed.

However, the need for shields which enforce the full class of *ω-regular specifications* naturally arises in CPS applications (e.g. in autonomous driving [36, 37], or mobile robot navigation [32,33]) and requires to not just ensure safety, i.e., that nothing bad every happens, but to also enforce liveness, i.e., that something good eventually happens. At the same time, autonomous CPS must frequently adapt to changing conditions at runtime, e.g. due to actuator failures, evolving mission objectives, or updated optimization criteria. Moreover, different operational contexts may demand different levels of specification enforcement: at times, liveness properties must be satisfied urgently (e.g., reaching a goal zone), while in other situations, adhering closely to mission-specific performance objectives may take priority. This motivates the need for dynamic post-shielding, where the shield can be dynamically adapted, modified or interference-tuned in real time, while still ensuring correctness of the shielded policy w.r.t. the full class of ω-regular specifications.

The STARs Approach. To close this gap, this paper presents STARs – *Strategy-Template-based-Adaptive-Runtime-Shields* – a dynamic post-shielding framework which enforces the full class of ω-regular specifications over pre-computed (e.g. learned) probabilistic policies, schematically depicted in Fig. 1. While it is known that deterministic policies suffice for optimality in finite MDPs with stationary rewards [44], modern RL algorithms typically learn stochastic policies for stability, generalization, and robustness [20, 49, 50]. Moreover, stochasticity enables graded runtime enforcement, which is essential for smoothly tuning interference levels in STARs. In addition, STARs can adapt to incrementally updated specifications or actuator failures at runtime, making them suitable for CPS applications.

Shielding for *ω-regular specifications* requires enforcing liveness properties in addition to safety shielding. This is challenging, as liveness properties (i) do not easily lend themselves to permissive strategies needed for minimal interference, and (ii) only manifest themselves in the infinite limit, hardening their enforcement at runtime given only a finite prefix. STARs overcome these challenges by utilizing *strategy templates* [4], which have recently been introduced as an alternative representation of strategies in two-player parity games (resulting from an ω-regular specification) and can be computed with the same worst-case complexity as classical strategies. Strategy templates condense an infinite number of winning strategies into a simple and efficiently computable data structure, which is (i) truly permissive, enabling minimal inference shielding, and (ii) localizes required future progress (over a known transition graph [1]), enabling a purely history-induced evaluation of liveness properties. In addition, strategy templates are known to be easily composable and robust to the sporadic or persistent unavailability of actions at runtime. This naturally enables the resulting shield to robustly adapt to such scenarios at specification updates and actuator failures at runtime.

Dynamic Interference. In addition to the robustness and adaptation properties that

[1] We only need access to the graph *structure*, not to transition probabilities, rewards or computed policies.

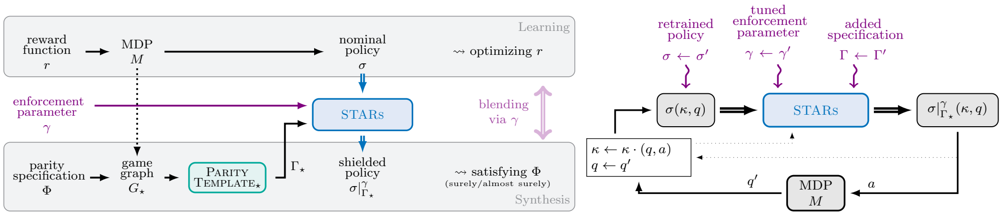


Figure 1 Overview of STARs synthesis (left) and runtime-application of STARs (right) as formalized in Sec. 3. The detailed operation of STARs is illustrated in Fig. 3. Cyan components are taken from the literature and purple components illustrate the dynamic adaptability of STARs.

STARs inherent from strategy templates, they implement an orthogonal and novel way of dynamic post-shield adaptation, that we term *dynamic interference*. Dynamic inference is inspired by multi-objective optimization, where different optimal policies are superposed – using weights the resulting *blended* policies can be biased towards particular optimization criteria. In the same spirit, we equip STARs with an *enforcement parameter* γ to bias the blending of a nominal policy with additional liveness obligations enforced by the shield (see Sec. 3.2 for a formalization and Fig. 3 for a visualization of this idea). Taking advantage of the fact that learning algorithms typically output a state-depended probability distribution over actions which classifies actions w.r.t. their achievable reward, STARs bias these probability distributions towards (additionally) satisfying liveness obligations 'eventually', allowing to remain ϵ-close to the optimal reward if a small enforcement parameter γ is chosen. At the same time, we allow γ to be tuned online as different operational contexts may demand different biases – e.g. enforcing to reach the charging station urgently when the energy level is critical, while allowing optimized mission-specific performance under low-risk conditions.

Following the STARs. To illustrate the effect of a policy which follows a STARs, we use a FactoryBot benchmark depicted in Fig. 2. In FactoryBot, the nominal agent policy optimizes a reward function and STARs are used to guide the agent towards satisfying (generalized) Büchi objectives[2]. FactoryBot simulates multiple OpenAI Gym [11] benchmarks, such as frozen lake, taxi or cliff walking, typically used to evaluate RL policies and represents a simplified version of the snake example used to evaluate a dynamic safety shield [31].

In contrast to [31], where only the underlying policy was allowed to change dynamically, we provide a GUI for the user to dynamically interact with the shielding mechanism via (i) manual adjustment of the enforcement parameter γ, (ii) manual selections of wall placements, which induce safety objectives, and (iii) manual addition of orange and green tiles, which induce co-Büchi and (generalized) Büchi objectives. This results in re-computations of STARs *online* and hence adapting the shields dynamically. In addition, whenever the robot follows the STARs, it is not only 'safety shielded' from bumping into walls, but also 'liveness shielded' towards visiting Büchi states with a frequency determined by the enforcement parameter γ.

In Fig. 2, the green cells denote the Büchi region $\mathcal{B}$, and the numbers inside the cells are the rewards received upon entering them. The agent must visit $\mathcal{B}$, while maximizing the average reward. The images show the agent's heatmap, when allowed to run without a shield (Fig. 2a), with a shield with low (Fig. 2b) and high γ (Fig. 2c), and on online addition of a Büchi objective (Fig. 2d).

Related Work. While the synthesis of permissive strategies for ω-regular objectives has received substantial attention in recent years [4,9,10,18,29], and strategy templates [4,5] have

---

[2] We note, that this choice of simple objectives is for illustration only. Our algorithm can handle the full class of ω-regular objectives compiled into a parity automaton.

(a) Unshielded **(b)** Low γ **(c)** High γ **(d)** Incremental

Figure 2 GUI showing STAR-shielded robot for an instance from FactoryBot. Recording of this shielding scenario is available at https://ritamraha.github.io/MARG/.

been applied to various problems in reactive synthesis [3, 6, 38, 39, 43, 48], these techniques have, to the best of our knowledge, not been used in the context of shielding.

Post-shielding approaches have so far focused mainly on *safety* shielding, where our work is the closest related to [2, 31, 46]. Similar ideas are also used for *policy repair* w.r.t. safety violations [42, 52, 53, 59] or via (partial) re-synthesis [41, 54]. In contrast, STARs directly inherent robustness and adaptability properties of strategy templates that typically circumvent the need for strategy repair and achieve necessary strategy adaptations directly via shielding. For general ω-regular specifications, our work is closely related to the runtime optimization [7] which propose a similar *blending* of a nominal policy with an additional liveness objective, however, via a very different shield synthesis technique. In contrast to our work, the shield synthesized in [7] uses a fixed enforcement parameter, does not exploit probabilistic policies, and allows no dynamic adaptation in the specification or in the graph.

In addition, this paper shares ideas with approaches that synthesize policies which satisfy a quantitative mean-payoff objective alongside an ω-regular constraint defined over the same game graph [1, 57], which are closely related to pre-shielding frameworks, e.g. [12, 21, 28, 34, 40, 56]. Achieving similar optimality results in post-shielding is much harder. Owing to the maximally permissive characteristics of STARs, the expected reward of shielded policies can still be brought arbitrary close to the non-shielded *optimal* value whenever the entire winning region is a strongly connected component (we discuss this formally in Sec. 4). We note that existing post-shielding frameworks also assume excess to the underlying MDP or do not guarantee correctness. To overcome this limitation, abstractions which both over- (for safety) and under- (for liveness) approximate the MDP while still allowing to quantify minimal interference, are needed. This is a challenging direction for future work.

Contributions and Outline. This paper presents a novel dynamic post-shielding framework for the full class of ω-regular objectives. In particular, we present STARs and prove their soundness and minimally interfering properties in Sec. 3. We then consider RL policies learned to maximize discounted or average reward in Sec. 4 and prove that we can always pick an enforcement parameter γ s.t. the expected reward of a shielded policy is arbitrarily close to the optimal reward achieved by the non-shielded policy. Finally, we present experimental results over the already introduced FactoryBot benchmark in Sec. 5. Due to page constrains, the formal proofs of all statements have been moved to the appendix.

2 Preliminaries

This section provides a brief overview of notation and basic concepts.

Notation. We denote by $\mathbb{R}$ the set of real numbers and $[a; b]$ represents the interval $\{a, a+1 \cdots, b\}$. We write Σ^* and Σ^ω to denote the set of finite and infinite sequences of elements from a set Σ, respectively. A *probability distribution* over a finite set S is denoted as a function $\mu : S \mapsto [0,1]$ such that $\sum_{s\in S}\mu(s) = 1$. The set of all probability distributions over S is denoted as $\mathcal{D}(S)$. The *support* of a distribution μ is the set $supp(\mu) = \{s \in S \mid \mu(s) > 0\}$. Given two distributions $\mu_1, \mu_2 \in \mathcal{D}(S)$, the *total variation distance* between μ_1 and μ_2 is defined as: $\mathtt{D_{TV}}(\mu_1, \mu_2) = \frac{1}{2}\sum_{s\in S}|\mu_1(s) - \mu_2(s)|$. Given any function $\mu : S \mapsto \mathbb{R}$, we use $\mathcal{N}(\mu) \in \mathcal{D}(S)$ to denote its normalized distribution: $\mathcal{N}(\mu)(s) = \frac{\mu'(s)}{\sum_{s'\in S}\mu'(s')}$, where $\mu'(s') = \max(\mu(s'), 0)$.

Markov Decision Process. A *Markov Decision Process* (MDP) is a tuple $M = \langle Q, A, \Delta, q_0\rangle$ where Q is a finite set of states, A is a finite set of actions, $\Delta : Q \times A \mapsto \mathcal{D}(Q)$ is a (partial) transition function and $q_0 \in Q$ is the initial state. For any state $q \in Q$, we let $A(q)$ denote the set of actions that can be selected in state q. A *strongly connected component (SCC)* of an MDP M is a maximal set of states $Q' \subseteq Q$ such that for every pair of states $q, q' \in Q'$, there exists a path from q to q' in Q' with non-zero probability.

Given a state q and an action $a \in A(q)$, we denote the probability of reaching the successor state q' from q by taking action a as $pr(q'|q,a)$. A run ρ of an MDP M is an infinite sequence in $Q \times (A \times Q)^\omega$ of the form $q_0a_0q_1\ldots$ such that $pr(q_{i+1}|q_i, a_i) > 0$. A *finite run* of *length* n is a finite such sequence $\kappa = q_0a_0\ldots q_na_n$ or $\kappa = q_0a_0\ldots q_n$. We write $\rho[i]$ to denote the i^{th} state-action pair (q_i, a_i) appearing in ρ, $\rho[i;j]$ to denote the infix $q_ia_i\ldots q_j$ for $j \geq i$, and $\rho[j;\infty]$ to denote the suffix $q_ja_j\ldots$. These notations extend to the case of finite runs analogously. We write Runs^M (resp. FRuns^M) to denote the set of all infinite (resp. finite) runs of M. We denote the last state of a finite run ρ as *last*(ρ).

A *policy* (or, a strategy) in an MDP M is a function $\sigma : \mathrm{FRuns}^M \mapsto \mathcal{D}(A)$ such that $supp(\sigma(\rho)) \subseteq A(last(\rho))$. Intuitively, a policy maps a finite run to a distribution over the set of available actions from the last state of that run. A policy is *stochastic* if $|supp(\sigma(\kappa q))| = A(q)$ for every history κq. A run $\rho = q_0a_0q_1\ldots$ is a σ-run if $a_i \in supp(\sigma(q_0a_0\ldots q_i))$. Given a measurable set of runs $P \subseteq \mathrm{Runs}^M$, $\Pr_\sigma[P]$ is the probability that a σ-run belongs to P. We use Runs^{M^σ} to denote the set all σ-runs and define the set of all policies over M as Π_M.

Let $\mathbb{M}$ be a set called *memory*. A policy σ with memory $\mathbb{M}$ is represented as a tuple $(\mathbb{M}, m_0, \alpha, \beta)$ where $m_0 \in \mathbb{M}$ is the initial memory value, $\alpha : \mathbb{M} \times Q \mapsto \mathbb{M}$ is the memory update function, and $\beta : \mathbb{M} \times Q \mapsto \mathcal{D}(A)$ is the function prescribing the distribution over the next set of available actions. A policy σ is said to be a *finite memory* policy if $\mathbb{M}$ is a finite set. It is called *stationary* if $\mathbb{M} = \emptyset$, i.e., the choice of action only depends on the state. Given a finite run (or history) κ, a state q and an action $a \in A(q)$, $\sigma(\kappa q, a) = pr(a|\kappa q)$ denotes the probability that σ assigns for choosing the action a from state q with history κ. If σ is stationary, we will write $\sigma(q,a)$ instead of $\sigma(\kappa q, a)$. Given a random variable $f : \mathrm{Runs}^M \mapsto \mathbb{R}$, we denote by $\mathbb{E}^q_{M^\sigma}(f)$ the expectation of f over the runs of M originating at state q that follow σ. We instead write $\mathbb{E}^q_\sigma(f)$ when M is clear from the context.

