Formal Verification of Stochastic Systems with ReLU Neural Network Controllers

Shiqi Sun, Yan Zhang, Xusheng Luo, Panagiotis Vlantis, Miroslav Pajic and Michael M. Zavlanos

***Abstract*— In this work, we address the problem of formal safety verification for stochastic cyber-physical systems (CPS) equipped with ReLU neural network (NN) controllers. Our goal is to find the set of initial states from where, with a predetermined confidence, the system will not reach an unsafe configuration within a specified time horizon. Specifically, we consider discrete-time LTI systems with Gaussian noise, which we abstract by a suitable graph. Then, we formulate a Satisfiability Modulo Convex (SMC) problem to estimate upper bounds on the transition probabilities between nodes in the graph. Using this abstraction, we propose a method to compute tight bounds on the safety probabilities of nodes in this graph, despite possible over-approximations of the transition probabilities between these nodes. Additionally, using the proposed SMC formula, we devise a heuristic method to refine the abstraction of the system in order to further improve the estimated safety bounds. Finally, we corroborate the efficacy of the proposed method with simulation results considering a robot navigation example and comparison against a state-of-the-art verification scheme.**

I. Introduction

In recent years, advances in the field of deep learning have furnished a new class of adept and adaptable control schemes for cyber-physical systems which considerably simplify the overall design process. Robot navigation is one such exemplar application where neural network controllers have been successfully employed for steering a variety of robotic platforms in a diversity of situations [1]–[5]. As these AI-enabled systems open up new possibilities for control, which are still considered understudied in the literature, the issues of safety and reliability of neural network controllers, become more pressing. In addition, as such control schemes are employed to address safety-critical real-world problems, the ability to formally verify the security of the neural networks becomes imperative [6].

To address these challenges, a significant effort has been recently directed to the robustification and verification of deep neural networks. Considering the former direction, Generative Adversarial Network (GAN) methodologies have been successfully employed to train networks with improved tolerance to disturbances [7], [8]. Although these methods may yield noticeably more robust networks, they provide no means of estimating the reliability of the system. On the other hand, verification schemes provide ways to estimate bounds on the output of already trained networks and answer reachability queries related to the corresponding closed-loop dynamics. In [9], [10], a reachability analysis method for neural networks is proposed that relies on semi-definite programming whereas, in [11], satisfiability modulo theory is employed to provide formal verification guarantees. In [12], [13], a hybrid system verification scheme is proposed to answer reachability queries concerning dynamical systems equipped with neural network controllers. Likewise, the Satisfiability Modulo Convex optimization (SMC) approach [14] is adopted in [15] in order to verify the safety of neural networks with ReLU activation functions for robotic platforms equipped with proximity sensors. Stochastic problems are also considered in [16]–[20]. Particularly, in [19], [20] a sampling-based method is proposed to ensure safety of a closed-loop system subject to randomness only in the initial conditions, whereas the methodologies in [17], [18] are limited to mixed monotone stochastic systems. Finally, guarantees on the safety of stochastic switched systems equipped with general nonlinear controllers are also derived in [16] using Internal Markov Chain (IMC) and Bound-Parameter Markov Decision Process (BMDP) methods which, however, require very fine partitions of the domain in order to furnish accurate safety probability bounds.

In this paper, we propose a new verification scheme for stochastic dynamical systems equipped with ReLU neural network controllers. Particularly, we consider discrete-time LTI systems with Gaussian noise and partition of the continuous state space into convex sets (e.g., constructed as in [15]). Then, we abstract the system by a graph and formulate a Satisfiability Modulo Convex problem which we solve using existing tools in order to estimate valid upper bounds on the transition probabilities between pairs of nodes in the graph. Using this transition graph, we also develop an algorithm to estimate tight upper bounds on the probability the system reaches the set of unwanted states after a specified amount of steps, even when the underlying transition probability bounds have been over-estimated. Unlike methodologies such as [19], [20] that yield probably correct estimations of the safety probability bounds, here we provide bounds that are correct by design. Additionally, we use the proposed SMC formula to devise a heuristic method to subdivide the cells in a given abstraction in order to further improve our safety probability estimations. Finally, we provide numerical simulations on a robot navigation problem corroborating the efficacy of our proposed verification method compared

Shiqi Sun, Yan Zhang, Xusheng Luo, Panagiotis Vlantis and Michael M. Zavlanos are with the Department of Mechanical Engineering and Materials Science, Duke University, Durham, NC 27708, USA. {shiqi.sun,yan.zhang2,xusheng.luo, panagiotis.vlantis,michael.zavlanos}@duke.edu
Miroslav Pajic is with the Department of Electrical and Computer Engineering Duke University, Durham, NC 27708, USA. miroslav.pajic@duke.edu . This work is supported in part by AFOSR under award #FA9550-19-1-0169 and by NSF under award CNS-1932011.

to [16], which provides looser bounds on the safety probability for the coarse partitions considered here.

We organize the paper as follows. In Section II, we formulate the problem under consideration while in Section III we elaborate on the construction of the graph and the methodology used for computing upper bounds on the transition probabilities. Then, in Section IV, we present the proposed verification scheme and in Section V we develop the proposed heuristic method to refine the selected state abstraction. Finally, in Section VI we conclude this work by presenting comparative results corroborating the efficacy of our scheme.

II. Problem Formulation

We consider an autonomous robot moving in a compact, polytopic workspace $\mathcal{W} \subset \mathbb{R}^p$ occupied by a set of zero or more inner obstacles $\{\mathcal{O}_i\}_{i=1}^{p_o}$. Let $\mathcal{W}_s = \mathcal{W} \backslash \cup_{i=1}^{p_o} \mathcal{O}_i$ be the set of safe robot positions and let $\mathcal{W}_o$ denote its complement. The robot's dynamics are described by the following linear discrete-time stochastic model

$$x^{t+1} = Ax^t + Bu^t + w^t, \tag{1}$$

where $x^t \in \mathcal{X} \subseteq \mathbb{R}^n$ and $u^t \in \mathcal{U} \subset \mathbb{R}^m$ denote the robot's state and control input at time t, respectively, and $w^t \sim \mathcal{N}(0, \sigma_t)$, $\sigma_t \in \mathbb{R}^n$ is externally induced Gaussian noise, applied at time t. We assume that the robot is equipped with one or more sensors that allow it to perceive the unoccluded region of the workspace around it. We shall use $d(x^t)$ to denote the sensor measurements obtained at configuration x^t, with $d : \mathbb{R}^n \mapsto \mathbb{R}^q$ being the measurement function. Also, we assume that a pre-trained, feed-forward neural network controller $f_{\text{NN}} : \mathbb{R}^q \mapsto \mathbb{R}^m$ is provided for steering the robot to a desired configuration, i.e., $u^t = f_{\text{NN}}(d(x^t))$. Particularly, we assume that the controller consists of L fully connected layers, i.e.,

$$\begin{aligned} h^1 &= \max\left(0, W_\phi^0 d(x^t) + w_\phi^0\right), \\ &\vdots \\ h^L &= \max\left(0, W_\phi^{L-1} h^{L-1} + w_\phi^{L-1}\right), \\ u^t &= W_\phi^L h^L + w_\phi^L \end{aligned}, \tag{2}$$

where $W_\phi^l \in \mathbb{R}^{M_i \times M_{l-1}}$, $w_\phi^l \in \mathbb{R}^{M_l}$ are pre-trained weight bias matrices and h^i denotes the output of the i-th layer.

