Generative Modeling of Residuals for Real-Time Risk-Sensitive Safety with Discrete-Time Control Barrier Functions

Ryan K. Cosner, Igor Sadalski, Jana K. Woo, Preston Culbertson, Aaron D. Ames

Abstract—A key source of brittleness for robotic systems is the presence of model uncertainty and external disturbances. Most existing approaches to robust control either seek to bound the worst-case disturbance (which results in conservative behavior), or to learn a deterministic dynamics model (which is unable to capture uncertain dynamics or disturbances). This work proposes a different approach: training a state-conditioned generative model to represent the distribution of error residuals between the nominal dynamics and the actual system. In particular we introduce the Online Risk-Informed Optimization controller (ORIO), which uses Discrete-Time Control Barrier Functions, combined with a learned, generative disturbance model, to ensure the safety of the system up to some level of risk. We demonstrate our approach in both simulations and hardware, and show our method can learn a disturbance model that is accurate enough to enable risk-sensitive control of a quadrotor flying aggressively with an unmodelled slung load. We use a conditional variational autoencoder (CVAE) to learn a state-conditioned dynamics residual distribution, and find that the resulting probabilistic safety controller, which can be run at 100Hz on an embedded computer, exhibits less conservative behavior while retaining theoretical safety properties.

I. INTRODUCTION

Robots operating in the real world face considerable uncertainty due to imperfect perception, approximate world and dynamics models, and random disturbances. These error sources are often a key failure cause for field-deployed robots, and in general can undermine classical safety and performance guarantees that rely on perfect models of the system and its environment. A natural framework to address this issue is risk-sensitivity: to operate effectively in novel, uncertain environments, robots should both be able to represent their uncertainty about the world (due to, e.g., modelling error, perception failures) and design controllers that can still (probabilistically) ensure some level of safety or performance, despite this uncertainty.

In this paper, we explore using deep generative models (DGMs), [1]–[3] which are a broad class of methods that use neural networks to approximate the probability distribution underlying a given dataset, to learn disturbance distributions for risk-sensitive control. These models can be used for density estimation, which provides a likelihood model for the data, and to sample new data points from the data distribution. Beyond their more traditional applications, like generating image [4] and text [5] data, these models have been applied to a broad range of robotics tasks including
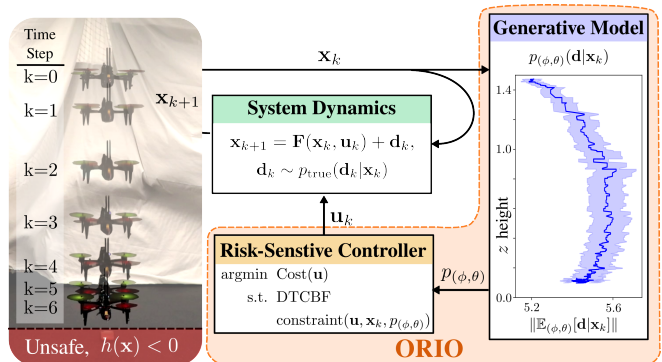
Fig. 1. The drone falling through the air and avoiding hitting the ground using our proposed method. The generative model learns the state dependent disturbance distribution and the mean of the disturbance norm expanded by the trace of the covariance both scaled by the time step plotted on the right.

SLAM [6], imitation learning [7], [8], motion planning [9], [10], human-robot-interaction [11], [12], anomaly detection [13], sim-to-real transfer [14], dynamics learning [15], [16], and reinforcement learning [17].

In this work we employ conditional variational autoencoders (CVAEs) [18] which are a generalization of variational autoencoders (VAEs) that allow one to condition the generating process on a context variable (e.g., the current state). CVAEs have been used to recreate hand-written images of numbers given the desired digit [18] or to predict trajectories given state and environment understanding [19]–[21]. Since the generative process for a CVAE only requires two neural network forward passes and normal distribution samples, they are computationally efficient and well suited for real-time robotics applications.

Stochastic control methods can make use of generative models to ensure constraint satisfaction (e.g., collision avoidance) up to a desired probability. In general, stochastic control methods assume prior knowledge of distribution such as value at risk (VaR) or conditional value at risk (CVaR) [22] values of a constraint function for the noisy dynamics, or the disturbance distribution's mean and covariance [23], [24]. While this assumption is less restrictive than the global upper bound on the disturbance magnitude common in classical, deterministic robust control [25]–[27], it is still unrealistic to assume one has perfect, *a priori* knowledge of the disturbance distribution before operating the system, and impractical / unprincipled to estimate these parameters by hand. To address these issues, we propose to learn a conditional generative model of the dynamics distribution instead of assuming its structure *a priori*.

This work on modeling dynamics residuals most closely

resembles a probabilistic generalization of [28], [29]. For generative modeling this work leverages [18] and for safety we rely on the probabilistic safety guarantees for DTCBFs with stochastic dynamics residuals presented in [23].

This work presents the Online Risk-Informed Optimization (ORIO) controller, a risk-based safety framework that learns to ensure safety in the presence of stochastic dynamics residuals using DGMs and DTCBFs. The main contributions of this work are (1) the proposed ORIO controller, which is a unified framework for dynamics distribution learning and usage of that distribution for ensuring probabilistic safety using DTCBFs; and (2) simulation and hardware demonstrations of the real-time application of these methods on a multi-rotor aerial robot for safe flight.

II. CONDITIONAL VARIATIONAL AUTOENCODERS (CVAEs) FOR GENERATIVE DISTURBANCE MODELING

In this work, we consider applications of safe control in the presence of unmodeled disturbances. Specifically, We consider the following systems with discrete time dynamics:

$$\mathbf{x}_{k+1} = \mathbf{F}(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{d}_k, \quad \forall k \in \mathbb{N} \tag{1}$$

with state $\mathbf{x}_k \in \mathbb{R}^n$, input $\mathbf{u}_k \in \mathbb{R}^m$, unmodeled residual dynamics $\mathbf{d}_k$ that take values in $\mathbb{R}^\ell$ and are sampled from some unknown distribution $p(\mathbf{d}|\mathbf{x})$, and modeled dynamics $\mathbf{F} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$. A state-feedback controller $\mathbf{k} : \mathbb{R}^n \to \mathbb{R}^m$ yields the discrete-time closed-loop system:

$$\mathbf{x}_{k+1} = \mathbf{F}(\mathbf{x}_k, \mathbf{k}(\mathbf{x}_k)) + \mathbf{d}_k, \quad \forall k \in \mathbb{N}. \tag{2}$$

The assumptions that the dynamics residuals are input-independent and unmatched is a general one often made in robust control theory for discrete-time systems [30]–[33].

A. Conditional Variational Inference

To account for the unmodeled disturbances, we first seek a generative model that can approximate the conditional distribution $p(\mathbf{d}|\mathbf{x})$ given a dataset $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{d}_i)\}_{i=1}^N$. We do this by fitting a parametric distribution to $\mathcal{D}$ which attempts to maximize the likelihood of the observed data with respect to the learned distribution.

While there exist many (learning- and learning-free) methods for generative modeling, in this paper, we look to Conditional Variational Autoencoders (CVAEs) [18], a variant of Variational Autoencoders (VAEs) [34] that allows the learned models to be conditioned on observations, $\mathbf{x}$. CVAEs assume there exists a latent variable $\mathbf{z}$ which captures the "unobserved" information explaining any non-random variation in the data distribution p. For example, if the setting of robot safety, the latent codes $\mathbf{z}$ could represent state-dependent modeling errors, or other hidden variables (e.g., higher-order dynamics, time delays) that could influence the difference between the observed next state $\mathbf{x}_{k+1}$, and the modeled dynamics $\mathbf{F}(\mathbf{x}_k, \mathbf{u}_k)$.

