Correct-by-Design Control Synthesis of Stochastic Multi-agent Systems: a Robust Tensor-based Solution ⋆

Ruohan Wang * **Siyuan Liu** * **Zhiyong Sun** ** **Sofie Haesaert** *

* *Department of Electrical Engineering, Control Systems Group, Eindhoven University of Technology, The Netherlands. Emails: {r.wang2, s.liu5@tue.nl, s.haesaert}@tue.nl.*
** *Department of Mechanics and Engineering Science & State Key Laboratory for Turbulence and Complex Systems, Peking University, Beijing, China. Email: zhiyong.sun@pku.edu.cn.*

Abstract: Discrete-time stochastic systems with continuous spaces are hard to verify and control, even with MDP abstractions, due to the curse of dimensionality. We propose an abstraction-based framework with robust dynamic programming mappings that deliver control strategies with provable lower bounds on temporal-logic satisfaction, quantified via approximate stochastic simulation relations. Exploiting decoupled dynamics, we reveal a Canonical Polyadic Decomposition tensor structure in value functions that makes dynamic programming scalable. The proposed method provides correct-by-design probabilistic guarantees for temporal logic specifications. We validate our results on continuous-state linear stochastic systems.

Keywords: Stochastic systems, temporal logic, dynamic programming

1. INTRODUCTION

Advances in computational power have enabled the development of large-scale systems in safety-critical domains like smart grids and traffic management Venayagamoorthy (2011). These systems involve numerous agents with complex or uncertain dynamics Venayagamoorthy (2011); Julian et al. (2019). While connectivity improves coordination, it also increases complexity and risk. Temporal-logic specifications provide a formal way to express safety objectives for stochastic multi-agent systems modeled as general Markov decision processes (gMDP), in which stochasticity is governed by a Markov kernel and observed through output mappings Wolff et al. (2012); Schön et al. (2023); Cai and Vasile (2025). In this framework, safety is encoded as a reachability problem on the product of the system and an automaton, which enables quantitative guarantees Baier and Katoen (2008); Belta et al. (2017). To formalize and certify such guarantees, the continuous nature of the state (and possibly action) spaces motivates finite abstractions, with stochastic simulation relations providing certified bounds that link abstract and original models Tabuada (2009); Abate et al. (2010); Desharnais et al. (2004).

Correct-by-design control synthesis for stochastic systems with temporal-logic specifications faces the curse of dimensionality, as computational cost grows exponentially with system size, rendering synthesis intractable Liu et al. (2021). Prior efforts mitigate this mainly via scalable abstractions rather than accelerating the synthesis step itself Mallik et al. (2018); Haesaert and Soudjani (2020); Liu et al. (2021); Schön et al. (2023). In single-agent settings, however, abstraction-based approaches can surpass abstraction-free methods in quantitative accuracy while retaining certified guarantees Zamani et al. (2015). Yet, due to scalability bottlenecks in compositional analysis, such abstraction-driven techniques have not been realized at scale for multi-agent systems.

Given a finite abstraction, the synthesis problem for a multi-agent system reduces to solving dynamic programming on a very large gMDP, with the computational bottleneck shifted from model construction to planning. This shift motivates structure-exploiting accelerations of dynamic programming: by leveraging low-rank tensor approximations in dynamic programming problems Gorodetsky et al. (2018); Rozada and Marques (2024), the optimal solution can be approximated while reducing the computational complexity and mitigating scalability issues. For finite Markov Decision Processes (MDP), recent work Wang et al. (2025) shows that a tree structure can be used to manage and prune the tensor rank in a provable fashion. This work advances on Alora et al. (2016) in which the low rank tensor approximations do not yield lower bounds on satisfaction probabilities. In this paper, we develop tensor-based methods that synthesize controllers on finite-state abstractions of multi-agent systems. We certify robustness by quantitatively relating the abstraction to its continuous-state system, so the synthesized controllers come with provable guarantees.

Organization. This paper is organized as follows. The next section outlines the framework. Sec. 3 constructs finite abstractions of multi-agent systems and quantifies them

⋆ This work is supported by the European project SymAware under the grant number 101070802 and COVER under the grant number 101086228.

via stochastic similarity relations. Sec. 4 recalls the tree-structured tensor-based value iteration and shows how to correct the resulting satisfaction probabilities using a-posteriori error bounds. Sec. 5 then introduces a robust variant of this tensor-based synthesis. Sec. 6 presents numerical case studies, and Sec. 7 concludes.

Contributions. To keep this paper self-contained, we briefly recap necessary results from our prior work Wang et al. (2025) in Sec. 4 that underpin the present work. Our main contribution is extending those results from discrete to continuous-state systems at scale. Specifically: (1) we extend formal abstractions and stochastic simulation relations to multi-agent systems; (2) we show how tensor computations can be corrected a-posteriori; (3) we show a robust version of the tensor computations with tailored policy optimization; (4) we validate the proposed methods on multi-agent systems.

2. FRAMEWORK AND APPROACH

Denote a Borel measurable space as $(\mathbb{S}, \mathcal{B}(\mathbb{S}))$ where $\mathbb{S}$ is a set and $\mathcal{B}(\mathbb{S})$ is the Borel sigma-algebra on $\mathbb{S}$. A probability measure $\mathbb{P}$ over $(\mathbb{S}, \mathcal{B}(\mathbb{S}))$ defines the probability space $(\mathbb{S}, \mathcal{B}(\mathbb{S}), \mathbb{P})$ and has realization $x \sim \mathbb{P}$. We assume all such spaces $\mathbb{S}$ are Polish Bogachev (2007). Let $d_{\mathbb{S}} : \mathbb{S} \times \mathbb{S} \to [0, \infty)$ be a *metric* [1] on $\mathbb{S}$. For vectors $x_i \in \mathbb{R}^{n_i}$ we define $\mathrm{col}(x_1, \ldots, x_N) := [x_1^\top \ldots x_N^\top]^\top \in \mathbb{R}^{\sum_{i=1}^N n_i}$.

2.1 System models: Stochastic difference equations

In this paper, we consider a multi-agent system $\mathbf{M}$ composed of N agents. Each agent $\mathbf{M}^i$, $i \in \{1, \ldots, N\}$, is modeled with a stochastic difference equation:

$$\begin{cases} x_i^+ &= f_i(x_i, u_i) + w_i \\ y_i &= h_i(x_i) \end{cases} \quad \forall i \in \{1, \ldots N\} \tag{1}$$

with state $x_i \in \mathbb{X}_i \subset \mathbb{R}^{n_i}$, the control input $u_i \in \mathbb{U}^i \subset \mathbb{R}^{m_i}$, output $y_i \in \mathbb{Y}_i \subset \mathbb{R}^{h_i}$, and the state disturbance $w_i \in \mathbb{R}^{n_i}$ an independent, identically distributed noise sequence with distribution $w_i \sim \mathbb{P}_{w^i}(\cdot)$, i.e., $w_i[t_1]$ and $w_i[t_2]$ are independent for all $t_1 \neq t_2$. Each agent $\mathbf{M}^i$ (1) is initialized with state $x_i[0]$.

For the combined multi-agent system $\mathbf{M}$, the collective state is given as $x = \mathrm{col}(x_1, \ldots, x_N) \in \mathbb{R}^n, n = \sum_{i=1}^N n_i$, together with the collective stochastic dynamics

$$\begin{cases} x^+ &= f(x, u) + w \\ y &= h(x) \end{cases} \tag{2}$$

where $u = \mathrm{col}(u_1 \ldots u_N) \in \mathbb{R}^m$ with $m = \sum_{i=1}^N m_i$, and $y = \mathrm{col}(y_1 \ldots y_N) \in \mathbb{R}^h$ with $h = \sum_{i=1}^N h_i$. The state disturbance $w \in \mathbb{R}^n$ is distributed as $w \sim \mathbb{P}_w(\cdot)$ with $\mathbb{P}_w := \prod_{i=1}^n \mathbb{P}_{w_i}$ the joint distribution. The random variables w_i and w_j are independent for all $i \neq j$, that is, the stochastic disturbances of the multi-agent system are independent. The multi-agent system (2) is initialized as $x[0] := \mathrm{col}(x_1[0] \ldots x_N[0])$ by construction.

The *execution (state trajectory)* of the system (2) is a sequence of states $\boldsymbol{x}_{[0,T]} := \{x[t] | t = 0, \ldots, T\}$ initialized with $x[0]$. Consecutive states $x[t+1]$ are obtained from realizations $x[t+1] \sim \mathbb{T}(\cdot | x[t], u[t])$ of the controlled Borel-measurable stochastic kernel, which is defined for system (2) as $\mathbb{T}(dx[t+1] | x[t], u[t]) := \mathbb{P}_w(dx[t+1] - f(x[t], u[t]))$. Meanwhile, the *output trajectory* of the system (2) is a sequence of outputs $\boldsymbol{y}_{[0,T]} := \{y[t] | t = 0, \ldots, T\}$, where $y[t]$ is obtained as $y[t] = h(x[t])$.

