

# Learning Vision-Based Neural Network Controllers with Semi-Probabilistic Safety Guarantees

Xinhang Ma<sup>1</sup>, Junlin Wu<sup>1</sup>, Hussein Sibai<sup>1</sup>, Yiannis Kantaros<sup>1</sup>, Yevgeniy Vorobeychik<sup>1</sup>

<sup>1</sup>Washington University in St. Louis  
{m.owen, junlin.wu, ioannisk, sibai, yvorobeychik}@wustl.edu

## Abstract

Ensuring safety in autonomous systems with vision-based control remains a critical challenge due to the high dimensionality of image inputs and the fact that the relationship between true system state and its visual manifestation is unknown. Existing methods for learning-based control in such settings typically lack formal safety guarantees. To address this challenge, we introduce a novel semi-probabilistic verification framework that integrates reachability analysis with conditional generative networks and distribution-free tail bounds to enable efficient and scalable verification of vision-based neural network controllers. Next, we develop a gradient-based training approach that employs a novel safety loss function, safety-aware data-sampling strategy to efficiently select and store critical training examples, and curriculum learning, to efficiently synthesize safe controllers in the semi-probabilistic framework. Empirical evaluations in X-Plane 11 airplane landing simulation, CARLA-simulated autonomous lane following, F1Tenth vehicle lane following in a physical visually-rich miniature environment, and Airsim-simulated drone navigation and obstacle avoidance demonstrate the effectiveness of our method in achieving formal safety guarantees while maintaining strong nominal performance.

**Code** — <https://github.com/xhOwenMa/SPVT>

**Extended version** — <https://arxiv.org/abs/2503.00191>

## 1 Introduction

Many real-world applications, such as self-driving cars and robotic navigation, require controllers that process high-dimensional image inputs to make real-time decisions. The centrality of visual inputs (particularly when other modalities are limited or unreliable) thus makes ensuring the safety of vision-based control an important problem in trustworthy AI. However, verifying the safety of such controllers remains a major open challenge due to the complexity of image-based inputs and the high computational cost of traditional verification methods (Huang et al. 2019; Katz et al. 2017; Huang et al. 2017).

While reinforcement learning (RL) with high-dimensional image inputs has shown promise in learning

control policies that optimize performance (Le et al. 2022; Kiran et al. 2021), most methods lack *formal guarantees* of safety (Kochdumper et al. 2023). Moreover, verifying the safety of neural network controllers operating in high-dimensional observation spaces remains computationally intractable. Existing approaches to safe control primarily focus on low-dimensional state inputs (Dawson et al. 2022) and empirical safety evaluations. Other approaches, such as verified safety over the entire input region and control barrier function-based methods, have also been explored (Wu, Zhang, and Vorobeychik 2024; Emam et al. 2022). However, these methods struggle when the controller operates on image inputs due to the high dimensionality of the observation space. Moreover, while dynamic behavior of many autonomous systems of interest has established models approximating their trajectories through the system state space, physics that map state to its visual representation are considerably more complex, and associated models far more complex and less reliable.

In this work, we integrate reachability analysis with generative modeling to enable efficient verification of neural network controllers operating on high-dimensional image spaces (Katz et al. 2022; Cai, Fan, and Bak 2024). Specifically, we employ a generative neural network with latent inputs representing environmental and perceptual variation to model the perceptual mapping from states to images, allowing us to verify safety properties in a structured and lower-dimensional latent space. To address the scalability challenges of verification, we introduce a *semi-probabilistic verification (SPV) framework*, where safety properties are verified over a sampled distribution of initial states (using distribution-free tail bounds) but for all possible latent environment representations of the trained generative perception model. In addition, we present a training algorithm that makes use of a novel safety loss as a differentiable proxy to this verification objective. A key component of this algorithm is our approach to adapt the training set, stochastically biasing it towards states for which safety is difficult to verify. Our experiments in simulated plane landing, simulated and physical autonomous lane following, and drone navigation and obstacle avoidance, demonstrate that the proposed approach yields policies that exhibit significantly stronger safety properties compared to state-of-the-art safe control baselines.

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

In summary, our key contributions are as follows:

- A novel semi-probabilistic safety verification (SPV) framework that provides formal safety guarantees while remaining computationally feasible in high-dimensional vision-based control settings.
- A novel training approach which uses a differentiable proxy loss for SPV and maintains a dynamic training set which adaptively prioritizes safety-critical states.
- Experimental evaluation in simulated and physical path following and simulated drone control that demonstrates effective empirical and provably-verified performance of the policies trained through our approach in comparison with state-of-the-art baselines.

## 2 Model

### 2.1 Problem Formulation

We consider a discrete-time dynamical system:

$$s_{t+1} = f(s_t, u_t), o_t = h(s_t, \omega), s_0 \sim \mathcal{D}, \omega \sim \Omega, \quad (1)$$

where  $s_t \in \mathcal{S}$  is the system state (e.g., position, steering angle of the vehicle),  $o_t \in \mathcal{O}$  is the vision-based (image) observation perceived by the agent,  $u_t \in \mathcal{U}$  the control action,  $h$  the mapping from state to observation, and  $\mathcal{D}$  a distribution over the initial state  $s_0$ . Notably,  $h$  takes as input a *perceptual environment*  $\omega$ , which models an unobserved source of environment variation distributed according to an unknown distribution  $\Omega$ . We assume that the dynamics  $f$  are known (for example, well-known dynamical system models for physical systems), while  $h$  and  $\mathcal{D}$  are both unknown. At execution time, we suppose that only observations  $o_t$  are known to the controller, with state  $s_t$  unobservable. Our goal is to synthesize a control policy  $\pi$  mapping visual observations  $o$  to control actions  $u$  which is *provably safe* in the sense we formalize next.

Let  $P$  denote a safety specification, which is a predicate  $P(s)$  indicating whether a state  $s \in \mathcal{S}$  is safe or not. Similarly, let  $P(\mathcal{T})$  indicate whether  $P(s)$  is true for all  $s \in \mathcal{T} \subseteq \mathcal{S}$ . We assume that both  $P(s)$  and  $P(\mathcal{T})$  can be evaluated efficiently (for example, safety is often described using linear inequalities and  $\mathcal{T}$  is a polyhedron). Given a policy  $\pi$ , the controlled dynamical system effectively becomes  $s_{t+1} = f(s_t, \pi(h(s_t, \omega)))$ . We say that this dynamical system is *safe* for an initial state  $s_0$  over a horizon  $K$  if  $P(s_t)$  is true for all  $0 \leq t \leq K$ .

