Safe Reinforcement Learning in Uncertain Contexts

Dominik Baumann and Thomas B. Schön, *Senior Member, IEEE*

***Abstract*—When deploying machine learning algorithms in the real world, guaranteeing safety is an essential asset. Existing safe learning approaches typically consider continuous variables, i.e., regression tasks. However, in practice, robotic systems are also subject to discrete, external environmental changes, e.g., having to carry objects of certain weights or operating on frozen, wet, or dry surfaces. Such influences can be modeled as discrete *context* variables. In the existing literature, such contexts are, if considered, mostly assumed to be known. In this work, we drop this assumption and show how we can perform safe learning when we cannot directly measure the context variables. To achieve this, we derive frequentist guarantees for multiclass classification, allowing us to estimate the current context from measurements. Furthermore, we propose an approach for identifying contexts through experiments. We discuss under which conditions we can retain theoretical guarantees and demonstrate the applicability of our algorithm on a Furuta pendulum with camera measurements of different weights that serve as contexts.**

***Index Terms*—Frequentist bounds, multiclass classification, safe reinforcement learning.**

I. Introduction

When learning on real and potentially expensive robotics hardware, respecting safety constraints is instrumental. In response to this need, safe learning algorithms have been developed that provide different forms of safety certificates [1]. One such algorithm is SafeOpt [2], which guarantees to find an optimal policy while never violating any safety constraints during the search with high probability.

Most safe learning algorithms consider only the internal dynamics of a robot (and possibly external constraints). However, when learning in the real world, robotic systems are subject to changes in their environment that influence their dynamics. Consider the illustrative example shown in Fig. 1. We have a Furuta pendulum and aim to learn a balancing policy. Different kinds of weights can be added to the top of the pole. These weights clearly influence the dynamics of the pendulum. However, the pendulum cannot control, which weight is added. Including the weight as an external parameter in the policy optimization is then challenging for two reasons. First, as the pendulum cannot control, which weight is added, it cannot actively generate informative data. Furthermore, we may have only a few examples (e.g., only three different weights in Fig. 1). This makes it hard to perform a regression. Second, directly including the weights in the policy optimization increases the dimensionality of the parameter space, which can be computationally demanding for safe learning algorithms. To account for that, Berkenkamp et al. [2] proposed to include such parameters as discrete context variables and showed that SafeOpt can successfully deal with a mixed continuous and discrete parameter space. Nevertheless, they assumed that the context variable, e.g., the weight, is known. In Fig. 1, we only have a camera that can capture the current

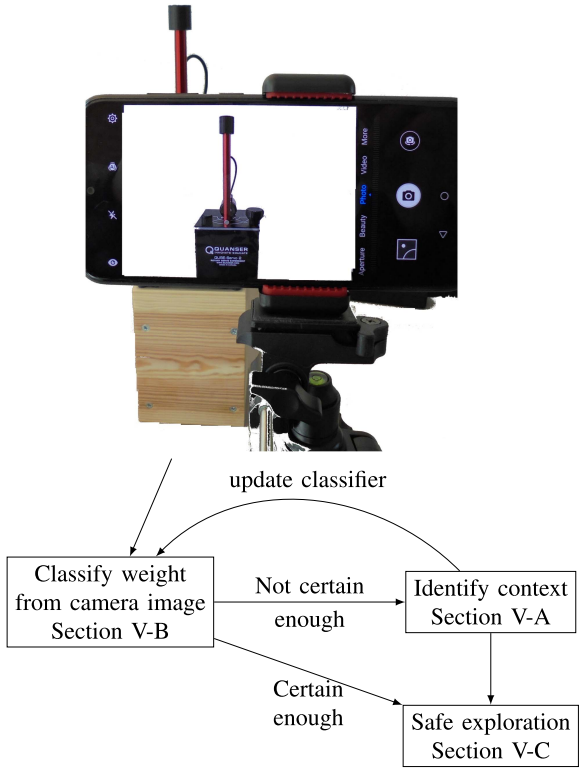

Fig. 1. Our experimental setup. We aim at optimizing a balancing controller for a Furuta pendulum whose dynamics can be altered by adding (removing) weights to (from) its pole. Our algorithm tries to infer the current weight from image data and resorts to identifying it through dedicated experiments if the image data is not sufficiently informative.