Specifically, CVAEs represent the conditional distributions $p_\theta(\mathbf{d}|\mathbf{x}, \mathbf{z})$ and $q_\varphi(\mathbf{z}|\mathbf{x}, \mathbf{d})$, and the latent prior $p_\phi(\mathbf{z}|\mathbf{x})$ as multilayer perceptions (MLPs) with corressponding parameters θ, ϕ, φ, and seek to optimize these parameters such that
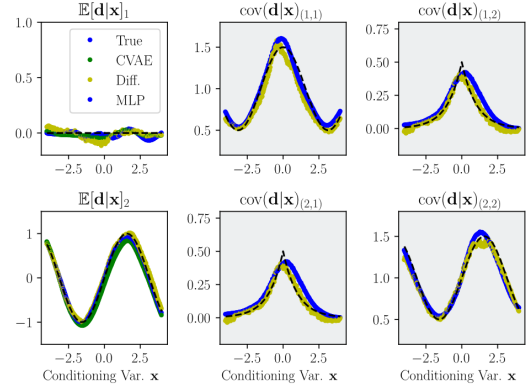

Fig. 2. Learning heteroschedastic disturbance of double integrator system using 3 minutes of data at 100 Hz (36 five second long trajectories). The approximated mean and covariance values are scaled by the time step and plotted against the true values in black using the CVAE in blue, diffusion model in yellow, and MLP (mean only) in green.

the data likelihood, $p_{\theta,\phi}(\mathbf{d})$, marginalized over all states $\mathbf{x}$ and latent codes $\mathbf{z}$, is maximized. Traditionally q_φ is called an "encoder," since it maps states $\mathbf{x}$ and disturbances $\mathbf{d}$ to distributions over the latent codes $\mathbf{z}$, and similarly p_θ is called a "decoder." While maximizing $p_{\theta,\phi}(\mathbf{d})$ exactly is intractable (and, since we do not have access to this distribution, as we are hoping to estimate it from data), we instead optimize the evidence lower bound (ELBO):

$$\log p_{\theta,\phi}(\mathbf{d}|\mathbf{x}) \geq \mathbb{E}_{q_\varphi}[\log p_\theta(\mathbf{d}|\mathbf{x}, \mathbf{z})] \\ - KL\big(q_\varphi(\mathbf{z}|\mathbf{x}, \mathbf{d})\|p_\phi(\mathbf{z}|\mathbf{x})\big) \tag{3}$$

where KL is the Kullback-Liebler divergence. In practice, each network represents its corresponding distribution as a conditional Gaussian, with, e.g. $p_\theta(\mathbf{d}|\mathbf{x}, \mathbf{z}) = \mathcal{N}(\mathbf{d}; \boldsymbol{\mu}_\theta(\mathbf{x}, \mathbf{z}), \boldsymbol{\Sigma}_\theta(\mathbf{x}, \mathbf{z}))$, where $\mathcal{N}(\cdot; \mu, \Sigma)$ is the probability density function of a multivariate Gaussian with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$, and $\boldsymbol{\mu}_\varphi, \boldsymbol{\Sigma}_\varphi$ are neural network outputs representing the sufficient statistics of this distribution.

B. Mean and Covariance Estimation using CVAEs

Once trained, a CVAE can be used for estimation of the disturbance's conditional likelihood, generate new samples, or estimate the mean and covariance of the true distribution.

In particular, the risk-sensitive DTCBF-based controller [23] we use requires us to compute the mean and covariance of the disturbance distribution $\mathbf{d}$. To do this, we use the following estimator[1] for $p_{\theta,\phi}(\mathbf{d}|\mathbf{x})$:

$$p_{\theta,\phi}(\mathbf{d}|\mathbf{x}) \approx \frac{1}{S} \sum_{s=1}^{S} p_\theta(\mathbf{d}|\mathbf{x}, \mathbf{z}^{(s)}) \tag{4}$$

$$= \frac{1}{S} \sum_{s=1}^{S} \mathcal{N}\left(\mathbf{d}; \mu_\theta(\mathbf{x}, \mathbf{z}^{(s)}), \Sigma_\theta(\mathbf{x}, \mathbf{z}^{(s)})\right)$$

---

[1]This estimator is best understood as a "Rao-Blackwellization" [35] (as used in the Markov Chain-Monte Carlo literature) of the simple two-step sampling estimator which uses the sample mean and covariance of $\mathbf{d}^{(s)} \sim p_\theta(\cdot \mid \mathbf{x}, \mathbf{z}^{(s)})$. This estimator is known to have stronger convergence ($O(\frac{1}{\sqrt{S}})$) than the two-step scheme.

	μ Err. Avg. $\pm 2\sigma$	Σ Err. Avg. $\pm 2\sigma$
GMM	0.04512 ± 0.00433	0.09518 ± 0.00296
Sampling	0.04604 ± 0.00989	0.09710 ± 0.01419
Diffusion	0.05866 ± 0.00942	0.1025 ± 0.01363

TABLE I. The statistics for the mean and covariance estimates of each estimation method obtained from 100 estimates at 201 states. The average error is similar for each model, but the GMM-based method has smaller variance which is important when using its outputs in closed-loop control. Estimates for each method are calculated using 10,000 samples.
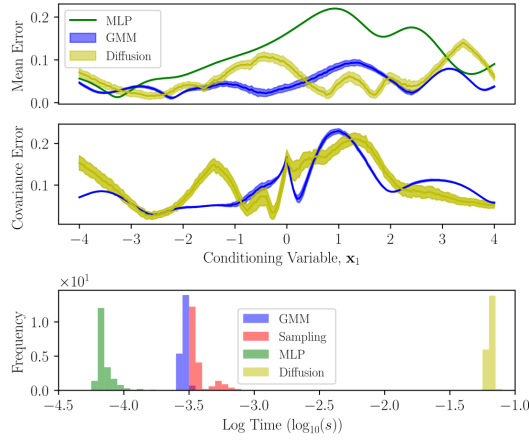
Fig. 3. (Top) Mean error $\|\mathbb{E}_{\text{predicted}}[\mathbf{d}|\mathbf{x}] - \mathbb{E}_{\text{true}}[\mathbf{d}|\mathbf{x}]\|$ vs. state for GMM-based CVAE sampling method (blue), MLP (green), diffusion model sampling mean estimate (yellow). One standard deviation estimated from 100 samples per state, $\mathbf{x}$, is plotted around the CVAE and diffusion model curves. **(Middle)** Covariance error $\|\text{cov}_{\text{predicted}}(\mathbf{d}|\mathbf{x}) - \text{cov}_{\text{true}}(\mathbf{d}|\mathbf{x})\|_2$ vs. state for GMM-based CVAE method (blue) and diffusion model sampling-mean estimate (yellow) both using 10,000 samples is shown with one standard deviation. **(Bottom)** Evaluation time for each model to make its apprpoximation. Calculations performed on a desktop computer with a Nvidia 3090Ti GPU with 10,000 samples for the stochastic methods.