This notion of safety, however, is limited for two reasons. First, we do not know  $h$ , so we cannot directly verify the dynamical system above. Second, we wish for a policy  $\pi$  to satisfy safety in a way that is not tied to a specific starting state, but with respect to the full set of initial states  $\mathcal{S}$ . We address the first challenge by leveraging a conditional generative neural network model to approximate  $h$ , and the second by using a *semi-probabilistic verification (SPV)* framework. We describe both of these ideas next.

### 2.2 Approximating the Visual Observation Model

We address the first challenge by using a conditional generative model  $g(s, z)$  which induces a distribution over obser-

vations  $o \in \mathcal{O}$  for a given state  $s \in \mathcal{S}$ , with  $z \in \mathcal{Z}$  a (typically distributed according to a uniform or Gaussian distribution) random vector, analogous to the approach proposed by Katz et al. (2022). Such a generator can be trained, for example, using the conditional generative adversarial network (cGAN) framework (Mirza and Osindero 2014; Isola et al. 2017) or a conditional diffusion model (Yang and Mandt 2023) from a collection of data  $(o, s)$  in which images  $o$  are annotated with associated states  $s$ . We can view the latent random vectors  $z$  as representations of natural environment variation (e.g., different perspectives, lighting, etc). The goal here is that  $g$  approximates  $h$ , but in practice this assumption is too strong. Instead, we make the following considerably weaker assumption about the relationship between  $h$  and  $g$ .

**Assumption 1.**  $\sup_{s, \omega} \inf_z \|h(s, \omega) - g(s, z)\| \leq \epsilon$ .

In practice, this assumption boils down to having (a) sufficient training data for the generator  $g$  and (b) a sufficiently rich representation (e.g., neural network) and latent dimension of  $z$ . We validate this assumption in Section 5 below.

## 3 Semi-Probabilistic Verification

Our notion of safety is based on  $K$ -reachability. In traditional  $K$ -reachability, safety is guaranteed for all states in some specified set  $\mathcal{S}_0 \subseteq \mathcal{S}$  from which dynamics may be initialized. However, such  $K$ -reachability proofs are generally conservative and typically suffer from significant scalability challenges. When controllers use vision, scalability can be a prohibitive barrier to verification. In practice, however, we can often obtain information about the distribution  $\mathcal{D}$  over initial states  $s_0$  of the dynamical system. For example, by collecting empirical visual data and annotating it with state-relevant information (Waymo (Sun et al. 2020) or KITTI (Geiger et al. 2013) datasets in the case of autonomous driving). On the other hand, the distribution over the initial state is often difficult to cleanly characterize (indeed, it may be heavy-tailed). It is, therefore, natural to appeal to distribution-free bounds to obtain probabilistic safety proofs with respect to the unknown distribution  $\mathcal{D}$  over  $s_0$  based on safety properties obtained for a finite sample of initial states. In contrast, the distribution of the visual environment induced by  $\omega$  is far more challenging to characterize or sample, particularly since we do not know  $h$ .

We propose to balance these considerations through a semi-probabilistic verification (SPV) framework, in which we aim to obtain provable distribution-free guarantees with respect to  $\mathcal{D}$ , but which hold in the worst case with respect to environment variation  $\omega$ .

To formalize, fix a policy  $\pi$  and let

$$\mathcal{S}_{t+1}(s_0, \pi) = \{f(s, \pi(o)) \mid o = h(s, \omega), \\ s \in \mathcal{S}_t(s_0, \pi), \omega \in \Omega\},$$

where  $\mathcal{S}_0(s_0, \pi) = \{s_0\}$ . Define  $\text{Reach}_K(s_0, \pi) = \cup_{t=0}^K \mathcal{S}_t(s_0, \pi)$ , that is, all states that can be reached from  $s_0$  for any perceptual environment  $\omega \in \Omega$ . Note that this form of reachability cannot be verified, since we do not know  $h$ . However, we can now leverage the generator  $g$  as a proxy,

with Assumption 1 allowing us to obtain sound safety guarantees. Specifically, let

$$\hat{\mathcal{S}}_{t+1}(s_0, \pi) = \{f(s, \pi(o)) \mid \|o - g(s, z)\| \leq \epsilon, \\ s \in \hat{\mathcal{S}}_t(s_0, \pi), z \in \mathcal{Z}\},$$

and define  $\text{Reach}_K(s_0, \pi, g) = \cup_{t=0}^K \hat{\mathcal{S}}_t(s_0, \pi)$ . The following result allows us to focus on verification with respect to  $\text{Reach}_K(s_0, \pi, g)$ .

**Theorem 1.** *Under Assumption (1),  $\text{Reach}_K(s_0, \pi) \subseteq \text{Reach}_K(s_0, \pi, g)$ . Therefore,  $P(\text{Reach}_K(s_0, \pi, g)) \Rightarrow P(\text{Reach}_K(s_0, \pi))$ .*

*Proof.* Suppose  $\mathcal{S}_t \subseteq \hat{\mathcal{S}}_t$  and let  $s \in \mathcal{S}_t$  and  $o = h(s, \omega)$  for some  $\omega \in \Omega$ . Then  $s \in \hat{\mathcal{S}}_t(s_0, \pi)$  and by Assumption (1),  $\|o - g(s, z)\| \leq \epsilon$  for some  $z \in \mathcal{Z}$ . Consequently,  $f(s, \pi(o)) \in \hat{\mathcal{S}}_{t+1}(s_0, \pi)$ . Since  $\hat{\mathcal{S}}_0(s_0, \pi) = \mathcal{S}_0(s_0, \pi) = \{s_0\}$ , the result follows by induction.  $\square$

In practice, we will make use of a verification tool that is able to efficiently obtain an over-approximation of  $\text{Reach}_K(s_0, \pi, g)$ , which maintains soundness.

Our next step is to combine this with a distribution-free tail bound with respect to the initial state distribution  $\mathcal{D}$ . Specifically, suppose that we have a finite sample of  $N$  initial states  $\{s_i\}_{i=1}^N$  i.i.d. from  $\mathcal{D}$ . We show that by verifying only with respect to this finite sample of  $N$  states, we can achieve a semi-probabilistic safety guarantee for the entire initial region with respect to the unknown distribution  $\mathcal{D}$ .