Given the stochastic system (1) and associated control law (2), let $P_k : \mathbb{R}^n \mapsto [0, 1]$ denote the probability that the robot will reach an unsafe state after k time steps starting from x^t, i.e.,

$$P_k(x^t) = P(\mathcal{P}_W(x^{t+k}) \in \mathcal{W}_o \mid x^t), \tag{3}$$

where $\mathcal{P}_W : \mathbb{R}^n \mapsto \mathbb{R}^p$ is a projection operator that returns the robot's current position. In the remainder, we shall say that a state x^t is (p, k)-safe if $P_k(x) \leq p$, given $p \in [0, 1]$. Note that, in practice, computing a precise approximation of P_k may generally be intractable. Therefore, in this work, we address the problem of computing a correct-by-design tight upper bound on P_k, which allows us to answer safety queries, albeit more conservatively.

Problem 1. *Given a robotic system obeying the closed-loop dynamics* (1) *and* (2)*, compute a tight upper bound of the probability function $P_k(x)$, for given horizon k and $\forall x \in \mathcal{X}$.*

III. Transition Graph

In order to address Problem 1, in this section we develop a methodology to construct a discrete abstraction of the system's dynamics and to compute upper bounds on the transition probabilities between different pairs of cells in this abstraction. Then, in Section IV, we utilize this transition graph to estimate upper bounds on the safety probability P_k.

We begin by partitioning[1] the state space $\mathcal{X}$ into a set $\mathcal{S} = \{\mathcal{S}_i\}_{i=1}^{p_s}$ of p_s non-overlapping convex polytopes such that $\mathcal{X} = \cup_{i=1}^{p_s} \mathcal{S}_i$ and $\mathcal{S}_i \cap \mathcal{S}_j = \emptyset$ for all $i \neq j$. Let $\mathcal{E} \subseteq \mathcal{S} \times \mathcal{S}$ consist of all the pairs $\mathcal{S}_i, \mathcal{S}_j$ such that there exists at least one $x^t \in \mathcal{S}_i$ for which $P(x^{t+1} \in \mathcal{S}_j) > 0$. Using the partition $\mathcal{S}$, we can model the dynamics of the stochastic system in (1) by a graph with edge weights that correspond to upper bounds on the transition probabilities between all pairs of abstract states $\mathcal{S}_i$ and $\mathcal{S}_j$. Specifically, we state the following definition.

Definition 1. *Given the system* (1)*,* (2) *and the partition of the space $\mathcal{S}$, an Upper Bound Probabilistic Transition Graph is a tuple $\mathcal{D} = (\mathcal{S}, \mathcal{E}, \hat{P})$ such that $P(x^{t+1} \in \mathcal{S}_j | x^t \in \mathcal{S}_i) \leq \hat{P}(\mathcal{S}_i, \mathcal{S}_j)$ for all $(\mathcal{S}_i, \mathcal{S}_j) \in \mathcal{E}$.*

In order to construct the transition graph in Definition 1, we require a function $\hat{P} : \mathcal{S} \times \mathcal{S} \to [0, 1]$ that upper bounds the transition probability from state $\mathcal{S}_i$ to $\mathcal{S}_j$ from above. To accomplish this, we extend the SMC encoding presented in [15] to the case of stochastic dynamical systems considered.

Specifically, we consider the evolution $X^{t+1} = AX^t + Bu^t$, where X^t is the expectation of x^t. Notice that x^{t+1} is normally distributed, i.e., $x^{t+1} \sim \mathcal{N}(X^{t+1}, \delta^{t+1})$, for all $X^t \in \mathcal{X}$. Notice also that any given convex polytope $\mathcal{S}_i$ can be defined as the intersection of a finite number of hyperplanes $a_{i,\ell}^T x \leq b_{i,\ell}$, $\ell \in \mathcal{G}(\mathcal{S}_i)$, with $\mathcal{G}(\mathcal{S}_i)$ some arbitrary indexing. Following a procedure similar to the one in [21], we can compute the augmented set

$$\bar{\mathcal{S}}_i(q) = \{X | \min_{\ell \in \mathcal{G}(\mathcal{S}_i)} (P(V_{i,\ell}(x) \leq 0)) \geq q,\ x \sim (X, \delta)\}, \tag{4}$$

with $V_{i,\ell}(x) = a_{i,\ell} x - b_{i,\ell}, \ell \in \mathcal{G}(\mathcal{S}_i)$, which is convex and also its complement consists only of states X^{t+1} such that $P(x^{t+1} \in \mathcal{S}_i | x^{t+1} \sim \mathcal{N}(X^{t+1}, \delta^{t+1})) < q$. This last fact can be derived by noticing that $P(x^{t+1} \in \mathcal{S}_i) < \min_{\ell \in \mathcal{G}(\mathcal{S}_i)}(P(a_{i,\ell}^T x^{t+1} \leq b_{i,\ell}))$. Next, let b_i^l indicate the activation status of the i-th node in the l-th layer of the neural network controller f_{NN}, i.e., b_i^l is false when $h_i^l = 0$. Then, given a probability threshold $q \in [0, 1]$ and a pair of $\mathcal{S}_i, \mathcal{S}_j \in \mathcal{S}$, we can define the following SMC problem,

[1] We assume that the state space $\mathcal{X}$ consisting of the viable configurations (i.e., states where the robot does not overlap with the obstacles) is either given as or sufficiently approximated by a polytope.