Manuscript received 26 June 2023; revised 29 October 2023; accepted 22 December 2023. Date of publication 15 January 2024; date of current version 22 February 2024. This paper was recommended for publication by Associate Editor M. Walter and Editor S. Behnke upon evaluation of the reviewers' comments. This work was supported in part by the Federal Ministry of Education and Research (BMBF) and the Ministry of Culture and Science of the German State of North Rhine-Westphalia (MKW) under the Excellence Strategy of the Federal Government and the Länder, in part by the project New LEADS - New Directions in Learning Dynamical Systems under Grant 621-2016-06079 funded by the Swedish Research Council, in part by the Kjell och Märta Beijer Foundation, and in part by the Mitsubishi Electric Research Laboratories (MERL) to purchase the equipment used for the experiments. *(Corresponding author: Dominik Baumann.)*

Dominik Baumann is with the Department of Electrical Engineering and Automation, Aalto University, 00076 Espoo, Finland, and also with the Department of Information Technology, Uppsala University, 75105 Uppsala, Sweden (e-mail: dominik.baumann@aalto.fi).

Thomas B. Schön is with the Department of Information Technology, Uppsala University, 75105 Uppsala, Sweden (e-mail: thomas.schon@it.uu.se).

Digital Object Identifier 10.1109/TRO.2024.3354176

weight. This is a typical industrial setting, where cameras are often used to track work pieces. Yet, inferring the weight of an object from image data is not possible in general [3].

An alternative to inferring the weight from data would be identifying the context through experiments. Such an algorithm is proposed in [4]. The downside of this approach is that it requires additional experimentation and cannot provide formal guarantees on identifying the correct context. Thus, it cannot straightforwardly be combined with a safe learning algorithm. Suppose we instead understand trajectories as samples from different probability distributions. In that case, we can investigate whether the distribution generating the current sample is identical to the distribution of a sample we saw in the past. One measure for comparing distributions is the maximum mean discrepancy (MMD) [5], whose applicability to dynamical systems has been discussed in [6]. In this article, we show how we can extend the formal guarantees the MMD provides for context identification in a safe learning algorithm.

Still, the problem of additional experimentation remains. Thus, over time, we would like to learn a classifier that can, at least in some cases, infer the current weight from image data. In particular, the classifier should output a probability that the camera image corresponds to a certain weight. This is a standard multiclass classification setting, for which various approaches exist, e.g., support vector machines [7], neural networks [8], or conditional mean embeddings (CMEs) [9]. However, if we want to combine them with a safe learning algorithm, we again require theoretical guarantees, which typical classification approaches cannot provide in the required form. Thus, we derive frequentist uncertainty bounds for multiclass classification based on CMEs and show how they can be used within a safe learning algorithm.

In summary, we propose an algorithm that mainly consists of two parts. We propose to train a classifier that yields a probabilistic score to relate, for instance, camera images to weights. For this classifier, we derive frequentist uncertainty bounds to use its output in an existing safe exploration algorithm, such as SafeOpt, and provide safety guarantees. If the uncertainty of our classifier is too high, we propose to identify the current context. Also here, we provide guarantees that can be used in safe exploration. Furthermore, we update our classifier to sharpen the uncertainty bounds whenever the current context has been identified.

The contributions of this article are as follows:

1) we derive frequentist uncertainty intervals for multiclass classification;
2) we develop a context identification method with statistical guarantees;
3) we combine multiclass classification with frequentist uncertainty intervals and context identification in an algorithm that allows for safe learning in uncertain contexts;
4) we evaluate the algorithm on a Furuta pendulum [10] with camera images of weights serving as contexts.