where $\mathbf{z}^{(s)} \sim p_\theta(\mathbf{z}|\mathbf{x})$. Since this MC approximation is a Gaussian mixture model (GMM) we can obtain its mean and expectation in closed form as:

$$\mathbb{E}[\mathbf{d}|\mathbf{x}] \approx \frac{1}{S} \sum_{s=1}^{S} \mu_\theta(\mathbf{x}, \mathbf{z}^{(s)}) \triangleq \overline{\mu}(\mathbf{x}), \qquad (5)$$

$$\text{cov}(\mathbf{d}|\mathbf{x}) \approx \frac{1}{S} \Big(\sum_{s=1}^{S} \Sigma_\theta(\mathbf{x}, \mathbf{z}^{(s)}) + \mu_\theta(\mathbf{x}, \mathbf{z}^{(s)})\mu_\theta(\mathbf{x}, \mathbf{z}^{(s)})^T \Big)$$
$$- \overline{\mu}(\mathbf{x})\overline{\mu}(\mathbf{x})^\top \triangleq \overline{\Sigma}(\mathbf{x}) \qquad (6)$$

C. Residual Dynamics Modeling: Simple Simulation

To demonstrate the capabilities of CVAEs to learn complex dynamics disturbances, we consider the simple double integrator system:

$$\mathbf{x}_k = \begin{bmatrix} x \\ v \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & \Delta_t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ v \end{bmatrix}_k + \mathbf{d}_k \qquad (7)$$

with a state dependent residual distribution, $\mathbf{d}_k \sim p(\mathbf{d}_k|\mathbf{x}_k)$ which is a state-dependent Gaussian distribution with mean $\boldsymbol{\mu}(\mathbf{x}) = \begin{bmatrix} 0, & \sin(x) \end{bmatrix}^\top$ and the covariance is $\boldsymbol{\Sigma}(\mathbf{x}) = \frac{1}{2}\begin{bmatrix} 2 + \cos(x) & \exp(-|x|) \\ \exp(-|x|) & 2 + \sin(x) \end{bmatrix}$. The system was initialized at $\mathbf{x}_0 = \mathbf{0}$, with $\Delta_t = 0.01$, and simulated for 35 five-second trials to collect data. Then the CVAE was trained to approximate the distribution. The CVAE accurately learns the nonlinear heteroschedastic disturbance with a relatively small amount (3 minutes, total, in simulation time) of data, as can be seen in Fig. 2,

We compare the CVAE to two baselines: a conditional diffusion model [36], which is a state-of-the-art generative model that has recently seen interest as a policy representation for robotics [8], and a simple MLP trained to map the state $\mathbf{x}$ to a fixed, deterministic disturbance $\mathbf{d}(\mathbf{x})^2$. The

---

[2] Code for this test and the MLP and diffusion models can be found here.

results of this are shown in Fig. 3 and Table I. There we can see that the MLP, despite being significantly faster, tends to overfit to the noise causing higher mean error. Alternatively, the diffusion model accurately learns the distribution, but is nearly two orders of magnitude slower than the CVAE. Additionally, two approximation methods are used for the CVAE: the GMM-based estimator in (5, 6) and a simple two-step sampling estimator using the population mean and covariance calculations from samples of $p_{\theta,\phi}(\mathbf{d}|\mathbf{x})$. The GMM-based method is shown to be slightly faster and results in less average error and variance.

III. SAFETY THEORY USING DISCRETE-TIME CONTROL BARRIER FUNCTIONS (DTCBFs)

Now, given a way to learn complex, heteroscedastic noise distributions from trajectory data, we need a way to perform risk-sensitive control under this uncertainty. We begin by first formalizing our definition of safety. We define the safety of a system to be the forward invariance of some user-defined "safe set", $\mathcal{C}$, as is common in robotics and control theory [37]–[40]

Definition 1 (Forward Invariance and Safety). *A set $\mathcal{C} \subset \mathbb{R}^n$ is forward invariant for the system (2) if $\mathbf{x}_0 \in \mathcal{C}$ implies that $\mathbf{x}_k \in \mathcal{C}$ for all $k \in \mathbb{N}$. In this case, we call system (2) safe with respect to the set $\mathcal{C}$.*

A. Deterministic Safety with DTCBFs

Discrete-time control barrier functions (DTCBFs) are tools for guaranteeing the safety of discrete time systems. Consider a set $\mathcal{C} \triangleq \{\mathbf{x} \in \mathbb{R}^n : h(\mathbf{x}) \geq 0\}$ expressed as the 0-superlevel set of a continuous function $h : \mathbb{R}^n \to \mathbb{R}$.

Definition 2 (Discrete-Time Control Barrier Function (DTCBF) [41]). *Let $\mathcal{C} \subset \mathbb{R}^n$ be the 0-superlevel set of a continuous function $h : \mathbb{R}^n \to \mathbb{R}$. The function h is a discrete-time control barrier function (DTCBF) for (1) on $\mathcal{C}$ if there exists an $\alpha \in [0, 1]$ such that for each $\mathbf{x} \in \mathbb{R}^n$, there exists a $\mathbf{u} \in \mathbb{R}^m$ such that:*

$$h(\mathbf{F}(\mathbf{x}, \mathbf{u})) \geq \alpha h(\mathbf{x}). \qquad (8)$$

Given a CBF h for (1) and a corresponding $\alpha \in [0, 1]$, we define the point-wise set of control values:

$$\mathscr{K}_{\text{CBF}}(\mathbf{x}) = \{\mathbf{u} \in \mathbb{R}^m : h(\mathbf{F}(\mathbf{x}, \mathbf{u})) \geq \alpha h(\mathbf{x})\} \qquad (9)$$

This yields the following theoretical result:

Theorem 1. *([41]) Let $\mathcal{C} \subset \mathbb{R}^n$ be the 0-superlevel set of a continuous function $h : \mathbb{R}^n \to \mathbb{R}$. If h is a DTCBF for (1) on $\mathcal{C}$, then the set $\mathscr{K}_{\text{CBF}}(\mathbf{x})$ is non-empty for all $\mathbf{x} \in \mathbb{R}^n$, and*

for any continuous state-feedback controller $\mathbf{k}$ *with* $\mathbf{k}(\mathbf{x}) \in \mathcal{K}_{\mathrm{CBF}}(\mathbf{x})$ *for all* $\mathbf{x} \in \mathbb{R}^n$, *then the system:*

$$\mathbf{x}_{k+1} = \mathbf{F}(\mathbf{x}_k, \mathbf{k}(\mathbf{x}_k)) \tag{10}$$

is safe with respect to $\mathcal{C}$.

Intuitively, the value of $h(\mathbf{x}_k)$ can only decay as fast as the geometric sequence $\alpha^k h(\mathbf{x}_0)$, which is lower-bounded by 0, thus ensuring the safety (i.e., forward invariance) of $\mathcal{C}$. This inequality mimics that of a discrete-time Lyapunov function [42] and similarly regulates the evolution of h based on its previous value.

Given a continuous nominal controller $\mathbf{k}_{\mathrm{nom}} : \mathbb{R}^n \times \mathbb{N} \to \mathbb{R}^m$ and a DTCBF h for (1) on $\mathcal{C}$, a controller $\mathbf{k}$ satisfying $\mathbf{k}(\mathbf{x}, k) \in \mathcal{K}_{\mathrm{CBF}}(\mathbf{x})$ for all $\mathbf{x} \in \mathbb{R}^n$ and $k \in \mathbb{N}$ can be specified via the following optimization problem (assuming feasibility for all $\mathbf{x} \in \mathcal{C}$):

$$\mathbf{k}(\mathbf{x}) = \underset{\mathbf{u} \in \mathbb{R}^m}{\operatorname{argmin}} \quad \|u - \mathbf{k}_{\mathrm{nom}}(\mathbf{x}, k)\|^2 \tag{11}$$
$$\text{s.t.} \quad h(\mathbf{F}(\mathbf{x}, \mathbf{u})) \geq \alpha h(\mathbf{x})$$

We note that unlike the continuous-time CBF constraint [38], the DTCBF inquality constraint (8) is not necessarily convex with respect to the input, preventing it from being integrated into a convex optimization-based controller. To solve this issue, it is often assumed that the function $h \circ \mathbf{F} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is concave with respect to its second argument [22], [41], [43]. This assumption was shown to be well motivated for concave h and sufficiently fast sampling times [44].