**Theorem 2.** *Suppose that  $\{s_i\}_{i=1}^N$  i.i.d. from  $\mathcal{D}$  and let  $V = \{s_i \mid P(\text{Reach}_K(s_i, \pi, g))\}$  be the subset of sampled initial states for which the reachable set is safe. Then under Assumption (1),*

$$\Pr_{s \sim \mathcal{D}} [P(\text{Reach}_K(s, \pi))] \geq \frac{|V|}{N} - \sqrt{\frac{1}{2N} \log \frac{2}{\delta}}$$

with probability at least  $1 - \delta$ .

*Proof.* Let  $\alpha = \Pr_{s \sim \mathcal{D}} [P(\text{Reach}_K(s, \pi, g))]$  and  $\hat{\alpha} = \frac{|V|}{N}$ . By the Chernoff-Hoeffding bound,  $\Pr(|\hat{\alpha} - \alpha| \geq \epsilon) \leq 2e^{-2N\epsilon^2}$ , where the probability is with respect to datasets of  $N$  initial states. Letting  $\delta = 2e^{-2N\epsilon^2}$ , we obtain the confidence bound:  $\Pr\left(\alpha \geq \hat{\alpha} - \sqrt{\frac{1}{2N} \log \frac{2}{\delta}}\right) \geq 1 - \delta$ . Finally, since by Theorem 1,  $P(\text{Reach}_K(s_0, \pi, g)) \Rightarrow P(\text{Reach}_K(s_0, \pi))$ , the result follows.  $\square$

The SPV framework above can thereby combine reachability over a finite sample of initial states to yield a rigorous tail bound guarantee for safety over a given safety horizon  $K$ . This, of course, is for a given policy  $\pi$ . In the next section, we turn to the main subject of our work: synthesizing control policies  $\pi$  for the dynamical system (1) that achieve strong semi-probabilistic guarantees of this kind.

## 4 Learning-Based Synthesis of Provably Safe Vision-Based Control

At the high level, our goal is to learn a policy  $\pi$  which has a long safety horizon  $K$  (that is, does not reach an unsafe state for any possible trajectory over as long a horizon  $K$  as possible) with high probability  $1 - \delta$ . Suppose that  $\pi_\theta$  is parametric with parameters  $\theta$  (e.g., a neural network), and  $K$  (i.e., the target safety horizon) is fixed. Our goal is to maximize the probability that a trajectory is safe for at least  $K$  steps, that is,

$$\max_{\theta} \Pr_{s \sim \mathcal{D}} [P(\text{Reach}_K(s, \pi_\theta))]. \quad (2)$$

To make this practical, we can only rely on a finite sample of initial states, as well as make use of the cGAN  $g$ . Consequently, the revised proxy objective is

$$\max_{\theta} \sum_i P(\text{Reach}_K(s_i, \pi_\theta, g)). \quad (3)$$

The previous section shows that this still enables rigorous semi-probabilistic verification. Additionally, we consider a special case in which safety properties are tied to a scalar *safety score* (for example, cross-track error in lane following). In particular, let  $\sigma(\text{Reach}_K(s_i, \pi_\theta, g))$  be a safety score function over the reachable set, with  $P(\mathcal{T})$  translating the safety score into a predicate (e.g., error exceeds a predefined threshold). We assume that we can obtain differentiable bounds on  $\sigma(\text{Reach}_K(s_i, \pi_\theta, g))$  (e.g., if we use  $\alpha, \beta$ -CROWN (Zhang et al. 2018; Xu et al. 2020b,a)).

### 4.1 The Learning Framework

A central challenge in synthesizing a provably safe policy  $\pi_\theta$  in our setting arises from the involvement of high-dimensional images generated by the generator  $g$  (as a proxy for the true perception model  $h$ ), which serve as inputs to the controller. Our overall approach is as follows. First, we begin with a pre-trained controller  $\pi_{\hat{\theta}}$  that is empirically safe, which we also use as the *anchor controller* to avoid sacrificing too much empirical performance as we train for safety verification. Next, starting with  $\hat{\theta}$ , we train (or fine-tune)  $\pi_\theta$  to minimize  $\sum_{i \in S_l} \mathcal{L}(s_i, \theta)$ , where  $S_l$  is a set of initial states which evolves over training iterations  $l$ , and  $\mathcal{L}(\theta)$  an appropriate loss function. The central algorithm design questions thus amount to 1) the choice of the loss function, and 2) the problem of selecting data  $S_l$  to use in each iteration, so as to ultimately obtain a provably (rather than merely empirically) safe policy. We address these questions next.

### 4.2 Loss Function

We propose a loss function that integrates both a performance-preserving loss and safety loss as follows:

$$\mathcal{L}(s_0, \theta) = \lambda_1 \mathcal{L}_{\text{perf}}(s_i, \theta) + \lambda_2 \mathcal{L}_{\text{safety}}(s_i, \theta). \quad (4)$$

The performance loss  $\mathcal{L}_{\text{perf}}$  aims to preserve the empirical performance with respect to the pre-trained anchor controller  $\pi_{\hat{\theta}}$ . Depending on the training method, it can be (1)  $\ell_2$  loss between predictions and ground truth for supervised

training, or (2) RL objective (e.g. policy gradient) to encourage high expected returns for RL training.

Turning next to the safety loss, recall that we assume that safety is quantified by a safety score function  $\sigma(\text{Reach}_K(s_i, \pi_\theta, g))$ . One candidate would simply be to use this score as part of the loss function. However, this is impractical, as it is typically intractable to compute at scale and to the extent that it can be done, the tools for doing so are not differentiable. However, neural network verification techniques exist which compute *differentiable sound upper and lower bounds on this quantity*, and these therefore make natural candidates to use in constructing a loss function. More precisely, let  $\underline{\sigma}(\text{Reach}_K(s_i, \pi_\theta, g)) \leq \sigma(\text{Reach}_K(s_i, \pi_\theta, g)) \leq \bar{\sigma}(\text{Reach}_K(s_i, \pi_\theta, g))$  (i.e.,  $\underline{\sigma}(\cdot)$  is the lower and  $\bar{\sigma}(\cdot)$  the upper bound on  $\sigma(\cdot)$ ). To simplify notation, we let  $\sigma_K^{(i)} = \sigma(\text{Reach}_K(s_i, \pi_\theta, g))$ , with  $\bar{\sigma}_K^{(i)}$  and  $\underline{\sigma}_K^{(i)}$  the corresponding upper and lower bounds. Then we define the safety loss as

$$\mathcal{L}_{safety}(s_i, \theta) = \frac{|\bar{\sigma}_K^{(i)}| + |\underline{\sigma}_K^{(i)}|}{K - 1}, \quad (5)$$

which measures the rate of change of the reachable region.