II. Related Work

This article proposes an algorithm for learning safe optimal policies in context-dependent dynamical systems with uncertain contexts. Here, we relate our contribution to the literature.

Safe learning: For a general overview of safe learning, we refer the reader to [1]. Most of these works focus on pure regression tasks without discrete parameters. Introducing discrete parameters as context variables initially emerged in the bandit setting [11], [12], [13], [14]. In [2], it was shown that the concept of discrete contexts could also benefit safe policy optimization for dynamical systems. In [2] and in the works on bandit settings, it is typically assumed that the current context is known to the learning agent. In reality, this may not always be the case. Thus, we here consider a setting where the agent receives some sensor information that it can use to infer the context, but it cannot directly measure the current context. Another approach that considers unknown contexts in a Bayesian framework is presented in [15]. However, the authors assume no information about the context, which differs from our setting, and they implicitly model the context. In contrast, we explicitly account for it in the learning problem.

Learning with unobserved context: Some works consider optimizing policies for context-dependent dynamical systems with unobserved contexts [15], [16]. The setting we consider is different in that we assume some information about the context to be available even though it cannot be unambiguously determined through the measurements. Other approaches, such as [17], assume access to some low-dimensional task parameter that is directly correlated with the context. This allows them to integrate the task parameter directly into the learning algorithm. In [17], the authors further assume that they can observe the task parameter over a range of values. Neither of these assumptions are satisfied in our setting. We assume access to a high-dimensional observation, such as image data, that is related to the context. Thus, we cannot directly include the observation in the learning algorithm due to the scalability properties of Gaussian process regression. Moreover, we consider a setting in which we receive discrete observations, such as images of traffic signs, where learning a classifier is more efficient than trying to learn a continuous model over all possible pixel values. In control theory, discrete contexts that alter the dynamics of a system may be interpreted as disturbances. Assuming knowledge of worst-case bounds for those disturbances, we can design robust controllers that provide stability guarantees while sacrificing performance [18]. We do not assume knowledge of worst-case bounds. Instead, we assume that there are measurements that can be used to estimate the current context. This idea is similar to disturbance observers that try to reconstruct the disturbance given the measurements and then compensate for them [19]. Both robust control techniques and disturbance observers require access to a mathematical model of the underlying system, while the approach we present herein is model-free.

Context identification: A dedicated context identification algorithm is proposed in [4]. Nevertheless, it lacks theoretical guarantees. Thus, we use the MMD initially introduced in [5] and extended in [6] to dynamical systems. Our main idea is that contexts change the dynamics of the system. Thus, by comparing the current trajectory with trajectories collected in the past, we can infer the current value of the context variable. This boils down to comparing the probability distributions generating the trajectories. For this, also other measures than the MMD are of

course possible; see [20] for an overview. We here choose the MMD since it allows us to compare probability distributions without actually estimating them, provides theoretical guarantees, and can be computed efficiently.

Classification: While various algorithms for classification exist, we here require one that provides input-dependent frequentist uncertainty intervals. This is especially important since classifiers tend to be overconfident [21], [22]. In the literature, we find general bounds for multiclass classification [9], [23], probably approximately correct bounds for Gaussian process classification [24], bounds for distribution-free calibration in binary classification [25], for general CMEs [26], [27], as well as their application to classification [9]. Nevertheless, all these bounds have in common that they are *uniform*. While uniform bounds are strong mathematical statements, they are not the most suitable objects for our problem setting. In particular, they typically only hold for the misclassification risk averaged over the entire input space. Instead, we require concrete statements about the misclassification risk at a particular input location. Intuitively, we should be more confident about our estimates in regions of the input space where we have already seen many data points. This is also pointed out as a potential extension in [9, Appendix]. Furthermore, existing learning theoretic bounds often aim to provide convergence rates, and actually applying those bounds to practical settings with finitely many data points is nontrivial. A main contribution of our work is the derivation of finite-sample frequentist bounds for classification with CMEs. Another approach that considers input-dependent convergence bounds is presented in [28]. In that work, the authors provide deterministic uncertainty intervals for kernel ridge and support vector regression, although not aiming at classification settings. In Section VI-D, we discuss how their bounds can be applied to classification and why they are noninformative in that setting.