B. Stochastic Safety with DTCBFs

For this work we are interested in generalizing beyond deterministic DTCBFs by ensuring safety in the presence of unbounded disturbances, where the disturbance $\mathbf{d}_k$ is sampled from some probability distribution which may be a function of the state ($\mathbf{d}_k \sim p(\mathbf{d}_k | \mathbf{x}_k)$).

Since system (2) will almost surely leave any compact set as time goes to infinity [45], [46], we focus on bounding safety probabilities for a finite time horizon.

Definition 3 (*K*-Step Exit Probability [23]). *Let* $h : \mathbb{R}^N \to \mathbb{R}$ *be a continuous function. For any* $K \in \mathbb{N}$, *and initial condition* $\mathbf{x}_0 \in \mathbb{R}^n$, *the K-step exit probability of the closed-loop system* (10) *is given by:*

$$P_u(K, \mathbf{x}_0) = \mathbb{P}\left\{ \min_{0 \leq k \leq K} h(\mathbf{x}_k) < 0 \right\} \tag{12}$$

In particular, this describes the probability that the system will leave $\mathcal{C}$ within K steps. We now present a key result that bounds this exit probability when a CBF condition is imposed in expectation.

Theorem 2 (Thm. 5 [23]). *Let* $h : \mathbb{R}^n \to \mathbb{R}$ *be a continuous, upper-bounded function with upper bound* $M \in \mathbb{R}_{>0}$. *If there exists an* $\alpha \in (0, 1)$ *such that the closed-loop system* (2) *satisfies:*

$$\mathbb{E}[\ h(\mathbf{F}(\mathbf{x}, \mathbf{k}(\mathbf{x})) + \mathbf{d}) \mid \mathbf{x}\ ] \geq \alpha h(\mathbf{x}), \tag{13}$$
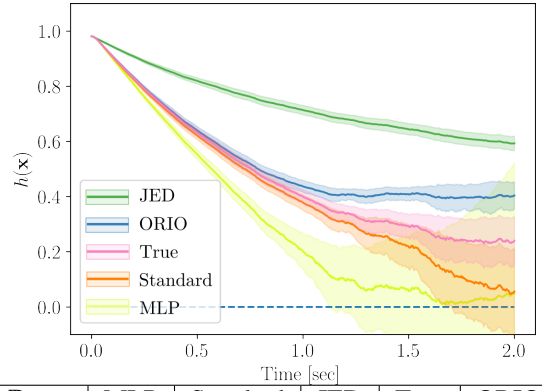
$\mathbb{P}_{\mathrm{bound}}$	MLP	Standard	JED	True	ORIO
0.82	0.69	0.56	0.00	0.32	0.11

Fig. 4. Quadrotor simulation Results. (**Figure**) The mean of 100 trajectories for each controller is plotted with 1/2 standard deviation around it. (**Table**) The $K-$step probability bound for the 2 second long trial from Thm. 3 and the approximated $K-$step probability experienced on in simulation over 100 trials.

for all $\mathbf{x} \in \mathbb{R}^n$, *with* $\mathbf{d} \sim p(\mathbf{d}|\mathbf{x})$, *then for any* $K \in \mathbb{N}$:

$$P_u(K, \mathbf{x}_0) \leq 1 - \frac{h(\mathbf{x}_0)}{M}\alpha^K. \tag{14}$$

Since this constraint contains the expectation of a nonlinear function of the disturbance distribution, it may be difficult to compute. To this end [23] provides control methods for ensuring that inequality is satisfied for systems where the disturbance mean and covariance are known:

Theorem 3 (Thm. 6 [23]). *Consider the system* (2) *and let* $h : \mathbb{R}^n \to \mathbb{R}$ *be a twice-continuously differentiable, concave function such that* $\sup_{\mathbf{x} \in \mathbb{R}^n} h(\mathbf{x}) \leq M$ *for* $M \in \mathbb{R}_{>0}$ *and* $\sup_{\mathbf{x} \in \mathbb{R}^n} \|\nabla^2 h(\mathbf{x})\|_2 \leq \lambda_{\max}$ *for* $\lambda_{\max} \in \mathbb{R}_{\geq 0}$. *Suppose there exists an* $\alpha \in (0, 1)$ *such that:*

$$h(\mathbf{F}(\mathbf{x}, \mathbf{k}(\mathbf{x})) + \mathbb{E}[\mathbf{d}|\mathbf{x}]) - \frac{\lambda_{\max}}{2}\mathrm{tr}(\mathrm{cov}(\mathbf{d}|\mathbf{x})) \geq \alpha h(\mathbf{x}) \tag{15}$$

for all $\mathbf{x} \in \mathcal{C}$ *with* $\mathbf{d} \sim p(\mathbf{d}|\mathbf{x})$. *Then we have that:*

$$\mathbb{E}[\ h(\mathbf{F}(\mathbf{x}, \mathbf{k}(\mathbf{x})) + \mathbf{d}) \mid \mathbf{x}\ ] \geq \alpha h(\mathbf{x}), \tag{16}$$

for all $\mathbf{x} \in \mathcal{C}$ *with* $\mathbf{d} \sim p(\mathbf{d}|\mathbf{x})$.

This provides a tractable way for enforcing safety once a model of the conditional mean $\mathbb{E}[\mathbf{d}|\mathbf{x}]$ and covariance $\mathrm{cov}(\mathbf{d}|\mathbf{x})$ is known.

In particular we propose the ORIO (Online Risk-Informed Optimization) controller:

$$\mathbf{k}(\mathbf{x}) = \underset{\mathbf{u} \in \mathbb{R}^m}{\operatorname{argmin}} \|u - \mathbf{k}_{\mathrm{nom}}(\mathbf{x}, k)\|^2 \tag{ORIO}$$
$$\text{s.t. } h(\mathbf{F}(\mathbf{x}, \mathbf{u}) + \overline{\boldsymbol{\mu}}(\mathbf{x}))$$
$$- \frac{\lambda_{\max}}{2}\mathrm{tr}(\overline{\boldsymbol{\Sigma}}(\mathbf{x})) \geq \alpha h(\mathbf{x})$$

where we approximate $\mathbb{E}[\mathbf{d}|\mathbf{x}]$ and $\mathrm{cov}(\mathbf{d}|\mathbf{x})$ using the outputs, $\overline{\boldsymbol{\mu}}_{(\phi, \theta)}(\mathbf{x})$ and $\overline{\boldsymbol{\Sigma}}_{(\phi, \theta)}(\mathbf{x})$, of the CVAE and use those approximations in conjunction with the simplified DTCBF constraint (15) which endows a system with the K-step exit probability guaranteed by Thm 2.

IV. SAFETY FOR QUADROTOR DRONE: THEORY AND EXPERIMENTS

In this section we show the utility of our method which combines the usefulness of CVAEs as tools for learning complex dynamics residual distributions and our controller (ORIO) which leverages those generative models for safe control. This framework is applied to safe flight on a quadrotor drone, and evaluated in both simulation and hardware.