### 4.3 Adaptive Training Data

Our adaptive training procedure performs gradient updates by sampling batches from an adaptive collection of training data  $S_l$  which consists of two disjoint and fixed-size components: the set of random initial states  $S_0$ , and  $S_A$ , maintained as a priority queue, containing initial states for which safety is a challenge to satisfy. Specifically, when training starts,  $S_A$  is empty, and we gradually populate  $S_A$  during the warmup period by adding the  $m\%$  most challenging datapoints (in the sense detailed below) from each training batch to it. To ensure  $S_0$  and  $S_A$  remain disjoint, whenever a datapoint is added to  $S_A$ , it is also deleted from  $S_0$ , and we generate another datapoint uniformly randomly to add to  $S_0$ .

We select datapoints to add to  $S_A$  based on the rate of change in the safety margin  $\sigma_i$  over an entire  $K$ -step trajectory. For example, a datapoint where the vehicle deviates from the center of the lane and drifts toward the margin at high speed is prioritized. This allows us to detect points that are likely to become unsafe ahead of time. The safety loss defined in Equation (5) can be directly used as this metric here. If training fails to improve safety on datapoints in  $S_A$ ,  $S_A$  effectively becomes fixed once full. Otherwise, if safety improves or we encounter new datapoints that are less safe,  $S_A$  adapts to reflect such changes.

After we have enough datapoints in  $S_A$ , future training batch  $T$  with size  $L$  consists of  $\lfloor p \cdot L \rfloor$  datapoints from  $S_A$  and the rest from  $S_0$ , where  $p$  is a tunable parameter during training. The portion of datapoints from  $S_0$  can be sampled uniformly. The portion of datapoints from  $S_A$  are sampled following the Efraimidis & Spirakis (Efraimidis and Spirakis 2006) weighted sampling approach to prioritize datapoints that are more difficult (less safe) than others. Specifically, for each datapoint  $i$  in  $S_A$ , we first calculate a safety

---

### Algorithm 1: Data Sampling Algorithm

---

```

1: Input:  $S_l = S_0 \cup S_A, \beta, \alpha, p, L$ 
2: Output: Sampled points  $T$ 
3:  $L_A \leftarrow \lfloor p \cdot L \rfloor; L_0 \leftarrow L - L_A$ 
4: for each datapoint  $x_i \in S_A$  do
5:   Compute  $\eta^{(i)}$  according to Equation 6
6:   Compute weight  $w_i = e^{\alpha \cdot \eta^{(i)}}$ 
7: end for
8: Normalize  $w_i$  and compute keys  $k_i$  for all  $x_i \in S_A$ 
9:  $T_A \leftarrow$  top  $L_A$  elements in  $S_A$  by  $k_i$  values
10:  $T_0 \leftarrow$  uniformly sample  $L_0$  points from  $S_0$ 
11:  $T \leftarrow T_A \cup T_0$ 
12: Return  $T$ 

```

---

parameter

$$\eta^{(i)} = \mathcal{L}_{safety}^{(i)} + \left( \max(0, |\bar{\sigma}_K^{(i)}| - \beta) + \max(0, |\underline{\sigma}_K^{(i)}| - \beta) \right), \quad (6)$$

where the second term further penalizes datapoints whose upper and lower bound is outside of the  $[-\beta, \beta]$  region, where  $\beta$  is a predefined safety threshold, and ensure such difficult-to-verify inputs are more likely to be sampled. We then assign a weight  $w_i = e^{\alpha \cdot \eta^{(i)}}$  (where  $\alpha$  is a hyperparameter) to each datapoint  $i$  and normalize the weights such that  $\sum_{i=1}^N w_i = 1$ . Finally we assign each element  $x_i$  a key  $k_i = w_i^{1/U_i}$ , where  $U_i \sim \text{Uniform}(0, 1)$ . Selecting the top  $L$  elements with the highest  $k_i$  values yields a weighted random sample without replacement, where each element  $x_i$  is included with probability:  $P(x_i \in S) = w_i / \sum_{j \in B} w_j$ .

The full algorithm for data sampling is shown in Algorithm 1. Line 3 calculates how many datapoints to sample from the buffer based on a tunable parameter  $p$ ; line 4 – 8 computes the weights and keys following (Efraimidis and Spirakis 2006); line 9 – 11 construct the training batch. This approach enables us to have an adaptive training set which maintains the controller’s average performances while improving its verifiability with respect to datapoints that are more difficult.

### 4.4 Curriculum Learning

Finally, as in prior work (Wu, Zhang, and Vorobeychik 2024), we use curriculum learning. Specifically, during training, we first target  $K_i$ -step verified safety. After a series of epochs, we progress to  $K_{i+1}$ -step verified safety, where  $1 \leq K_1 < K_2 < \dots < K_{n-1} < K_n = K$ . This gradual increase in the verification horizon helps improve training stability and enables the controller to learn more complex safety constraints.

## 5 Experiments

### 5.1 Experiment Setup

We evaluate our approach in four settings: 1) the X-Plane 11 Flight Simulator, 2) the CARLA Simulator, 3) a mini-city miniature urban physical autonomous driving platform with

an F1Tenth racing car, and 4) the Airsim Simulator for drone control. We use two evaluation metrics: 1) empirical performance and 2) lower bound probability for safety guarantee as a function of  $K$ . We use three baselines for comparison: 1) RESPO (Ganai et al. 2024), a safe reinforcement learning framework using iterative reachability analysis; 2) SAC-RCBF (Emam et al. 2022), which incorporates safety as a robust-control-barrier-function layer into training; and 3) VSRL (Wu, Zhang, and Vorobeychik 2024), which guarantees finite-horizon safety by integrating incremental reachability verification into safe reinforcement learning.