III. Problem Setting and Background

We consider a setting in which a reinforcement learning (RL) agent seeks to optimize a policy but needs to guarantee that it satisfies safety guarantees throughout exploration. Furthermore, the dynamics are influenced by an unobserved context variable. Our main contributions are a context identification method with statistical guarantees and frequentist uncertainty intervals for multiclass classification that together allow the context to be inferred from external measurements. We propose to combine these two contributions with a safe learning algorithm. The concrete safe learning algorithm that is actually in charge of optimizing the policy is, thus, *independent* of our contributions, and basically, any RL algorithm that provides safety guarantees could be used. Nevertheless, to make the problem setting more concrete, we will here consider the algorithm from [2] and introduce this setting in Section III-A. The problem that we address in this article is formulated in Section III-B.

A. Background

We consider a dynamic system

$$x(k+1) = z(x(k), u(k), c) \tag{1}$$

with discrete time index $k \in \mathbb{N}$, state $x(k) \in \mathcal{X} \subseteq \mathbb{R}^\ell$, and input $u(k) \in \mathbb{R}^m$, whose dynamics depend on a context parameter $c \in \mathcal{C} \subseteq \mathbb{N}$. For this system, we want to learn a policy $u(k) = \pi(x(k), c, a)$, parametrized by parameters $a \in \mathcal{A} \subseteq \mathbb{R}^d$ that maximizes an unknown reward function $f : \mathcal{A} \times \mathcal{C} \to \mathbb{R}$ while guaranteeing safety. Safety is encoded through (unknown) constraint functions $g_i : \mathcal{A} \times \mathcal{C} \to \mathbb{R}$, $i \in \{1, \dots, q\}$.

We assume that the reward and constraint functions are unknown. However, we can receive (noisy) measurements of both by doing experiments: we select a parameterization a, perform an experiment, and afterward receive measurements of f and g_i for all $i \in \{1, \dots, q\}$. That way, we can, over time, find the optimum of f. Importantly, we seek to provide safety guarantees for each exploration experiment. Thus, the overall optimization problem is

$$\max_{a \in \mathcal{A}} f(a, c) \text{ s.t. } g_i(a, c) \geq 0 \text{ for all } i \in \{1, \dots, q\}. \tag{2}$$

We need a few assumptions to enable safe exploration despite unknown dynamics and constraints [2].

Assumption 1: The reward function f and the constraint functions g_i, with $i \in \{1, \dots, q\}$, have bounded norm in a reproducing kernel Hilbert space (RKHS).

Assumption 2: After each exploration experiment, we receive noisy measurements of the reward and constraint functions. Those measurements are perturbed by σ-sub Gaussian measurement noise.

Assumption 3: We are given at least one safe parameter vector a, for which we have, for all contexts $c \in \mathcal{C}$ and for all $i \in \{1, \dots, q\}$ that $g_i(a, c) > 0$ with probability at least $1 - \delta_{\text{safe}}$.

Assumption 3 may seem relatively strong as it requires parameter vectors that are safe under all contexts. It is required since we need some policy to start with, and at least during the first experiment, we cannot estimate the context without prior knowledge. In practice, such an initial safe parameter can have arbitrarily bad performance. For instance, when considering a mobile robot supposed to reach some target while not colliding with obstacles where the contexts are different surfaces, an initial safe policy could barely move the robot. Such a policy would be safe but have a close to minimum reward. Still, it would be enough as a starting point for our algorithm.