Algorithm 1 Estimation of $\hat{P}(\mathcal{S}_i, \mathcal{S}_j)$

Input: $\mathcal{S}_i$, $\mathcal{S}_j$, dq
Output: $\hat{P}(\mathcal{S}_i, \mathcal{S}_j)$

$q_l \leftarrow 0$, $q_r \leftarrow 1$
while $q_r - q_l > dq$ **do**
  $q \leftarrow 0.5(q_l + q_l)$
  if SNN-SMC$(\mathcal{S}_i, \mathcal{S}_j, q)$ not satisfiable **then**
    $q_r \leftarrow q$
  else
    $q_l \leftarrow q$
return q_r

which we refer to as Stochastic Neural Network SMC (SNN-SMC):

$$\exists\ X^t, X^{t+1} \in \mathbb{R}^n, u^t \in \mathbb{R}^m, d \in \mathbb{R}^{2N} \quad (b^l, h^l, t^l) \in \mathbb{B}^{M_l} \times \mathbb{R}^{M_l} \times \mathbb{R}^{M_l} \tag{5}$$

subject to:

$$X^t \in \mathcal{S}_i \wedge X^{t+1} \in \bar{\mathcal{S}}_j(q) \wedge X^{t+1} = AX^t + Bu \tag{6}$$

$$\wedge \big(t^1 = W^0_\phi d(X^t) + w^0_\phi\big) \cap \big(\bigwedge_{l=2}^{L} t^l = W^{l-1}_\phi h^{l-1} + w^l_\phi\big) \tag{7}$$

$$\wedge \big(u^t = W^L_\phi h^L + w^L_\phi\big) \tag{8}$$

$$\wedge \bigwedge_{l=1}^{L}\bigwedge_{i=1}^{M_i} b^l_i \rightarrow \big[\big(h^l_i = t^l_i\big) \wedge \big(t^l_i \geq 0\big)\big] \tag{9}$$

$$\wedge \bigwedge_{l=1}^{L}\bigwedge_{i=1}^{M_i} \neg b^l_i \rightarrow \big[\big(h^l_i = 0\big) \wedge \big(t^l_i < 0\big)\big]. \tag{10}$$

In the above definition, (6) encodes the transition from state $\mathcal{S}_i$ to $\mathcal{S}_j$ and (7)-(10) encode the behavior imposed by the neural network controller.

Using this SNN-SMC encoding, we now present our proposed algorithm to compute the upper bounds $\hat{P}$ in Definition 1 on the underlying transition probabilities. Specifically, observe that for a given pair of cells $\mathcal{S}_i, \mathcal{S}_j$ if the selected threshold q is large, then there may not be x^t such that $P(x^{t+1} \in \mathcal{S}_j | x^t \in \mathcal{S}_i) \geq q$, which would render the SNN-SMC problem unsatisfiable. Therefore, any such threshold q is a valid upper bound on the transition probability from $\mathcal{S}_i$ to $\mathcal{S}_j$, i.e., $P(\mathcal{S}_i, \mathcal{S}_j) \leq q$. Based on this fact, we propose an iterative algorithm outlined in Algorithm 1, which, given a user-specified precision $dq \in (0, 1)$, employs binary search for finding q such that the SNN-SMC problem for $\mathcal{S}_i, \mathcal{S}_j$ can no longer be satisfied. By executing Algorithm 1 for every pair of cells $\mathcal{S}_i$ and $\mathcal{S}_j$ in the partition $\mathcal{S}$, we can obtain the desired function $\hat{P}$.

IV. Safety Probability Bounds

In this section, we elaborate on how to compute upper bounds on the safety probability P_k given a valid transition graph $\mathcal{D}$. First, we extend the definition of a (p, k)-safe state x to a (p, k)-safe cell. Specifically, we say that a cell $\mathcal{S}_i$ is (p, k)-safe if all the states in $\mathcal{S}_i$ are (p, k)-safe. Let $\hat{P}_k : \mathcal{S} \mapsto [0, 1]$ denote any function that bounds the safety probability P_k from above, i.e., $\hat{P}_k(\mathcal{S}_i) \geq \max_{x \in \mathcal{S}_i}(P_k(x))$ for all $\mathcal{S}_i \in \mathcal{S}$. Then, it is simple to see that any cell $\mathcal{S}_i \in \mathcal{S}$ is (p, k)-safe if $\hat{P}_k(\mathcal{S}_i) \leq p$. Let $\mathcal{N}_{\mathcal{S}_i} = \{\mathcal{S}_j | (\mathcal{S}_i, \mathcal{S}_j) \in \mathcal{E}\}$ denote the set of nodes $\mathcal{S}_j$ reachable from $\mathcal{S}_i$. Next, we discuss how to compute tight safety probability bounds $\hat{P}_k$ for all cells in the partition $\mathcal{S}$. We begin by presenting the following proposition which provides a straightforward method to compute $\hat{P}_{k+1}$ given $\hat{P}_k$.

Proposition 1. *Assume that the transition and safety probability bounds $\hat{P}$ and $\hat{P}_k$ are known. Then, for any $\mathcal{S}_i \in \mathcal{S}$ the following holds:*

$$P_{k+1}(x) \leq \sum_{\mathcal{S}_j \in \mathcal{N}_{\mathcal{S}_i}} \hat{P}_k(\mathcal{S}_i)\hat{P}(\mathcal{S}_i, \mathcal{S}_j), \quad \forall x \in \mathcal{S}_i \tag{11}$$

Proof. Let $x^t \in \mathcal{S}_i$. Then, by definition, it holds that $P_{k+1}(x^t) = P(x^{t+k+1} \in \mathcal{W}_o | x^t \in \mathcal{S}_i) = \sum_{\mathcal{S}_j \in \mathcal{N}_{\mathcal{S}_i}} P(x^{t+k+1} \in \mathcal{W}_o | x^{t+1} \in \mathcal{S}_i) P(x^{t+1} \in \mathcal{S}_i | x^t \in \mathcal{S}_j) \leq \sum_{\mathcal{S}_j \in \mathcal{N}_{\mathcal{S}_i}} \hat{P}_k(\mathcal{S}_j)\hat{P}(\mathcal{S}_j, \mathcal{S}_i)$. □

We remark that a valid choice of $\hat{P}_0$ is to let $\hat{P}_0(\mathcal{S}_i) = 1$ if there exists $x \in \mathcal{S}_i$ such that $P_W(x) \in \mathcal{W}_o$ and $\hat{P}_0(\mathcal{S}_i) = 0$ otherwise. Using this choice for $\hat{P}_0$, we can recursively compute safety probability $\hat{P}_k(\mathcal{S}_i)$ for every cell $\mathcal{S}_i \in \mathcal{S}$ as

$$\hat{P}_{k+1}(\mathcal{S}_i) = \hat{P}'_{k+1}(\mathcal{S}_i) \triangleq \sum_{\mathcal{S}_j \in \mathcal{N}_{\mathcal{S}_i}} \hat{P}_k(\mathcal{S}_i)\hat{P}(\mathcal{S}_i, \mathcal{S}_j) \tag{12}$$

until the desired horizon $k = T$ is reached. However, note that (12) is expected to furnish loose bounds on the safety probabilities when the estimations of the underlying transition probabilities are not tight or the partition is coarse. The reason for the latter case is that Algorithm 1 computes a worst case upper bound on the transition probability from cell $\mathcal{S}_i$ to cell $\mathcal{S}_j$ that is close to the transition probability from the worst case state $X^\star = \text{argmax}_{X^t \in \mathcal{S}_i} P(x^{t+1} \in \mathcal{S}_j | X^t))$ with $x^{t+1} \sim \mathcal{N}(X^{t+1}, \delta^{t+1})$ and $X^{t+1} = AX^t + Bf_{\text{NN}}(d^t(X^t))$. Note that if the partition is coarse, it is likely that there are many other states in $\mathcal{S}_i$ that have much lower transition probabilities to $\mathcal{S}_j$ but are effectively treated the same as $X^\star$. If the partition is finer, many of these states can be grouped in a different cell with lower transition probability to the cell $\mathcal{S}_j$. We discuss a way to refine the partition $\mathcal{S}$ in Section V. But first we describe how to improve the safety probability bounds $\hat{P}_k$ in (12) for a given partition $\mathcal{S}$.