(Stochastic) Games on Graphs. A *stochastic game graph* is a tuple $G = (Q = Q_{\bigcirc} \cup Q_{\Box} \cup Q_{\triangle}, E)$ where (Q, E) is a finite directed graph. For every state $q \in Q$, we denote the set of all available edges from q as $E(q)$ and assume $|E(q)| > 0$ for all $q \in Q$. Further, for $\Diamond \in \{\bigcirc, \Box, \triangle\}$, we define $E_\Diamond = \{(q, q') \in E \mid q \in Q_\Diamond\}$.

A stochastic game involves three players: 'system' ($\bigcirc$), 'environment' ($\Box$), and 'random' ($\triangle$). They take turns moving a token along states, forming a path. When the token is at a state in $Q_{\bigcirc}$ (resp. $Q_{\Box}$), the system (resp. environment) player chooses one of its successors to move the token. At a state in $Q_{\triangle}$, the random player moves the token to one of its successors following a known or unknown probability distribution, selecting uniformly at random. Stochastic game graphs are often called $2\frac{1}{2}$-player game graphs. If $Q_{\triangle} = \emptyset$, $Q_{\Box} = \emptyset$, or $Q_\square = Q_\triangle = \emptyset$, they reduce to 2-player, $1\frac{1}{2}$-player, and 1-player game graphs, respectively. Game graphs without a random player are called *deterministic*.

Given a stochastic game graph G, a *run* (or play) ρ over G is an infinite sequence of states $q_0 q_1 \ldots \in Q^\omega$. We write $\mathtt{Inf}_Q(\rho)$ (resp. $\mathtt{Inf}_E(\rho)$) to denote the set of all states (resp. edges) which occur infinitely often along ρ. We collect all runs over G in the set Runs^G. A *strategy* for player $\Diamond \in \{\bigcirc, \square, \triangle\}$ over G is a function $\sigma_\Diamond : Q^* \times Q_\Diamond \to \mathcal{D}(Q)$ that describes a probability distribution over next available moves to the successor states based on the history of the current run. Given a system player strategy σ, a run ρ is said to comply with σ, i.e., be a σ-run, if $q_{i+1} \in supp(\sigma(q_0 \ldots q_i))$ holds for all $q_i \in Q_\bigcirc$ along ρ. Given a measurable set of infinite runs $P \subseteq \mathrm{Runs}^G$, $\Pr_\sigma[P]$ is the probability that a σ-run belongs to P. We use Runs^{G^σ} to denote the set of all σ-runs over G.

ω-Regular Objectives and (Almost) Sure Winning. Given a game graph G, a *winning condition* (or *objective*) is defined as a set of runs $\Phi \subseteq \mathrm{Runs}^G$. An ω-regular objective can be canonically represented by a *parity objective* (possibly with a larger set of states [8]) $\Phi = \textsc{Parity}[c]$ which is defined using a coloring function $c : Q \to [0; d]$ that assigns each state a *color*. The parity objective $\textsc{Parity}[c]$ contains all runs $\rho \in \mathrm{Runs}^G$ for which the highest color (as assigned by the coloring function c) appearing infinitely often is even. Formally, $\textsc{Parity}[c] = \{\rho \in \mathrm{Runs}^G \mid \max\{c(q) \mid q \in \mathtt{Inf}_Q(\rho)\} \text{ is even}\}$. The parity objective $\textsc{Parity}[c]$ reduces to a *Büchi* objective, if the domain of c is restricted to two colors $\{1, 2\}$.

Given a game graph G and an objective $\Phi \subseteq \mathrm{Runs}^G$, a run is said to *satisfy* Φ if it belongs to Φ. A system player strategy σ is said to be *surely* (resp. *almost surely*) *winning* from a state q in the game (G, Φ), if every σ-run from q satisfies Φ (resp. $\Pr(\rho \in \Phi \mid \rho \text{ is a } \sigma\text{-run from } q) = 1$). We collect all such states from which a surely (resp. almost surely) winning strategy exists in the winning region $\mathcal{W}^\bullet_\Phi$ (resp. $\mathcal{W}^\circ_\Phi$). Furthermore, we say a strategy σ is *surely* (resp. *almost surely*) *winning* in the game (G, Φ), denoted by $(G, \sigma) \models_\bullet \Phi$ (resp. $(G, \sigma) \models_\circ \Gamma$), if it is surely (resp. almost surely) winning from every state in the winning region.

Strategy Templates. *Strategy templates* [4] collect an infinite number of system player strategies over a (stochastic) game in a concise data structure consisting of three types of local conditions on the system player moves: *safety*, *co-live* and *live-group* templates. Formally, given a game $G = (Q, E)$, a strategy template (for the system player) is a tuple $\Gamma = (S, D, H_\ell)$ comprising a set of *unsafe* edges $S \subseteq E_\bigcirc$, a set of *co-live* edges $D \subseteq E_\bigcirc$ and a set of live-groups $H_\ell \subseteq 2^{E_\bigcirc}$. A strategy template Γ over G induces a set of infinite runs

$$\mathrm{Runs}^\Gamma := \left\{ \rho \in \mathrm{Runs}^G \;\middle|\; \begin{array}{ll} & \forall e \in S : e \notin \rho \\ \wedge & \forall e \in D : e \notin \mathtt{Inf}_E(\rho) \\ \wedge & \forall H \in H_\ell : src(H) \cap \mathtt{Inf}_Q(\rho) \neq \emptyset \to H \cap \mathtt{Inf}_E(\rho) \neq \emptyset \end{array} \right\}$$

Intuitively, a run $\rho \in \mathrm{Runs}^\Gamma$ satisfies the following objectives: (i) ρ never uses the unsafe edges in S, and (ii) ρ stops using the co-live edges in D eventually, and (iii) if ρ visits the set of source states of a live-group $H \in H_\ell$ infinitely often, then it also uses the edges in H infinitely many times. Given a game graph G, a strategy σ in G *follows* a template Γ if $(G, \sigma) \models_\bullet \mathrm{Runs}^\Gamma$. If G is clear from the context we often abuse notation and write $\sigma \models_\bullet \Gamma$ if σ follows Γ. A strategy template Γ is said to be *surely* (resp. *almost surely*) winning in the ω-regular game (G, Φ) if every strategy that follows Γ is surely (resp. almost surely) winning.

► **Proposition 1** ([4, 43]). *Given a 2-player (resp. $2\frac{1}{2}$-player) parity game (G, Φ), a* surely *(resp.* almost surely*) winning strategy template $\Gamma_\bullet$ (resp. $\Gamma_\circ$) for (G, Φ) can be computed in time $\mathcal{O}(|Q|^{d+\mathcal{O}(1)})$ (resp. $\mathcal{O}((|Q|d)^{d+\mathcal{O}(1)})$). We denote the algorithm that computes a winning strategy template by $\textsc{ParityTemplate}_\star(G, \Phi)$ with $\star \in \{\bullet, \circ\}$, realized by [4, Alg.3] for $\Gamma_\bullet$ and by [43, Alg.8] for $\Gamma_\circ$.*

Note that such winning strategy templates are always *conflict-free*, i.e., from any state, there is always an edge that is neither unsafe nor co-live and for every source of a live-group, there is always a live edge that is neither unsafe nor co-live (see [4] for details). For simplicity, from now on, we assume that strategy templates are conflict-free.

3 STARs – Dynamic Post-Shielding for ω-Regular Objectives

This section formalizes our novel dynamic post-shielding framework via Strategy-Template-based-Adaptive-Runtime-Shields (STARs) schematically depiced in Fig. 1.

3.1 Synthesizing STARs

As depicted in Fig. 1, an essential step in the construction of STARs is the synthesis of winning strategy templates via the ParityTemplate algorithm from [4, 43] in a game derived from the transition structure of the underlying MDP M. Dependent on whether STARs should enforce the additional ω-regular specification Φ almost surely or surely, we abstract M either into a $1\frac{1}{2}$- or a 2-player game graph. Thereby, sure satisfaction treats the randomness of the MDP fully adversarial, i.e., by a second deterministic player, while almost sure satisfaction allows to keep the process random[3].

▶ **Definition 2.** *Given an MDP $M = \langle Q, A, \Delta, q_0\rangle$ we define the 2-player (resp. $1\frac{1}{2}$-player) game graph induced by M as the tuple $G^M_\bullet = (Q_\bigcirc \cup Q_\square, E_\bigcirc \cup E_\square)$ (resp. $G^M_\circ = (Q_\bigcirc \cup Q_\triangle, E_\bigcirc \cup\ E_\triangle)$) s.t. for $\diamondsuit \in \{\square, \triangle\}$:*

- $Q_\bigcirc := Q, \quad Q_\diamondsuit := \{q^a_\diamondsuit \mid a \in A(q) \text{ and } q \in Q\}$,
- $E_\bigcirc := \{(q, q^a_\diamondsuit) | a \in A(q)\}$, *and* $E_\diamondsuit := \{(q^a_\diamondsuit, q') | pr(q'|q, a) > 0\}$.

Given a game graph $G^M_\star$ and a parity objective Φ over $G^M_\star$, we use ParityTemplate$_\star$ (Proposition 1) to compute winning strategy templates $\Gamma_\star$ for $(G^M_\star, \Phi)$. Hence, any strategy that follows $\Gamma_\star$ will be surely (if $\star = \bullet$) or almost surely (if $\star = \circ$) winning in $(G_\star, \Phi)$.

▶ Remark 3. We remark that assuming Φ to be directly defined over $G^M_\star$ is not restrictive. Any ω-regular property φ with propositions interpretable as subsets over Q can be converted into a parity game $(G_\varphi, \Phi_\varphi)$ which can be combined with $G^M_\star$ through a simple product.

By restricting attention to the setting discussed in Rem. 3, we slightly abuse notation and interpret a template $\Gamma_\star$, computed over a game graph $G^M_\star$, directly over the original MDP M. That is, we convert every edge $(q, q^a_\diamondsuit)$ constrained by the template into a constrained state-action pair (q, a). This results in a template $\Gamma_\star = (S, D, H_\ell)$ where $S \subseteq Q \times A$, $D \subseteq Q \times A$ and $H_\ell \subseteq 2^{Q\times A}$. Further, we say that a policy σ in M follows a template $\Gamma_\star$ if the corresponding strategy in $G^M_\star$ follows the template.

3.2 Dynamical Interference via STARs

Given a strategy template $\Gamma_\star = (S, D, H_\ell)$ which is winning for the parity objective Φ interpret over an MDP M, STARs dynamically *blend* a given nominal policy σ with the safety and liveness obligations of Φ localized in $\Gamma_\star$ as depicted in Fig. 3.

Intuitively, to comply with the safety template S, STARs set the probabilities of unsafe actions to zero, thereby preventing runs to reach states from where Φ cannot be satisfied. For each edge in the co-live group D, STARs maintain a counter that tracks how many times the

[3] The $1\frac{1}{2}$ game is still constructed without access to Δ – almost-sure winning is independent of probabilities.

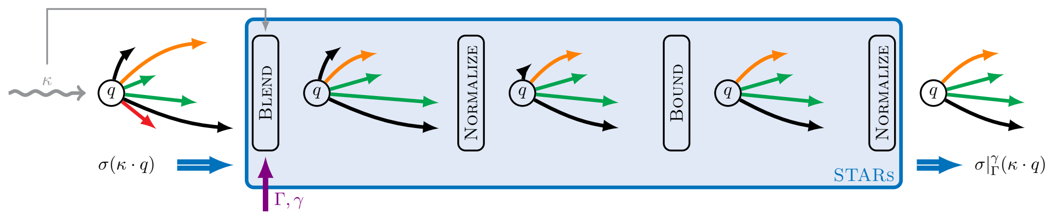


Figure 3 Illustration of dynamic interference via STARs. Length of arrows indicate the relative probability of the corresponding action in $\mu \in \mathcal{D}(A(q))$. The strategy template $\Gamma = (S, D, H_\ell)$ is illustrated via colors red (S), orange (D) and green (H_ℓ). Blending applies (1b) in Def. 5, bounding applies (1a) in Def. 5 and normalizing applies standard normalization, respectively.

edge has been sampled. Each time the edge is sampled, its probability is reduced, ensuring that runs eventually avoid co-live edges. Similarly, for each live group H_ℓ, STARs maintain a counter to track how many times the policy visits the source states of the group without sampling any of its corresponding actions. With each such visit, the shield incrementally increases the probability of sampling these actions based on the counter value. This guarantees that eventually, one of the actions in the live group is sampled (with probability close to 1) if its source states are visited often enough. Once an action is sampled, the corresponding counter is reset, and the process repeats.

In order to formalize the above intuition, we first formally define the history-dependent counter function for co-live and live groups discussed above.

▶ **Definition 4.** *Let M be an MDP and $\Gamma = (S, D, H_\ell)$ a strategy template interpreted over M. Further, let $\kappa = q_0 a_0 q_1 a_1 \ldots q_n a_n \in FRuns^M$ be a finite run over M. Then we define for all $(q,a) \in D$: $count_{(q,a)}(\kappa) := |\{i \mid \kappa[i] = (q,a)\}|$, and for all $H \in H_\ell$: $count_H(\kappa) := |\{i > maxpref^{\kappa}_H \mid \kappa[i] \in \{(q,a) \mid q \in src(H)\}\}|$, where $maxpref^{\kappa}_H := j$ such that $\kappa[j] \in H$ and $\kappa[j;\infty] \cap H = \emptyset$.*

The counters defined in Def. 4 allow us to formally define how a STAR modifies the probability distribution $\mu(A(q))$ over actions chosen by σ in the current state q reached with history κ using a template $\Gamma_\star$. Intuitively, the *bias* towards satisfying Φ introduced by the counter-based modification of $\mu(A(q))$ can be *tuned* by an enforcement parameter γ, and a threshold parameter θ, which can be changed dynamically at runtime.