We denote with $\mathbf{C}_{\boldsymbol{\pi}}$ a general control strategy that maps each finite history $x[0], u[0], x[1], u[1], \ldots, x[t]$ to an action $u[t]$. We denote a system $\mathbf{M}$ controlled by $\mathbf{C}_\pi$ as $\mathbf{M} \times \mathbf{C}_\pi$. A more specific type of control strategy is a Markov policy.

Definition 1 (Markov policy $\boldsymbol{\pi}$)**.** *A Markov policy $\boldsymbol{\pi}$ is a sequence $\boldsymbol{\pi} := \{\pi[t] | t = 0, 1, \ldots\}$ of maps $\pi[t] : \mathbb{X} \to \mathbb{U}$ that assign an action $u \in \mathbb{U}$ to each state $x \in \mathbb{X}$.*

2.2 Specification

To each agent $\mathbf{M}^i$, we associate a set of unique atomic propositions $\mathrm{AP}_i := \{p_1, \ldots, p_{|\mathrm{AP}_i|}\}$ for which $\mathrm{AP}_i \cap \mathrm{AP}_j = \emptyset$, if $i \neq j, \forall i, j \in \{1, \ldots, N\}$. A set of atomic propositions AP_i defines the alphabet $\Sigma^i := 2^{\mathrm{AP}_i}$. For each output y_i, the atomic propositions can either be true or false. The set of true propositions for a given output is defined by the labeling map $L^i : \mathbb{Y}_i \to 2^{\mathrm{AP}_i}$.

The output trajectory $\boldsymbol{y} = y[0], y[1], \ldots$ of system (2) is mapped to the word $\boldsymbol{l} := l_0, l_1, \ldots$ using the labelling map $L : \mathbb{Y} \to 2^{\mathrm{AP}}$ defined as $L\big(\mathrm{col}(y_1 \ldots y_N)\big) := \bigcup_{i=1}^N L^i(y_i) \in 2^{\mathrm{AP}}$, that translates each output to a specific letter $l_t = L(y[t])$. Similarly, output suffixes $\boldsymbol{y}_t := y[t], y[t+1], \ldots$ are translated to word suffixes $\boldsymbol{l}_t := l_t, l_{t+1}, \ldots$.

In this paper, we are interested in temporal specifications ϕ, e.g., syntactically co-safe linear temporal logic (*scLTL*) Schön et al. (2024) and linear temporal logic on finite traces (*LTL$_f$*) Jagtap et al. (2020), that can be automatically translated to a deterministic finite automaton (DFA) via off-the-shelf tools, e.g., SPOT Duret-Lutz and Poitrenaud (2004) and SCHECK2 Latvala (2003). In the sequel, we drop the subscript ϕ for the sake of simple presentation. We give the definition of Deterministic Finite Automaton (DFA) as follows:

Definition 2 (DFA)**.** *A deterministic finite automata (DFA) is defined by the tuple $\mathcal{A} = (Q, q_0, \Sigma_\mathcal{A}, \tau_\mathcal{A}, q_f)$. Here, Q, q_0, and q_f denote the set of states, initial state, and accepting state, respectively. Furthermore, $\Sigma_\mathcal{A} = 2^{\mathrm{AP}}$ denotes the input alphabet and $\tau_\mathcal{A} : Q \times \Sigma_\mathcal{A} \to Q$ is a transition function.*

We denote a DFA translated from a specification ϕ as $\mathcal{A}_\phi$.

Boolean-formula-labelled transitions of DFA. Given a DFA $\mathcal{A} = (Q, q_0, \Sigma_\mathcal{A}, \tau_\mathcal{A}, q_f)$, let us define a set $\boldsymbol{\alpha} := \{\alpha\}$ composed of Boolean formulae α, that allows the rewriting of DFA transition as $\tau_\mathcal{A} : Q \times \boldsymbol{\alpha} \to Q$. Each α is a conjunction of (possibly negated) atomic propositions: $\alpha ::= p | \neg p | \alpha_1 \wedge \alpha_2$. Furthermore, for every letter $l \in \Sigma_\mathcal{A}$ there exists exactly one such α with $l \models \alpha$. See Figure 2 for an intuitive grasp on α. To agent i, we define α_i as $\alpha_i := \{\bigwedge_{l \models \alpha} l^i\}$, with $l^i := L^i(y_i)$.

For a given word $\boldsymbol{l} = l_0, l_1, \ldots$, a run of a DFA $\mathcal{A}$ defines a trajectory $q_0, q_1, \ldots$ that starts with q_0 and evolves according to $q_{t+1} = \tau_\mathcal{A}(q_t, \alpha_t)$. The word $\boldsymbol{l}_{[0,T]}$ is accepted by $\mathcal{A}_\phi$ if $\exists t \in [0, T]$ such that $q_t = q_f$ for the corresponding trajectory $q_0, q_1, \ldots, q_T$. This is denoted as $\boldsymbol{l}_{[0,T]} \models \phi$. For

[1] We recall the axioms of a *metric*: $d_{\mathbb{S}}(x, y) \geq 0$, $d_{\mathbb{S}}(x, y) = 0 \Leftrightarrow x = y$, $d_{\mathbb{S}}(x, y) = d_{\mathbb{S}}(y, x)$, and $d_{\mathbb{S}}(x, z) \leq d_{\mathbb{S}}(x, y) + d_{\mathbb{S}}(y, z)$ for all $x, y, z \in \mathbb{S}$.

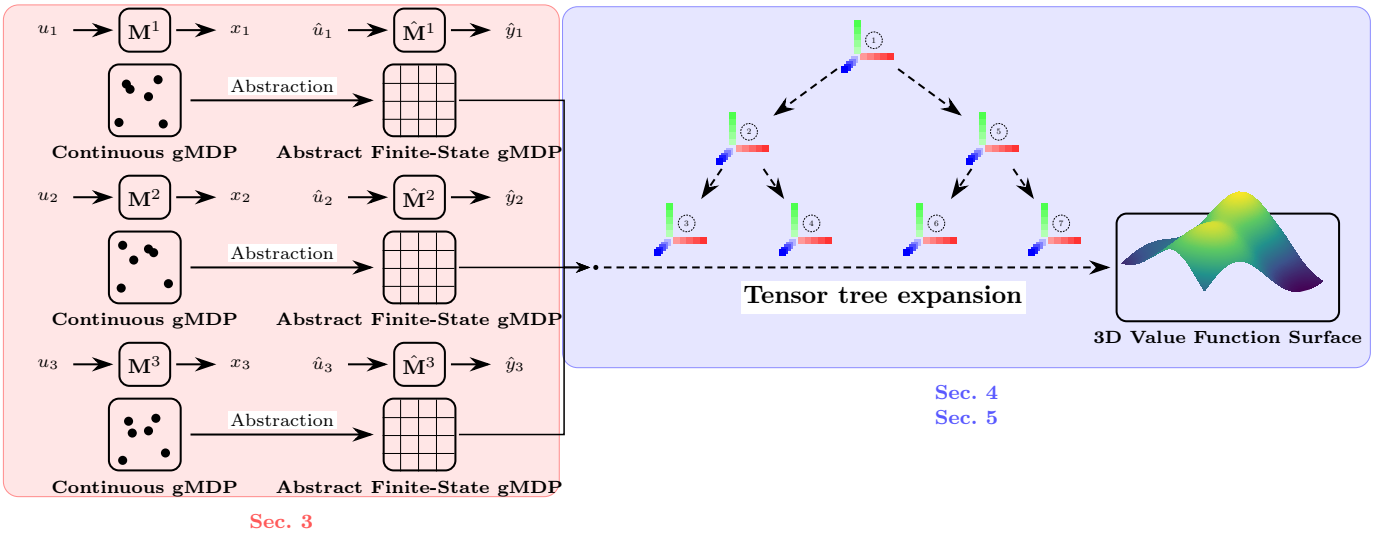


Fig. 1. Control-synthesis (three-agent example). Each agent $\mathbf{M}^i$ (1) is abstracted as a gMDP $\hat{\mathbf{M}}^i$ (red-shaded block; Sec. 3). The per-agent value components are then merged in every tree node and propagated through the tensor tree (blue-shaded block; Sec. 4 and Sec. 5), yielding a high-dimensional satisfaction probability.

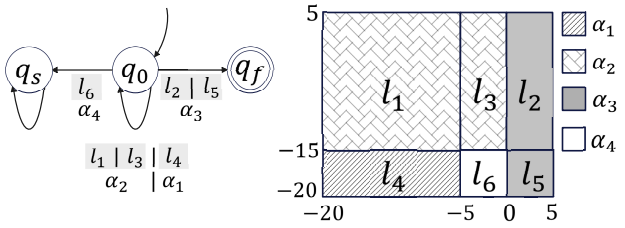


Fig. 2. Left: Letter- or Boolean-formula-triggered transitions of DFA (q_s denotes q_{sink} for simplicity), right: output space labeled by l or α.