A. Modeling and DTCBF Synthesis for a Multirotor Drone

We consider the safety of a quadrotor drone. We model the dynamics of this system as:

$$\underbrace{\frac{d}{dt}\begin{bmatrix}\mathbf{p}\\\mathbf{q}\\\mathbf{v}\end{bmatrix}}_{\dot{\mathbf{x}}} = \begin{bmatrix}\mathbf{v}\\\mathbf{0}\\-\mathbf{e}_z g\end{bmatrix} + \begin{bmatrix}\mathbf{0}\\\boldsymbol{\omega}\\\frac{1}{m}\mathbf{R}(\mathbf{q})\mathbf{e}_z\tau\end{bmatrix} \quad (17)$$

where the state $\mathbf{x} = (\mathbf{p} \in \mathbb{R}^3, \mathbf{q} \in \mathrm{SO}(3), \mathbf{v} \in \mathbb{R}^3)$ represents the position, orientation, and velocity, g is gravity, m is the drone mass, and the system has inputs of angular rate $\boldsymbol{\omega} \in \mathfrak{so}(3)$ and thrust force $\tau \in \mathbb{R}$. Here $\mathbf{e}_z$ is a unit vector in the z direction and $\mathbf{R} : \mathrm{SO}(3) \to \mathbb{R}^{3\times3}$ maps the quaternion representation of orientation to the respective rotation matrix. For simulation, these dynamics are approximated in discrete time using Euler integration on manifolds and for the dynamics approximation in the DTCBF, standard Euler integration is used to ease computation, [47] shows that this approximation is theoretically well justified for DTCBFs with short time steps.

The safety criteria for our quadrotor is to avoid collisions with the ground or roof. We can encode this safety as the 0-superlevel set of the function:

$$h_{\text{des}}(\mathbf{x}) = C - \boldsymbol{\zeta}^\top \mathbf{P}\boldsymbol{\zeta} \quad (18)$$

for some $C > 0$ where $\boldsymbol{\zeta} = \begin{bmatrix} z - z_0, & v_z \end{bmatrix}$ and $V(\boldsymbol{\zeta}) = \boldsymbol{\zeta}^T \mathbf{P}\boldsymbol{\zeta}$ is a Lyapunov function generated by the Discrete-time Algebraic Ricatti Equation (DARE) for discrete-time double integrator dynamics. However, this is not necessarily a DTCBF since the quadrotor's orientation may render it unable to track double integrator trajectories.

To avoid this problem, we add an penalty term to ensure correct orientation when $h(\mathbf{x}) = 0$.

$$h(\mathbf{x}) = (C - \boldsymbol{\zeta}^\top \mathbf{P}\boldsymbol{\zeta}) - \lambda(1 - \mathbf{e}_z R(\mathbf{q})\mathbf{e}_z) \quad (19)$$

This DTCBF is motivated by the differential flatness of the multi-rotor dynamics [48] since the system can track linear system trajectories.

Importantly, this is a valid DTCBF for the Euler-approximated dynamics and there are bounds for $\overline{\mu}$ and $\frac{\lambda_{\max}}{2}\mathrm{tr}(\overline{\boldsymbol{\Sigma}}(\mathbf{x}))$ such that ORIO controller is feasible for all $\mathbf{x}$ such that $h(\mathbf{x}) \geq 0$. This is stated formally in the following theorem:

Theorem 4. *Consider h as in (19) for $\alpha > 0$. If $C \geq 2\lambda$, then there exists $\mathbf{u} \in \mathbb{R}^4$ and $M_{\boldsymbol{\delta}}, M_{\mathbf{c}} > 0$ such that*

$$h(\mathbf{F}_{\text{Eul}}(\mathbf{x}, \mathbf{u}) + \boldsymbol{\delta}) + c \geq \alpha h(\mathbf{x}) \quad (20)$$

for all $\boldsymbol{\delta} < M_\delta, c < M_c$ and all $\mathbf{x}$ such that $h(\mathbf{x}) \geq 0$.

See the appendix in section VI for the proof of this theorem.

B. Comparison Controllers

For comparison, we implement several controllers in addition to ORIO. Each controller has the structure:

$$\mathbf{k}(\mathbf{x}) = \underset{\mathbf{u}\in\mathbb{R}^m}{\arg\min} \quad \|u - \mathbf{k}_{\text{Nom}}(\mathbf{x}, k)\|^2 \quad (21)$$
$$\text{s.t.} \quad h(\mathbf{F}_{\text{Eul}}(\mathbf{x}, \mathbf{u}) + \mathbf{m}(\mathbf{x})) - c(\mathbf{x}) \geq \alpha h(\mathbf{x})$$

with the following ablations:

- **(Standard)** where $\mathbf{m}(\mathbf{x}) = 0$ and $c(\mathbf{x}) = 0$. This is the standard DTCBF controller [41] where the modeled dynamics are assumed to be correct.
- **(JED)** where $\mathbf{m}(\mathbf{x})$ is the constant sample mean of the $\mathcal{D}$ and $c(\mathbf{x})$ is the trace of the sample covariance times $\sup_{\mathbf{x}\in\mathbb{R}^n} \|\nabla^2 h(\mathbf{x})\|_2$. This is the "Jensen-Enhanced DTCBF" as presented in [23].
- **(MLP)** where $\mathbf{m}(\mathbf{x})$ is an MLP that is trained on the dataset $\mathcal{D}$ to approximate the dynamics residuals and $c(\mathbf{x}) = 0$.
- **(True)** where $\mathbf{m}(\mathbf{x})$ is the true dynamics residual mean and $c(\mathbf{x})$ is the trace of the true covariance times $\sup_{\mathbf{x}\in\mathbb{R}^n} \|\nabla^2 h(\mathbf{x})\|_2$.

C. Simulation Results

For simulation we use the dynamics residual model:

$$p(\mathbf{d}|\mathbf{x}) = \mathcal{N}\left(\mathbf{d}; \underbrace{\mathbf{0}_9}_{\boldsymbol{\mu}(\mathbf{x})}, \underbrace{I_9 \times (1 + 50e^{-30z^2}) \times 10^{-5}}_{\text{cov}(\mathbf{x})}\right) \quad (22)$$

where the disturbance grows as the drone approaches the ground to approximate complicated ground effects.

We fly the drone using an SE(3) stabilization controller [49] from 1 meter in the air to the ground 20 times for 2 seconds each with a control and data collection frequency of 333Hz to collect training data (13320 data points). Each controller was simulated for 100 two-second long trajectories at 333Hz with $\alpha = 0.9975$. Results for these simulations are shown in Fig. 2. The looseness of the probability bound is in part due to the fact that the covariance is small for a large portion of the trajectory, which is not accounted for by the martingale-based bound. Despite the risk probability bound not being tight (ORIO), produces behavior which is similar to (True) and which is less conservative than (JED) while still being more robust than (Standard) and (MLP).