▶ **Definition 5** (STARs). *Fix an MDP M, a finite run $\kappa \in FRuns^M$ with $q = last(\kappa)$, a strategy template Γ interpreted over M, an enforcement parameter γ, and a threshold θ. Then, the probability distribution $\mu \in \mathcal{D}(A(q))$ induces a* shielded distribution $\overline{\mu} \in \mathcal{D}(A(q))$ *with $\overline{\mu} := \mathcal{N}(\mu')$ s.t. for all $a \in A(q)$*

$$\mu'(a) := \begin{cases} 0 & \text{if } \mathcal{N}(\mu'')(a) \leq \theta, \\ \mathcal{N}(\mu'')(a) & \text{otherwise.} \end{cases} \quad \text{(1a)} \qquad \mu''(a) := \begin{cases} 0 & \text{if } (q,a) \in S, \\ \mu(a) - \gamma \cdot count_{(q,a)}(\kappa) & \text{if } (q,a) \in D, \\ \mu(a) + \gamma \cdot count_H(\kappa) & \text{if } (q,a) \in H, H \in H_\ell. \end{cases} \quad \text{(1b)}$$

We write $\overline{\mu} = STARs(\mu, \kappa, \Gamma, \gamma, \theta)$ to denote that $\overline{\mu}$ is obtained from μ via (1).

The effect of shielding a policy as formalized in Def. 5 is illustrated in Fig. 3. Intuitively, (1b) ensures that the probability of taking certain actions in state q is adapted via the counters induced by the history of the current run and the enforcement parameter γ. If γ is close to 1 these updates are very aggressive. If γ is close to 0 they are very mild. As the resulting function μ'' is not a probability distribution anymore (as probabilities over $A(q)$ do not sum up to 1), we use its normalized version in (1a) to impose the threshold $\theta > 0$ to make sure that a live edge is surely taken after a finite number of time steps (dependent on γ and σ). In the end, we normalize the resulting distribution to obtain the final distribution.

▶ Remark 6. We note that if there is an unsafe action a in S such that the current distribution μ assigns a probability of 1 to it, i.e., $\mu(a) = 1$, then $\overline{\mu}$ as in Def. 5 will be not well-defined as it assigns zero probability to all actions. This corner case can be handled by perturbing the distribution μ slightly, e.g., by adding a small $\varepsilon > 0$ to all actions before applying STARs.

Given the formalization of a shielded probability distribution in Def. 5 the definition of a *shielded policy* immediately follows.

▶ **Definition 7.** *Given any MDP M, a strategy template Γ interpreted over M, a threshold θ, and an enforcement parameter $\gamma > 0$, a stochastic policy σ in M induces the shielded policy $\sigma|_{\gamma}^{\Gamma,\theta} : FRuns^M \mapsto \mathcal{D}(A)$ s.t. $\sigma|_{\gamma}^{\Gamma,\theta}(\kappa) = STARs(\sigma(\kappa), \kappa, \Gamma, \gamma, \theta)$.*

In order to avoid the corner case discussed in Rem. 6, the definition assumes that the initial policy σ is stochastic, i.e., $supp(\sigma(\kappa q)) = A(q)$ for all histories κq. This is without loss of generality as any deterministic policy can be converted into a stochastic one as discussed in Rem. 6. Furthermore, as the resulting shielded policy is dependent on the history of a run, a policy is actually shielded via STARs *online* while generating a *shielded run*, i.e., a $\sigma|_{\gamma}^{\Gamma_\star,\theta}$-run, as illustrated in Fig. 1 (right). We emphasize that STARs never modify the underlying policy and thereby maximize modularity between the nominal policy and constraint enforcement.

3.3 Correctness and Minimally Interference of STARs

This section shows that STARs indeed implement the fundamental shielding paradigm of correct but minimal interference shielding.

Correctness of STARs follows directly from the fact that they are based on *winning strategy templates* $\Gamma_\star$, which implies that the shielded policy $\sigma|_{\gamma}^{\Gamma_\star,\theta}$ satisfies the objective Φ (almost) surely, if it follows the template. It therefore remains to show that the shielded policy indeed follows the template. As (1b) ensures that the shielded policy assigns zero probability to unsafe edges and that the probability of taking co-live edges is reduced with each visit, the shielded policy will never take an unsafe edge and will eventually avoid co-live edges. Furthermore, as (1a) increments the counter for live groups each time the source states are visited without any action from the group being taken, the shielded policy will eventually take an action from the live group. In total, the shielded policy will follow the template $\Gamma_\star$ and therefore ensure that the shielded run satisfies Φ as formalized below.

▶ **Theorem 8.** *Given the premises of Def. 7 it holds that $\sigma|_{\gamma}^{\Gamma_\star,\theta}$ follows $\Gamma_\star$.*

▶ **Corollary 9.** *Given any MDP M, a stochastic policy σ over M and an enforecement parameter $\gamma > 0$, let $\Gamma_\star := \textsc{ParityTemplate}_\star(G_\star^M, \Phi)$. Then, every $\sigma|_{\gamma}^{\Gamma_\star,\theta}$-run from the winning region of Φ satisfies Φ surely/almost surely (depending on $\star$).*

Minimal interference of STARs is, unfortunately, less straight forward to formalize. Based on existing notions of minimal interference, we characterise two orthogonal notions: (i) a minimal deviation in the distribution of observed histories, and (ii) a minimal expected average shielding cost measured in the expected number of non-optimal action choices.

History-based minimal interference is inspired by a similar notion from [17] for safety shields: an action must be deactivated after a history κ, if *and only if* there exists a nonzero probability that the safety constraint would be violated in a bounded extension of κ, regardless of the agent's policy. This argument extends to STARs for both safety and co-live templates, ensuring minimal interference in these settings. However, defining minimal interference for liveness templates is more challenging due to their inherently infinite nature, making bounded violations inapplicable. Instead, we establish minimal interference by showing that for any bounded execution κ that can be extended to a run that satisfies the liveness template, the probability of observing κ in the shielded execution remains close to its probability under the nominal policy. Formally, for $\kappa \in \mathrm{FRuns}^M$, we define $\kappa \models \mathit{pref}(\Phi)$ if there exists an infinite run $\rho \in \mathrm{Runs}^M$ such that $\rho \models \Phi$ and κ is a prefix of ρ. We let $\Pr_\sigma(\kappa)$ denote the probability of observing κ in the execution of a policy σ in M and establish the following result.

▶ **Theorem 10.** *Given the premises of Cor. 9 with $\sigma' = \sigma|_\gamma^{\Gamma_\star,\theta}$, for any $\varepsilon > 0$ and for all length $l \in \mathbb{N}$, there exist parameters $\gamma, \theta > 0$ such that for all histories κ of length l with $\kappa \models \mathit{pref}(\Phi)$, it holds that $\Pr_{\sigma'}(\kappa) > \Pr_\sigma(\kappa) - \varepsilon$.*

Minimal shielding costs are inspired by a similar notion in [7], where a cost function is used to measure how much a (deterministic) shield changes the action choices of a (pure) nominal policy. A natural extension of this notion to stochastic policies is to define a shielding cost based on the distance between the distributions of the actions taken by the shielded and the nominal policies. Thereby, the shielding cost can also vary depending on the history of the run, allowing for a more fine-grained analysis as in [7]. To formalize this intuition, we define a history-based cost function $\mathtt{cost} : \mathrm{FRuns} \to [0, W]$ that assigns a cost to the history κ and the cost of shielding the policy σ at κ is defined as $\mathtt{cost}(\kappa, \sigma, \sigma') = \mathtt{cost}(\kappa) \cdot \mathtt{D}_{\mathtt{TV}}(\sigma(\kappa), \sigma'(\kappa))$. This cost captures the interference as the difference between the action distributions of the original and shielded policies based on the cost of shielding the policy at κ. This can be generalized to the cost of a run ρ as $\mathtt{cost}(\rho, \sigma, \sigma') = \limsup_{l \to \infty} \frac{1}{l} \sum_{i=0}^{l-1} \mathtt{cost}(\rho[0; i], \sigma, \sigma')$, which captures the average cost of the differences between the two policies over the run.

In the following, we show that the expected average cost of these differences stays below ε for liveness templates. More precisely, we show that the expected average cost of the shielded policy is bounded whenever the template considered only contains liveness templates.

These restrictions are needed because the shielded policy may have to stop (or eventually stop) taking certain actions—such as unsafe or co-live ones—to satisfy the ω-regular constraint. So, we can guarantee minimal interference with respect to the cost function as long as the original policy doesn't include any of these actions. This is ensured by using a strategy template that has no unsafe or co-live actions.

▶ **Theorem 11.** *Given the premises of Cor. 9 with $\sigma' = \sigma|_\gamma^{\Gamma_\star,\theta}$ such that $\Gamma_\star = (\emptyset, \emptyset, H_\ell)$, and a cost function $\mathtt{cost} : \mathit{FRuns}^M \to [0, W]$, for any $\varepsilon > 0$, there exist parameters $\gamma, \theta > 0$ such that the following holds: $\mathbb{E}_{\rho \sim \sigma'} \mathtt{cost}(\rho, \sigma, \sigma') < \varepsilon$.*

We note that the corner case discussed in Rem. 6 does not compromise these results. In Thm. 10, any history κ satisfying $\kappa \models \mathit{pref}(\Phi)$ inherently avoids unsafe actions. In Thm. 11, the assumption that the strategy template excludes unsafe and co-live actions ensures that the bound on the expected average cost remains valid.

3.4 Dynamic Adaptations of STARs Beyond Dynamic Interference

So far, we have considered a shielding scenario for a *static* parity objective Φ. However, a major strength of strategy templates is their *efficient compositionality* and *fault-tolerance*, which allow for further dynamic adaptations of STARs.

Compositionality facilitates the incremental integration of multiple ω-regular specifications into STARs. By using the existing algorithm ComposeTemplate from [4, Alg.4] we can compute STARs for *generalized parity constraints* of the form $\Phi = \wedge_{i=1}^{k} \Phi_i$, where each Φ_i represents a parity constraint over $G_\star^M$. Crucially, these objectives Φ_i may not be available all at once but might arrive incrementally over time, leading to the need to update the applied shield at runtime. As ComposeTemplate simply combines strategy templates for all objectives into a single (non-conflicting) template, Thm. 8 and Thm. 10 also apply in this case, as long as the run is in the combined winning region of all objectives during the update.

Fault-tolerance ensures that STARs can handle the occasional or persistent unavailability of actions correctly. Concretely, persistent faults are addressed by marking actions as unsafe and resolving conflicts as needed (see [4, Alg.5]), while occasional faults are handled by temporarily excluding the unavailable actions from the template (see [4, Sec.5.2]).

▶ Remark 12. A common assumption in robotic applications is that (incrementally arriving) *liveness* specifications are satisfiable from every safe node in the workspace [15, 19, 27, 41, 54] – most robotic systems can simply invert their path by suitable motions to return to all relevant positions in the workspace. Using the common decomposition of ω-regular objectives φ into a safety part φ_s and a liveness part φ_ℓ, one can restrict the newly arriving objectives to liveness obligations only. In this case, incremental synthesis never leads to an decreased winning region in such robot applications. This is in fact the case in the incremental instances considered in the FactoryBot benchmark used for evaluation (see Sec. 1 and Sec. 5).

4 Maintaining Optimal Rewards while Shielding

While Sec. 3 formalizes and proves *correctness and minimal interference* of STARs, this section strengthens this result further, i.e., we show that whenever nominal policies have been computed to optimize a given reward function (under certain assumptions), STARs produce a shielded policy which achieves a reward which is ϵ-close to the nominal one while additionally guiding the agent to (almost) surely satisfy an ω-regular correctness specification.

In order to formalize these strong optimality properties of STARs, we introduce additional notation. An optimal policy over an MDP is typically computed (e.g. via reinforcement learning (RL)) by associating transitions with so-called *reward functions*. Typically, these reward functions are Markovian, which assign utility to state-action pairs. Formally, a *rewardful MDP* is denoted as a tuple (M, r) where M is an MDP equipped with a reward function $r : Q \times A \mapsto \mathbb{R}$. Such an MDP under a policy σ determines a sequence of random rewards $r(X_i, Y_i)$ for $i \geq 0$, where X_i and Y_i are the random variables denoting the i^{th} state and i^{th} action, respectively. Given a rewardful MDP (M, r) with initial state q_0 and a policy σ, we define the *discounted reward* and the *average reward* via

$$\mathrm{Disc}_\sigma^{q_0}(\lambda) := \lim_{N\to\infty} \mathbb{E}_\sigma^{q_0}\left(\sum_{0\leq i\leq N} \lambda^i r(X_i, Y_i)\right) \text{ with } \lambda \in [0,1], \text{ and} \tag{2a}$$

$$\mathrm{Avg}_\sigma^{q_0} := \limsup_{N\to\infty} \tfrac{1}{N}\mathbb{E}_\sigma^{q_0}\left(\sum_{0\leq i\leq N} r(X_i, Y_i)\right). \tag{2b}$$

For any reward function, we define the *optimal reward* to be the maximum reward achievable by a policy σ. We call every policy that achieves the optimal reward an *optimal policy*. A policy σ is ε-*optimal* if it achieves a reward that is at least ε less than the optimal reward.

4.1 Discounted Rewards

When policies are trained to maximize a discounted reward, as formalized in (2a), the impact of obtained rewards decreases with time. Therefore, the optimal reward achievable by a policy over a given MDP significantly depends on the bounded (initial) executions possible over this MDP. As Thm. 10 shows that the probability of observing a bounded execution remains close to its probability under the original policy, the ε-optimality of a shielded optimal policy can be obtained as a direct consequence of Thm. 10 when discounted rewards are used during training. In particular, the discounted reward of the shielded policy remains close to the discounted reward of the original policy as formalized below.