$T \to \infty$, we denote $\boldsymbol{l} \models \phi$. Given a specification ϕ and its DFA $\mathcal{A} = (Q, q_0, \Sigma_{\mathcal{A}}, \tau_{\mathcal{A}}, q_f)$, we label the output trajectory $\boldsymbol{y}$ of a controlled system $\mathbf{M} \times \mathbf{C}_\pi$ and obtain a word $\boldsymbol{l}$. The controlled system satisfying the specification can be translated to an acceptance problem, i.e., the probability that the word $\boldsymbol{l}$ is accepted.

2.3 Problem statement & Approach

Correct-by-design methods for infinite-state (particularly continuous-state) stochastic systems typically rely on finite-state abstractions Tabuada (2009) and on approximate simulation relations Haesaert and Soudjani (2020); Schön et al. (2024) to carry guarantees across models. Our prior work Wang et al. (2025) mitigated the curse of dimensionality in finite-state control synthesis while preserving provable guarantees. This paper extends previous results to continuous-state multi-agent systems. In this paper, we focus on continuous-state systems and develop an approach that preserves correct-by-design guarantees while remaining computationally and memory efficient as state spaces grow.

Problem statement: *Given a specification ϕ with DFA $\mathcal{A}_\phi$ and a probability $p \in [0,1]$, synthesize control strategies $\mathbf{C}_{\boldsymbol{\pi}}$ for a multi-agent system $\mathbf{M}$ such that the controlled system satisfies ϕ with a lower bound, i.e.,*

$$\mathbb{P}_{\mathbf{M}\times\mathbf{C}_{\boldsymbol{\pi}}}(\boldsymbol{l} \models \phi) \geq p. \tag{3}$$

Fig. 1 presents a schematic overview of our approach, exemplified on a three-agent system. Red-shaded: the continuous-state three-agent system is abstracted by three finite gMDPs with per-agent similarity bounds. Blue-shaded: per-agent components are composed at nodes over a tensor tree, and then propagated along tree edges to yield the overall (high-dimensional) satisfaction probability, depicted as the 3-D surface on the right.

3. ROBUST DYNAMIC PROGRAMMING FOR MULTI-AGENT SYSTEMS

A stochastic difference equation can also be modelled as a general Markov decision process, which is defined as follows.

Definition 3 (general Markov decision processes (gMDP)). *A gMDP is denoted as $\mathbf{M} := (\mathbb{X}, \mathbb{U}, \mathbb{Y}, \mathbb{T}, x_0, h)$ with*

- *a Polish state space $\mathbb{X}$ with states $x \in \mathbb{X}$;*
- *a Polish input space $\mathbb{U}$ with inputs/actions $u \in \mathbb{U}$;*
- *an output space $\mathbb{Y}$ decorated with metric $\mathbf{d}_{\mathbb{Y}}$;*
- *a stochastic kernel $\mathbb{T} : \mathbb{X} \times \mathbb{U} \times \mathcal{B}(\mathbb{X}) \to [0,1]$, that assigns to each state-action pair $x \in \mathbb{X}$ and $u \in \mathbb{U}$ a probability measure $\mathbb{T}(\cdot \mid x, u)$ over $(\mathbb{X}, \mathcal{B}(\mathbb{X}))$;*
- *an initial state $x_0 \in \mathbb{X}$;*
- *a measurable output map $h : \mathbb{X} \to \mathbb{Y}$.*

We say that a gMDP is finite if $\mathbb{X}$, $\mathbb{Y}$, and $\mathbb{U}$ are finite. In this paper, each agent i (1) can be written as a gMDP

$$\mathbf{M}^i = (\mathbb{X}_i, \mathbb{U}_i, \mathbb{Y}_i, \mathbb{T}_i, x_{i,0}, h_i), \quad i = 1, \ldots, N, \tag{4}$$

where $\mathbb{T}_i(x_{i,t+1} | x_{i,t}, u_{i,t}) = \mathbb{P}_{w_i}(x_{i,t+1} - f(x_{i,t}, u_{i,t}))$. Similarly, the associated multi-agent gMDP can be modeled (2) via Cartesian composition:

$$\mathbf{M} = (\mathbb{X}, \mathbb{U}, \mathbb{Y}, \mathbb{T}, x_0, h) \tag{5}$$

with $\mathbb{X} = \prod_{i=1}^N \mathbb{X}_i$, $\mathbb{U} = \prod_{i=1}^N \mathbb{U}_i$, $\mathbb{Y} = \prod_{i=1}^N \mathbb{Y}_i$, $\mathbb{T} = \prod_{i=1}^N \mathbb{T}^i$, $x_0 := \text{col}(x_{1,0}, ..., x_{N,0})$, and $h = \text{col}(h_1, ..., h_N)$.

3.1 Dynamic programming

For a multi-agent system $\mathbf{M}$ and a specification ϕ with its DFA $\mathcal{A}$, we define value functions $\mathcal{V}^{\boldsymbol{\pi}}_{q,T} : \mathbb{X} \to [0,1]$

as the probability that the generated word $\boldsymbol{l}_{[0,T]}$ gets accepted Wang et al. (2025); Bertsekas and Shreve (1996). Given a Markov policy $\boldsymbol{\pi} := \{\pi[t] | t = 0, 1, \ldots, T-1\}$, value functions can iteratively be computed starting from $\mathcal{V}_{q,0} = 1$ if $q = q_f$, and 0, otherwise and iterated as

$$\mathcal{V}^{\boldsymbol{\pi}}_{q,t+1}(x) = \mathbf{1}_{q_f}(q) + \mathbf{1}_{Q\setminus\{q_f\}}(q)\mathbb{E}^{\pi[T-1-t]}_{x,q}[\mathcal{V}^{\boldsymbol{\pi}}_{q',t}(x')] \quad (6)$$

where $\mathbb{E}^{\pi}_{x,q}[f] := \mathbb{E}[f(x', q') | (x, q), u = \pi(x, q)]$ with $q' := \tau(q, \alpha)$ with $\alpha = L(h(x'))$. $\mathbf{1}_A(a)$ is an indicator function, defined as $\mathbf{1}_A(a) := 1$ if $a \in A$, and 0, otherwise. As in Wang et al. (2025); Haesaert and Soudjani (2020), the subsequent proposition follows.

Proposition 1 (Specification satisfaction.)**.** *For a given policy $\boldsymbol{\pi} = \{\pi[t] | t = 0, 1, \ldots, T-1\}$, the satisfaction probability within time horizon T for specification ϕ of the multi-agent system $\mathbf{M}$ is defined as*

$$\mathbb{P}_{\mathbf{M}\times\boldsymbol{\pi}}(\boldsymbol{l}_{[0,T]} \models \phi) := \mathcal{V}^{\boldsymbol{\pi}}_{\bar{q}_0,T}(x_0) \quad (7)$$

with $\bar{q}_0 = \tau_{\mathcal{A}}(q_0, L(h(x_0)))$, and computed as in (6).

A general form of DFA-informed operators. Value functions $\mathcal{V}^{\boldsymbol{\pi}}_{q,T}(x)$ can be recursively computed as

$$\mathcal{V}^{\boldsymbol{\pi}}_{q,t+1}(x) = \sum_{(\alpha,q')\in \mathrm{Succ}(q)} \mathbf{T}^{\pi_q[T-1-t]}_{\alpha}(\mathcal{V}^{\boldsymbol{\pi}}_{q',t})(x) \quad (8)$$

for all $q \in Q \setminus \{q_f\}$, where $t = 0, \ldots, T-1$, $\mathrm{Succ}(q)$ is defined for DFA as $\mathrm{Succ}(q) := \{(\alpha, q') | q' = \tau_{\mathcal{A}}(q, \alpha)\}$, and $\mathbf{T}^{\pi_q}_{\alpha}$ is the DFA-informed operator originally proposed in Wang et al. (2025), defined as

$$\mathbf{T}^{\pi}_{\alpha}(\mathcal{V})(x) := \mathbb{E}^{\pi}_{x,q}[\mathcal{L}_{\alpha}(h(x'))\mathcal{V}(x')], \quad (9)$$

with the indicator functions $\mathcal{L}_{\alpha} : \mathbb{Y} \to \{0, 1\}$ defined for each formula α as

$$\mathcal{L}_{\alpha}(y') = \begin{cases} 1, & \text{if } L(y') \models \alpha \\ 0, & \text{otherwise.} \end{cases} \quad (10)$$

Dynamic programming is generally intractable on gMDPs with infinite state spaces, particularly uncountable ones (e.g., Euclidean state spaces), yet feasible on finite gMDPs. In this paper, we approximate the continuous-state multi-agent system using a finite-state model and perform value iteration on it. To soundly refine the synthesized controller back to the continuous-state system, we quantify the abstraction error via approximate simulation relations, as introduced in the next section.