B. Problem Setting

In this work, we drop the often-adopted assumption that the discrete context parameter c is known or can be precisely measured. Instead, we assume that we receive measurements $y \in \mathcal{Y} \subseteq \mathbb{R}^s$ that reveal information about the context. These could be temperature measurements that induce a probability of whether or not the road is frozen for an autonomous car or camera images that allow us to reason about the weight of an object that a robot is supposed to manipulate. To automate this reasoning while still guaranteeing safety as above, we need an efficient way to estimate the probability of being in a particular context that itself also provides guarantees.

We define the underlying probability space as $(\Omega, \mathcal{F}, P)$ with random variables $Y : \Omega \to \mathcal{Y}$ and $C : \Omega \to \mathcal{C}$ that take values in

$\mathcal{Y}$ and $\mathcal{C}$, respectively, and whose respective probability distributions we denote by P_Y and P_C. In this work, we aim at estimating the probability $p_c(y) := \mathbb{P}(C = c \mid Y = y)$, i.e., the probability of context $c \in \mathcal{C} \subset \mathbb{N}$ given the current measurements $y \in \mathcal{Y} \subseteq \mathbb{R}^s$. Furthermore, we aim at deriving high probability uncertainty bounds for the estimate $\hat{p}_c$. Suppose we are given n tuples (y, c) with $n > 0$. For each context c, we generate a vector $\mathbf{c} \in \mathbb{R}^n$, where entry c_j is 1 if in the corresponding tuple of sample j the context was c_j and 0 otherwise. We aim at finding upper bounds $\epsilon_c(y, \delta, n)$ such that for each c we have with probability at least $1 - \delta_{\text{class}}$, where $\delta_{\text{class}} \in (0, 1)$

$$|p_c(y) - \hat{p}_c(y)| \leq \epsilon_c(y, \delta, n) \tag{3}$$

at the current input y.

However, in the beginning, when we have not gathered any data, the bounds (3) may be arbitrarily large and not enable confident classification decisions and, hence, meaningful safety guarantees. Thus, we additionally require a possibility to *identify* the current context. As with any identification procedure based on finitely many data points, this identification will come with some uncertainty. Suppose the current context is c^*. We then seek to guarantee that with probability at least $1 - \delta_{\text{MMD}}$, we have $\hat{c} = c^*$, where $\hat{c}$ is the identified or estimated context. By enhancing the dataset of (y, c) tuples with these estimated contexts $\hat{c}$, we can then, over time, create a dataset that we can use to train a classifier. Nevertheless, this classifier must acknowledge that each $\hat{c}$ is only correct with probability at least $1 - \delta_{\text{MMD}}$. Thus, we are tackling the following two problems.

1) Show that we can guarantee safety when we need to identify the context.
2) Incorporate uncertainty about identified contexts into the classifier and show that we can guarantee safety when our classification algorithm is confident enough about its decision.

IV. Preliminaries

Having introduced the problem setting, we now present the required mathematical foundations to develop the safe learning algorithm.

A. Context Identification

We need to identify the context whenever the probability estimate (5) for the current context is too low or too uncertain. For this, we collect trajectory data for each context that we encounter. Then, if we need to identify the current context, we use the safe policy from Assumption 3 to excite the system and compare the generated data with all collected trajectories. To compare trajectory data, we use the MMD [5]. Given a stored dataset X_c of context c and trajectory data X of the current context, we can calculate a finite-sample approximation of the squared MMD as

$$\text{MMD}^2(X, X_c) = \frac{1}{r^2} \sum_{i,j=1}^{r} k_{\text{mmd}}(X_i, X_j) + \frac{1}{r^2} \sum_{i,j=1}^{r} k_{\text{mmd}}(X_{c_i}, X_{c_j}) - \frac{2}{r^2} \sum_{i=1}^{r} \sum_{j=1}^{r} k_{\text{mmd}}(X_i, X_{c_j})$$

with $k_{\text{mmd}}(\cdot, \cdot)$ a characteristic kernel and r the length of the two datasets. For $r \to \infty$ and if the trajectory data were independent and identically distributed (i.i.d.), we could now guarantee that $\text{MMD}^2(X, X_c) = 0$ if, and only if, data samples X and X_c were generated in the same context [5]. However, data generated by dynamical systems are naturally non-i.i.d. To arrive at similar statements for our setting, we require an assumption on the trajectories created in each context.