D. Hardware Result

Finally, we deploy our risk-informed controller (ORIO) on a quadrotor drone flying aggressively near the ground. For all tests, we provide the drone with real-time pose measurements from a motion capture system. The drone is equipped with a Nvidia Tx2 that is used to perform onboard computation of all neural network forward passes and optimization-based controllers. The mean and covariance of the dynamics residuals are approximated using the CVAE
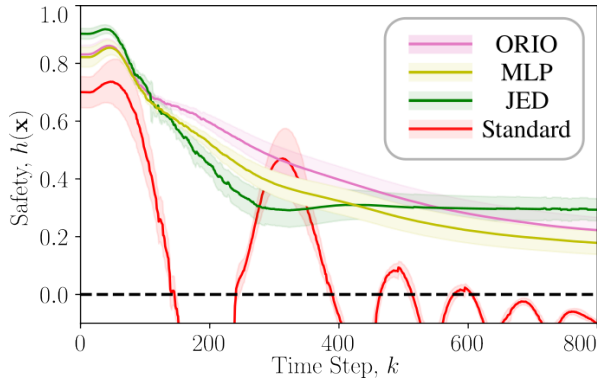
Fig. 5. Mean and one standard deviation of $h(\mathbf{x})$ for the "drop" test case in hardware, which drops the drone with $\mathbf{k}_{\text{nom}} = 0$ from a height of roughly two meters. We compare our proposed controller (ORIO) with three ablations: a deterministic MLP, a simple aggregate across all trajectory data (JED), and a standard CBF controller. All controllers except the standard CBF satisfy the safety constraint $h(\mathbf{x}) \geq 0$, where the standard CBF fails due to the unmodelled dynamics. While the residual dynamics include complex aerodynamic effects, in this case they are low-variance, and so we expect both the MLP and (ORIO) to perform similarly.

with 200 samples at 100 Hz and the optimization problem in ORIO is an SOCP which is solved using en embedded conic solver [50] at 300 Hz. Approximately 2 minutes of training data is collected via human-operated flight for both experiments.

Our first experiment is a drop test (shown in Figure 1) where we drop the drone from a hover at approximately 2m, and enforce the barrier constraint (19) with $\alpha = 0.9975$ for positions above the ground; this case has low noise but requires accurate estimation of the quadrotor's thrust / ground effects for the barrier to be effective in preventing ground collision. Figure 5 plots the mean barrier value $h(\mathbf{x})$ over at least 50 trials for each ablation of our method, with one standard deviation shaded around the mean. All controllers except the standard CBF (which nearly immediately becomes unsafe due to the unmodeled dynamics) exhibit conservative behavior and settle relatively far from the boundary. Of particular interest is the extremely similar behavior of the simple MLP and CVAE methods; this result is intuitive since the low-variance disturbance source allows the MLP to accurately capture the unmodeled dynamics. This provides an interesting insight: learning residual dynamics via simple regression, as proposed in [28], [29], [51], is well-posed for systems subject to deterministic, low-variance disturbances, and can yield safe, performant behavior without reasoning about stochasticity.

In our second test, the quadrotor is carrying a slung, unmodeled load of 0.55kg, which induces large disturbances that are not uniquely determined by the current state $\mathbf{x}$; this test requires the residual dynamics models to capture high-variance behavior to accurately model the slung load's effect on the dynamics. Here we again enforce the barrier constraint (19) which is adjusted to prevent the slung mass from contacting the ground and which has $\alpha = 0.995$. For this test, because all controllers can only condition their disturbance on the current state (which does not include the position or velocity of the slung load), the disturbances
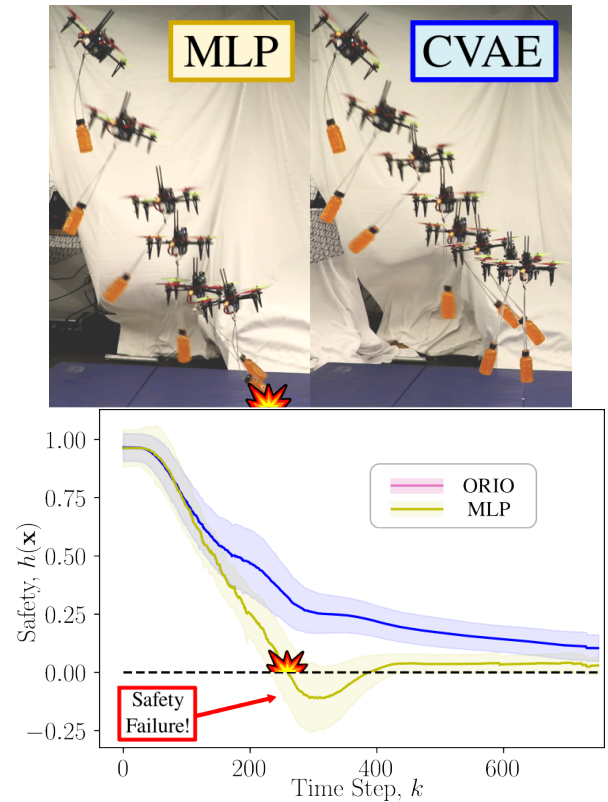

Fig. 6. (Top) The drone is dropped from the top left and moves to the right as it falls while carrying an orange payload. The left shows a failure case when controlled by the MLP controller and the right shows a success from the same initial condition when controlled by the ORIO controller. **(Bottom)** The average of 14 trajectories is plotted with one standard deviation shading. The ORIO controller successfully keeps the system safe while the MLP-based controller results in safety failures. The video of these experiments can be found here.

appear to be random and high variance, since they are large and depend on the history of states visited previously. Here we compare only our proposed method (ORIO) and the MLP; as expected, in this noisy case the our CVAE-based method performs remarkably better as seen in Fig. 6, and has no safety violations. A video of the experiments can be found with our code here. This experiment demonstrates both that the CVAE can learn an accurate stochastic model of highly noisy dynamics (including trajectories where the slung load reached nearly 90 degree angles dynamics), and also is necessary to ensure safety in such noisy cases.

V. Conclusions

We present a unified framework for risk-senstive control that combines CVAEs, which learn stochastic disturbance models from trajectory data, with DTCBFs, which provide probabilistic safety guarantees for stochastic systems. We demonstrate the real-time utility of this framework by running the full pipeline (after training) at 100Hz onboard a quadrotor drone performing aggressive flight, including a free fall and flight with a slung load.

Future work involves extending the method to handle other forms of uncertainty such as those generated by perception, reduced order models, contact dynamics, or human-robot-interaction.

VI. Appendix

First we restate Theorem 4:

Theorem. *Consider h as in (19) for $\alpha > 0$. If $C \geq 2\lambda$, then there exists $\mathbf{u} \in \mathbb{R}^4$ and $M_{\boldsymbol{\delta}}, M_{\boldsymbol{c}} > 0$ such that*

$$h(\mathbf{F}_{\text{Eul}}(\mathbf{x}, \mathbf{u}) + \boldsymbol{\delta}) + c \geq \alpha h(\mathbf{x}) \tag{23}$$

for all $\boldsymbol{\delta} < M_\delta, c < M_c$ and all $\mathbf{x}$ such that $h(\mathbf{x}) \geq 0$.

Now we state the proof of this theorem. This proof relies on the differential-flatness of the quadrotor system by considering two cases: one when the drone's thrust vector is at least partially aligned with the z-direction and the other when it is normal to the z-direction. By finding feasible control actions in both cases, we prove the feasibility of the constraint in general.

Importantly, this proof relies on the use of the Euler approximation of the drone dynamics with the rotation-matrix representation of the system orientation. The effect of using this dynamics approximation was analyzed in [47].

Proof. Let $\varphi(\mathbf{R}) \triangleq \widehat{\mathbf{e}}_z^\top \mathbf{R} \widehat{\mathbf{e}}_z$.