Specifically, we first consider the autonomous aircraft taxiing problem using the **X-Plane 11 Flight Simulator** (Laminar Research 2011). For training the image generator, we collected 20,000 state-image pairs. Each sample consists of an image captured by a forward-facing camera and the corresponding state information (lateral offset  $d$  ranging from  $-10$  to  $10$  meters and heading error  $\theta$  between  $-0.5$  and  $0.5$  radians). Data was collected while the aircraft operated at a constant ground speed.

Our second set of experiments consider autonomous lane following using the **CARLA Simulator** (Dosovitskiy et al. 2017) version 0.9.14. For training the image generator, we sample initial states consisting of the lateral distance  $d$  from the lane center, the heading error  $\theta$  relative to the lane direction, and the global coordinates  $(x, y, z)$  within the CARLA map. The dataset includes trajectories with  $d \in [-0.8, 0.8]$  meters and  $\theta \in [-0.15, 0.15]$  radians. Final dataset contains 20,000 state-image pairs collected across different towns (maps) and environmental conditions.

Our third experiment extend the CARLA experiments to a **F1Tenth racing car**, an open-source 1:10 scale autonomous vehicle, for lane following in a **miniature city physical testbed**. The F1Tenth vehicle is equipped with a front-facing camera that provides visual input for lane following and can achieve scaled speeds comparable to full-scale autonomous vehicles. This platform enables us to evaluate the transferability of our training framework to real-world physical systems. We collected 400 images in the mini-city and manually annotated their state information. We then finetuned the image generator from the CARLA experiment on this dataset to obtain the image generator for the F1Tenth experiments.

The final problem we consider is drone navigation and obstacle avoidance in **Airsim Simulator** (Shah et al. 2017). The drone is initialized at a fixed hovering position in front of an obstacle observable by its camera. The goal is to learn a smooth control policy while avoid collisions. The image generator is trained on a dataset with 20,000 samples. The state input consists of the drone’s 3D position  $(x, y, z)$  and orientation represented as a quaternion  $(q_w, q_x, q_y, q_z)$ . The dataset was collected while the drone was manually controlled by a remote controller.

For all experiments, we pretrain an anchor controller first. For the three path following experiments, the training is done by imitation learning from a tuned PID controller for the corresponding task, with learning rate of 0.0005, batch size 256, and 200 epochs. For the drone experiment, the anchor controller is trained using the standard PPO algorithm with  $\gamma$  0.99, learning rate 0.0003 for a total of 200000 steps.

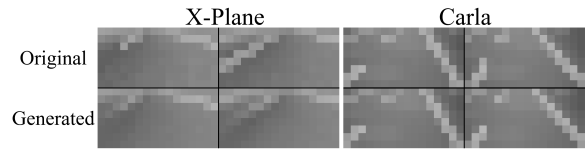


Figure 1: X-Plane and Carla generator illustrations.

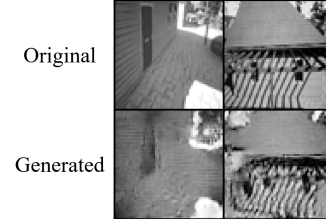


Figure 2: Drone generator illustration.

Then, during safety training, we used supervised training for the path following experiments where we collected a dataset of 30,000 samples of (state, action) pair. We train the controller for 100 epochs, with batch size 128 and learning rate 0.0002. For the drone experiment, we integrated the safety loss directly into RL training using the PPO algorithm. The controller is trained for an additional 200000 steps with learning rate 0.0002. Detailed hyperparameters and settings can be found in appendix of the extended version.

## 5.2 Image Generator Training and Evaluation

The image generators are implemented as conditional GAN (cGAN) that maps from state information and a dimension 10 latent vector to grayscale images; in ablations below we also consider diffusion models for this purpose. The latent vector  $z$  is sampled from a uniform distribution  $\mathcal{U}(-1, 1)$ , aiming to capture semantic variations that are not explicitly represented in our state information (e.g., lighting conditions, road textures, environmental elements). The generators output  $8 \times 16$  images in the three path following experiments, and  $64 \times 64$  images in the drone experiment.

For the path following experiments, we use a 4-layer fully connected neural network with hidden layers of 256 neurons each, using ReLU activation for the hidden layers and Tanh activation for the output layer. For the drone experiment, we use a more complex convolutional generator—its exact architecture is documented in appendix of the extended version. The discriminator in all experiments is a convolutional neural network. During training, we apply spectral normalization (Miyato et al. 2018) to all discriminator layers and orthogonal regularization (Brock et al. 2017) to the generator loss function. Both generator and discriminator networks are initialized using orthogonal initialization and trained with batch size 128, learning rate  $7e-4$ , for 100 epochs.

Figure 1 and Figure 2 show two ground truth images from the test set for each experimental setting alongside their corresponding generated images. Images from F1Tenth are excluded here as they are visually similar to the Carla im-

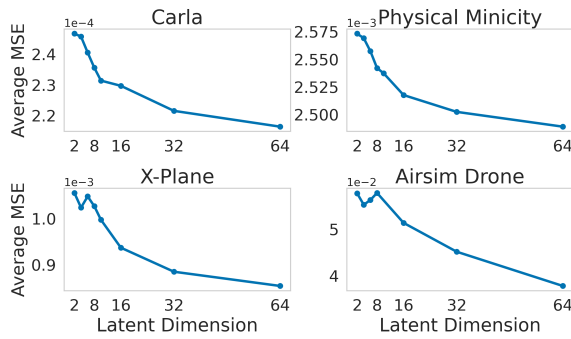


Figure 3: Empirical validation of Assumption 1.

ages. From visual inspections, the generators perform well for the Carla and Plane experiments, which is largely due to the low-dimensional nature of the images in these settings. For the Drone experiment where we have higher resolution images, on the other hand, while the generated images are less precise, they still capture the essential semantic features needed for effective controller learning.

An important ingredient in our approach is Assumption 1, that for true image inputs we can find a latent representation  $z$  as input to the generator that results in its close approximation. We validate this empirically for 8 choices of latent dimensions: 2, 4, 6, 8, 10, 16, 32, 64. We evaluate the generators on a separate test set: for each ground-truth image, we perform gradient-based optimization to find the best latent variable  $z$  that minimizes the mean squared error (MSE). Notably, since this is not guaranteed to find a globally optimal  $z$ , our results provide a pessimistic evaluation (i.e., it is possible that such a  $z$  exists even if we fail to find it using gradient-based optimization).