3.2 Approximate simulation relations

Consider a **concrete** system modeled as a gMDP

$$\mathbf{M} = (\mathbb{X}, \mathbb{U}, \mathbb{Y}, \mathbb{T}, x_0, h)$$

and its **approximation**

$$\hat{\mathbf{M}} = (\hat{\mathbb{X}}, \hat{\mathbb{U}}, \hat{\mathbb{Y}}, \hat{\mathbb{T}}, \hat{x}_0, \hat{h}).$$

To quantify the error caused by $\hat{\mathbf{M}}$, we first quantify the similarity between probability distributions $\mathbb{T}$ and $\hat{\mathbb{T}}$. Let $\mathcal{R}$ be a measurable relation $\mathcal{R} \subset \hat{\mathbb{X}} \times \mathbb{X}$ that relates states of the concrete system with those of the approximation. To use this relation, we need to define its lifted version, which is denoted as $\bar{\mathcal{R}}_{\delta}$ where δ quantifies the probability mass that falls outside of the lifting. More precisely we say that (Δ, Θ) belongs to the δ-lifted relation $\bar{\mathcal{R}}_{\delta}$ if there exists a probability measure $\mathbb{W}$, referred to as a lifting, with probability space $(\hat{\mathbb{X}} \times \mathbb{X}, \mathcal{B}(\hat{\mathbb{X}} \times \mathbb{X}), \mathbb{W})$ such that

L1. $\mathbb{W}(\hat{A} \times \mathbb{X}) = \Delta(\hat{A}) \quad \forall \hat{A} \in \mathcal{B}(\hat{\mathbb{X}})$,

L2. $\mathbb{W}(\hat{\mathbb{X}} \times A) = \Theta(A) \quad \forall A \in \mathcal{B}(\mathbb{X})$,

L3. $\mathbb{W}(\mathcal{R}) \leq 1 - \delta$.

If the pair of probability distribution belongs to the lifted relation $(\Delta, \Theta) \in \bar{\mathcal{R}}_{\delta}$, then we denote this as $\Delta \bar{\mathcal{R}}_{\delta} \Theta$. δ quantifies the amount of the probability distribution that cannot be coupled into the relation $\mathcal{R}$. In the remainder, we will consider lifting of the stochastic kernels of $\hat{\mathbf{M}}$ and $\mathbf{M}$, that is, $\hat{\mathbb{T}}\bar{\mathcal{R}}_{\delta}\mathbb{T}$. Consider an *interface function* Girard and Pappas (2009); Haesaert and Soudjani (2020) that refines control actions as follows:

$$\mathcal{U} : \hat{\mathbb{U}} \times \hat{\mathbb{X}} \times \mathbb{X} \to \mathcal{P}(\mathbb{U}, \mathcal{B}(\mathbb{U})). \quad (11)$$

Here, an interface function refines any control action $\hat{u}$ synthesized over the approximation $\hat{\mathbf{M}}$ to an action that's applied to the concrete system $\mathbf{M}$. This interface is used with δ-lifting to define approximate stochastic simulation relations on gMDPs as defined in Haesaert and Soudjani (2020); Schön et al. (2023, 2024) and recalled next.

Definition 4 ((ϵ, δ)-stochastic simulation relation)**.** *Consider a concrete system $\mathbf{M} = (\mathbb{X}, \mathbb{U}, \mathbb{Y}, \mathbb{T}, x_0, h)$ and its approximation $\hat{\mathbf{M}} = (\hat{\mathbb{X}}, \hat{\mathbb{U}}, \hat{\mathbb{Y}}, \hat{\mathbb{T}}, \hat{x}_0, \hat{h})$. If there exist a measurable relation $\mathcal{R} \subset \hat{\mathbb{X}} \times \mathbb{X}$, a measurable interface function $\mathcal{U} : \hat{\mathbb{U}} \times \hat{\mathbb{X}} \times \mathbb{X} \to \mathcal{P}(\mathbb{U})$, and a measurable stochastic kernel $\mathbb{W}_{\mathbb{T}} : \hat{\mathbb{U}} \times \hat{\mathbb{X}} \times \mathbb{X} \to \mathcal{P}(\hat{\mathbb{X}} \times \mathbb{X})$ such that*

A1. $(\hat{x}_0, x_0) \in \mathcal{R}$,

A2. $\forall (\hat{x}, x) \in \mathcal{R} : \boldsymbol{d}_{\mathbb{Y}}(\hat{y}, y) \leq \epsilon$ *with* $y = h(x)$ *and* $\hat{y} = \hat{h}(\hat{x})$

A3. $\hat{\mathbb{T}}(\cdot|\hat{x}, \hat{u})\mathcal{R}_{\delta}\mathbb{T}(\cdot|x, \mathcal{U}(\hat{u}, \hat{x}, x))\forall \hat{u} \in \hat{\mathbb{U}} \ \forall (\hat{x}, x) \in \mathcal{R}$ *with the lifted kernel* $\mathbb{W}_{\mathbb{T}}(\cdot|\hat{u}, \hat{x}, x)$,

then $\hat{\mathbf{M}}$ is (ϵ, δ)-stochastically simulated by $\mathbf{M}$. We denote this by $\hat{\mathbf{M}} \preceq^{\delta}_{\epsilon} \mathbf{M}$.

Composable simulations for multi-agent systems. Consider agent i modeled as a gMDP $\mathbf{M}^i$ in (4) with state space $\mathbb{X}_i$. Let $\hat{\mathbf{M}}^i$ be its approximation with a measurable relation $\mathcal{R}_i \subset \hat{\mathbb{X}}_i \times \mathbb{X}_i$ that yields a similarity quantification

$$\hat{\mathbf{M}}^i \preceq^{\delta_i}_{\epsilon_i} \mathbf{M}^i$$

with respect to metric $d_{\mathbb{Y}_i}(\hat{y}_i, y_i) := \|\hat{y}_i - y_i\|$. We immediately compose the agent approximations to obtain $\hat{\mathbf{M}}$ with state space $\hat{\mathbb{X}} := \prod_{i=1}^{N} \hat{\mathbb{X}}_i$ that approximates the concrete multi-agent system $\mathbf{M}$ in (5). Then there exists a multi-agent-level relation $\mathcal{R} \subset \hat{\mathbb{X}} \times \mathbb{X}$ induced by agent-wise relations $\mathcal{R}^i$ as $\mathcal{R} := \{(\hat{x}, x) \in \hat{\mathbb{X}} \times \mathbb{X} | (\hat{x}_i, x_i) \in \mathcal{R}^i, \forall i\}$ for which

$$\hat{\mathbf{M}} \preceq^{\boldsymbol{\delta}}_{\boldsymbol{\epsilon}} \mathbf{M}, \quad \boldsymbol{\delta} := 1 - \prod_{i=1}^{N}(1 - \delta^i), \quad \boldsymbol{\epsilon} := \max_i(\epsilon^i), \quad (12)$$

with respect to the distance metric defined as

$$d_{\mathbb{Y}}(\hat{y}, y) := \max_i(d_{\mathbb{Y}_i}(\hat{y}_i, y_i)). \quad (13)$$

$\boldsymbol{\delta}$ quantifies the amount of probability distribution that cannot be coupled into the relation $\mathcal{R}$.

We proceed to address output deviation ϵ by modifying indicator functions $\mathcal{L}_{\alpha}$ in (10). Similar existing approaches include signal-space robustness degree Fainekos and Pappas (2009) and approximation metrics for transition systems Girard and Pappas (2007). For probabilistic models, metrics

on labelled Markov processes quantify behavioural closeness, with zero distance yielding probabilistic bisimulation Desharnais et al. (2004). In contrast, we define ϵ-robust indicator functions by tightening labeling via contracting (and, when needed, inflation) of predicate regions. The same robust-labeling viewpoint is also used in Liu and Ozay (2016), Section 3.

ϵ-robust indicator functions. Given an $\epsilon \geq 0$, we define the ϵ-robust indicator function $\mathcal{L}^{\epsilon}_{\alpha} : \mathbb{Y} \to \{0, 1\}$ as

$$\mathcal{L}^{\epsilon}_{\alpha}(y) = \begin{cases} 1, & \text{if } L(y) \vDash \alpha \text{ and } \forall y^{\alpha} \in \mathbb{Y}, d_{\mathbb{Y}}(y, y^{\alpha}) \leq \epsilon, \\ 0, & \text{otherwise,} \end{cases} \tag{14}$$

where $y^{\alpha} := \{y \in \mathbb{Y} | L(y) \vDash \alpha\}$.