Assumption 4: For any context $c \in \mathcal{C}$, X_c is stationary and there exists a time shift a^* and a threshold $\kappa(\epsilon, r)$ such that for $X_c = (x_c(a^*), x_c(2a^*), \ldots, x_c(na^*))$, we have

$$\mathbb{P}[\text{MMD}^2(X_c, \bar{X}_c) \geq \kappa] < \epsilon$$

where $\bar{X}_c$ is data from an independent trajectory.

The intuition behind this assumption is that if we subsample from the trajectory, we end up with approximately i.i.d. data. For a more detailed discussion on estimating a^* from data, we refer the reader to [6]. In there, the authors also show empirically that Assumption 4 seems to be satisfied for human walking.

B. Classification

For classification, we leverage the concept of CMEs. We first define positive definite kernel functions $k : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ and $\ell : \mathcal{C} \times \mathcal{C} \to \{0, 1\}$ for the input space (the measurements y) and the output space (the contexts c), respectively. In our setting, it is natural to choose ℓ as the Kronecker delta kernel. That is, we assume equal contexts have unit similarity while different contexts have no similarity. The Kronecker delta kernel is integrally strictly positive definite on $\mathcal{C}$ and, therefore, characteristic [29, Th. 7]. For the input space, we choose the Gaussian kernel, which is also characteristic, and then have that both k and ℓ uniquely define the RKHSs $\mathcal{H}_k$ and $\mathcal{H}_\ell$.

Following [30], the conditional kernel mean embedding operator $\mathcal{U}_{C|Y=y}$ is the operator $\mathcal{U} : \mathcal{H}_k \to \mathcal{H}_\ell$ and the CME is $\mu_{C|Y=y} = \mathcal{U}k(y, \cdot) := \mathbb{E}[\ell(C, \cdot) \mid Y = y]$. We further define the cross-covariance operators $R_{CY} := \mathbb{E}[\ell(C, \cdot) \otimes k(Y, \cdot)] : \mathcal{H}_k \to \mathcal{H}_\ell$ and $R_{YY} := \mathbb{E}[k(Y, \cdot) \otimes k(Y, \cdot)] : \mathcal{H}_k \to \mathcal{H}_k$. Given that $k(y, \cdot)$ is in the image of R_{YY}, we now have that $\mathcal{U}_{C|Y} = R_{CY} R_{YY}^{-1}$. However, since this assumption is not necessarily satisfied for continuous input spaces $\mathcal{Y}$ [31], with which we generally deal in our problem setting, we instead use the regularized version $\mathcal{U}_{C|Y} = R_{CY}(R_{YY} + \lambda I)^{-1}$ with regularization parameter λ and I the identity matrix of appropriate dimensions.

Ultimately, we seek to infer the classification probabilities $p_c(y)$, i.e., the probability of context c given specific measurements y. We can write this probability in terms of the indicator function $\mathbb{1}(\cdot)$ as

$$p_c(y) := \mathbb{P}(C = c \mid Y = y) = \mathbb{E}[\mathbb{1}_c(C) \mid Y = y]. \tag{4}$$