As shown in Figure 3, as latent dimension increases, the averaged MSE rapidly decreases, i.e., the generator  $g$  better approximates the unknown  $h$  in all four experiment settings we consider in this work. Moreover, we can find effective approximations of the true image inputs for even relatively small latent dimension.

### 5.3 Results

**Empirical Performance** To evaluate the controllers’ empirical performance, we simulate trajectories starting from 100 random initial states for 200 steps. We use the undiscounted cumulative rewards  $\sum r_t$  as the evaluation metric. For the path following tasks (X-Plane, Carla, and Physical Minicity), the reward at each step  $t$  is defined as  $r_t = 1 - \min(1, |d_t|/\beta)$ , where  $d_t$  is the cross-track error at step  $t$  and  $\beta$  is a predefined safety threshold. We set  $\beta = 1.0\text{m}$  for Carla and Physical Minicity evaluations, and  $\beta = 12.5\text{m}$  for X-Plane evaluations. For drone control in Airsim, we used the same reward function in both training and evaluation. The reward includes components for forward velocity, attitude stability, and control smoothness. The complete reward function is explained in appendix of the extended version.

As shown in Figure 4, the proposed SPVT approach maintains strong empirical performance across all experi-

mental settings. Notably, in the Drone experiment, which has the highest-dimensional image input, SPVT demonstrates significantly superior performance compared to all safe training baselines. This result suggests that SPVT is the only scalable safe training method capable of handling high-dimensional image inputs. It is worth noting that VSRL training fails entirely in the Drone experiment because VSRL requires formal verification over the controller’s complete input space, which becomes computationally infeasible for high-dimensional image data.

**Safety Guarantee** For the path following tasks, the safety property we consider is the vehicle not leaving the current lane, equivalently,  $|d_i| \leq \beta$  for  $0 \leq i \leq K$ . For the drone experiment, safety properties include altitude bounds ( $z \in [0.2, 15]$  meters) and attitude limits (roll and pitch angles  $\leq 1.2$  radians).

To obtain the lower-bound safety probability under our semi-probabilistic verification (SPV) framework, we collected a dataset of 2000 initial states sampled i.i.d. from the initial state distribution. We used  $\alpha, \beta$ -CROWN (Zhang et al. 2018; Xu et al. 2020a,b) to verify the model iteratively. At each step,  $\alpha, \beta$ -CROWN computes bounds on the model output, which, after dynamics calculations, gives the reachable state regions at the next timestep. We then check whether this state region intersects with unsafe states. We iterate this forward reachability analysis for  $K$  steps, obtaining empirical verified safety result, i.e. the proportion of initial states in our dataset that are verified safe at each step. These empirical results are then used in Theorem 2 to compute formal lower-bound safety probabilities over the entire initial state space. We set the confidence parameter  $\delta = 0.05$  for all experiments.

Figure 5 shows the resulting safety probability lower bounds as a function of horizon  $K$ . Compared to all baseline methods, the proposed SPVT approach significantly increases these probabilistic safety bounds.

**Ablations** A natural alternative to our cGAN-based state-to-image mapping is to use diffusion-based image generation. In this case,  $g(s, z) = z - \phi(x, z)$ , where the diffusion model  $\phi(x, z)$  predicts the noise in the randomly generated input. To provide finer control over the dimension of  $z$ , we make use of an autoencoder  $\psi(z)$  that takes randomly generated vector  $z$  of a given dimension as an input, and outputs a decoded noisy image  $z' = \psi(z)$  with the dimension of the image, so that  $g(s, z) = \psi(z) - \phi(x, \psi(z))$ . To evaluate this design choice, we implement a diffusion-based variant of SPVT and compare it against our cGAN approach on both controller performance and safety verification. Figure 6 shows results in the X-Plane environment. We find that controller trained in the cGAN setup achieve higher rewards and maintain better safety probability bounds. We refer the readers to the extended version for more results and detailed analysis.

## 6 Related Work

Formally verifying the safety of vision-based controllers is extremely challenging. Traditional verification tools (Katz

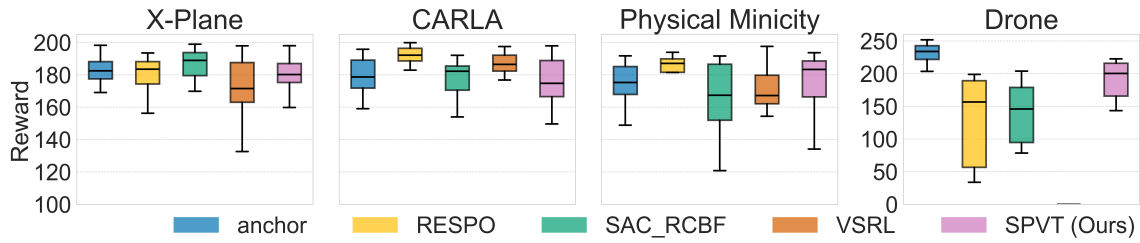


Figure 4: Empirical performance comparison of controllers. Box plots show the distribution of episode rewards over 100 evaluation episodes.

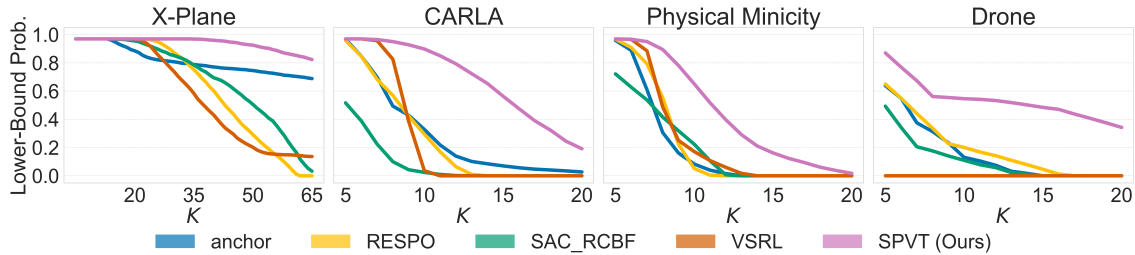


Figure 5: *Semi-Probabilistic Verification (SPV)* results:  $x$ -axis marks the target verification trajectory length ( $K$ ).  $y$ -axis is the lower bound safety probability.