A. Partition Merging

In this section, we present a method to merge nodes in $\mathcal{S}$ in order to improve the transition probability bounds used in (12). Specifically, we provide the following result.

Proposition 2. *Let $\mathcal{S}_o \in \mathcal{S}$, $\mathcal{S}_i, \mathcal{S}_j \in \mathcal{N}_{\mathcal{S}_o}$ and $p \in (0, 1)$. Consider the set $\mathcal{S}'_{ij} = \mathcal{S}_i \cup \mathcal{S}_j$. If $\bar{\mathcal{S}}_i(p) \cap \bar{\mathcal{S}}_j(p) = \emptyset$, then $\mathcal{D}' = (\mathcal{S}', \mathcal{E}', \hat{P}')$ with $\mathcal{S}' = \mathcal{S} \cup \{\mathcal{S}'_{ij}\}$, $\mathcal{E}' = \mathcal{E} \bigcup \{(\mathcal{S}_o, \mathcal{S}'_{ij})\} \setminus \{(\mathcal{S}_o, \mathcal{S}_i), (\mathcal{S}_o, \mathcal{S}_j)\}$, $\hat{P}'(\mathcal{S}_o, \mathcal{S}'_{ij}) = \max(\max(\hat{P}(\mathcal{S}_o, \mathcal{S}_i), \hat{P}(\mathcal{S}_o, \mathcal{S}_j)) + p, 2p)$ and $\hat{P}'(\mathcal{S}_o, \mathcal{S}_i) = \hat{P}'(\mathcal{S}_o, \mathcal{S}_j) = 0$ is a valid transition graph.*

Proof. Let $q_i, q_j \in (0,1)$. Recall that the complement of $\bar{\mathcal{S}}_i(q_i)$ (resp. $\bar{\mathcal{S}}_j(q_j)$) consists only of states $X \in \mathcal{X}$ such that $P(x \in \mathcal{S}_i) < q_i$ (resp. $P(x \in \mathcal{S}_j) < q_j$) for $x \sim \mathcal{N}(X, \delta)$. Let X^t be a state in $\mathcal{S}_o$. Assuming $\bar{\mathcal{S}}_i(q_i) \cap \bar{\mathcal{S}}_j(q_j) = \emptyset$, then X^{t+1} must lie either in $\bar{\mathcal{S}}_i(q_i)$, $\bar{\mathcal{S}}_j(q_j)$, or the complement of their union. As such, the following inequalities hold:

$$P(x^{t+1} \in \mathcal{S}'_{ij}) < \begin{cases} \hat{P}(\mathcal{S}_o, \mathcal{S}_i) + q_j, & \text{if } X^{t+1} \in \bar{\mathcal{S}}_i(q_i) \\ \hat{P}(\mathcal{S}_o, \mathcal{S}_j) + q_i, & \text{if } X^{t+1} \in \bar{\mathcal{S}}_j(q_j) \\ q_i + q_j, & \text{if } X^{t+1} \in \mathcal{X} \setminus \{\bar{\mathcal{S}}_j, \bar{\mathcal{S}}_j\}. \end{cases} \tag{13}$$

Setting $q_i = q_j = p$ and choosing the worst of these cases concludes the proof. □

Proposition 2 introduces a new node $\mathcal{S}'_{ij}$ to the transition graph $\mathcal{D}$ that is the result of "merging" cells $\mathcal{S}_i, \mathcal{S}_j \in \mathcal{N}_{\mathcal{S}_o}$ which are far enough from each other so that their augmented sets $\bar{\mathcal{S}}_i(p)$ and $\bar{\mathcal{S}}_j(p)$ do not overlap, where $p \in (0,1)$ is a user specified probability threshold. The algorithm is illustrated in Algorithm 2. Notice that neither $\mathcal{S}_i$ nor $\mathcal{S}_j$ are removed from $\mathcal{D}$; what is removed is their edges with $\mathcal{S}_o$. Consequently, by repeatedly merging cells in the graph $\mathcal{D}$ until there are no more cells that can be merged, one can obtain a new graph $\mathcal{D}'$ which has more nodes than $\mathcal{D}$ but is not fully connected, i.e., $|\mathcal{N}'_{\mathcal{S}_o}| \leq |\mathcal{N}_{\mathcal{S}_o}|$ for all $\mathcal{S}_o \in \mathcal{S} \subseteq \mathcal{S}'$. Note also that $\max_{X^t \in \mathcal{S}_o} P(x^{t+1} \in \mathcal{S}_i \cup \mathcal{S}_j | X^t)$ is bounded from above by $\max_{X_i^t \in \mathcal{S}_o} P(x_i^{t+1} \in \mathcal{S}_i | X_i^t) + \max_{x_j^t \in \mathcal{S}_o} P(x_j^{t+1} \in \mathcal{S}_j | X_j^t)$ for all cells $\mathcal{S}_i, \mathcal{S}_j \in \mathcal{N}_{\mathcal{S}_o}$[2]. In addition, Proposition 2 informs us that $\hat{P}(\mathcal{S}_o, \mathcal{S}'_{ij}) \ll \hat{P}(\mathcal{S}_o, \mathcal{S}_i) + \hat{P}(\mathcal{S}_o, \mathcal{S}_j)$ for those cells $\mathcal{S}_i, \mathcal{S}_j$ which are reachable from $\mathcal{S}_o$ with high transition probabilities but are far away from each other, given small probability threshold p, because there exists no single state X^t in $\mathcal{S}_o$ which is as likely to transition to $\mathcal{S}_i$ as to $\mathcal{S}_j$ (see Figure 1). Finally, in order to guarantee that the new bounds we obtain on the safety probabilities P_k are tighter than the original ones, two cells $\mathcal{S}_i$ and $\mathcal{S}_j$ are merged only if the following condition holds:

$$P(\mathcal{S}_o, \mathcal{S}'_{ij}) \max(\hat{P}_k(\mathcal{S}_i), \hat{P}_k(\mathcal{S}_j)) < P(\mathcal{S}_o, \mathcal{S}_i)\hat{P}_k(\mathcal{S}_i) + P(\mathcal{S}_o, \mathcal{S}_j)\hat{P}_k(\mathcal{S}_j). \tag{14}$$

Particularly, this condition ensures that the terms of (12) effectively removed by Algorithm 2 get replaced by strictly lesser ones.