For the Kronecker delta kernel, we have $\mathbb{1}_c(y) = \ell(c, y)$. Thus, the indicator function $\mathbb{1}_c = \ell(c, \cdot)$ is the canonical feature map of $\mathcal{H}_\ell$ and we can estimate it using the CME [9]

$$\mathbb{E}[\mathbb{1}_c(C) \mid Y = y] \approx \langle \hat{\mu}_{C|Y=y}, \mathbb{1} \rangle_k = \mathbb{1}^{\mathsf{T}}(K + n\lambda I)^{-1} K_y := \hat{p}_c(y) \tag{5}$$

with n being the number of data points, $K_y := (k(y, y_1), \dots, k(y, y_n))$, $\mathbb{1} := \mathbb{1}_c(c_j)_{j=1}^n$, and the entry (a, b) in the matrix $K \in \mathbb{R}^{n\times n}$ is $k(y_a, y_b)$. It then also follows that we can essentially estimate the probabilities for all contexts using kernel ridge regression [9].

Theorem 1 (see [9, Th. 1]): The classifier (5) is consistent if $k(y, \cdot)$ is in the image of R_{YY}.

Remark 1: For finite sample sizes, (5) may yield context probabilities above one or below zero, which can be avoided by applying a normalization [9].

To obtain practically useful bounds, we make an assumption about the regularity of the true context probability functions $p_c(y)$ that is similar to Assumption 1 and generally common in the safe learning literature [2], [32], [33].

Assumption 5: The true probability functions $p_c(y)$ are all in the Hilbert space $\mathcal{H}_k$ and have bounded norm, $\|p_c(y)\|_k^2 \leq \Gamma$ for all $c \in \mathcal{C}$ with known bound Γ.

V. Safe Reinforcement Learning in Uncertain Contexts

We now present our safe reinforcement learning algorithm. We start by deriving high-probability guarantees for context identification. Then, we develop frequentist uncertainty intervals for multiclass classification based on CMEs. Lastly, we integrate both ingredients into the safe learning algorithm. Before starting, we make the notion of a context more precise.

Definition 1: For any two contexts $c_\mathrm{a} \neq c_\mathrm{b}$, we have that $\mathrm{MMD}^2(X_\mathrm{a}, X_\mathrm{b}) > \eta$ in the large sample limit (i.e., when the number of data points $r \to \infty$).

That is, we define contexts as external environmental changes that cause a significant change in the system dynamics. Since the definition via the MMD might seem abstract, we make the notion explicit in Appendix B for the Furuta pendulum with weights used in the evaluation in Section VI.

A. Context Identification With Guarantees

The MMD, as presented in Section IV-A, provides guarantees in the infinite sample limit. For finitely many data samples, Gretton et al. [5] presented various test statistics that can be used for hypothesis testing. Those hypothesis tests come with two challenges for our setting. First, they assume data to be i.i.d. Data drawn from dynamical systems are naturally correlated and, thus, not i.i.d. Second, the hypothesis test can only reject the null hypothesis $c_\mathrm{a} = c_\mathrm{b}$ based on a chosen significance level but cannot provide guarantees for detecting that $c_\mathrm{a} \neq c_\mathrm{b}$. In fact, Gretton et al. [5] showed through an example that providing such guarantees for distinguishing probability distributions is generally impossible, independent of the distance measure.

We address the first challenge by employing the subsampling strategy from [6]. That is, from every collected trajectory, we subsample data such that the time shift equals a^* from Assumption 4. For the second challenge, we leverage Definition 1. Then, we arrive at the following statement.

Proposition 1: Under Assumption 4 and given r data samples X and X_c, subsampled from the whole trajectory such that the time shift is a^*. Set

$$\eta = 4\sqrt{\frac{2\,K}{r}}\left(1 + \sqrt{2\ln\frac{2}{\delta_\mathrm{MMD}}}\right)$$

in Definition 1. Assuming a characteristic kernel k_MMD with $0 \leq k_\mathrm{MMD}(x, y) \leq K$, we have, with probability at least $1 - \delta_\mathrm{MMD}$

$$\mathrm{MMD}^2(X, X_c) < 2\sqrt{\frac{2\,K}{r}}\left(1 + \sqrt{2\ln\frac{2}{\delta'_\mathrm{MMD}}}\right)$$

where $\delta_\mathrm{MMD} = \frac{1}{3}(\delta'_\mathrm{MMD} + 2\epsilon)$ with ϵ from Assumption 4, if, and only if, the probability distribution that generated the data of the current context is the same that generated the trajectory data of context c.