Let $\mathcal{A} = (Q, q_0, \Sigma_{\mathcal{A}}, \tau_{\mathcal{A}}, q_f)$ be a DFA with Boolean-formula-labeled transitions. For $\epsilon \geq 0$ we define robust indicator functions $\mathcal{L}^{\epsilon}_{\alpha}$ as in (14). The ϵ-robust satisfaction probabilities can be obtained directly as in Proposition 1 based on computing value functions iteratively using operators $\mathbf{T}^{\epsilon}_{\alpha}$ ($\mathcal{L}_{\alpha}$ in (9) replaced by (14)).

Section summary. In this section, we introduced δ-lifted approximate simulation and ϵ-robust indicator functions, implying that (ϵ, δ)-robust satisfaction probability can be computed. The next section shows how an a posteriori correction refines value functions back to certified lower bounds for the concrete continuous system. For ease of presentation, we assume $\epsilon = 0$ for the next two sections. The extension to $\epsilon > 0$ follows directly by replacing nominal indicator functions with robust ones, such that acceptance on the approximate model implies acceptance on the concrete system under (ϵ, δ)-similarity relation. This extension leaves all of our main results intact.

4. TENSOR TREES, VALUE ITERATION, AND A-POSTERIORI CORRECTION

In this section, we revisit the tree-structured tensor value-iteration method, as introduced in our earlier work Wang et al. (2025), for efficient computation of satisfaction probabilities of the abstract system $\hat{\mathbf{M}}$. We will then show that these satisfaction probabilities can be corrected a-posteriori to compensate for approximation-induced robustness gaps. The resulting corrected values provide a provable lower bound on the satisfaction probabilities of the concrete system $\mathbf{M}$.

4.1 Tensor and tensor tree

For $\mathbb{X} = \prod_{i=1}^{N} \mathbb{X}_i$ with $\mathbb{X}_i := \{1, \ldots, n_i\}$, the value functions $\mathcal{V}^{\boldsymbol{\pi}}_{q,t} : \mathbb{X} \to [0, 1]$, can also be represented by an order-N tensor, which is a multi-dimensional array in $\mathbb{R}^{n_1 \times n_2 \times \ldots \times n_N}$. We say that there exists a **Canonical Polyadic Decomposition** (CPD) representation of $\mathcal{V}_{q,t}$ Hitchcock (1927), if there exists arrays $v^{(i)}(z) \in \mathbb{R}^{n_i}$ s.t.

$$\mathcal{V}_{q,t} = \sum_{z=1}^{|\mathcal{Z}|} v^{(1)}(z) \otimes v^{(2)}(z) \otimes \ldots \otimes v^{(N)}(z) \tag{15}$$

where $\otimes$ denotes the outer product and where the CPD rank of $\mathcal{V}_{q,t}$ is $|\mathcal{Z}| \in \mathbb{N}^{+}$ and is denoted as $\text{rank}(\mathcal{V}_{q,t})$.

Algorithm 1 Tensor-tree value iteration $\mathcal{T}^{\pi}(\mathcal{G})$ Wang et al. (2025)

procedure $\mathcal{T}^{\pi}(\mathcal{G})$
  $\mathcal{G}^{+} \leftarrow \text{Grow}(\mathcal{G})$ ▷ Grow tree
  for $(z, \alpha, z') \in \mathcal{G}^{+}.\mathcal{E}$ **do** ▷ Update tree values
    $v(z') = \mathbf{T}^{\pi}_{\alpha}(v(z))$
  end for
end procedure

Algorithm 2 Growing tensor tree $\text{Grow}(\mathcal{G})$ Wang et al. (2025)

procedure Grow($\mathcal{G}$) ▷ Grow tree
  for $z \in \mathcal{G}.\textit{leaves}$ **do**
    for $\{(q, \alpha, q') \in \tau_{\mathcal{A}} | q' = \mathcal{L}_Q(z)\}$ **do**
      $\bar{z} \leftarrow$ CreateNewVertex
      $\mathcal{G}.\mathcal{Z} \leftarrow \mathcal{Z} \cup \{\bar{z}\}$
      $\mathcal{G}.\mathcal{E} \leftarrow \mathcal{E} \cup \{(z, \alpha, \bar{z})\}$
      $\mathcal{G}.\mathcal{L}_Q(\bar{z}) \leftarrow q$
    end for
  end for
end procedure

We are interested in doing value iterations based on low rank CPD representations of our value functions. To this end, we introduce the notion of a tensor-tree value function.

Definition 5 (Tensor-tree value function $\mathcal{G}$ Wang et al. (2025))**.** *For a given DFA $\mathcal{A} = (Q, q_0, \Sigma_{\mathcal{A}}, \tau_{\mathcal{A}}, q_f)$, a tensor-tree value function $\mathcal{G} = (\mathcal{Z}, \mathcal{E}, \mathcal{L}_Q, v)$ has*

- *a set of vertices $\mathcal{Z}$ with elements $z \in \mathcal{Z}$;*
- *a set of (labelled) edges $\mathcal{E}$ with elements $(z, \alpha, z') \in \mathcal{E}$ that defines a* rooted tree*.*
- *a DFA-state mapping $\mathcal{L}_Q : \mathcal{Z} \to Q$ that maps a vertex $z \in \mathcal{Z}$ to a DFA state $q \in Q$.*
- *a vertex value mapping $v : \mathcal{Z} \to \mathbb{R}^{\prod |\mathbb{X}_i|}$ that maps a vertex $z \in \mathcal{Z}$ to a* ***rank-1 tensor***

$$v(z) = \bigotimes_{i=1}^{N} v^{(i)}(z).$$

For a given DFA mode q, the tensor-tree value function maps to a value function

$$\mathcal{V}_q = \sum_{z : \mathcal{L}_Q(z) = q} v(z) \in \mathbb{R}^{\prod |\mathbb{X}^{(i)}|},$$

with $\mathcal{V}_q = \mathbf{0}$ if $\{z | \mathcal{L}_Q(z) = q\} = \emptyset$.

We show that the tensor-tree format is preserved under the recursive computation of $\mathcal{V}_{q,t}$ in (8) if the strategy of the agents only depends on their own state and the mode q, that is if the policy $\boldsymbol{\pi} = \{\pi[t] | t = 0, 1, \ldots, T-1\}$ has elements $\pi[t] \in \Pi_{\mathbf{M}}$, denoted as $\boldsymbol{\pi} \in \Pi_{\mathbf{M}}$, and defined as

$$\Pi_{\mathbf{M}} := \{\pi = \text{col}(\pi^{(1)}, ..., \pi^{(N)}) | \pi^{(i)} : \mathbb{X}_i \times Q \to \mathbb{U}_i\}. \tag{16}$$

First note that the initial value function $\mathcal{V}_{q,0}$ can be trivially represented by $\mathcal{G}_0 = \{\mathcal{Z}_0, \mathcal{E}_0, v_0, \mathcal{L}_{Q_0}\}$, with $\mathcal{Z}_0 = \{1\}$, $\mathcal{E}_0 = \emptyset$, $v_0(1) = \bigotimes_{i=1}^{N} \mathbf{1}$, and $\mathcal{L}_Q(1) = q_f$. Based on Algorithm 2, we can grow the number of leafs $\mathcal{G}_0^{+} = \text{Grow}(\mathcal{G}_0)$ after which we can compute the updated vertex values based on the operator $\mathbf{T}^{\pi_q}_{\alpha}$ in eq. (9). For a policy sequence $\boldsymbol{\pi} = \{\pi[0], \ldots, \pi[T-1]\}$, this yields the tensor-tree value iteration defined in Alg.1 with

$$\mathcal{G}_{t+1} = \mathcal{T}^{\pi[T-1-t]}(\mathcal{G}_t). \quad (17)$$

As in Wang et al. (2025), we can show that the tensor-tree value iterations preserve vertices with rank-1 tensor values with whose value the value function in can be recovered.