B. Transition Probability Normalization

Given the transition graph $\mathcal{D}$ constructed in subsection IV-A, next we present an alternative way to recursively compute tight bounds $\hat{P}_k$ on the safety probabilities when the underlying transition probability bounds are over-approximated. Specifically, we provide the following result, which is similar to the one derived in [16], to truncate the sum in (12) while ensuring that the new estimation remains a valid upper bound of P_k.

[2] We recall that x^{t+1} is a random variable sampled from the normal distribution $\mathcal{N}(X^{t+1}, \delta^{t+1})$ and the probability of x^{t+1} lying in a set $\mathcal{S}_o$ is obtained by integrating the corresponding density function over $\mathcal{S}_o$.

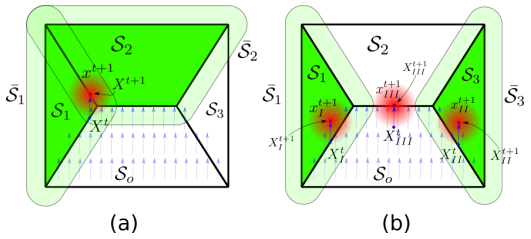

Fig. 1. Example of mergeable and not mergeable pairs of cells according to Proposition 2. The probability density function of each x^{t+1} is depicted using red-colored gradient. a) Cells $\mathcal{S}_1$ and $\mathcal{S}_2$ are not mergeable because the intersection of their augmented sets $\bar{\mathcal{S}}_1$ and $\bar{\mathcal{S}}_2$ is not empty and there exists $X^t \in \mathcal{S}_o$ whose X^{t+1} lies in $\bar{\mathcal{S}}_1 \cap \bar{\mathcal{S}}_2$. As such, it is not possible to deduce a tight bound on the probability $P(\mathcal{S}_1 \cup \mathcal{S}_2 | \mathcal{S}_o)$ based on readily available bounds on $P(\mathcal{S}_1|\mathcal{S}_o)$ and $P(\mathcal{S}_2|\mathcal{S}_o)$; in this particular case, $P(\mathcal{S}_1 \cup \mathcal{S}_2|\mathcal{S}_o) \approx 1$. b) Cells $\mathcal{S}_1$ and $\mathcal{S}_3$ can be merged and a valid bound on the transition probability $P(\mathcal{S}_1 \cup \mathcal{S}_3|\mathcal{S}_o)$ is approximately equal to $\max(P(\mathcal{S}_1|\mathcal{S}_o), P(\mathcal{S}_3|\mathcal{S}_o))$, since there is no $X^t \in \mathcal{S}_o$ with a good probability of landing in $\mathcal{S}_1$ and a good probability of landing in $\mathcal{S}_2$.

Algorithm 2 Partition Merging

Input: $\mathcal{D} = (\mathcal{S}, \mathcal{E}, \hat{P}), \mathcal{S}_o, p, \hat{P}_k$
Output: $\mathcal{D}' = (\mathcal{S}', \mathcal{E}', \hat{P}'), \hat{P}'_k$

$\mathcal{S}' \leftarrow \mathcal{S}$, $\mathcal{E}' \leftarrow \mathcal{E}$, $\hat{P}'_k \leftarrow \hat{P}_k$
for $\mathcal{S}_i$ in $\mathcal{S}$ **do**
  for $\mathcal{S}_j$ in $\mathcal{S} \setminus \{\mathcal{S}_i\}$ **do**
    if $\bar{\mathcal{S}}_i(p) \cap \bar{\mathcal{S}}_j(p)$ **then**
      $\mathcal{S}'_{ij} \leftarrow \mathcal{S}_i \cup \mathcal{S}_j$
      $\mathcal{P}'(\mathcal{S}_o, \mathcal{S}'_{ij}) \leftarrow \max(\hat{P}(\mathcal{S}_o, \mathcal{S}_i), \hat{P}(\mathcal{S}_o, \mathcal{S}_j), p) + p$
      if (14) holds **then**
        $\mathcal{S}' \leftarrow \mathcal{S}' \cup \{\mathcal{S}'_{ij}\}$
        $\mathcal{E}' \leftarrow (\mathcal{E}' \setminus \{(\mathcal{S}_o, \mathcal{S}_i), (\mathcal{S}_o, \mathcal{S}_j)\}) \cup (\mathcal{S}_o, \mathcal{S}'_{ij})$

Proposition 3. *Let $\mathcal{S}_o \in \mathcal{S}$ and $\hat{\kappa}_o : \mathbb{N} \mapsto \mathbb{N}$ such that $\hat{P}_k(\mathcal{S}_{\hat{\kappa}_o}(i)) \leq \hat{P}_k(\mathcal{S}_{\hat{\kappa}_o}(j))$, for all $i \leq j$ with $\mathcal{S}_{\hat{\kappa}_o(i)}, \mathcal{S}_{\hat{\kappa}_o(j)} \in \mathcal{N}_{\mathcal{S}_o}$. Also, let $n = |\mathcal{N}_{\mathcal{S}_o}|$ and $\hat{m}$ such that $\sum_{i=\hat{m}+1}^{n} \hat{P}(\mathcal{S}_o, \mathcal{S}_{\hat{\kappa}_o(i)}) \leq 1$. Then, $P_{k+1}(x) \leq \hat{P}''_{k+1}(x)$ for all $x \in \mathcal{S}_o$, where*

$$\hat{P}''_{k+1}(x) = \sum_{i=\hat{m}+1}^{n} \hat{P}(\mathcal{S}_o, \mathcal{S}_{\hat{\kappa}_o(i)}) \hat{P}_k(\mathcal{S}_{\hat{\kappa}_o(i)}) + \left(1 - \sum_{i=\hat{m}+1}^{n} \hat{P}(\mathcal{S}_o, \mathcal{S}_{\hat{\kappa}_o(i)})\right) \hat{P}_k(\mathcal{S}_{\hat{\kappa}_o(\hat{m})}), \tag{15}$$

Additionally, if $\hat{m}$ such that $\sum_{i=\hat{m}}^{n} \hat{P}(\mathcal{S}_o, \mathcal{S}_{\hat{\kappa}_o(i)}) > 1$, then $\hat{P}''_{k+1}(x) < \hat{P}'_{k+1}(x)$ for all $x \in \mathcal{S}_o$.

The proof of Proposition 3 can be found in [22]. Proposition 3 provides an alternative formula to (12) to recursively estimate the safety probability bounds $\hat{P}_k$ that uses the $n - \hat{m}$ cells with the largest estimated safety probabilities $\hat{P}_k$ in order to obtain a tighter bound on the safety probability by mitigating the over-approximation of the transition probability bounds.