▶ **Theorem 13.** *Given the premises of Cor. 9 with $\sigma' = \sigma|_{\gamma}^{\Gamma^\star,\theta}$ and $\mathcal{W}_\Phi^\star = Q$, for every $\varepsilon > 0$, there exist parameters $\theta, \gamma > 0$ such that the following holds: $Disc_{\sigma'}^{q_0}(\lambda) > Disc_{\sigma}^{q_0}(\lambda) - \varepsilon$.*

▶ Remark 14. We remark that the assumption $\mathcal{W}_\Phi^\star = Q$ in Thm. 13 is not restrictive for two main reasons. First, if $\mathcal{W}_\Phi^\star \subsetneq Q$, we can use $\mathcal{W}_\Phi^\star$ as an additional constraint in existing safe reinforcement learning frameworks. For instance, one can use *preemptive safety shielding* proposed in [2]. This allows to learn an optimal policy within $\mathcal{W}_\Phi^\star$ which directly allows to transfer the results from Thm. 13. Second, we recall the discussion of Rem. 12 to note that including the safety-part of the objective into the learning process does not harm the incremental adaptation of STARs when new (liveness) specifications arrive.

4.2 Average Rewards

We now consider the scenario that policies where trained to maximize the average reward, as formalized in (2b). In contrast to policies which optimize a discounted reward, optimal average reward policies do not put special emphasis on bounded (initial) executions. On the contrary, the optimal average reward is equivalently impacted by rewards collected over the entire (infinite) length of runs compliant with the policy. This implies, that a policy can only satisfy an ω-regular property (almost) surely *and* optimize the average reward, if it can 'switch' between their satisfaction by alternating infinitely often between finite intervals which satisfy either one. This, however, is only possible if the underlying MDP is 'nice' enough to allow for this alternation. In particular, to retain the modularity of shielding with STARs, we demand to be able to shield a policy over an MDP 'blindly', i.e. without assuming access to the reward-structure over M or knowledge of the actual policy. We therefore demand, in addition to $\mathcal{W}_\Phi^\star = Q$ assumed in Thm. 13 (further discussed in Rem. 14), that $\mathcal{W}_\Phi^\star$ is a strongly connected component (SCC). With this assumption, we achieve ε-optimality of the average reward for the special class of Büchi objectives as a consequence of Thm. 11.

▶ **Theorem 15.** *Given the premises of Cor. 9 with $\sigma' = \sigma|_{\gamma}^{\Gamma^\star,\theta}$, Büchi objective Φ, and SCC $\mathcal{W}_\Phi^\star = Q$, for every $\varepsilon > 0$, there exist $\theta, \gamma > 0$ such that $Avg_{\sigma'}^{q_0} > Avg_{\sigma}^{q_0} - \varepsilon$ holds.*

Unfortunately, the ε-optimality of shielded policies established for Büchi objectives in Thm. 15 does not directly generalize to parity conditions. To 'blindly' shield for the latter, we require the following definition.

▶ **Definition 16.** *Let M be an MDP and $c : Q \to [0; d]$ be a coloring function induced by a parity condition Φ over M. Let $\widetilde{Q} \subseteq Q$ be an SCC. We say $\tilde{Q}$ is $\star$-good w.r.t. Φ if $q \in \{\widetilde{Q} \mid c(q)$ is odd$\}$ implies $q \in \mathcal{W}_{\varphi_{good}}^\star$, where $\varphi_{good} := \{\rho \in Runs^M|_{\widetilde{Q}} \mid \exists n \geq 0 : c(\rho[n]) > c(q)$ and is even$\}$, i.e., from every odd state in $\widetilde{Q}$, the system player can surely/almost surely visit a higher even state in $\widetilde{Q}$.*

It is known that the optimal average reward achievable over an MDP M while (almost) surely satisfying a parity condition Φ reduces to (i) finding all $\star$-good SCC's of M w.r.t. Φ, (ii) computing the optimal average reward of each $\star$-good SCC, and (iii) enforcing reaching a $\star$-good SCC with the highest achievable optimal average reward (see [1] for details). As a consequence, the results of Thm. 15 carry over to parity objectives if Q is a $\star$-good SCC.

In addition, within $\star$-good SCCs, the full expressive power of strategy templates is not required—adding co-live templates D does not yield additional winning strategies. This is because for any state q, if q' is a state with the maximal color reachable (almost) surely from q, then $c(q')$ must be even; otherwise, by definition of a $\star$-good SCC, a higher even-colored state would be (almost) surely reachable. Consequently, winning strategy templates over $\star$-good SCCs only require live-groups to reach maximum (even) color states from each state (see App. B for details). It follows that $\textsc{ParityTemplate}_\star(G^M_\star|_{\widetilde{Q}}, \Phi)$ contains only live-groups and unsafe edges, as in the Büchi case. The following result is thus a direct corollary of Thm. 15.

▶ **Corollary 17.** *Given the premises of Cor. 9 with $\sigma' = \sigma|^{\Gamma_\star,\theta}_{\gamma}$, parity objective Φ, and $\star$-good SCC $\mathcal{W}^\star_\Phi = Q$, for every $\varepsilon > 0$, there exist $\theta, \gamma > 0$ such that $Avg^{q_0}_{\sigma'} > Avg^{q_0}_{\sigma} - \varepsilon$ holds.*

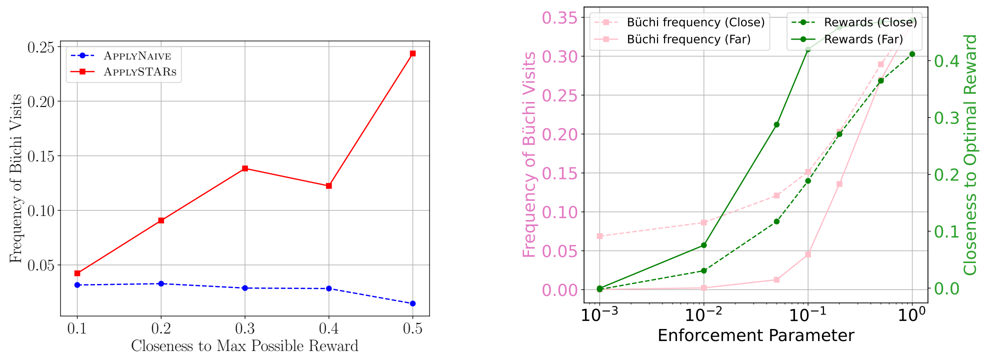
▶ Remark 18. We note that Rem. 14 directly transfers to Thm. 15 and Cor. 17. That is, we can restrict the training of an optimal average reward policy to a ($\star$-good-)SCCs and shield the resulting policy with STARs therein. If we would like to maximize the state space over which our shield is applicable, we can only restrict learning to $\mathcal{W}^\star_\Phi$, as in the discounted reward case. We can then compute separate STARs for every $\star$-good SCC of a given MDP M and an additional STARs synthesized for the objective to reach some $\star$-good SCC. However, in order to ensure that the resulting shielded policy is optimal in the above sense, we would need to enforce reaching the $\star$-good SCC with the highest achievable reward. Determining this SCC would, however, need access to the reward structure of the MDP M which we assume not to have. Hence, shielding 'blindly' in this case would still be minimally interfering in the sense of Thm. 10 but not necessarily optimal in the sense of Cor. 17.

4.3 A Note on the Quality of Shielded Policies

Given the fact that Thm. 15 and Cor. 17 restrict attention to ($\star$-good-)SCCs $\widetilde{Q} \subseteq Q$ we note that any stochastic policy σ satisfies Φ almost surely within $\widetilde{Q}$ already without any shielding. This is due to the fact that under stochastic policies all edges have positive probability of being sampled and, therefore, infinite runs reach all states in the SCC almost surely. As the maximum color in a ($\star$-good-)SCCs is even, all runs satisfy Φ almost- surely. In practice, however, the frequency with which even color vertices are seen is extremely low. While one might suggest that perturbing the nominal policy σ might increase the frequency of visiting even color states, this is actually not the case, as this perturbed policy would explore the *entire state space* more aggressively. We show this effect via experiments in Sec. 5, where we call the algorithm that implements the discussed naive perturbation of σ $\textsc{ApplyNaive}$.

In contrast, STARs modify probabilities in a targeted fashion. This (i) avoids visiting odd-color vertices which are not optimal, and (ii) allows to tune the desired frequency in which even color vertices are visited via the enforcement parameter. This can be formalized using the notion of *frequency* of a run ρ visiting a set T of states which can be defined as $\texttt{freq}(\rho, T) = \frac{1}{|Q|} \limsup_{l \to \infty} \frac{1}{l} \left|\{i \in [0; l] \mid \rho[i] \in T\}\right|$. With this definition, the following theorem[4] ensures that the frequency of a run ρ visiting even color states can be increased by tuning the enforcement parameter γ.

[4] Note that this shows the existence of such parameters only for the case of surely satisfying Φ. For the case of almost-sure satisfaction, the frequency would also depend on the transition probabilities of the MDP and hence, we cannot guarantee the existence of such parameters for every δ, while the same intuition still holds.



(a) Trade-off between frequency of Büchi region visits and average reward.

(b) Relating γ to frequency of Büchi region visits and the average reward.

Figure 4 Experimental evaluation summary (larger plots in App. C).

▶ **Theorem 19.** *Given the premises of Thm. 15 with $\sigma' = \sigma|_{\gamma}^{\Gamma_\bullet,\theta}$ and T being the set of even color states in the objective Φ, for every frequency $0 < \delta \leq 1$, there exists parameters $\theta, \gamma > 0$ such that for every run $\rho \sim \sigma'$, it holds that $\texttt{freq}(\rho, T) \geq \delta$.*

5 Experiments

This section shows the utility of STARs for (post-)shielding a learned policy to satisfy an ω-regular objective over the benchmark FactoryBot of grid worlds (see Fig. 2). As described already in Sec. 1, the robot in FactoryBot is following a policy σ trained to maximize the average reward collected over a randomly generated grid and a user can define additional safety and liveness obligations (incrementally) through the provided GUI.

As the tools implementing the existing shielding approaches closest to STARs from [31] and [7] are not accessible, we do not provide an empirical comparison. Instead, we have implemented both ApplySTARs (our shielding algorithm depicted in Fig. 1) and ApplyNaive (discussed in Sec. 4.3) in our Python-based prototype tool `MARG` (Monitoring and Adaptive Runtime Guide) to compare their performance. Our experiments focus on the evaluation of dynamic interference via the online-tuning of the enforcement parameter γ, which is not supported by any existing technique.

Benchmark. The benchmark suite generates random grids based on the values of the key parameters `size`, `min_distance` ($\in [0,1]$) and `max_distance` ($\in [0,1]$). The generated instance is a `size`$\times$`size` grid with randomly generated walls between the cells, in which the region with positive rewards is minimum `min_distance`$\cdot$`size` and maximum `max_distance`$\cdot$`size` l_1-distance away from every cell in the Büchi region. Due to resource constraints, we generate grids with `size` between 5 and 13. We also generate two sets of instances — `Far` with `min_distance` $= 0.7$ and `max_distance` $= 0.9$, and `Close` with `min_distance` $= 0.1$ and `max_distance` $= 0.2$. Fig. 2 in Sec. 1 shows example instances for a 6×6 grid in the category `Far` and `Close`.

Experimental Setup. We test ApplySTARs and ApplyNaive on 383 instances (189 in `Far` and 194 in `Close`) to guide a robot. We first find a policy to maximize the average reward without considering the Büchi objective. Then we apply ApplySTARs and ApplyNaive on the learned policy. For every instance we measure the number of times the robot visits the Büchi region and the average reward in 100,000 steps starting from a random initial position in the grid for both approaches. The experiments were run on a 32-core Debian machine with an Intel Xeon E5-V2 CPU (3.3 GHz) and up to 256 GB of RAM.

ApplySTARs vs ApplyNaive. We evaluate the trade-off between the frequency of visiting the Büchi region (henceforth, the Büchi frequency) and the average reward obtained by the two shielding approaches. We observe that, in order to ensure higher average reward, the robots need to spend more and more time in the region giving them positive reward (henceforth, high payoff region, $\mathcal{R}$). While ApplyNaive ensures visiting the Büchi region $\mathcal{B}$ infinitely often in the long run, the robot aimlessly wanders around the grid always eventually reaching $\mathcal{B}$ 'accidentally'. However, since STARs use strategy templates to guide the robot, the steps towards $\mathcal{B}$ are guided. This allows ApplySTARs to waste fewer steps not collecting the reward while visiting $\mathcal{B}$.

This observation is substantiated by our evaluation. Fig. 4a plots the average Büchi frequency (y-axis) for all instances that obtained an average reward that is ε-close to the maximal possible reward in that instance, where $\varepsilon \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6\}$ (x-axis). The red line represents the averages for the robot shielded by ApplySTARs, and the dashed blue line represents the same for the one shielded by ApplyNaive. We observe that ApplySTARs can maintain a similar average reward as ApplyNaive, while ensuring a much higher Büchi frequency. In addition, ApplyNaive does not allow to increase the Büchi frequency beyond a very low level. On the other hand, by sacrificing on the average reward, ApplySTARs allows attaining very high Büchi frequency.

Tuning γ. The previous section shows that ApplySTARs can be used to reach a very high Büchi frequency when sacrificing optimality w.r.t. the obtained average reward. To understand the actual trade-off between the Büchi frequency and obtained average reward, we evaluated the effects of the enforcement parameter γ on both measures for instances grouped into the categories `Far` and `Close`. As noted above, in order to increase the average reward of a run, the robot needs to stay longer and longer in $\mathcal{R}$. As the distance to $\mathcal{B}$ affects the time it spends away from $\mathcal{R}$, this measure directly impacts the average reward: the higher the distance between $\mathcal{B}$ and $\mathcal{R}$, the smaller should γ be to restrict the Büchi frequency, to attain a given closeness ε to the average reward.