Proposition 2 (Tensor-tree value iterations Wang et al. (2025))**.** *For a horizon T, a Markov policy $\boldsymbol{\pi} = \{\pi[0], \ldots, \pi[T-1]\} \in \Pi_{\mathbf{M}}$ and a tree $\mathcal{G}_t = \{\mathcal{Z}_t, \mathcal{E}_t, \mathcal{L}_{Q_t}, v_t\}$ computed with tensor-tree value iterations (17), we have that*

- *$\mathcal{G}_t$ is a rank-1 tensor tree as defined in Def. 5, that is,*
$$\text{rank}(v_t(z)) = 1 \quad \forall z \in \mathcal{Z}$$
- *$\mathcal{G}_t$ represents the value functions $\mathcal{V}^{\boldsymbol{\pi}}_{q,t} \quad \forall q \in Q$, computed based on (8), that is,*
$$\mathcal{V}^{\boldsymbol{\pi}}_{q,t}(x) = \sum_{z:\mathcal{L}_Q(z)=q} v_t(z)(x) \quad \forall x \in \mathbb{X}, \forall q \in Q. \quad (18)$$

Policy optimization. As in Wang et al. (2025), for each mode q and each subsystem $i \in \{1, \ldots, N\}$, we can develop efficient heuristic optimization of the policy with

$$\pi_q^{(i)*}(x_i, u_i) \in \arg\max_{\pi_q^{(i)}} \sum_{e \in \mathcal{E}_q} \mathbf{T}_{\alpha^i}^{\pi_q^{(i)}}(v^{(i)})(z) c_e^i \quad (19)$$

where $\mathcal{E}_q := \{e = (z, \alpha, z') \in \mathcal{E} | \mathcal{L}_Q(z') = q\}$ and $c_e^i := \prod_j^{\{1,\ldots,N\}\setminus\{i\}} \|\mathbf{T}_{\alpha^j}^{\pi_q^{(j)}}(v^{(j)})(z)\|_1$.

4.2 A-posteriori corrected satisfaction probability

Given a multi-agent gMDP $\mathbf{M}$ as in (5), we require that each agent $\mathbf{M}^i$ has an abstract agent $\hat{\mathbf{M}}^i$ such that for all $i \in \{1, \ldots, N\}$, $\hat{\mathbf{M}}^i \preceq_0^{\delta^i} \mathbf{M}^i$. Together, these abstract gMDPs define the abstract multi-agent gMDP $\hat{\mathbf{M}}$ which is $(\boldsymbol{\delta}, 0)$-simulated by $\mathbf{M}$ with $\boldsymbol{\delta} := 1 - \prod_{i=1}^N (1 - \delta^i)$.

Leveraging the tensor-tree value iterations in Algorithm 1, we can efficiently use the tensor tree to compute value functions $\mathcal{V}^{\boldsymbol{\pi}}_{q,T}(\hat{x})$. In the rest of this section, we use the approximate simulation relation between $\mathbf{M}$ and $\hat{\mathbf{M}}$ to relate the abstract computations back to satisfaction probabilities of the concrete multi-agent system $\mathbf{M}$. Under the adopted similarity quantification, probability losses accumulate up to the acceptance event. Its effect can be quantified with the *first hitting time* Haesaert and Soudjani (2020), defined as follows. For a given execution $\hat{\boldsymbol{x}}$, the first hitting time of a set $\{q_f\}$ is a random variable conditioned on the initial state $\hat{x}_0$ and defined as

$$H_{\{q_f\}}(\hat{x}_0) := \{\inf\{t \in \mathbb{N} \cup \{\infty\} : q_t \in \{q_f\}\} - 1 | \hat{x}_0\}, \quad (20)$$

where $q_{k+1} := \tau_{\mathcal{A}}(q_k, L(h(\hat{x}_k)))$ for $k \in \mathbb{N}$, with $q_0 := q_0$. The support of first hitting time is $\mathbb{N} \cup \{\infty\}$. It can be infinity if the execution does not hit the set $\{q_f\}$. $H_{\{q_f\}}(\hat{x}_0)$ is 0 with probability $\mathbb{P}(H_{\{q_f\}}(\hat{x}_0) = 0) = 1$ if $\tau_{\mathcal{A}}(q_0, L(h(\hat{x}_0))) = q_f$.

Proposition 3 ($\mathbf{C}_{\boldsymbol{\pi}}$ with satisfaction probability)**.** *For multi-agent gMDP $\mathbf{M}$ and its abstraction $\hat{\mathbf{M}}$ with $\hat{\mathbf{M}} \preceq_0^{\delta} \mathbf{M}$, we can* construct *a control strategy $\mathbf{C}_{\boldsymbol{\pi}}$ such that the controlled system $\mathbf{M} \times \mathbf{C}_{\boldsymbol{\pi}}$ satisfies ϕ with lower bound*

$$\mathbb{P}_{\mathbf{M}\times\mathbf{C}_{\boldsymbol{\pi}}}(\boldsymbol{l}_{[0,T]} \models \phi) \geq \mathcal{V}^{\boldsymbol{\pi}}_{\bar{q}_0,T}(\hat{x}_0) - \boldsymbol{\delta}\sum_{h=1}^{T} \mathbb{P}(H_{\{q_f\}}(\hat{x}_0) \geq h),$$

for any $T \geq 1$ with $\bar{q}_0 = \tau_{\mathcal{A}}(q_0, L(h(\hat{x}_0)))$.

Proof. The proof follows the same reasoning as in the proof of Thm. 3 Haesaert and Soudjani (2020) and is omitted here due to space limitation. □

In general, $\mathbb{P}(H_{\{q_f\}}(\hat{x}_0) \geq h)$ can be very difficult to compute. However, given the information encapsulated in the tensor tree we can compute a bound on it as follows.

Theorem 1 ($\mathbf{C}_{\boldsymbol{\pi}}$ with a-posteriori corrected satisfaction probability)**.** *For multi-agent gMDP $\mathbf{M}$ and its abstraction $\hat{\mathbf{M}}$ with $\hat{\mathbf{M}} \preceq_0^{\delta} \mathbf{M}$, we can* construct *a control strategy $\mathbf{C}_{\boldsymbol{\pi}}$ such that the controlled system $\mathbf{M} \times \mathbf{C}_{\boldsymbol{\pi}}$ satisfies ϕ with lower bound*

$$\mathbb{P}_{\mathbf{M}\times\mathbf{C}_{\boldsymbol{\pi}}}(\boldsymbol{l}_{[0,T]} \models \phi) \geq \mathcal{V}^{\boldsymbol{\pi}}_{\bar{q}_0,T}(\hat{x}_0) - \boldsymbol{\delta}(T - \sum_{h=1}^{T}\sum_{i=0}^{h-1}\sum_{z\in\mathcal{D}_i^{\bar{q}_0}} v(z)) \quad (21)$$

for any $T \geq 1$ with $\bar{q}_0 = \tau_{\mathcal{A}}(q_0, L(h(\hat{x}_0)))$ where $v(z)$ is the value of the vertex that belongs to $\mathcal{D}_i^{\bar{q}_0}$. $\mathcal{D}_i^{\bar{q}_0}$ is the set containing $\bar{q}_0$-labeled vertices, each of which has exactly i predecessors, defined as

$$\mathcal{D}_i^{\bar{q}_0} = \{z | z \text{ has } i \text{ predecessors and } \mathcal{L}_Q(z) = \bar{q}_0\}. \quad (22)$$

Proof. It is trivial that

$$\mathbb{P}(H_{\hat{\mathbb{X}}\times\{q_f\}}(\hat{x}, q) \geq h) = 1 - \mathbb{P}(H_{\hat{\mathbb{X}}\times\{q_f\}}(\hat{x}, q) < h).$$

Based on the results in Proposition 3, we get

$$\mathcal{V}^{\boldsymbol{\pi}}_{q,T}(\hat{x}, x) \geq \mathcal{V}^{\boldsymbol{\pi}}_{q,T}(\hat{x}) - \boldsymbol{\delta}\sum_{h=1}^{T}\Big(1 - \mathbb{P}(H_{\hat{\mathbb{X}}\times\{q_f\}}(\hat{x}, q) < h)\Big),$$

where $\mathbb{P}(H_{\hat{\mathbb{X}}\times\{q_f\}}(\hat{x}, q) < h)$ denotes the probability the controlled system product reaches the accepted state within $h-1$ control steps. Based on to be updated Theorem 2/Proposition 2, such probability can be lower bounded by the vertices' values, i.e.,

$$\mathbb{P}(H_{\hat{\mathbb{X}}\times\{q_f\}}(\hat{x}, q) < h) \geq \sum_{i=0}^{h-1}\sum_{z\in\mathcal{D}_i^q} v(z) \quad (23)$$

where $\mathcal{L}^{-1}_{Q_{h-1}}(q) : Q \to \mathcal{Z}$ gives the vertices set in $\mathcal{G}_{h-1}$ which is labeled q, and $\mathcal{D}_i^q$ is the set containing q-labeled vertices, each of which has exactly i predecessors, defined as in (22). □

A-posteriori corrected policy optimization. For each mode q and each agent i, we extend the policy optimization in (19) for $\pi_q^i \in \Pi_{\mathbf{M}}$ to account for the a-posteriori correction by multiplying a weighted term as follows.

$$\pi_q^{(i)*}(x_i, u_i) \in \arg\max_{\pi_q^{(i)}} \sum_{e \in \mathcal{E}_q} \zeta_{z'} \mathbf{T}_{\alpha^i}^{\pi_q^{(i)}}(v^{(i)})(z) c_e^i \quad (24)$$

where $\mathcal{E}_q := \{e = (z, \alpha, z') \in \mathcal{E} | \mathcal{L}_Q(z') = q\}$, $\zeta_{z'} := 1 + \boldsymbol{\delta}(T - \text{Pred}(z'))$ with $\text{Pred}(z')$ the number of predecessors of z', and $c_e^i := \prod_j^{\{1,\ldots,N\}\setminus\{i\}} \|\mathbf{T}_{\alpha^j}^{\pi_q^{(j)}}(v^{(j)})(z)\|_1$.