C. Verification Framework

Given a graph $\mathcal{D}$, upper bounds on the transition and safety probabilities and a user specified horizon T, we now

Algorithm 3 Verifcation Framework

Input: $\mathcal{D}$, $\hat{P}_0$, p
Output: $\hat{P}_T$

1: **for** i in $1, 2, \ldots, T$ **do**
2: $\quad \mathcal{D}' \leftarrow \mathcal{D}$
3: $\quad$ **repeat**
4: $\quad\quad \mathcal{D}'' \leftarrow \mathcal{D}'$
5: $\quad\quad$ **for** j in $1, 2, \ldots, |\mathcal{S}'|$ **do**
6: $\quad\quad\quad \mathcal{D}'', \hat{P}_{i-1} \leftarrow \text{MergeCells}(\mathcal{D}'', \mathcal{S}'_j, p, \hat{P}_{i-1})$
7: $\quad\quad$ flag $\leftarrow \mathcal{D}'' = \mathcal{D}'$, $\mathcal{D}' \leftarrow \mathcal{D}''$
8: $\quad$ **until not** flag
9: $\quad$ **for** j in $1, 2, \ldots, |\mathcal{S}|$ **do**
10: $\quad\quad \hat{P}_i(\mathcal{S}'_j) \leftarrow \text{PropagateSafetyProb}(\mathcal{S}_j, \mathcal{N}'_{\mathcal{S}_j}, \hat{P}_{i-1})$

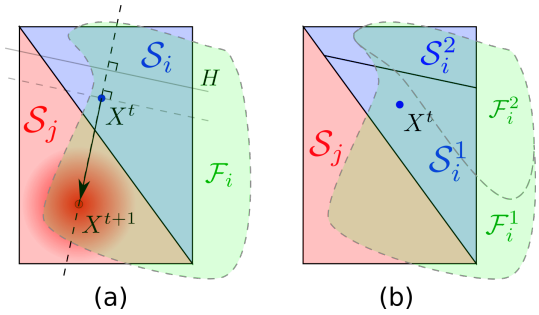

Fig. 2. Subdivision of cell $\mathcal{S}_i$ w.r.t. the transition probability $P(\mathcal{S}_i, \mathcal{S}_j)$ into two subcells $\mathcal{S}_i^1$ and $\mathcal{S}_i^2$. The forward reachable sets $\mathcal{F}_i, \mathcal{F}_i^1, \mathcal{F}_i^2$ of cells $\mathcal{S}_i, \mathcal{S}_i^1, \mathcal{S}_i^2$ under the dynamics of the closed-loop system $X^{t+1} = AX^t + Bf_{\text{NN}}(d^t(X^t))$. are depicted in green, respectively. We notice that the selection of hyperplane H minimizes the probability of a state $X^t \in \mathcal{S}_i^2$ to land in $\mathcal{S}_j$, by placing a sufficiently large neighborhood of $X^t \in \mathcal{X}_{ij}^\star$ strictly inside $X^t \in \mathcal{S}_i^1$.

present an algorithm to compute $\hat{P}_k$, which is illustrated in Algorithm 3. Let $p \in (0, 1)$ denote a desired probability threshold. Then, for each $i \in 1, 2, \ldots, T$, firstly we apply Algorithm 2 iteratively until no cells remain in the new graph $\mathcal{D}'$ which can be merged (lines 2-8). Finally, for each cell $\mathcal{S}_j$ of the original graph $\mathcal{D}$, we compute the safety probability bound $\hat{P}_i(\mathcal{S}_j)$ using (15) (lines 9-10). As such, by virtue of Proposition 2 and Proposition 3, the bounds $\hat{P}_T$ obtained using Algorithm 3 are valid upper bounds on the safety probability and are guaranteed to be at least as tight as the ones obtained by applying (12) on $\mathcal{D}$. We validate this result in Section VI.

V. SNN-SMC Based Refinement

In this section, we present a method to refine the cells of a given partition $\mathcal{S}$ in order to obtain tighter upper bounds on the safety probabilities $\hat{P}_k$ compared to those obtained for a coarser initial partition. We begin by presenting the following proposition which provides bounds on the transition probabilities of the cells $\mathcal{S}_i^1, \mathcal{S}_i^2$ obtained by cutting a cell $\mathcal{S}_i \in \mathcal{S}$ into two disjoint cells separated by a hyperplane H.

Proposition 4. *Given the transition graph $\mathcal{D}$ and safety probabilities $\hat{P}_k$, let H be a hyperplane splitting $\mathcal{S}_i \in \mathcal{S}$ into $\mathcal{S}_i^1$ and $\mathcal{S}_i^2$. For all $\mathcal{S}_j \in \mathcal{N}_{\mathcal{S}_i}$, if $\hat{P}(\mathcal{S}_i^1, \mathcal{S}_j) = \hat{P}(\mathcal{S}_i, \mathcal{S}_j)$, then $\hat{P}(\mathcal{S}_i^2, \mathcal{S}_j) \leq \hat{P}(\mathcal{S}_i, \mathcal{S}_j)$. Also, the transition graph $\mathcal{D}' = (\mathcal{S}', \mathcal{E}', \hat{P}')$ is valid, where $\mathcal{S}'$ and $\mathcal{E}'$ are obtained by replacing the cell $\mathcal{S}_i$ with $\mathcal{S}_i^1$ and $\mathcal{S}_i^2$. Moreover, $\hat{P}_{k+1}(\mathcal{S}_i^l) \leq \hat{P}_{k+1}(\mathcal{S}_i)$ for all $l \in \{1, 2\}$.*

In words, given $\mathcal{S}_j \in \mathcal{N}_{\mathcal{S}_i}$, the transition probability bound $\hat{P}(\mathcal{S}_i^\ell, \mathcal{S}_j)$ computed using Algorithm 1 is the same as $\hat{P}(\mathcal{S}_i, \mathcal{S}_j)$ for at least one $\ell \in \{1, 2\}$. The reason is that satisfiability of the SNN-SMC problem in Algorithm 1 implies that there exists at least one $X^t \in \mathcal{S}_i = \mathcal{S}_i^1 \cup \mathcal{S}_i^2$ that marginally satisfies the inequality $P(x^{t+1} \in \mathcal{S}_j | X^t) \geq \hat{P}(\mathcal{S}_i, \mathcal{S}_j) - dq$. Let $\mathcal{X}_{ij}^\star$ denote the set of all these X^t. Assuming that $\mathcal{X}_{ij}^\star$ lies strictly in the interior of $\mathcal{S}_i$, it can be placed in one of the subcells (e.g., $\mathcal{S}_i^1$) through a proper selection of the hyperplane H so that the transition probability of the other subcell (e.g., $\hat{P}(\mathcal{S}_i^2, \mathcal{S}_j)$) becomes strictly less than $\hat{P}(\mathcal{S}_i, \mathcal{S}_j)$ (see Figure 2). To find this hyperplane, we consider a state $X^t \in \mathcal{X}_{ij}^\star$ that marginally satisfies the SNN-SMC problem for $\mathcal{S}_i, \mathcal{S}_j$ and define the hyperplane H that is perpendicular to the line connecting X^t, X^{t+1} and contains X^t, where $X^{t+1} = AX^t + Bf_{\text{NN}}(d(X^t))$. Assuming that all $X^t \in \mathcal{X}_{ij}^\star$ lie on the same half-space defined by H, translating this hyperplane away from $\mathcal{X}_{ij}^\star$ will decrease the transition probability bound from one of the new cells $\mathcal{S}_i^1$ or $\mathcal{S}_j^2$ to $\mathcal{S}_j$. Thus, given a pair of cells $\mathcal{S}_i \in \mathcal{S}$ and $\mathcal{S}_j \in \mathcal{N}_{\mathcal{S}_i}$, we can refine the partition S by translating the hyperplane H so that either $\hat{P}(\mathcal{S}_i^1, \mathcal{S}_j)$ or $\hat{P}(\mathcal{S}_i^2, \mathcal{S}_j)$ is minimized.