- If $\varphi(\mathbf{R}) \neq 0$, then $\tau = \frac{1}{\varphi(\mathbf{R})}\left(\Delta_t g - \mathbf{k}_{\text{DARE}}(\boldsymbol{\zeta}_k)\right)$ with $[\boldsymbol{\omega}]_\times = -\frac{1}{\Delta_t}\delta_3$ yields:

$$h(\mathbf{F}_{\text{Eul}}(\mathbf{x},\mathbf{u})+\boldsymbol{\delta}) \geq h(\mathbf{x}) + \boldsymbol{\zeta}^\top Q \boldsymbol{\zeta} - \gamma\left(\left\|\begin{bmatrix}\mathbf{e}_z^\top \boldsymbol{\delta_1} \\ \mathbf{e}_z^\top \boldsymbol{\delta_3}\end{bmatrix}\right\|\right) \tag{24}$$

The inequality comes from using the scaled DARE controller and the existence of $\gamma \in \mathcal{K}$ is guaranteed by the exponential stability nominal system using the DARE controller. Since $(Q - (1-\alpha)P) \in \mathbb{S}_{>0}^n$ and $R \in SO(3)$, we can expand this inequality as:

$$\geq \alpha h(\mathbf{x}) + (1-\alpha)(C - 2\lambda) - \gamma\left(\left\|\begin{bmatrix}\mathbf{e}_z^\top \boldsymbol{\delta_1} \\ \mathbf{e}_z^\top \boldsymbol{\delta_3}\end{bmatrix}\right\|\right) \tag{25}$$

Pick $M_{\boldsymbol{\delta}} < \gamma^{-1}((1-\alpha)(C-2\lambda))$ and $M_c \leq (1-\alpha)(C-2\lambda) - \gamma(M_\delta)$, then

$$h(\mathbf{F}_{\text{Eul}}(\mathbf{x},\mathbf{u})+\boldsymbol{\delta}) \geq \alpha h(\mathbf{x}) + c \tag{26}$$

- If $\varphi(\mathbf{R}) = 0$, then affect of ω on $h(\mathbf{F}_{\text{Eul}}(\mathbf{x},\mathbf{u})+\boldsymbol{\delta})$ is

$$\lambda \Delta_t \mathbf{e}_z R[\boldsymbol{\omega}]_\times \mathbf{e}_z = \mathbf{e}_z \mathbf{R}\boldsymbol{\varpi} \tag{27}$$

where $\boldsymbol{\varpi} = \begin{bmatrix}\omega_x & \omega_y & 0\end{bmatrix}^\top$. Since $\varphi(\mathbf{R}_k) = 0$, $\mathbf{e}_z \mathbf{R}_k$ is perpendicular to $\mathbf{e}_z$ and thus there exists ω_x, ω_y such that: $h(\mathbf{x}_{k+1}) = \alpha h(\mathbf{x}_k) + c$ for any $c \in \mathbb{R}$. $\qquad\square$

References

[1] D. J. Rezende and S. Mohamed, "Variational Inference with Normalizing Flows."

[2] D. P. Kingma, T. Salimans, B. Poole, and J. Ho, "Variational Diffusion Models."

[3] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," Dec. 2022, arXiv:1312.6114 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1312.6114

[4] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, "Zero-shot text-to-image generation," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 8821–8831. [Online]. Available: https://proceedings.mlr.press/v139/ramesh21a.html

[5] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving Language Understanding by Generative Pre-Training."

[6] X. Huang, Z. Li, Y. Xiang, Y. Ni, Y. Chi, Y. Li, L. Yang, X. B. Peng, and K. Sreenath, "Creating a Dynamic Quadrupedal Robotic Goalkeeper with Reinforcement Learning," arXiv, Tech. Rep. arXiv:2210.04435, Oct. 2022, arXiv:2210.04435 [cs, eess] type: article. [Online]. Available: http://arxiv.org/abs/2210.04435

[7] J. Urain, M. Ginesi, D. Tateo, and J. Peters, "ImitationFlow: Learning Deep Stable Stochastic Dynamic Systems by Normalizing Flows," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Las Vegas, NV, USA: IEEE, Oct. 2020, pp. 5231–5237. [Online]. Available: https://ieeexplore.ieee.org/document/9341035/

[8] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, "Diffusion Policy: Visuomotor Policy Learning via Action Diffusion," in *Robotics: Science and Systems XIX*. Robotics: Science and Systems Foundation, July 2023. [Online]. Available: http://www.roboticsproceedings.org/rss19/p026.pdf

[9] J. Carvalho, "Motion Planning Diffusion: Learning and Planning of Robot Motions with Diffusion Models."

[10] B. Ichter, J. Harrison, and M. Pavone, "Learning Sampling Distributions for Robot Motion Planning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 7087–7094, iSSN: 2577-087X.

[11] W. Chi, G. Dagnino, T. M. Y. Kwok, A. Nguyen, D. Kundrat, M. E. M. K. Abdelaziz, C. Riga, C. Bicknell, and G.-Z. Yang, "Collaborative Robot-Assisted Endovascular Catheterization with Generative Adversarial Imitation Learning," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. Paris, France: IEEE, May 2020, pp. 2414–2420. [Online]. Available: https://ieeexplore.ieee.org/document/9196912/

[12] A. Xie, D. P. Losey, R. Tolsma, C. Finn, and D. Sadigh, "Learning Latent Representations to Influence Multi-Agent Interaction."

[13] D. Park, Y. Hoshi, and C. C. Kemp, "A Multimodal Anomaly Detector for Robot-Assisted Feeding Using an LSTM-Based Variational Autoencoder," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1544–1551, July 2018. [Online]. Available: http://ieeexplore.ieee.org/document/8279425/

[14] T. Inoue, S. Chaudhury, G. D. Magistris, and S. Dasgupta, "Transfer learning from synthetic to real images using variational autoencoders for robotic applications."

[15] H. Ren and P. Ben-Tzvi, "Learning inverse kinematics and dynamics of a robotic manipulator using generative adversarial networks," *Robotics and Autonomous Systems*, vol. 124, p. 103386, Feb. 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0921889019303501

[16] M. Watter, J. Springenberg, J. Boedecker, and M. Riedmiller, "Embed to control: A locally linear latent dynamics model for control from raw images," in *Advances in neural information processing systems*, 2015, pp. 2746–2754.

[17] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi, "Dream to control: Learning behaviors by latent imagination," *arXiv preprint arXiv:1912.01603*, 2019.

[18] K. Sohn, H. Lee, and X. Yan, "Learning Structured Output Representation using Deep Conditional Generative Models."

[19] B. Ivanovic, K. Leung, E. Schmerling, and M. Pavone, "Multimodal Deep Generative Models for Trajectory Prediction: A Conditional Variational Autoencoder Approach," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 295–302, Apr. 2021, conference Name: IEEE Robotics and Automation Letters.

[20] T. Barbié, T. Nishio, T. Nishida, and Department of Mechanical and Control Engineering, Kyushu Institute of Technology 1-1 Sensui, Tobata, Kitakyushu, Fukuoka 804-8550, Japan, "Trajectory Prediction with a Conditional Variational Autoencoder," *Journal of Robotics and Mechatronics*, vol. 31, no. 3, pp. 493–499, June 2019. [Online]. Available: https://www.fujipress.jp/jrm/rb/robot003100030493

[21] M. Noseworthy, R. Paul, S. Roy, D. Park, and N. Roy, "Task-

Conditioned Variational Autoencoders for Learning Movement Primitives."

[22] M. Ahmadi, A. Singletary, J. W. Burdick, and A. D. Ames, "Barrier Functions for Multiagent-POMDPs with DTL Specifications," in *2020 59th IEEE Conference on Decision and Control (CDC)*, Dec. 2020, pp. 1380–1385, iSSN: 2576-2370.