This theoretical dependence is supported by Fig. 4b which shows the Büchi frequency (pink) and the proximity to the maximum average reward (green) attained by ApplySTARs for a given enforcement parameter γ, over instances from `Far` (dashed) and `Close` (solid), respectively. We observe that as the enforcement parameter increases, the Büchi frequency increases while the average reward gets further away from the optimal for both classes of instances. As expected, these trends have a higher slope on `Far` instances.

► Remark 20. We chose to only report on experiments with optimal average reward policies, as discounted rewards are less challenging from a shielding perspective. Optimal discounted reward policies crucially depend on the beginning of the robot trace, while ω-regular objectives can be satisfied independently of any finite prefix (when started in the winning region). This leads to a trivial shielding approach in the FactoryBot benchmark: one first chooses a very low γ value in the beginning which is cranked up in the tail of the execution. While this naturally gives a high performance w.r.t. both objectives, we want to emphasize that this natural dynamic shielding of optimal discounted reward policies is only possible because STARs allow for dynamic modifications of γ during runtime and is therefore also a major advantage of our framework over existing techniques.

References

1 Shaull Almagor, Orna Kupferman, and Yaron Velner. Minimizing expected cost under hard boolean constraints, with applications to quantitative synthesis. In *CONCUR*, volume 59 of *LIPIcs*, pages 9:1–9:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016.

2 Mohammed Alshiekh, Roderick Bloem, Rüdiger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu. Safe reinforcement learning via shielding. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

3 Ashwani Anand, Kaushik Mallik, Satya Prakash Nayak, and Anne-Kathrin Schmuck. Computing adequately permissive assumptions for synthesis. In *TACAS (2)*, volume 13994 of *Lecture Notes in Computer Science*, pages 211–228. Springer, 2023.

4 Ashwani Anand, Satya Prakash Nayak, and Anne-Kathrin Schmuck. Synthesizing permissive winning strategy templates for parity games. In *CAV (1)*, volume 13964 of *Lecture Notes in Computer Science*, pages 436–458. Springer, 2023.

5 Ashwani Anand, Satya Prakash Nayak, and Anne-Kathrin Schmuck. Strategy templates - robust certified interfaces for interacting systems. In *ATVA*, volume 15054 of *Lecture Notes in Computer Science*, pages 22–41. Springer, 2024.

6 Ashwani Anand, Anne-Kathrin Schmuck, and Satya Prakash Nayak. Contract-based distributed logical controller synthesis. In *HSCC*, pages 11:1–11:11. ACM, 2024.

7 Guy Avni, Roderick Bloem, Krishnendu Chatterjee, Thomas A. Henzinger, Bettina Könighofer, and Stefan Pranger. Run-time optimization for learned controllers through quantitative games. In Isil Dillig and Serdar Tasiran, editors, *Computer Aided Verification - 31st International Conference, CAV 2019, New York City, NY, USA, July 15-18, 2019, Proceedings, Part I*, volume 11561 of *Lecture Notes in Computer Science*, pages 630–649. Springer, 2019. doi:10.1007/978-3-030-25540-4_36.

8 Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking*. The MIT Press, 2008.

9 Julien Bernet, David Janin, and Igor Walukiewicz. Permissive strategies: from parity games to safety games. *RAIRO Theor. Informatics Appl.*, 36(3):261–275, 2002. doi:10.1051/ita:2002013.

10 Patricia Bouyer, Nicolas Markey, Jörg Olschewski, and Michael Ummels. Measuring permissiveness in parity games: Mean-payoff parity games revisited. In Tevfik Bultan and Pao-Ann Hsiung, editors, *Automated Technology for Verification and Analysis, 9th International Symposium, ATVA 2011, Taipei, Taiwan, October 11-14, 2011. Proceedings*, volume 6996 of *Lecture Notes in Computer Science*, pages 135–149. Springer, 2011. doi:10.1007/978-3-642-24372-1_11.

11 Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016. arXiv:arXiv:1606.01540.

12 Damien Busatto-Gaston, Debraj Chakraborty, Shibashis Guha, Guillermo A. Pérez, and Jean-François Raskin. Safe learning for near-optimal scheduling. In *QEST*, volume 12846 of *Lecture Notes in Computer Science*, pages 235–254. Springer, 2021.

13 Krishnendu Chatterjee and Laurent Doyen. Energy parity games. In *ICALP (2)*, volume 6199 of *Lecture Notes in Computer Science*, pages 599–610. Springer, 2010.

14 Krishnendu Chatterjee and Laurent Doyen. Games and markov decision processes with mean-payoff parity and energy parity objectives. In *MEMICS*, volume 7119 of *Lecture Notes in Computer Science*, pages 37–46. Springer, 2011.

15 Ziyang Chen, Mingyu Cai, Zhangli Zhou, Lin Li, and Zhen Kan. Fast motion planning in dynamic environments with extended predicate-based temporal logic. *IEEE Transactions on Automation Science and Engineering*, pages 1–15, 2024. doi:10.1109/TASE.2024.3418409.

16 David Dalrymple, Joar Skalse, Yoshua Bengio, Stuart Russell, Max Tegmark, Sanjit Seshia, Steve Omohundro, Christian Szegedy, Ben Goldhaber, Nora Ammann, et al. Towards guaranteed safe ai: A framework for ensuring robust and reliable ai systems. *arXiv preprint arXiv:2405.06624*, 2024.

17 Ingy Elsayed-Aly, Suda Bharadwaj, Christopher Amato, Rüdiger Ehlers, Ufuk Topcu, and Lu Feng. Safe multi-agent reinforcement learning via shielding. In *AAMAS*, pages 483–491. ACM, 2021.

18 Daniel J. Fremont and Sanjit A. Seshia. Reactive control improvisation. In Hana Chockler and Georg Weissenbacher, editors, *Computer Aided Verification*, pages 307–326, Cham, 2018. Springer International Publishing.

19 David Gundana and Hadas Kress-Gazit. Event-based signal temporal logic synthesis for single and multi-robot tasks. *IEEE Robotics and Automation Letters*, 6(2):3687–3694, 2021. doi:10.1109/LRA.2021.3064220.

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

21 Ernst Moritz Hahn, Mateo Perez, Sven Schewe, Fabio Somenzi, Ashutosh Trivedi, and Dominik Wojtczak. Multi-objective ω-regular reinforcement learning. *Form. Asp. Comput.*, 35(2), July 2023. doi:10.1145/3605950.

22 Ernst Moritz Hahn, Mateo Perez, Sven Schewe, Fabio Somenzi, Ashutosh Trivedi, and Dominik Wojtczak. Omega-regular reward machines. In *ECAI*, volume 372 of *Frontiers in Artificial Intelligence and Applications*, pages 972–979. IOS Press, 2023.

23 Klaus Havelund and Grigore Roşu. Synthesizing monitors for safety properties. In *Tools and Algorithms for the Construction and Analysis of Systems: 8th International Conference, TACAS 2002 Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2002 Grenoble, France, April 8–12, 2002 Proceedings 8*, pages 342–356. Springer, 2002.

24 Kai-Chieh Hsu, Haimin Hu, and Jaime F. Fisac. The safety filter: A unified view of safety-critical control in autonomous systems. *Annual Review of Control, Robotics, and Autonomous Systems*, 7(Volume 7, 2024):47–72, 2024. URL: https://www.annualreviews.org/content/journals/10.1146/annurev-control-071723-102940, doi:10.1146/annurev-control-071723-102940.

25 Loïc Hélouët, Nicolas Markey, and Ritam Raha. Reachability games with relaxed energy constraints. In Jérôme Leroux and Jean-François Raskin, editors, *Proceedings Tenth International Symposium on Games, Automata, Logics, and Formal Verification, GandALF 2019, Bordeaux, France, 2-3rd September 2019*, volume 305 of *EPTCS*, pages 17–33, 2019. doi:10.4204/EPTCS.305.2.

26 Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the Nineteenth International Conference on Machine Learning*, ICML '02, page 267–274, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.

27 Samarth Kalluraya, George J. Pappas, and Yiannis Kantaros. Multi-robot mission planning in dynamic semantic environments. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1630–1637, 2023. doi:10.1109/ICRA48891.2023.10160344.

28 Milad Kazemi, Mateo Perez, Fabio Somenzi, Sadegh Soudjani, Ashutosh Trivedi, and Alvaro Velasquez. Translating omega-regular specifications to average objectives for model-free reinforcement learning. In *AAMAS*, pages 732–741. International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS), 2022.

29 Joachim Klein, Christel Baier, and Sascha Klüppelholz. Compositional construction of most general controllers. *Acta Informatica*, 52(4-5):443–482, 2015. doi:10.1007/s00236-015-0239-9.

30 Bettina Könighofer, Roderick Bloem, Rüdiger Ehlers, and Christian Pek. *Correct-by-Construction Runtime Enforcement in AI – A Survey*, pages 650–663. Springer Nature Switzerland, Cham, 2022.

31 Bettina Könighofer, Julian Rudolf, Alexander Palmisano, Martin Tappler, and Roderick Bloem. Online shielding for reinforcement learning. *Innovations in Systems and Software Engineering*, 19(4):379–394, 2023.

32 Hadas Kress-Gazit, Georgios E. Fainekos, and George J. Pappas. Temporal-logic-based reactive mission and motion planning. *IEEE Transactions on Robotics*, 25(6):1370–1381, 2009. doi:10.1109/TRO.2009.2030225.

33 Hadas Kress-Gazit, Morteza Lahijanian, and Vasumathi Raman. Synthesis for robots: Guarantees and feedback for robot behavior. *Annual Review of Control, Robotics, and Autonomous Systems*, 1(1):211–236, 2018.

34 Jan Kretínský, Guillermo A. Pérez, and Jean-François Raskin. Learning-based mean-payoff optimization in an unknown MDP under omega-regular constraints. In *CONCUR*, volume 118 of *LIPIcs*, pages 8:1–8:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.

35 Insup Lee, Sampath Kannan, Moonzoo Kim, Oleg Sokolsky, and Mahesh Viswanathan. Runtime assurance based on formal specifications. In *International Conference on Parallel and Distributed Processing Techniques and Applications*, pages 279–287. PDPTA', 1999.

36 Noushin Mehdipour, Matthias Althoff, Radboud Duintjer Tebbens, and Calin Belta. Formal methods to comply with rules of the road in autonomous driving: State of the art and grand challenges. *Automatica*, 152:110692, 2023. URL: https://www.sciencedirect.com/science/article/pii/S0005109822005568, doi:10.1016/j.automatica.2022.110692.

37 Noushin Mehdipour, Matthias Althoff, Radboud Duintjer Tebbens, and Calin Belta. Formal methods to comply with rules of the road in autonomous driving: State of the art and grand challenges. *Automatica*, 152:110692, 2023. URL: https://www.sciencedirect.com/science/article/pii/S0005109822005568, doi:10.1016/j.automatica.2022.110692.

38 Satya Prakash Nayak, Lucas N. Egidio, Matteo Della Rossa, Anne-Kathrin Schmuck, and Raphael M. Jungers. Context-triggered abstraction-based control design. *IEEE Open Journal of Control Systems*, 2:277–296, 2023. doi:10.1109/OJCSYS.2023.3305835.

39 Satya Prakash Nayak and Anne-Kathrin Schmuck. Most general winning secure equilibria synthesis in graph games. In *TACAS (3)*, volume 14572 of *Lecture Notes in Computer Science*, pages 173–193. Springer, 2024.

40 Haritz Odriozola-Olalde, Maider Zamalloa, and Nestor Arana-Arexolaleiba. Shielded reinforcement learning: A review of reactive methods for safe learning. In *SII*, pages 1–8. IEEE, 2023.

41 Adam Pacheck and Hadas Kress-Gazit. Physically feasible repair of reactive, linear temporal logic-based, high-level tasks. *IEEE Transactions on Robotics*, 39(6):4653–4670, 2023. doi:10.1109/TRO.2023.3304009.

42 Shashank Pathak, Luca Pulina, and Armando Tacchella. Verification and repair of control policies for safe reinforcement learning. *Applied Intelligence*, 48(4):886–908, April 2018. doi:10.1007/s10489-017-0999-8.

43 Kittiphon Phalakarn, Sasinee Pruekprasert, and Ichiro Hasuo. Winning strategy templates for stochastic parity games towards permissive and resilient control. In Chutiporn Anutariya and Marcello M. Bonsangue, editors, *Theoretical Aspects of Computing – ICTAC 2024*. Springer Nature Switzerland, 2025.

44 Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., USA, 1st edition, 1994.

45 Peter JG Ramadge and Walter Murray Wonham. The control of discrete event systems. *Proceedings of the IEEE*, 77(1):81–98, 1989.

46 Robert Reed, Hanspeter Schaub, and Morteza Lahijanian. Shielded deep reinforcement learning for complex spacecraft tasking. *arXiv preprint arXiv:2403.05693*, 2024.

47 Yagiz Savas, Christos K. Verginis, Michael Hibbard, and Ufuk Topcu. On minimizing total discounted cost in mdps subject to reachability constraints. *IEEE Transactions on Automatic Control*, 69(9):6466–6473, 2024. doi:10.1109/TAC.2024.3384834.

48 Anne-Kathrin Schmuck, Philippe Heim, Rayna Dimitrova, and Satya Prakash Nayak. Localized attractor computations for infinite-state games. In *CAV (3)*, volume 14683 of *Lecture Notes in Computer Science*, pages 135–158. Springer, 2024.

49 John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.

50 John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017.

51 Sanjit A Seshia, Dorsa Sadigh, and S Shankar Sastry. Toward verified artificial intelligence. *Communications of the ACM*, 65(7):46–55, 2022.

52 Martin Tappler, Andrea Pferscher, Bernhard K. Aichernig, and Bettina Konighofer. Learning and Repair of Deep Reinforcement Learning Policies from Fuzz-Testing Data . In *2024 IEEE/ACM 46th International Conference on Software Engineering (ICSE)*, pages 38–50, Los Alamitos, CA, USA, April 2024. IEEE Computer Society. URL: https://doi.ieeecomputersociety.org/10.1145/3597503.3623311, `doi:10.1145/3597503.3623311`.