Remark 1. *The proposed cell merging and refinement schemes discussed in Sections III, IV and V can also be easily extended to deal with agents with nonlinear dynamics. Specifically, we can compute the reachable regions of nonlinear transition functions using methods, e.g., in [23] and build the transition graph in a similar way as in the SNN-SMC approach. Then, the approaches in Section IV and V can be directly applied.*

VI. Numerical Experiments

In this section, we present simulation results to validate the proposed bounds on the probability that a point-sized robot collides with the boundary of the non-convex planar workspace $\mathcal{W}$. Particularly, we consider a scenario similar to the one in [15] and assume that the robot's dynamics can be modeled as a single-integrator, i.e.:

$$x_{t+1} = x_t + f_{NN}(d(x_t)) + w^t, \tag{16}$$

where $x^t \in \mathbb{R}^2$ denotes the robot's position at time step t and $w^t \sim \mathcal{N}(0, 3)$. Additionally, we assume that the robot is equipped with a LiDAR scanner that emits a set of q lasers evenly distributed in a 2π rad fan, i.e., $d(x^t) = [(d_0(x^t))^T, (d_1(x^t))^T, ..., (d_q(x^t))^T]^T$, where $d_i(x^t) = [r_i(x^t)\cos\theta_i, r_i(x^t)\sin\theta_i]^T$ and $r_i(x^t)$ denotes the distance measured between the robot and the closest obstacle in the direction $[\cos(\theta_i), \sin(\theta_i)]$, for all $i \in \{1, 2, \ldots q\}$. To drive the robotic system to a predetermined goal position using only the feedback $d(x^t)$, we employed a ReLU neural network controller f_{NN} consisting of three hidden layers and a total of 32 neurons. Lastly, we used the partitioning method

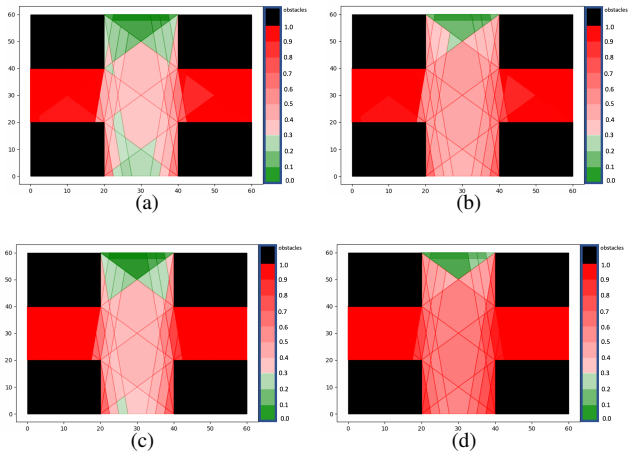

Fig. 3. Safety probability bound $\hat{P}_T$ for various horizons computed using Algorithm 3 ((a), (b)) and the algorithm proposed in [16] ((c), (d)). Left column corresponds to $T = 6$ and right column to $T = 9$.

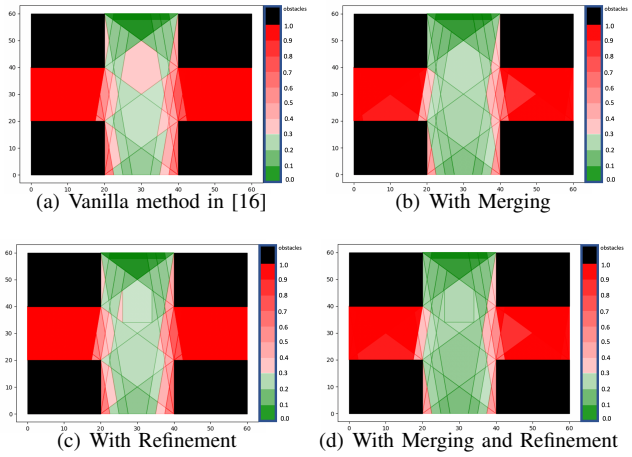

Fig. 4. Safety probability bound $\hat{P}_6$ computed using (12) (left column) and Algorithm 3 for horizon $T = 7$ on the original and refined graphs, respectively.

proposed in [15] to partition the domain $\mathcal{X}$, which in this scenario coincides with the workspace $\mathcal{W}$.

In Figure 3 we provide a comparison between the bound $\hat{P}_T$ computed over the given domain using Algorithm 3 and the method proposed in [16], for various horizons T. We observe that our method provides tighter bounds compared to [16] for every horizon, even for the coarse partition considered in this scenario. Additionally, in Figure 4, we show the bounds on the safety probability P_6 computed over this domain using (12) and Algorithm 3 on the original and refined transition graphs, respectively. The refined graph was obtained using the heuristic subdivision scheme presented in Section V to the largest cell located near the center of the workspace. It can be seen in Figure 4 that the safety probability bounds estimated using merging and/or refinement are noticeably tighter than the ones without. Finally, we certify the correctness of the proposed safety probability bounds by comparing them to the true safety probability of for the cell that is adjacent to the one subject to the refinement. To estimate the true safety probability, we simulate a sufficiently large number ($\approx$ 10000 of robot trajectories starting from

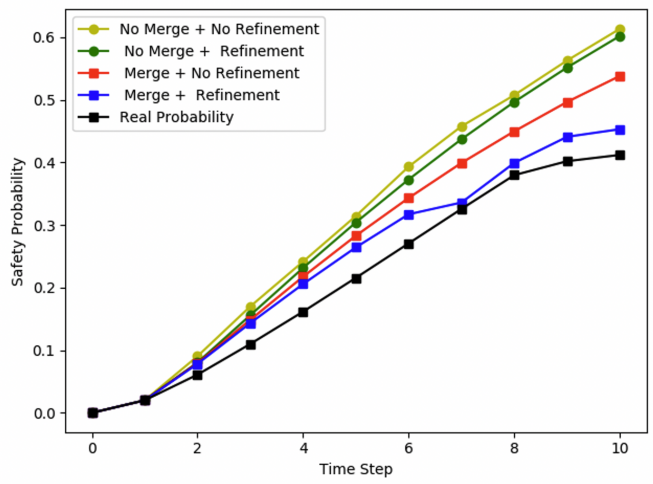

Fig. 5. Bound on and actual safety probability P_k of a given cell $\mathcal{S}_i$ for different horizons.

states within that cell and compute the percentage of those that end up violating the safety requirements as a result of the applied disturbances. In Figure 5 we present the safety probability bounds returned by our method for that given cell for different horizons and compare these bounds to the estimated true safety probability. We observe that all bounds returned by our method correctly upper bound the true probability, while the bounds obtained by using both merging and refinement are the tightest. We also remark that the gap between the estimated bounds and the true safety probability becomes larger as the horizon increases.