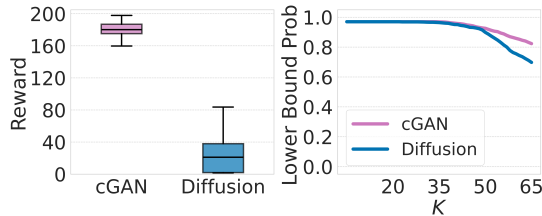


Figure 6: Ablation comparing cGAN vs. diffusion-based image generator in X-Plane. Left: controller performance over 100 episodes. Right: SPV verified safety probability bounds.

et al. 2017; Huang et al. 2017; Ehlers 2017; Sinha et al. 2017) are too computationally demanding for this purpose. Recent progress has enabled verification of larger neural networks through over-approximation (Zhang et al. 2018; Gowal et al. 2018; Xu et al. 2020b), and abstract interpretation (Gehr et al. 2018; Singh et al. 2019; Katz et al. 2019). However, verifying vision-based controllers remains challenging due to high dimensionality, as well as the difficulty of defining safety properties in the image space.

A practical approach to verify vision-based controllers is by approximating the perception module. This reduces verification complexity by projecting the problem from image space to lower-dimensional state space (Katz et al. 2022; Cai, Fan, and Bak 2024; Hsieh et al. 2022). These methods, however, only focus on post-hoc verification and cannot be integrated into the training process for learning a safer controller. For a more comprehensive survey on the verification of vision-based controllers, see (Mitra et al. 2024).

Safe reinforcement learning encodes safety based on the constrained Markov decision processes (Altman 2021) but it

is often too soft to enforce strict safety constraints (Wang et al. 2023). On the other hand, control-theoretic methods such as barrier functions (Dawson et al. 2022; Emam et al. 2022; Xiao et al. 2023), and reachability analysis (Yu et al. 2022; Ganai et al. 2024) are more powerful but they have limited scalability to high-dimensional system such as vision-based controllers. Recently, some works extend these methods to the image space (Abdi, Raja, and Ghabcheloo 2023; Tong, Dawson, and Fan 2023; Yang and Sibai 2024; Tabbara and Sibai 2024). However, they function more like a safety check around the controller, giving yes/no answers for passed in reference control, without providing formal safety guarantees about the controller itself.

## 7 Conclusion and Limitations

We introduced a semi-probabilistic verification framework for efficiently training and verifying vision-based neural network controllers. Our method models the perceptual mapping from state to image with a conditional generator and uses distribution-free tail bounds to get safety guarantees over the entire initial state space. We designed a differentiable proxy to the safety verification objective under the SPV framework that can be directly incorporated into gradient-based training, and an adaptive training set that prioritizes states for which safety property is difficult to verify. While our experiments demonstrate the efficacy of our approach, many limitations remain. For the moment, our controllers and image generator require images to be grayscale and relatively low resolution, and latent dimension of the generator is relatively small. Further research is needed to handle high-resolution visual inputs, as well as to extend to multi-modal sensing.

## Acknowledgements

This work was partially supported by NSF (IIS-2214141, CCF-2403758), ARO (W911NF-25-1-0059), ONR (N00014-24-1-2663), Foresight Institute, and Amazon.