53 Sanne van Waveren, Christian Pek, Jana Tumova, and Iolanda Leite. Correct me if i'm wrong: Using non-experts to repair reinforcement learning policies. In *2022 17th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 493–501, 2022. `doi:10.1109/HRI53351.2022.9889604`.

54 Vasileios Vasilopoulos, Yiannis Kantaros, George J. Pappas, and Daniel E. Koditschek. Reactive planning for mobile manipulation tasks in unexplored semantic environments. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6385–6392, 2021. `doi:10.1109/ICRA48506.2021.9561958`.

55 Kim P. Wabersich, Andrew J. Taylor, Jason J. Choi, Koushil Sreenath, Claire J. Tomlin, Aaron D. Ames, and Melanie N. Zeilinger. Data-driven safety filters: Hamilton-jacobi reachability, control barrier functions, and predictive methods for uncertain systems. *IEEE Control Systems Magazine*, 43(5):137–177, 2023. `doi:10.1109/MCS.2023.3291885`.

56 Wen-Chi Yang, Giuseppe Marra, Gavin Rens, and Luc De Raedt. Safe reinforcement learning via probabilistic logic shields. In *IJCAI*, pages 5739–5749. ijcai.org, 2023.

57 Christopher K. Zeitler, Kristina Miller, Sayan Mitra, John Schierman, and Mahesh Viswanathan. Optimizing rewards while meeting ω-regular constraints. *RLJ*, 5:2492–2514, 2024.

58 Yiming Zhang and Keith W. Ross. On-policy deep reinforcement learning for the average-reward criterion. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 12535–12545. PMLR, 2021. URL: http://proceedings.mlr.press/v139/zhang21q.html.

59 Weichao Zhou, Ruihan Gao, BaekGyu Kim, Eunsuk Kang, and Wenchao Li. Runtime-safety-guided policy repair. *CoRR*, abs/2008.07667, 2020. URL: https://arxiv.org/abs/2008.07667, `arXiv:2008.07667`.

A Proofs

▶ **Theorem 8.** *Given the premises of Def. 7 it holds that* $\sigma|_{\gamma}^{\Gamma_\star,\theta}$ *follows* $\Gamma_\star$.

Proof. Let $\sigma' := \sigma|_{\gamma}^{\Gamma_\star,\theta}$ (for notational convinience) and ρ a σ'-run of M. We show that ρ satisfies the template $\Gamma_\star = (S, D, H_\ell)$. As probability of an unsafe edge $(q, a) \in S$ is set to zero by σ', the safety template S is satisfied.

Let's assume that ρ does not satisfy the co-live template D. Then, there exists a co-live edge $e = (q, a)$ that appears in ρ infinitely many times. Let κq be a finite prefix of ρ such that a has been sampled from q more than $1/\gamma$ times. Then, $count_{(q,a)}(\kappa q) > 1/\gamma$, which means $\sigma(\kappa q, a) - \gamma \cdot count_{(q,a)}(\kappa q) < 0$. Consequently, the probability of choosing e after history κq under σ' becomes zero. Hence, ρ can not visit q more than $1/\gamma$ times, which contradicts the assumption that ρ visits (q, a) infinitely many times. Thus, ρ satisfies the co-live template D.

Next, suppose ρ does not satisfy the live-group template H_ℓ. Then, there exists a live group $H \in H_\ell$ such that ρ visits the source states of H infinitely many times but does not sample any action from H infinitely many times. Let $q \in src(H)$ be a state that is visited infinitely many times by ρ, and let q be a source state for live-groups $H_1, \ldots, H_l$ (with $H_1 = H$). Suppose ρ does not satisfy the live-group template H_i for all $i \le l'$ and satisfies H_i for all $l' < i \le l$. Note that $l' > 1$ as ρ does not satisfy $H = H_1$. Then, for every history κq of ρ after which ρ does not contain any co-live edge and for every action $a' \in A' = \{a' \in A(q) \mid (q, a') \notin \bigcup_{i=1}^{l'} H_i\}$, we have:

$$\sigma'(\kappa q, a') \le \frac{\sigma(\kappa q, a) + \gamma \cdot \sum_{i=l'}^{l} count_{H_i}(\kappa q)}{1 + \gamma \cdot \sum_{i=1}^{l} count_{H_i}(\kappa q)}.$$

Furthermore, there exists a history κ' of ρ after which ρ (does not use any co-live edge and) visits q infinitely many times but never samples an action from any of the live-groups $H_1, \ldots, H_{l'}$. From that point on, the counter $count_{H_i}$ for $i \le l'$ is incremented unboudedly, whereas the counter $count_{H_i}$ for $i \ge l'$ is reset to zero infinitely many times. Consequently, there exists a history κq of ρ (that is an extension of κ') such that for every $a' \in A'$, we have:

$$\frac{\sigma(\kappa q, a') + \gamma \cdot \sum_{i=l'}^{l} count_{H_i}(\kappa q)}{1 + \gamma \cdot \sum_{i=1}^{l} count_{H_i}(\kappa q)} < \theta.$$

By construction, $\sigma'(\kappa q, a') = 0$ for every $a' \in A'$, and hence, σ' has to sample an action from $A \setminus A'$ at history κq. As all actions in $A \setminus A'$ are from the live-groups $\bigcup_{i=1}^{l'} H_i$, this contradicts the assumption that ρ does not sample from $\bigcup_{i=1}^{l'}$ after κ'. Thus, ρ satisfies the live-group template H_ℓ. ◀

▶ **Theorem 10.** *Given the premises of Cor. 9 with* $\sigma' = \sigma|_{\gamma}^{\Gamma_\star,\theta}$, *for any* $\varepsilon > 0$ *and for all length* $l \in \mathbb{N}$, *there exist parameters* $\gamma, \theta > 0$ *such that for all histories* κ *of length* l *with* $\kappa \models pref(\Phi)$, *it holds that* $\Pr_{\sigma'}(\kappa) > \Pr_\sigma(\kappa) - \varepsilon$.

Proof. Let us fix an $\varepsilon > 0$ and a length l. Let $\kappa = q_0 a_0 \ldots q_l \in \mathrm{FRuns}^M$ be a history such that $\kappa \models pref(\Phi)$. Clearly, for edges $(q, a) \in S$, (q, a) cannot appear in κ as then any extension of κ will not satisfy Φ.

First, let $\sigma(\kappa[0; i], a_i) = x_i$ and $\Pr(q_{i+1}|q_i, a_i) = y_i$. Then it holds that $\Pr_\sigma(\kappa) = \prod_{0 \le i \le l-1} x_i y_i$. Note that whenever $x_i = 0$ for any $i \le l$, then $\Pr_\sigma(\kappa) = 0$ and hence, $\Pr_{\sigma'}(\kappa) > \Pr_\sigma(\kappa) - \varepsilon$ trivially holds. Let us therefore now consider the case where $x_i > 0$ for all $i \le l$.

Now, for every $i \leq l$, we have $\sum_{e \in D} count_e(\kappa[0;i]) \leq l$ and $\sum_{H \in H_\ell} count_H(\kappa[0;i]) \leq |H_\ell|\, l$. Hence, if $\sigma'(\kappa[0;i], a_i) = x'_i$, by taking small enough θ, it holds that

$$x'_i \geq \frac{x_i - l \cdot \gamma}{1 + |H_\ell|\, l \cdot \gamma}.$$

Thus, we have:

$$\Pr_{\sigma'}(\kappa) = \prod_{0 \leq i \leq l-1} x'_i y_i \geq \prod_{0 \leq i \leq l-1} \frac{x_i - l \cdot \gamma}{1 + |H_\ell|\, l \cdot \gamma} \cdot y_i.$$

For $x = \min_{0 \leq i \leq l-1} x_i$, we have:

$$\begin{aligned}
\Pr_{\sigma}(\kappa) - \Pr_{\sigma'}(\kappa) &\leq \prod_{0 \leq i \leq l-1} x_i y_i - \prod_{0 \leq i \leq l-1} \frac{x_i - l \cdot \gamma}{1 + |H_\ell|\, l \cdot \gamma} \cdot y_i \\
&= \Pr_{\sigma}(\kappa) \cdot \left(1 - \prod_{0 \leq i \leq l-1} \frac{1 - \frac{l \cdot \gamma}{x_i}}{1 + |H_\ell|\, l \cdot \gamma}\right) \\
&\leq \Pr_{\sigma}(\kappa) \cdot \left(1 - \left(\frac{1 - \frac{l \cdot \gamma}{x}}{1 + 1\, |H_\ell|\, l \cdot \gamma}\right)^l\right)
\end{aligned}$$

It now follows that by fixing an appropriate value for γ, one can bound the above expression by ε. As the above expression is independent of the choice of history κ, we can conclude that for every history κ of length l such that $\kappa \models \mathit{pref}(\Phi)$, it holds that: $\Pr_{\sigma'}(\kappa) > \Pr_{\sigma}(\kappa) - \varepsilon$. ◀

▶ **Theorem 11.** *Given the premises of Cor. 9 with $\sigma' = \sigma|_{\gamma}^{\Gamma_\star, \theta}$ such that $\Gamma_\star = (\emptyset, \emptyset, H_\ell)$, and a cost function $\mathtt{cost} : FRuns^M \to [0, W]$, for any $\varepsilon > 0$, there exist parameters $\gamma, \theta > 0$ such that the following holds: $\mathbb{E}_{\rho \sim \sigma'} \mathtt{cost}(\rho, \sigma, \sigma') < \varepsilon$.*

Proof. First let us show that the tuple of counter values can be bounded based on the parameters θ and γ. As $\Gamma_\star = (\emptyset, \emptyset, H_\ell)$, we only have counters for the live-groups in H_ℓ. Hence, the following claim formalizes the bound on the counter values.

▷ Claim 21. For parameters $K_{\theta,\gamma} = \max\left(1, \frac{1/\theta - 1}{\gamma}\right)$, every counter value $count_H(\kappa) \leq K_{\theta,\gamma}$ for every history κ and live-group $H \in H_\ell$.

Proof. By the construction of ParityTemplate$_\star$, as in [4, Alg. 3], every state q can be a source state for at most one live-group $H \in H_\ell$. Hence, for a history κ ending in such a state q, for every action $a \in A(q) \setminus H$, we have

$$\sigma'(\kappa, a) \leq \frac{\sigma(\kappa, a)}{1 + count_H(\kappa) \cdot \gamma} \leq \frac{1}{1 + count_H(\kappa) \cdot \gamma}.$$

Hence, whenever $count_H(\kappa) \geq K_{\theta,\gamma}$, we have $\sigma'(\kappa, a) \leq \theta$ and hence, by (1a), $\sigma'(\kappa, a) = 0$. Therefore, for $count_H(\kappa) \geq K_{\theta,\gamma}$, probability of sampling an action not in H becomes zero and hence, an action from H will be sampled and hence, $count_H(\kappa)$ will be reset to zero. So, the counter value $count_H(\kappa) \leq K_{\theta,\gamma}$ for every history κ. ◁

Now, w.l.o.g, let us assume that σ is a stationary policy in M as otherwise we can take the product of M with the memory set of σ to ensure that σ is stationary in the product MDP. Furthermore, σ' is a stationary policy w.r.t. the extended MDP $M' = \langle Q', A, \Delta', q'_0 \rangle$

obtained by taking product of the MDP with the tuples of counter values. Since σ is a stationary stochastic policy in M, it is also a stationary stochastic policy in M'.

Now, given an $\varepsilon > 0$ and a cost function $\mathtt{cost} : \mathrm{FRuns}^M \to [0, W]$, it holds that $\mathtt{cost}(\kappa) \leq W$ for every history κ. Hence, the following holds for stationary distribution $d_{\sigma'}$ of σ' in M':

$$\begin{aligned}
\mathbb{E}_{\rho\sim\sigma'}\mathtt{cost}(\rho,\sigma,\sigma') &= \mathbb{E}_{\rho\sim\sigma'}\left[\limsup_{l\to\infty}\frac{1}{l}\sum_{i=0}^{l-1}\mathtt{cost}(\rho[0;i])\cdot \mathtt{D_{TV}}(\sigma(\rho[0;i]),\sigma'(\rho[0;i]))\right]\\
&= W\cdot\mathbb{E}_{\rho\sim\sigma'}\left[\limsup_{l\to\infty}\frac{1}{l}\sum_{i=0}^{l-1}\mathtt{D_{TV}}(\sigma(\rho[0;i]),\sigma'(\rho[0;i]))\right]\\
&= W\cdot\limsup_{l\to\infty}\frac{1}{l}\cdot\mathbb{E}_{\rho\sim\sigma';|\rho|=l}\left[\sum_{i=0}^{l-1}\mathtt{D_{TV}}(\sigma(\rho[0;i]),\sigma'(\rho[0;i]))\right]\\
&= W\cdot\limsup_{l\to\infty}\frac{1}{l}\cdot\sum_{i=0}^{l-1}\mathbb{E}_{(q,C)\sim d_{\sigma'}}\left[\mathtt{D_{TV}}(\sigma(q),\sigma'((q,C)))\right]\\
&= W\cdot\mathbb{E}_{(q,C)\sim d_{\sigma'}}\left[\mathtt{D_{TV}}(\sigma(q),\sigma'((q,C)))\right].
\end{aligned}$$

Evaluating the total variation distance $\mathtt{D_{TV}}(\sigma(q),\sigma'((q,C)))$ gives us the following:

$$\mathbb{E}_{\rho\sim\sigma'}\mathtt{cost}(\rho,\sigma,\sigma') = \frac{1}{2}W\cdot\mathbb{E}_{(q,C)\sim d_{\sigma'}}\sum_{a\in A(q)}\left|\sigma(q,a)-\sigma'((q,C),a)\right|. \tag{3}$$