5. ROBUST TENSOR TREE

This section presents a robust version of (8) by proposing a robust operator via which we embed probability loss quantification into value iteration, following the same idea as δ-robust mapping in Haesaert and Soudjani (2020). Unlike the a-posteriori correction as introduced in the

Algorithm 3 Robust tensor-tree value iteration $\mathcal{T}^{\pi}(\mathcal{G})$

procedure $\mathcal{T}^{\pi}(\mathcal{G})$
  $\mathcal{G}^{+} \leftarrow \text{Grow}(\mathcal{G})$ ▷ Grow tree
  for $(z, \alpha, z') \in \mathcal{G}^{+}.\mathcal{E}$ **do** ▷ Update tree values
    $v^{(i)}(z') = \mathbf{T}^{\pi^i}_{\delta^i,\alpha^i}(v^{(i)})(z)\ \forall i \in \{1, \ldots, N\}$
  end for
end procedure

previous section, this online robust correction accounts for probability loss during iteration rather than until acceptance, such that guaranteed lower bounds are guaranteed throughout recursive computation.

We define the δ^i-robust DFA-informed operator $\mathbf{T}^{\delta^i}$ as

$$\mathbf{T}^{\pi}_{\delta^i,\alpha^i}(v^{(i)})(x_i) := \mathbf{L}\left(\mathbb{E}^{\pi^i_q}_{x_i'}[\mathcal{L}_{\alpha^i}(x_i')v^{(i)}(x_i')] - \delta^i\right). \tag{25}$$

Unlike in the previous subsection, we present the proposed operator explicitly for each $\mathbf{M}^i$ because δ^i is $\mathbf{M}^i$-specific; as reusing the generic form (9) would obscure this dependence. We perform robust dynamic programming on $\hat{\mathbf{M}}$ by leveraging the robust tensor-tree value iteration in Algorithm 3. We use the approximate relation between $\mathbf{M}$ and $\hat{\mathbf{M}}$ to relate the computed robust value functions back to satisfaction probabilities of the controlled multi-agent system $\mathbf{M} \times \mathbf{C}_{\boldsymbol{\pi}}$. The robust tensor tree provides a direct certified lower bound as follows.

Theorem 2 ($\mathbf{C}_{\boldsymbol{\pi}}$ with robust-tree corrected satisfaction probability)**.** *For multi-agent gMDP* $\mathbf{M}$ *and its abstraction* $\hat{M}$ *with* $\hat{\mathbf{M}} \preceq^{\delta}_{0} \mathbf{M}$*, we can construct a control strategy* $\mathbf{C}_{\boldsymbol{\pi}}$ *such that the controlled system* $\mathbf{M} \times \mathbf{C}_{\boldsymbol{\pi}}$ *satisfies* ϕ *with lower bound*

$$\mathbb{P}_{\mathbf{M}\times\mathbf{C}_{\boldsymbol{\pi}}}(\boldsymbol{l}_{[0,T]} \vDash \phi) \geq \sum_{n \in \mathcal{L}_{Q_T}^{-1}(\bar{q}_0)} v^{\delta}_{T}(z) \tag{26}$$

for any $T \geq 1$ *with* $\bar{q}_0 := \tau_{\mathcal{A}}(q_0, L(h(\hat{x}_0)))$*, where* $\mathcal{L}_{Q_T}^{-1}(\bar{q}_0) : Q \to \mathcal{Z}$ *gives the vertice set in* $\mathcal{G}^{\delta}_{T}$ *which is labeled by* $\bar{q}_0$*.* $\mathcal{G}^{\delta}_{T}$ *is robustly expanded as in Algorithm 3.*

Proof. The proof of Theorem 2 follows along the same lines as the proof of Theorem 2 in Haesaert and Soudjani (2020) with $\bigotimes_{i=1}^{N} \mathbf{T}^{\pi^i_q}_{\delta^i,\alpha^i}$ instead of $\mathbf{T}$. For a complete proof, we refer to our website to be updated. □

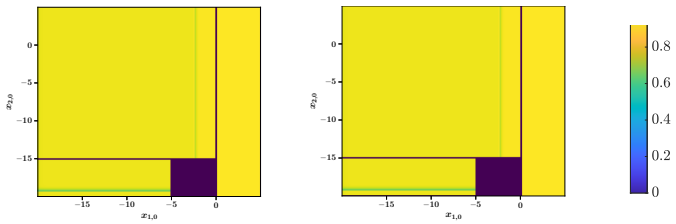
Robust-tree policy optimization. To account for the robust-tree correction, for each mode q and each agent i, we optimize policy $\pi^i_q \in \Pi_{\mathbf{M}}$ as

$$\pi^{(i)*}_q(x_i, u_i) \in \arg\max_{\pi^{(i)}_q} \sum_{e \in \mathcal{E}_q} \mathbf{T}^{\pi^{(i)}_q,\delta^i}_{\alpha^i}(v^{(i)})(z)c^i_e \tag{27}$$

where $\mathcal{E}_q := \{e = (z, \alpha, z') \in \mathcal{E} | \mathcal{L}_Q(z') = q\}$ and $c^i_e := \prod_j^{\{1,\ldots,N\}\setminus\{i\}} \|\mathbf{T}^{\pi^{(j)}_q,\delta^j}_{\alpha^j}(v^{(j)})(z)\|_1$. Compared to policy optimization in (24), this correction uniformly subtracts δ^i for all nodes and all states, and is therefore more conservative.

6. BENCHMARKING

All simulations were conducted in Python on a MacBook Air with Apple M4 chip, and 16 GM RAM.



(a) A-posteriori corrected optimal value functions. (b) Robust-tree corrected optimal value functions.

Fig. 3. Lower-bounded satisfaction probabilities for initial states $x_0 \in [-20, 5] \times [-20, 5]$ with specification horizon $T = 6$.

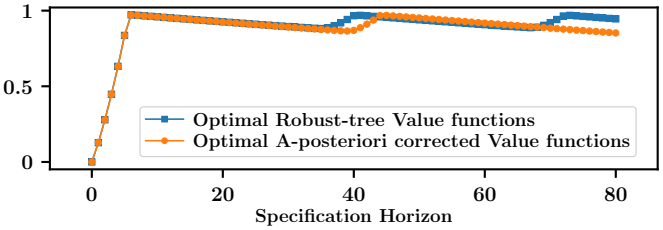


Fig. 4. Lower bounds on the satisfaction probability of a reach-avoid specification on system (28) as a function of the specification horizon. Orange: a-posteriori corrected (Theorem 1, $\Pi^{*}_{\mathbf{M}}$ optimized as in (24)); blue: robust-tree corrected (Theorem 2, $\Pi^{*}_{\mathbf{M}}$ optimized as in (27)).

6.1 Two dimensional case

Consider a 2-agent system:

$$\begin{aligned} x_i^{+} &= 0.9x_i + 0.5u_i + 0.5w_i, \\ y_i &= x_i, \qquad\qquad \text{for } i = 1, \ldots, N \end{aligned} \tag{28}$$

with $N = 2$, $w_i \sim \mathcal{N}(0, I)$, $\mathbb{X}_i : [-20, 5]$, $\mathbb{U}_i : [-5, 5]$. Consider a reach-avoid specification $\phi := (\neg p_2 \wedge \neg p_3) \mathsf{U} p_1$ with $p_1 =$ true iff $x_1 \in [0, 5]$, $p_2 =$ true iff $x_1 \in [-5, 0]$, and $p_3 =$ true iff $x_2 \in [-20, -15]$. We construct a finite-state abstraction for each agent (28) $\hat{\mathbf{M}}^1, \hat{\mathbf{M}}^2$ by gridding the state space $\mathbb{X}_i$ with 1000 grid cells and the input space $\mathbb{U}_i$ with 10 grid cells. With a chosen interface function $u_i = \hat{u}_i$ and relation $\mathcal{R}^i$ with $\epsilon^i = 0.1$, we obtain the approximate simulation relation $\hat{\mathbf{M}}^i \preceq^{0.002}_{0.1} \mathbf{M}^i$ using **SySCoRe** tool Van Huijgevoort et al. (2023). We visualize lower-bounded satisfaction probabilities computed via a-posteriori correction as in Theorem 1 and via robust tree as in Theorem 2 for system state initialized as $x_0 \in [-20, 5] \times [-20, 5]$ in Fig. 3. We also compute the average of probabilities for system states initialized as $x_0 \in [-20, 0] \times [-5, -2.5]$, visualized in Fig. 4. The robust-tree robust satisfaction probabilities exhibit regular small oscillations in Fig. 4. We hypothesize that this non-monotonicity is an artifact of optimizing over the constrained policy $\Pi_{\mathbf{M}}$.