VII. Conclusions

In this work, we addressed the problem of formal safety verification of stochastic cyber-physical systems (CPS) equipped with a ReLU neural network (NN) controllers. Particularly, we presented a method to compute sets of initial states which that ensure that the system trajectories are safe within a specified horizon. To do this, we designed a suitable discrete abstraction of the system and formulated an SMC problem to estimate upper bounds on the transition probabilities between cells in this discrete abstraction. Additionally, we proposed a method to obtain tighter bounds on the corresponding safety probability as well as a heuristic for refining the abstraction in a way that may further improve the results. Finally, we presented simulation results verifying the efficacy of our method compared to existing methodologies proposed in the literature.

References

[1] Amini, A., Rosman, G., Karaman, S., & Rus, D. (2019, May). Variational end-to-end navigation and localization. In 2019 International Conference on Robotics and Automation (ICRA) (pp. 8958-8964). IEEE.

[2] Shamsfakhr, F., & Bigham, B. S. (2017). A neural network approach to navigation of a mobile robot and obstacle avoidance in dynamic and unknown environments. Turkish journal of electrical engineering & computer sciences, 25(3), 1629-1642.

[3] Nägeli, T., Alonso-Mora, J., Domahidi, A., Rus, D., & Hilliges, O. (2017). Real-time motion planning for aerial videography with dynamic obstacle avoidance and viewpoint optimization. IEEE Robotics and Automation Letters, 2(3), 1696-1703.

[4] Chen, Y. F., Everett, M., Liu, M., & How, J. P. (2017, September). Socially aware motion planning with deep reinforcement learning. In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 1343-1350). IEEE.

[5] Schmuck, V., & Meredith, D. (2019). Training networks separately on static and dynamic obstacles improves collision avoidance during indoor robot navigation. In European Symposium on Artificial Neural Networks, Computational Intelligence and Machine LearningEuropean Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (pp. 655-660). Ciaco-i6doc. com.

[6] Xiang, W., Musau, P., Wild, A. A., Lopez, D. M., Hamilton, N., Yang, X., ... & Johnson, T. T. (2018). Verification for machine learning, autonomy, and neural networks survey. arXiv preprint arXiv:1810.01989.

[7] Charikar, M., Steinhardt, J., & Valiant, G. (2017, June). Learning from untrusted data. In Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing (pp. 47-60).

[8] Li, B., Chen, C., Wang, W., & Carin, L. (2019). Certified Adversarial Robustness with Additive Noise. In Advances in Neural Information Processing Systems (pp. 9459-9469).

[9] Fazlyab, M., Morari, M., & Pappas, G. J. (2019). Probabilistic Verification and Reachability Analysis of Neural Networks via Semidefinite Programming. arXiv preprint arXiv:1910.04249.

[10] Fazlyab, M., Morari, M., & Pappas, G. J. (2019). Safety verification and robustness analysis of neural networks via quadratic constraints and semidefinite programming. arXiv preprint arXiv:1903.01287.

[11] Katz, G., Barrett, C., Dill, D. L., Julian, K., & Kochenderfer, M. J. (2017, July). Reluplex: An efficient SMT solver for verifying deep neural networks. In International Conference on Computer Aided Verification (pp. 97-117). Springer, Cham.

[12] Ivanov, R., Weimer, J., Alur, R., Pappas, G. J., & Lee, I. (2019, April). Verisig: verifying safety properties of hybrid systems with neural network controllers. In Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control (pp. 169-178).

[13] Ivanov, R., Carpenter, T. J., Weimer, J., Alur, R., Pappas, G. J., & Lee, I. (2019). Case Study: Verifying the Safety of an Autonomous Racing Car with a Neural Network Controller. arXiv preprint arXiv:1910.11309. controllers

[14] Shoukry, Y., Nuzzo, P., Sangiovanni-Vincentelli, A. L., Seshia, S. A., Pappas, G. J., & Tabuada, P. (2017, April). SMC: Satisfiability modulo convex optimization. In Proceedings of the 20th International Conference on Hybrid Systems: Computation and Control (pp. 19-28).

[15] Sun X, Khedr H, Shoukry Y. Formal verification of neural network controlled autonomous systems, Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control. 2019: 147-156.

[16] Lahijanian, M., Andersson, S. B., & Belta, C. (2015). Formal verification and synthesis for discrete-time stochastic systems. IEEE Transactions on Automatic Control, 60(8), 2031-2045.

[17] Dutreix, M., & Coogan, S. (2018, April). Efficient verification for stochastic mixed monotone systems. In 2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS) (pp. 150-161). IEEE.

[18] Dutreix M, Coogan S. Specification-guided verification and abstraction refinement of mixed monotone stochastic systems[J]. IEEE Transactions on Automatic Control, 2020.

[19] Zarei, M., Wang, Y., & Pajic, M. (2020, April). Statistical verification of learning-based cyber-physical systems. In Proceedings of the 23rd International Conference on Hybrid Systems: Computation and Control (pp. 1-7).

[20] Wang, Y., Zarei, M., Bonakdarpour, B., & Pajic, M. (2019). Statistical verification of hyperproperties for cyber-physical systems. ACM Transactions on Embedded Computing Systems (TECS), 18(5s), 1-23.

[21] Blackmore L, Ono M, Williams B C. Chance-constrained optimal path planning with obstacles[J]. IEEE Transactions on Robotics, 2011, 27(6): 1080-1094.

[22] Sun, S., Zhang, Y., Luo, X., Vlantis, P., Pajic, M., & Zavlanos, M. M. (2021). Formal Verification of Stochastic Systems with ReLU Neural Network Controllers. arXiv preprint arXiv:2103.05142.

[23] Xiang, W., Tran, H. D., & Johnson, T. T. (2018). Output reachable set estimation and verification for multilayer neural networks. IEEE transactions on neural networks and learning systems, 29(11), 5777-5783.