Then, as $\Gamma_\star = (\emptyset,\emptyset,H_\ell)$, if $\sigma''((q,C),a)$ is the distribution obtained as in (1b) before bounding the probabilities by θ, then the following holds:

$$\sigma''((q,C),a) \geq \frac{\sigma(q,a)}{1+|C|\,\gamma} \quad\text{and}\quad \sigma''((q,C),a) \leq \frac{\sigma(q,a)+|C|\,\gamma}{1+|C|\,\gamma},$$

where $|C|$ denotes the sum of the counters in C.
This means, after bounding the probabilities by θ, if $\sigma''((q,C),a) < \theta$, then $\sigma'((q,C),a) = 0 = \sigma''((q,C),a) - \theta$, and if some probabilities are bounded by θ, then $\sigma'((q,C),a) \leq \sigma''((q,C),a) + |A(q)|\cdot\theta$. Hence, it holds that:

$$\begin{aligned}
\sigma'((q,C),a) \geq \sigma''((q,C),a)-\theta \quad&\text{and}\quad \sigma'((q,C),a)\leq\sigma''((q,C),a)+|A(q)|\cdot\theta\\
\Rightarrow \sigma'((q,C),a) \geq \frac{\sigma(q,a)}{1+|C|\,\gamma}-\theta \quad&\text{and}\quad \sigma'((q,C),a)\leq\frac{\sigma(q,a)+|C|\,\gamma}{1+|C|\,\gamma}+|A(q)|\cdot\theta\\
\Rightarrow \sigma'((q,C),a) \geq \frac{\sigma(q,a)}{1+|C|\,\gamma}-|A(q)|\cdot\theta \quad&\text{and}\quad \sigma'((q,C),a)\leq\frac{\sigma(q,a)+|C|\,\gamma}{1+|C|\,\gamma}+|A(q)|\cdot\theta.
\end{aligned}$$

Therefore, we have:

$$\begin{aligned}
\left|\sigma(q,a)-\sigma'((q,C),a)\right| &\leq \max\left\{\sigma(q,a)-\sigma'((q,C),a),\quad \sigma'((q,C),a)-\sigma(q,a)\right\}\\
&\leq \max\left\{\sigma(q,a)\left(\frac{|C|\,\gamma}{1+|C|\,\gamma}\right),\frac{|C|\,\gamma}{1+|C|\,\gamma}(1-\sigma(q,a))\right\}+|A(q)|\cdot\theta\\
&\leq |C|\,\gamma+|A|\,\theta.
\end{aligned}$$

Hence, the expected cost in (3) can be bounded as follows:

$$\begin{aligned}
\mathbb{E}_{\rho\sim\sigma'}\texttt{cost}(\rho,\sigma,\sigma') &\leq \frac{1}{2}W\cdot\mathbb{E}_{(q,C)\sim d_{\sigma'}}\sum_{a\in A(q)}(|C|\,\gamma+|A|\cdot\theta)\\
&\leq \frac{1}{2}W\cdot\sum_{a\in A}\mathbb{E}_{(q,C)\sim d_{\sigma'}}[|C|\,\gamma+|A|\cdot\theta]\\
&\leq \frac{1}{2}W\cdot|A|^2\cdot\theta+W\,|A|\,\gamma\cdot\mathbb{E}_{(q,C)\sim d_{\sigma'}}[|C|]
\end{aligned}$$

From the construction of the shielded policy σ', we know that $|C|$ increases only if the current state q is a source state of some live edge in a live-group $H\in H_\ell$ and none of the live edges in H are sampled. From such a state (q,C) with high enough counter value $count_H(C)$, the probability of sampling an action that is not in H can be expressed in terms of $\sigma(q,A(q)\cap H)=\sum_{a\in A(q)\cap H}\sigma(q,a)$ as follows:

$$\begin{aligned}
\Pr[(q,C)\to C+1] &= 1-\sum_{a\in A(q)\cap H}\sigma'((q,C),a)\\
&\leq 1-\frac{\sigma(q,A(q)\cap H)+count_H(C)}{1+count_H(C)\cdot\gamma}\\
&\leq \frac{1-\sigma(q,A(q)\cap H)}{1+count_H(C)\cdot\gamma}\\
&\leq 1-\sigma(q,A(q)\cap H).
\end{aligned}$$

If $\sigma(q,A(q)\cap H)=1$, then $\Pr[(q,C)\to C+1]=0$. Hence, if $\Pr[(q,C)\to C+1]>0$, then the probability $\Pr[(q,C)\to C+1]$ can be bounded by

$$m=\max\{1-\sigma(q,A(q)\cap H)\mid q\in Q,H\in H_\ell,A(q)\cap H\neq\emptyset,\sigma(q,A(q)\cap H)<1\}.$$

Note that $m>0$ as σ is a stochastic policy and $m<1$ by the above construction. Hence, the probability that the counter sum $|C|$ increases can be bounded by m. Therefore, the expected value of $|C|$ can be bounded as follows:

$$\mathbb{E}_{(q,C)\sim d_{\sigma'}}[|C|]\leq\sum_{i=0}^{\infty}i\cdot m^i=\frac{m}{(1-m)^2}\leq\frac{1}{(1-m)^2}.$$

Therefore, the expected cost in (3) can be bounded as follows:

$$\mathbb{E}_{\rho\sim\sigma'}\texttt{cost}(\rho,\sigma,\sigma')\leq\frac{1}{2}W\cdot|A|^2\cdot\theta+W\,|A|\,\gamma\cdot\frac{1}{(1-m)^2}.$$

By fixing θ and γ appropriately, we can ensure that the above expression is less than ε. ◀

▶ **Theorem 13.** *Given the premises of Cor. 9 with $\sigma'=\sigma|_{\gamma}^{\Gamma^\star,\theta}$ and $\mathcal{W}_\Phi^\star=Q$, for every $\varepsilon>0$, there exist parameters $\theta,\gamma>0$ such that the following holds: $Disc_{\sigma'}^{q_0}(\lambda)>Disc_{\sigma}^{q_0}(\lambda)-\varepsilon$.*

Proof. Let us first compute a bound on the length of the runs that are significant to get an ε-optimal reward. Let r_{max} be the maximum reward in the MDP M, and let $\mathrm{Disc}_{\sigma'}^{q_0}(\lambda,k)$ be the expected bounded discounted reward of policy σ' for the first k steps:

$$\mathrm{Disc}_{\sigma'}^{q_0}(\lambda,k)=\sum_{1\leq i\leq k}\lambda^i\mathbb{E}_{\sigma}^{q_0}(r(X_i,Y_i))$$

Now, let's choose B such that $\lambda^B \cdot r_{max} < (1-\lambda)\cdot\varepsilon/2$. Note that such a B exists as $\lambda^B \to 0$ when $B \to \infty$. This gives us the following for any policy σ':

$$\begin{aligned}
\mathrm{Disc}^{q_0}_{\sigma'}(\lambda) &= \lim_{N\mapsto\infty} \sum_{0\le i\le N-1} \lambda^i \mathbb{E}^{q_0}_{\sigma}(r(X_i,Y_i)) \\
&= \mathrm{Disc}^{q_0}_{\sigma'}(\lambda,B) + \lim_{N\mapsto\infty} \sum_{B+1\le i\le N} \lambda^i \mathbb{E}^{q_0}_{\sigma}(r(X_i,Y_i)) \\
&\le \mathrm{Disc}^{q_0}_{\sigma'}(\lambda,B) + \lim_{N\mapsto\infty} \sum_{B+1\le i\le N} \lambda^i \cdot r_{max} \\
&= \mathrm{Disc}^{q_0}_{\sigma'}(\lambda,B) + \frac{\lambda^B\cdot r_{max}}{1-\lambda} \\
&< \mathrm{Disc}^{q_0}_{\sigma'}(\lambda,B) + \frac{\varepsilon}{2}
\end{aligned}$$

Now, let $r(\kappa) = \sum_{0\le i\le|\kappa|-1} \lambda^i r(X_i,Y_i)$ be the reward of a run κ. Then, we can rewrite $\mathrm{Disc}^{q_0}_{\sigma'}(\lambda,k)$ in terms of the expected reward of k-length runs as follows:

$$\mathrm{Disc}^{q_0}_{\sigma'}(\lambda,k) = \sum_{\kappa\in\mathrm{FRuns}^M;|\kappa|=k} \mathbb{E}^{q_0}_{\sigma}(\Pr(\kappa))\cdot r(\kappa)$$

Hence, the difference between bounded discounted reward for optimal policy σ and shielded policy σ' is the following:

$$\begin{aligned}
\mathrm{Disc}^{q_0}_{\sigma}(\lambda,B) - \mathrm{Disc}^{q_0}_{\sigma'}(\lambda,B) &= \sum_{\kappa\in\mathrm{FRuns}^M;|\kappa|=B} (\mathbb{E}^{q_0}_{\sigma}(\Pr(\kappa)) - \mathbb{E}^{q_0}_{\sigma'}(\Pr(\kappa)))\cdot r(\kappa) \\
&= \sum_{\kappa\in\mathrm{FRuns}^M;|\kappa|=B} (\Pr_{\sigma'}(\kappa) - \Pr_{\sigma}(\kappa))\cdot r(\kappa) \\
&\le |Q|^B\cdot(\Pr_{\sigma'}(\kappa) - \Pr_{\sigma}(\kappa))\cdot r(\kappa).
\end{aligned}$$

As $\mathcal{W}^\star_\Phi = Q$, every history $\kappa \models \mathit{pref}(\Phi)$. Hence, using Thm. 10, for length B, there exists parameters $\theta,\gamma > 0$ such that we can bound the above term by $\varepsilon/2$. Using the property of bound B, this gives us:

$$\begin{aligned}
&\mathrm{Disc}^{q_0}_{\sigma}(\lambda,B) - \mathrm{Disc}^{q_0}_{\sigma'}(\lambda,B) < \varepsilon/2 \\
\implies &\mathrm{Disc}^{q_0}_{\sigma'}(\lambda,B) + \varepsilon/2 > \mathrm{Disc}^{q_0}_{\sigma}(\lambda,B) > \mathrm{Disc}^{q_0}_{\sigma}(\lambda) - \varepsilon/2 \\
\implies &\mathrm{Disc}^{q_0}_{\sigma'}(\lambda,B) > \mathrm{Disc}^{q_0}_{\sigma}(\lambda) - \varepsilon.
\end{aligned}$$

As $\mathrm{Disc}^{q_0}_{\sigma'}(\lambda) \ge \mathrm{Disc}^{q_0}_{\sigma'}(\lambda,B)$, we have that $\mathrm{Disc}^{q_0}_{\sigma'}(\lambda) > \mathrm{Disc}^{q_0}_{\sigma}(\lambda) - \varepsilon$. ◀

▶ **Theorem 15.** *Given the premises of Cor. 9 with $\sigma' = \sigma|^{\Gamma^\star,\theta}_{\gamma}$, Büchi objective Φ, and SCC $\mathcal{W}^\star_\Phi = Q$, for every $\varepsilon > 0$, there exist $\theta,\gamma > 0$ such that $Avg^{q_0}_{\sigma'} > Avg^{q_0}_{\sigma} - \varepsilon$ holds.*

Proof. First, note that $\Gamma = (S, D, H_\ell)$ such that $S = D = \emptyset$ as $\mathcal{W}^\star_\Phi = Q$ and Φ is a Büchi objective [4]. By Claim 21, we know that each counter value is bounded by $K_{\theta,\gamma}$. Now, as in the proof of Thm. 11, w.l.o.g, let us assume that σ is a stationary policy in M and hence, one can show that both σ' and σ are stationary policies w.r.t. the extended MDP $M' = \langle Q', A, \Delta', q'_0\rangle$ obtained by taking product of the MDP with the tuples of bounded counter values. Furthermore, as Q is an SCC, using an extension of the well-known policy difference lemma [26] for average rewards (see [44, 58] for more details), the difference in average rewards of σ and σ' can be expressed as[5]:

$$\Delta\mathrm{Avg} = \mathrm{Avg}_{\sigma}^{q_0} - \mathrm{Avg}_{\sigma'}^{q_0} = \mathbb{E}_{\substack{(q,C)\sim d_{\sigma'} \\ a\sim\sigma'}}\Big[V^{\sigma}((q,C)) - B^{\sigma}((q,C),a)\Big], \tag{4}$$

where $d_{\sigma'}$ is the stationary distribution of σ' in M'; B^σ and V^σ are the action-bias and state-bias functions of σ defined as:

$$B^{\sigma}((q,C),a) = B^{\sigma}(q,a) = \mathbb{E}_{\rho\sim\sigma}\Big[\sum_{i=0}^{\infty} r(\rho[i]) - \mathrm{Avg}_{\sigma}^{q_0} \;\Big|\; \rho[0] = (q,a)\Big],$$

$$V^{\sigma}((q,C)) = V^{\sigma}(q) = \mathbb{E}_{\rho\sim\sigma}\Big[\sum_{i=0}^{\infty} r(\rho[i]) - \mathrm{Avg}_{\sigma}^{q_0} \;\Big|\; \rho[0] \in q\times A(q)\Big].$$

Then, (4) can be rewritten as:

$$\begin{aligned}
\Delta\mathrm{Avg} &= \mathbb{E}_{(q,C)\sim d_{\sigma'}} \sum_{a\in A(q)} \sigma'((q,C),a)\cdot\Big[V^{\sigma}((q,C)) - B^{\sigma}((q,C),a)\Big] \\
&= \mathbb{E}_{(q,C)\sim d_{\sigma'}} \sum_{a\in A(q)} \sigma'((q,C),a)\cdot\Big[\sum_{a'\in A(q)} \sigma(q,a')B^{\sigma}(q,a') - B^{\sigma}(q,a)\Big] \\
&= \mathbb{E}_{(q,C)\sim d_{\sigma'}} \sum_{a'\in A(q)} \sigma(q,a')B^{\sigma}(q,a') - \sum_{a\in A(q)} \sigma'((q,C),a)\cdot B^{\sigma}(q,a) \\
&= \mathbb{E}_{(q,C)\sim d_{\sigma'}} \sum_{a\in A(q)} (\sigma(q,a) - \sigma'((q,C),a))\cdot B^{\sigma}(q,a).
\end{aligned}$$