## References

- Abdi, H.; Raja, G.; and Ghabcheloo, R. 2023. Safe Control using Vision-based Control Barrier Function (V-CBF). In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 782–788.
- Altman, E. 2021. *Constrained Markov decision processes*. Routledge.
- Brock, A.; Lim, T.; Ritchie, J.; and Weston, N. 2017. Neural Photo Editing with Introspective Adversarial Networks. In *International Conference on Learning Representations*.
- Cai, F.; Fan, C.; and Bak, S. 2024. Scalable surrogate verification of image-based neural network control systems using composition and unrolling. *arXiv preprint arXiv:2405.18554*.
- Dawson, C.; Qin, Z.; Gao, S.; and Fan, C. 2022. Safe nonlinear control using robust neural lyapunov-barrier functions. In *Conference on Robot Learning*, 1724–1735. PMLR.
- Dosovitskiy, A.; Ros, G.; Codevilla, F.; Lopez, A.; and Koltun, V. 2017. CARLA: An Open Urban Driving Simulator. In *Conference on Robot Learning*, 1–16.
- Efraimidis, P. S.; and Spirakis, P. G. 2006. Weighted random sampling with a reservoir. *Information processing letters*, 97: 181–185.
- Ehlers, R. 2017. Formal verification of piece-wise linear feed-forward neural networks. In *International Symposium Automated Technology for Verification and Analysis*, 269–286.
- Emam, Y.; Notomista, G.; Glotfelter, P.; Kira, Z.; and Egerstedt, M. 2022. Safe reinforcement learning using robust control barrier functions. *IEEE Robotics and Automation Letters*.
- Ganai, M.; Gong, Z.; Yu, C.; Herbert, S.; and Gao, S. 2024. Iterative Reachability Estimation for Safe Reinforcement Learning. In *Neural Information Processing Systems*.
- Gehr, T.; Mirman, M.; Drachler-Cohen, D.; Tsankov, P.; Chaudhuri, S.; and Vechev, M. 2018. AI2: Safety and robustness certification of neural networks with abstract interpretation. In *IEEE Symposium on Security and Privacy*, 3–18.
- Geiger, A.; Lenz, P.; Stiller, C.; and Urtasun, R. 2013. Vision meets robotics: The kitti dataset. *The international journal of robotics research*, 32(11): 1231–1237.
- Gowal, S.; Dvijotham, K.; Stanforth, R.; Bunel, R.; Qin, C.; Uesato, J.; Mann, T.; and Kohli, P. 2018. On the effectiveness of interval bound propagation for training verifiably robust models. *arXiv preprint arXiv:1810.12715*.
- Hsieh, C.; Li, Y.; Sun, D.; Joshi, K.; Misailovic, S.; and Mitra, S. 2022. Verifying controllers with vision-based perception using safe approximate abstractions. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 41(11): 4205–4216.
- Huang, C.; Fan, J.; Li, W.; Chen, X.; and Zhu, Q. 2019. Reachnn: Reachability analysis of neural-network controlled systems. *ACM Transactions on Embedded Computing Systems (TECS)*, 18(5s): 1–22.
- Huang, X.; Kwiatkowska, M.; Wang, S.; and Wu, M. 2017. Safety verification of deep neural networks. In *International Conference on Computer Aided Verification*, 3–29.
- Isola, P.; Zhu, J.-Y.; Zhou, T.; and Efros, A. A. 2017. Image-to-image translation with conditional adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1125–1134.
- Katz, G.; Barrett, C.; Dill, D. L.; Julian, K.; and Kochenderfer, M. J. 2017. Reluplex: An efficient SMT solver for verifying deep neural networks. In *International Conference on Computer Aided Verification*, 97–117.
- Katz, G.; Huang, D. A.; Ibeling, D.; Julian, K.; Lazarus, C.; Lim, R.; Shah, P.; Thakoor, S.; Wu, H.; Zeljić, A.; et al. 2019. The marabou framework for verification and analysis of deep neural networks. In *International Conference Computer Aided Verification*, 443–452.
- Katz, S. M.; Corso, A. L.; Strong, C. A.; and Kochenderfer, M. J. 2022. Verification of image-based neural network controllers using generative models. *Journal of Aerospace Information Systems*, 19(9): 574–584.
- Kiran, B. R.; Sobh, I.; Talpaert, V.; Mannion, P.; Al Sallab, A. A.; Yogamani, S.; and Pérez, P. 2021. Deep reinforcement learning for autonomous driving: A survey. *IEEE transactions on intelligent transportation systems*, 23(6): 4909–4926.
- Kochdumper, N.; Krasowski, H.; Wang, X.; Bak, S.; and Althoff, M. 2023. Provably safe reinforcement learning via action projection using reachability analysis and polynomial zonotopes. *IEEE Open Journal of Control Systems*, 2: 79–92.
- Laminar Research. 2011. X-Plane Flight Simulator. <https://www.x-plane.com>.
- Le, N.; Rathour, V. S.; Yamazaki, K.; Luu, K.; and Savvides, M. 2022. Deep reinforcement learning in computer vision: a comprehensive survey. *Artificial Intelligence Review*, 1–87.
- Mirza, M.; and Osindero, S. 2014. Conditional Generative Adversarial Nets. *ArXiv*, abs/1411.1784.
- Mitra, S.; Păsăreanu, C.; Prabhakar, P.; Seshia, S. A.; Mangal, R.; Li, Y.; Watson, C.; Gopinath, D.; and Yu, H. 2024. Formal Verification Techniques for Vision-Based Autonomous Systems—A Survey. In *Principles of Verification: Cycling the Probabilistic Landscape: Essays Dedicated to Joost-Pieter Katoen on the Occasion of His 60th Birthday, Part III*, 89–108. Springer.
- Miyato, T.; Kataoka, T.; Koyama, M.; and Yoshida, Y. 2018. Spectral Normalization for Generative Adversarial Networks. In *International Conference on Learning Representations*.
- Shah, S.; Dey, D.; Lovett, C.; and Kapoor, A. 2017. AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles. In *Field and Service Robotics*.

Singh, G.; Gehr, T.; Püschel, M.; and Vechev, M. 2019. An abstract domain for certifying neural networks. In *ACM Symposium on Principles of Programming Languages*.

Sinha, A.; Namkoong, H.; Volpi, R.; and Duchi, J. 2017. Certifying some distributional robustness with principled adversarial training. *arXiv preprint arXiv:1710.10571*.

Sun, P.; Kretzschmar, H.; Dotiwalla, X.; Chouard, A.; Patnaik, V.; Tsui, P.; Guo, J.; Zhou, Y.; Chai, Y.; Caine, B.; et al. 2020. Scalability in perception for autonomous driving: Waymo open dataset. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2446–2454.

Tabbara, I.; and Sibai, H. 2024. Learning Ensembles of Vision-based Safety Control Filters. *arXiv:2412.02029*.

Tong, M.; Dawson, C.; and Fan, C. 2023. Enforcing safety for vision-based controllers via Control Barrier Functions and Neural Radiance Fields. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 10511–10517.

Wang, Y.; Zhan, S. S.; Jiao, R.; Wang, Z.; Jin, W.; Yang, Z.; Wang, Z.; Huang, C.; and Zhu, Q. 2023. Enforcing hard constraints with soft barriers: safe reinforcement learning in unknown stochastic environments. In *International Conference on Machine Learning*.

Wu, J.; Zhang, H.; and Vorobeychik, Y. 2024. Verified Safe Reinforcement Learning for Neural Network Dynamic Models. In *Neural Information Processing Systems*.

Xiao, W.; Wang, T.-H.; Hasani, R.; Chahine, M.; Amini, A.; Li, X.; and Rus, D. 2023. BarrierNet: Differentiable Control Barrier Functions for Learning of Safe Robot Control. *IEEE Transactions on Robotics*, 39(3): 2289–2307.

Xu, K.; Shi, Z.; Zhang, H.; Wang, Y.; Chang, K.-W.; Huang, M.; Kailkhura, B.; Lin, X.; and Hsieh, C.-J. 2020a. Automatic perturbation analysis for scalable certified robustness and beyond. *Advances in Neural Information Processing Systems*, 33.

Xu, K.; Zhang, H.; Wang, S.; Wang, Y.; Jana, S.; Lin, X.; and Hsieh, C.-J. 2020b. Fast and complete: Enabling complete neural network verification with rapid and massively parallel incomplete verifiers. *arXiv preprint arXiv:2011.13824*.

Yang, R.; and Mandt, S. 2023. Lossy image compression with conditional diffusion models. In *Neural Information Processing Systems*, 64971–64995.

Yang, Y.; and Sibai, H. 2024. Pre-Trained Vision Models as Perception Backbones for Safety Filters in Autonomous Driving. *arXiv:2410.22585*.

Yu, D.; Ma, H.; Li, S.; and Chen, J. 2022. Reachability constrained reinforcement learning. In *International Conference on Machine Learning*, 25636–25655. PMLR.

Zhang, H.; Weng, T.-W.; Chen, P.-Y.; Hsieh, C.-J.; and Daniel, L. 2018. Efficient Neural Network Robustness Certification with General Activation Functions. In *Neural Information Processing Systems*.