6.2 Scalability benchmark

Consider a 20-agent system with dynamics as in (28) with $N = 20$, $w_i \sim \mathcal{N}(0, I)$, $\mathbb{X}_i : [-10, 10]$, $\mathbb{U}_i : [-2, 2]$. Consider an invariance specification $\phi := \bigwedge_{t=0}^{T} \bigcirc (\bigwedge_{i=1}^{N=20} p^i_1)$ with p^i_1 is true iff $x_i \in [-5, 5]$. We construct a finite-state abstraction for each agent (28) by gridding the state space $\mathbb{X}_i$ with 100 grid cells and the input space $\mathbb{U}_i$ with 10 grid cells. We choose interface function $u_i = \hat{u}_i$ and relation $\mathcal{R}^i$ with $\epsilon^i = 0.1$. We obtain $\hat{\mathbf{M}}^i \preceq^{0.0717}_{0.1} \mathbf{M}^i$ using

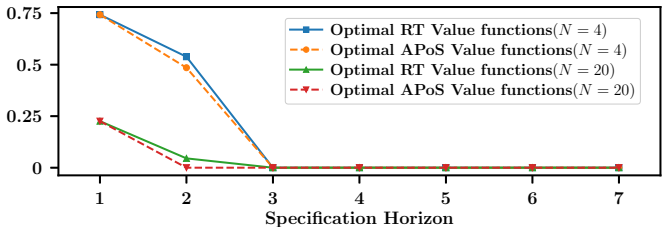


Fig. 5. Lower bounds on the satisfaction probability of an invariance specification on system (28) as a function of the specification horizon T. Orange, red: a-posteriori (**APoS**) corrected (Theorem 1, $\Pi^*_{\mathbf{M}}$ optimized as in (24)) for 4-agent system and 20-agent system; blue, green: robust-tree (**RT**) corrected (Theorem 2, $\Pi^*_{\mathbf{M}}$ optimized as in (27)) for 4-agent system and 20-agent system.

SySCoRe tool Van Huijgevoort et al. (2023). We present the average of (5 sampled initial states) of the lower bounds of satisfaction probability computed via a-posteriori as in Thm. 1 and robust tree as in Thm. 2 in Fig. 5 for a sweep of specification horizon, i.e., $T = 1, \dots, 7$. It is expected that as the number of agents increases, the probability that all agents remain in the invariant set over a fixed horizon decreases, highlighting the difficulty in satisfying an invariance specification as number of agents grows. However, robust computations do not rely on all agents staying in the simulation relation are therefore of interest for future work. For the 20-agent system with specification horizon $T = 3$, computing the optimal value functions for sampled points using a-posteriori corrected computation peaked at 233.95MiB and completed in 1.38 seconds.

7. CONCLUSIONS

This paper presents two correct-by-design control synthesis methods for continuous-state stochastic multi-agent systems that leverage tensor-structured value functions for scalable computation. Abstraction error is quantified via (ϵ, δ) simulation relation and two ways of correction, i.e., a-posteriori and robust-tree propagation. We validate our results on linear systems. Future work includes specification that requires interactions among agents.

REFERENCES

Abate, A., Katoen, J.P., Lygeros, J., and Prandini, M. (2010). Approximate model checking of stochastic hybrid systems. *European Journal of Control*, 16(6), 624–641.

Alora, J.I., Gorodetsky, A., Karaman, S., Marzouk, Y., and Lowry, N. (2016). Automated synthesis of low-rank control systems from sc-ltl specifications using tensor-train decompositions. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, 1131–1138. IEEE.

Baier, C. and Katoen, J.P. (2008). *Principles of model checking*. MIT press.

Belta, C., Yordanov, B., and Gol, E.A. (2017). *Formal methods for discrete-time dynamical systems*, volume 89. Springer.

Bertsekas, D. and Shreve, S.E. (1996). *Stochastic optimal control: the discrete-time case*, volume 5. Athena Scientific.

Bogachev, V.I. (2007). *Measure theory*. Springer.

Cai, M. and Vasile, C.I. (2025). Safety-critical learning of robot control with temporal logic specifications. *IEEE Transactions on Automatic Control*.

Desharnais, J., Gupta, V., Jagadeesan, R., and Panangaden, P. (2004). Metrics for labelled markov processes. *Theoretical computer science*, 318(3), 323–354.

Duret-Lutz, A. and Poitrenaud, D. (2004). Spot: an extensible model checking library using transition-based generalized bu/spl uml/chi automata. In *The IEEE Computer Society's 12th Annual International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems, 2004.(MASCOTS 2004). Proceedings.*, 76–83. IEEE.

Fainekos, G.E. and Pappas, G.J. (2009). Robustness of temporal logic specifications for continuous-time signals. *Theoretical Computer Science*, 410(42), 4262–4291.

Girard, A. and Pappas, G.J. (2007). Approximation metrics for discrete and continuous systems. *IEEE Transactions on Automatic Control*, 52(5), 782–798.

Girard, A. and Pappas, G.J. (2009). Hierarchical control system design using approximate simulation. *Automatica*, 45(2), 566–571.

Gorodetsky, A., Karaman, S., and Marzouk, Y. (2018). High-dimensional stochastic optimal control using continuous tensor decompositions. *The International Journal of Robotics Research*, 37(2-3), 340–377.

Haesaert, S. and Soudjani, S. (2020). Robust dynamic programming for temporal logic control of stochastic systems. *IEEE Transactions on Automatic Control*, 66(6), 2496–2511.

Hitchcock, F.L. (1927). The expression of a tensor or a polyadic as a sum of products. *Journal of Mathematics and Physics*, 6(1-4), 164–189.

Jagtap, P., Soudjani, S., and Zamani, M. (2020). Formal synthesis of stochastic systems via control barrier certificates. *IEEE Transactions on Automatic Control*, 66(7), 3097–3110.

Julian, K.D., Kochenderfer, M.J., and Owen, M.P. (2019). Deep neural network compression for aircraft collision avoidance systems. *Journal of Guidance, Control, and Dynamics*, 42(3), 598–608.

Latvala, T. (2003). Efficient model checking of safety properties. In *International SPIN Workshop on Model Checking of Software*, 74–88. Springer.

Liu, J. and Ozay, N. (2016). Finite abstractions with robustness margins for temporal logic-based control synthesis. *Nonlinear Analysis: Hybrid Systems*, 22, 1–15.

Liu, S., Noroozi, N., and Zamani, M. (2021). Symbolic models for infinite networks of control systems: A compositional approach. *Nonlinear Analysis: Hybrid Systems*, 43, 101097.

Mallik, K., Schmuck, A.K., Soudjani, S., and Majumdar, R. (2018). Compositional synthesis of finite-state abstractions. *IEEE Transactions on Automatic Control*, 64(6), 2629–2636.

Rozada, S. and Marques, A.G. (2024). Tensor low-rank approximation of finite-horizon value functions. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5975–5979. IEEE.

Schön, O., van Huijgevoort, B., Haesaert, S., and Soudjani, S. (2023). Verifying the unknown: Correct-by-design control synthesis for networks of stochastic uncertain systems. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, 7035–7042. IEEE.

Schön, O., van Huijgevoort, B., Haesaert, S., and Soudjani, S. (2024). Bayesian formal synthesis of unknown systems via robust simulation relations. *IEEE Transactions on Automatic Control*.

Tabuada, P. (2009). *Verification and control of hybrid systems: a symbolic approach*. Springer Science & Business Media.

Van Huijgevoort, B., Schön, O., Soudjani, S., and Haesaert, S. (2023). Syscore: Synthesis via stochastic coupling relations. In *Proceedings of the 26th ACM international conference on hybrid systems: Computation and control*, 1–11.

Venayagamoorthy, G.K. (2011). Dynamic, stochastic, computational, and scalable technologies for smart grids. *IEEE Computational Intelligence Magazine*, 6(3), 22–35.

Wang, R., Sun, Z., and Haesaert, S. (2025). Unraveling tensor structures in correct-by-design controller synthesis. In *Proceedings of the 64th IEEE Conference on Decision and Control (CDC)*, to appear. IEEE.

Wolff, E.M., Topcu, U., and Murray, R.M. (2012). Robust control of uncertain markov decision processes with temporal logic specifications. In *51st IEEE Conference on decision and control (CDC)*, 3372–3379. IEEE.

Zamani, M., Abate, A., and Girard, A. (2015). Symbolic models for stochastic switched systems: A discretization and a discretization-free approach. *Automatica*, 55, 183–196.