As it is knows that $|B^\sigma|$ is bounded by some B for such policy σ [44], we have the following:

$$\Delta\mathrm{Avg} \leq B\cdot\mathbb{E}_{(q,C)\sim d_{\sigma'}} \sum_{a\in A(q)} |\sigma(q,a) - \sigma'((q,C),a)|\,.$$

As the above expression is similar to the expected cost in (3), using similar arguments as in the proof of Thm. 11, using appropriate values for θ and γ, we can ensure that the above expression is less than ε. ◀

▶ **Theorem 19.** *Given the premises of Thm. 15 with $\sigma' = \sigma|_{\gamma}^{\Gamma_\bullet,\theta}$ and T being the set of even color states in the objective Φ, for every frequency $0 < \delta \leq 1$, there exists parameters $\theta, \gamma > 0$ such that for every run $\rho \sim \sigma'$, it holds that $\mathtt{freq}(\rho, T) \geq \delta$.*

Proof. Since Φ is a Büchi objective, by the construction of ParityTemplate$_\bullet$, as in [4, Alg. 3], $\Gamma_\bullet = (\emptyset, \emptyset, H_\ell = \{H_i \mid i \in [1;n]\})$ such that $\mathcal{W}_\Phi^\bullet = Q$ can be partitioned into n groups $\{Q_i\}_{i=0}^n$ with $Q_0 = T$ and H_i being the edges from Q_i to Q_{i-1}. Intuitively, the indices of the groups measure the closeness to the even color states T and the live-groups are the edges that are used to ensure that a run progresses from higher index groups to lower index groups leading to the final group T. Furthermore, by Claim 21, we know that each counter value of the live-groups is bounded by $K_{\theta,\gamma}$. As visiting Q_i ensures visiting the source states of the live-group H_i, it must hold that within $K_{\theta,\gamma}$ many visits to the group Q_i, the counter value of the live-group H_i is reset to zero, i.e., an edge from H_i is sampled at least once. Hence, applying the above argument recursively, we can show that within $K_{\theta,\gamma}^{n+1}$ many visits to the group Q_n will ensure visiting the group Q_n at least $K_{\theta,\gamma}^n$ times, which in turn ensures that

[5] We use $x \sim \mu$ to denote that x is sampled from distribution μ.

the group Q_{n-1} is visited at least $K_{\theta,\gamma}^{n-1}$ times and so on until the group $Q_0 = T$ is visited at least once. Finally, as every cycle of length $|Q|$ visits some group at least twice, for every run ρ sampled from σ', it holds that

$$\texttt{freq}(\rho, T) \geq \frac{1}{K_{\theta,\gamma}^{n-1}} = \min\left(1, \left(\frac{\gamma}{1/\theta - 1}\right)^{n-1}\right).$$

Hence, with appropriate values for θ and γ, we can ensure that the above expression is at least δ. ◀

B Strategy Templates in $\star$-good SCCs

▶ **Lemma 22.** *Let M be an MDP and $\Phi = \textsc{Parity}[c]$ be a parity objective such that $G_\star^M = (Q, E)$ is the corresponding $\star$-game graph. If Q is a $\star$-good SCC, then the strategy template $\Gamma_\star = \textsc{ParityTemplate}_\star(G_\star^M, \Phi)$ is such that $\Gamma_\star = (S, D, H_\ell)$ with $S = D = \emptyset$.*

Proof. First, let us consider the case when $\star = \bullet$. Let $\Gamma_\bullet = (S, D, H_\ell)$ be the strategy template obtained by the procedure $\textsc{ParityTemplate}_\bullet(G_\bullet^M, \Phi)$. Note that as Q is a $\bullet$-good SCC, by results in [1], $\mathcal{W}_\Phi^\bullet = Q$. As the procedure $\textsc{ParityTemplate}_\bullet$ marks all edges going out of $\mathcal{W}_\Phi^\bullet$ as unsafe, it follows that $S = \emptyset$. We only need to show that $D = \emptyset$.

Let $\Phi = \textsc{Parity}[c]$ be a parity objective with coloring function $c : Q \to [0; d]$. Let $P_i = \{q \in Q \mid c(q) = i\}$ be the set of states with color i. Let us first show that the maximal color d can only be even. Suppose d is odd, then there exists a state q with $c(q) = d$. As Q is a $\bullet$-good SCC, it holds that $q \in \mathcal{W}_\varphi^\bullet$, where

$$\varphi = \{\rho \in \mathrm{Runs}^M \mid \exists n \geq 0 : c(\rho[n]) > d \text{ and is even}\}.$$

However, as d is the maximum color, there are no states with $c(q) > d$, and hence, $\varphi = \emptyset$. So, $\mathcal{W}_\varphi^\bullet = \emptyset$, which is a contradiction to the assumption that $q \in \mathcal{W}_\varphi^\bullet$.

Now, let us show that $D = \emptyset$ using induction on d, i.e., the maximal color in c. For the base case, if $d = 0$, then the statement holds trivially.

Suppose the statement holds for $d = 2k$ for some $k \geq 0$. Now, let us consider the case when $d = 2k + 2$. Consider the procedure $\textsc{ParityTemplate}$ given in [4, Algorithm 3] that computes the strategy template $\Gamma_\bullet$. As d is even, the procedure $\textsc{ParityTemplate}$ starts by computing $A = \texttt{attr}^0(P_d)$ (in line 14) which is the set of states from which system player can force the play to visit a state with color d. If $A = Q$, then the procedure terminates (in line 15) and returns the output of a procedure $\textsc{ReachTemplate}$ (provided in [4, Algorithm 1]), which only returns a live group template. Hence, $D = \emptyset$. Suppose $A \neq Q$, then the procedure $\textsc{ParityTemplate}$ is called recursively on the subgame G' obtained by removing the states in A (in line 17). As $A = \texttt{attr}^0(P_d) \supseteq P_d$, the maximal color in G' is at most $d-1$. If there is a state q in G' with color $d-1$ (which is odd), then as Q is a $\bullet$-good SCC, it holds that $q \in \mathcal{W}_{\varphi'}^\bullet$, where

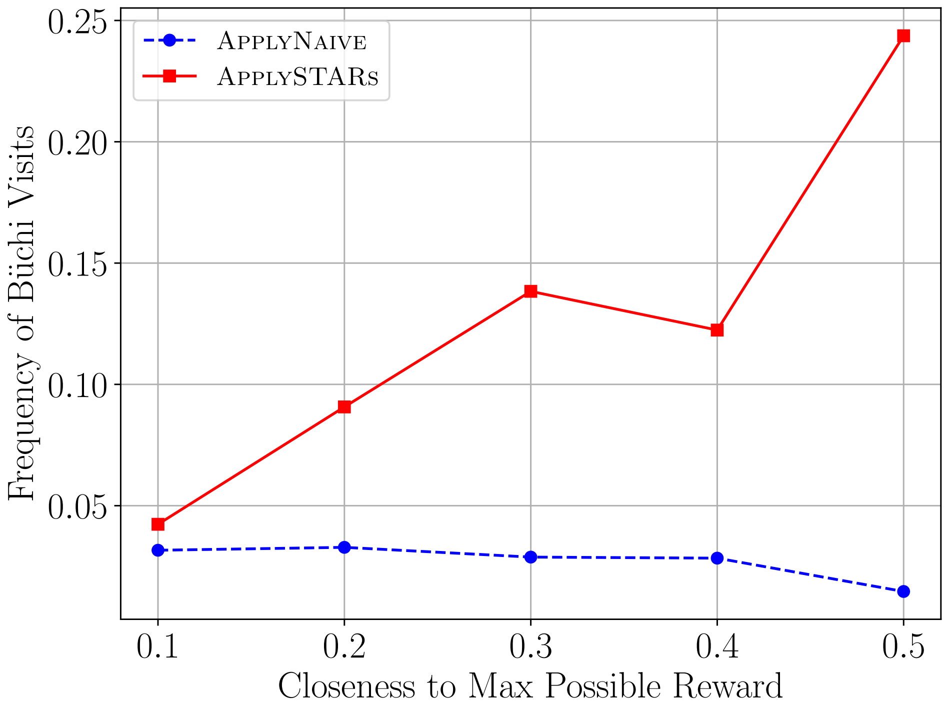
$$\varphi' = \{\rho \in \mathrm{Runs}^M \mid \exists n \geq 0 : c(\rho[n]) > d - 1 \text{ and is even}\}.$$

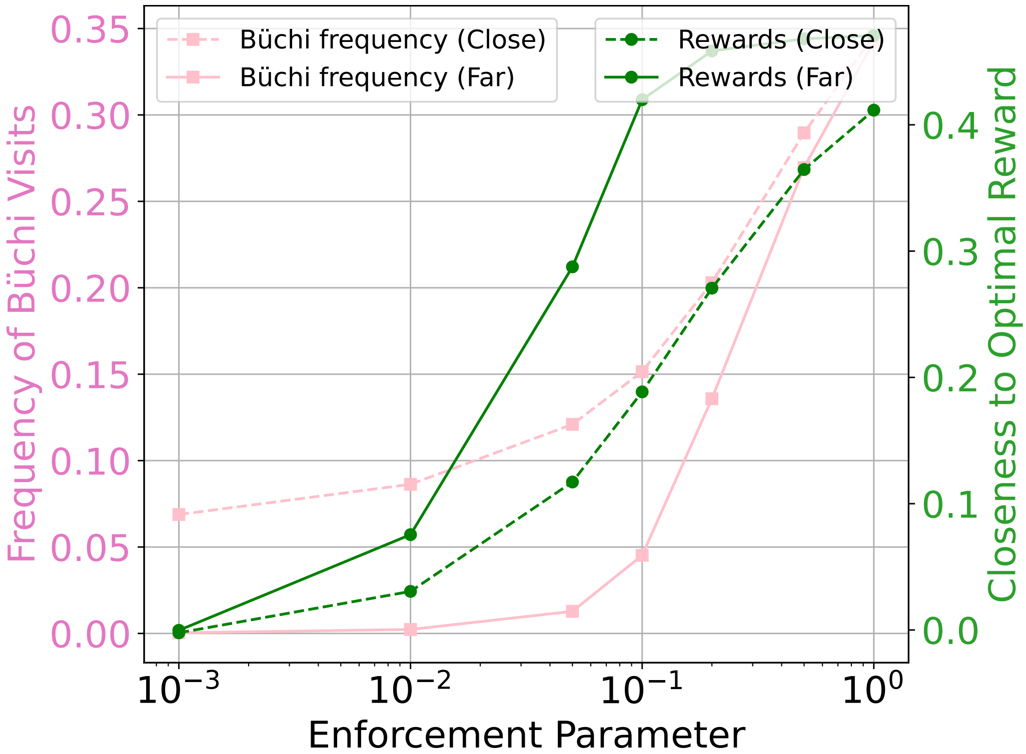
Hence, system player can force the run from q to visit a state with color d, and hence, $q \in \texttt{attr}^0(P_d) = A$. This contradicts the assumption that q is a node in G'. Therefore, there is no state in A with color $d-1$, and hence, the maximal color in G' is at most $d-2$. By induction hypothesis, the procedure $\textsc{ParityTemplate}$ returns a template with $D = \emptyset$.

For the case when $\star = \circ$, the procedure $\textsc{ParityTemplate}$ uses Algorithm 8 provided in [43]. This procedure first converts the $1\frac{1}{2}$-player game graph $G_\circ^M$ to a 2-player game graph

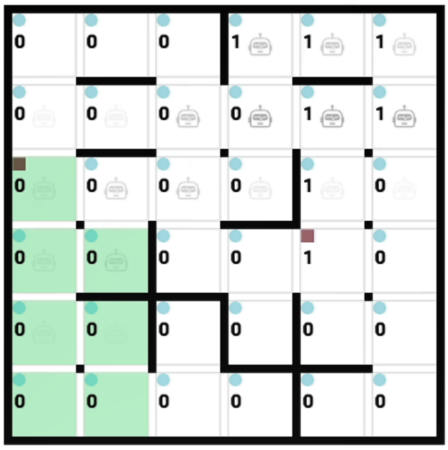
G' and then uses the procedure ParityTemplate provided in [4, Algorithm 3] to compute the strategy template. Using similar arguments as above, we can show that the procedure ParityTemplate provided in [4, Algorithm 3] for the 2-player game graph G' returns a template with $D = \emptyset$. ◀

C Bigger Figures



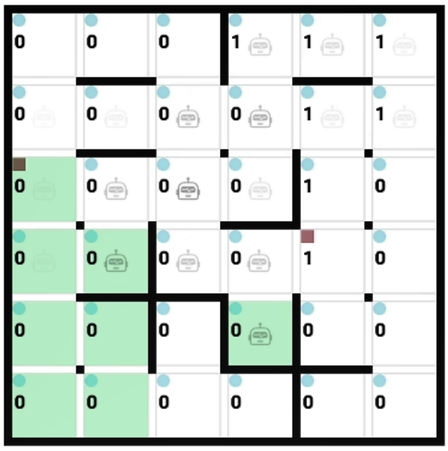

Büchi frequency (Close)
Büchi frequency (Far)
Rewards (Close)
Rewards (Far)
Frequency of Büchi Visits
Closeness to Optimal Reward
Enforcement Parameter
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0.4
0.3
0.2
0.1
0.0
10^{-3}
10^{-2}
10^{-1}
10^{0}

(a) Unshielded



(b) Low γ

(c) High γ



(d) Online objective addition

Figure 5 Screenshots of UI showing `MARG` controlled robot for an instance from FactoryBot.