[23] R. K. Cosner, P. Culbertson, A. J. Taylor, and A. D. Ames, "Robust Safety under Stochastic Uncertainty with Discrete-Time Control Barrier Functions," Feb. 2023, arXiv:2302.07469 [cs, eess]. [Online]. Available: http://arxiv.org/abs/2302.07469

[24] S. X. Wei, A. Dixit, S. Tomar, and J. W. Burdick, "Moving Obstacle Avoidance: a Data-Driven Risk-Aware Approach," Mar. 2022, arXiv:2203.14913 [cs]. [Online]. Available: http://arxiv.org/abs/2203.14913

[25] E. Sontag, "On the Input-to-State Stability Property," *European Journal of Control*, vol. 1, Nov. 1995.

[26] R. K. Cosner, A. W. Singletary, A. J. Taylor, T. G. Molnar, K. L. Bouman, and A. D. Ames, "Measurement-Robust Control Barrier Functions: Certainty in Safety with Uncertainty in State," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Prague, Czech Republic: IEEE, Sept. 2021, pp. 6286–6291. [Online]. Available: https://ieeexplore.ieee.org/document/9636584/

[27] S. Bansal, M. Chen, S. Herbert, and C. J. Tomlin, "Hamilton-Jacobi Reachability: A Brief Overview and Recent Advances," Sept. 2017, arXiv:1709.07523 [cs, math]. [Online]. Available: http://arxiv.org/abs/1709.07523

[28] G. Shi, X. Shi, M. O'Connell, R. Yu, K. Azizzadenesheli, A. Anandkumar, Y. Yue, and S.-J. Chung, "Neural Lander: Stable Drone Landing Control Using Learned Dynamics," in *2019 International Conference on Robotics and Automation (ICRA)*, May 2019, pp. 9784–9790, iSSN: 2577-087X.

[29] M. O'Connell, G. Shi, X. Shi, K. Azizzadenesheli, A. Anandkumar, Y. Yue, and S.-J. Chung, "Neural-Fly Enables Rapid Learning for Agile Flight in Strong Winds," *Science Robotics*, vol. 7, no. 66, p. eabm6597, May 2022, arXiv:2205.06908 [cs, eess]. [Online]. Available: http://arxiv.org/abs/2205.06908

[30] S. V. Rakovic and D. Q. Mayne, "Set robust control invariance for linear discrete time systems," in *Proceedings of the 44th IEEE Conference on Decision and Control*. IEEE, 2005, pp. 975–980.

[31] D. L. Marruedo, T. Alamo, and E. F. Camacho, "Input-to-state stable mpc for constrained discrete-time nonlinear systems with bounded additive uncertainties," in *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, vol. 4. IEEE, 2002, pp. 4619–4624.

[32] J. Lam, Z. Shu, S. Xu, and E.-K. Boukas, "Robust control of descriptor discrete-time markovian jump systems," *International Journal of control*, vol. 80, no. 3, pp. 374–385, 2007.

[33] L. Xie, C. E. De Souza, and Y. Wang, "Robust control of discrete time uncertain dynamical systems," *Automatica*, vol. 29, no. 4, pp. 1133–1137, 1993.

[34] D. P. Kingma and M. Welling, "An Introduction to Variational Autoencoders," *Foundations and Trends® in Machine Learning*, vol. 12, no. 4, pp. 307–392, 2019, arXiv:1906.02691 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1906.02691

[35] A. E. Gelfand and A. F. M. Smith, "Sampling-based approaches to calculating marginal densities," *Journal of the American Statistical Association*, vol. 85, no. 410, pp. 398–409, 1990. [Online]. Available: http://www.jstor.org/stable/2289776

[36] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *arXiv preprint arxiv:2006.11239*, 2020.

[37] L. Brunke, M. Greeff, A. W. Hall, Z. Yuan, S. Zhou, J. Panerati, and A. P. Schoellig, "Safe Learning in Robotics: From Learning-Based Control to Safe Reinforcement Learning," *arXiv:2108.06266 [cs, eess]*, Aug. 2021, arXiv: 2108.06266. [Online]. Available: http://arxiv.org/abs/2108.06266

[38] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control Barrier Functions: Theory and Applications," in *2019 18th European Control Conference (ECC)*, June 2019, pp. 3420–3431.

[39] K. P. Wabersich and M. N. Zeilinger, "A predictive safety filter for learning-based control of constrained nonlinear dynamical systems."

[40] S. L. Herbert, M. Chen, S. Han, S. Bansal, J. F. Fisac, and C. J. Tomlin, "FaSTrack: a Modular Framework for Fast and Guaranteed Safe Motion Planning," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, Dec. 2017, pp. 1517–1522, arXiv:1703.07373 [cs]. [Online]. Available: http://arxiv.org/abs/1703.07373

[41] A. Agrawal and K. Sreenath, "Discrete Control Barrier Functions for Safety-Critical Control of Discrete Systems with Application to Bipedal Robot Navigation," in *Robotics: Science and Systems XIII*. Robotics: Science and Systems Foundation, July 2017. [Online]. Available: http://www.roboticsproceedings.org/rss13/p73.pdf

[42] N. Bof, R. Carli, and L. Schenato, "Lyapunov Theory for Discrete Time Systems," Sept. 2018, arXiv:1809.05289 [math]. [Online]. Available: http://arxiv.org/abs/1809.05289

[43] J. Zeng, B. Zhang, and K. Sreenath, "Safety-Critical Model Predictive Control with Discrete-Time Control Barrier Function," in *2021 American Control Conference (ACC)*. New Orleans, LA, USA: IEEE, May 2021, pp. 3882–3889. [Online]. Available: https://ieeexplore.ieee.org/document/9483029/

[44] A. J. Taylor, V. D. Dorobantu, R. K. Cosner, Y. Yue, and A. D. Ames, "Safety of Sampled-Data Systems with Control Barrier Functions via Approximate Discrete Time Models," June 2022, arXiv:2203.11470 [cs, eess]. [Online]. Available: http://arxiv.org/abs/2203.11470

[45] J. Steinhardt and R. Tedrake, "Finite-time regional verification of stochastic non-linear systems," *The International Journal of Robotics Research*, vol. 31, no. 7, pp. 901–923, June 2012, publisher: SAGE Publications Ltd STM. [Online]. Available: https://doi.org/10.1177/0278364912444146

[46] P. Culbertson, R. K. Cosner, M. Tucker, and A. D. Ames, "Input-to-State Stability in Probability," Apr. 2023, arXiv:2304.14578 [cs, eess]. [Online]. Available: http://arxiv.org/abs/2304.14578

[47] M. Tayal and S. Kolathaya, "Control Barrier Functions in Dynamic UAVs for Kinematic Obstacle Avoidance: A Collision Cone Approach," Mar. 2023, arXiv:2303.15871 [cs]. [Online]. Available: http://arxiv.org/abs/2303.15871

[48] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *2011 IEEE international conference on robotics and automation*. IEEE, 2011, pp. 2520–2525.

[49] T. Lee, M. Leok, and N. H. McClamroch, "Geometric tracking control of a quadrotor UAV on SE(3)," in *49th IEEE Conference on Decision and Control (CDC)*. Atlanta, GA: IEEE, Dec. 2010, pp. 5420–5425. [Online]. Available: http://ieeexplore.ieee.org/document/5717652/

[50] A. Domahidi, E. Chu, and S. Boyd, "Ecos: An socp solver for embedded systems," in *2013 European control conference (ECC)*. IEEE, 2013, pp. 3071–3076.

[51] A. Zeng, S. Song, J. Lee, A. Rodriguez, and T. Funkhouser, "TossingBot: Learning to Throw Arbitrary Objects with Residual Physics," May 2020, arXiv:1903.11239 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1903.11239