Proof: Follows from combining [6, Th. 3] with Definition 1.□

That is, the guarantee for context identification we provide is essentially the same as in [6], except that we can also guarantee to detect $c_\mathrm{a} \neq c_\mathrm{b}$. This is possible because of Definition 1. Since different contexts, by definition, change the dynamics significantly, we can guarantee to detect this change with the proposed hypothesis test. In practice, we might also have external changes in the environment that only cause minor changes in the system dynamics. Then, learning a new policy would be inefficient, and it is more sensible to consider this still the same context. The design parameter η in Definition 1 quantifies when we consider an environmental change as *significant*, and we provide some intuition for it in Appendix B.

B. Classifier With Frequentist Bounds

Our goal is to develop a safe learning algorithm in uncertain contexts where we perform context identification only if necessary since it consumes time and causes wear and tear to the hardware. If we assume no prior knowledge about how measurements relate to contexts, we have, in the beginning, no other choice than to identify contexts. Over time, we can then learn a model that relates those identified contexts to received measurements. This is a standard classification setting. If we want to include the classification in the safe learning algorithm, it has to provide frequentist guarantees.

For classification, we need to consider the following three types of uncertainties:

i) uncertainty from estimating a function with limited amount of data;
ii) uncertainty from not obtaining samples of the true probability function $p_c(y)$ but only discrete labels;
iii) uncertainty that stems from context identification, which provides us with the correct context with probability $1 - \delta_\mathrm{MMD}$ (see Proposition 1).

In the following, we show how we can bound all three types of uncertainties and then combine them to obtain the required overall frequentist guarantees.

First, we define a variant of the *power function* [28], which will occur at several stages during the derivations.

Definition 2: The power function is defined as $\varrho(y) := \sqrt{k(y, y) - K_y^\mathrm{T}(K + n\lambda I)^{-1}K_y}$.

We now first address uncertainty (i) by introducing a virtual estimate $\bar{p}_c$ of $\hat{p}_c$. This virtual estimate is the estimate we would get if we could use measurements of the true context probability function $p_c(y)$ in (5) instead of discrete labels.

Lemma 1: Under Assumption 5, and given perfect measurements $\mathbf{p_y} \in \mathbb{R}^n$ of $p_c(y)$, for any $n > 0$, we have for all $y \in \mathcal{Y}$ and $c \in \mathcal{C}$

$$|p_c(y) - \bar{p}_c(y)| \leq \sqrt{\Gamma}\varrho(y)$$

where $\bar{p}_c$ is estimated using $\mathbf{p_y}$ and with $\varrho(y)$, the power function from Definition 2.

Proof: Similar in nature to that of [33, Th. 2]. Details are provided in Appendix A2. □

However, in practice, we only receive discrete labels. Thus, we next analyze the uncertainty from not measuring $p_c(y)$. We first establish that centered Bernoulli random variables are σ-sub-Gaussian.

Lemma 2: Let c be a Bernoulli random variable with success probability p_c. We have that the random variable $c - p_c$ is σ-sub Gaussian with $\sigma \leq \frac{1}{4}$.

Proof: Follows from [34, Theorem 2.1 and Lemma 2.1]. □

For bounding the measurement uncertainty, we similarly introduce a virtual estimate $\check{p}_c$, which is the estimate we would get if the context identification always returned the actual context.

Lemma 3: We have for any $n > 0$ and all $y \in \mathcal{Y}$, with probability at least $1 - \delta$

$$|\bar{p}_c(y) - \check{p}_c(y)| \leq \frac{\varrho(y)}{4\sqrt{n\lambda}} \sqrt{\log(\det(K + \bar{\lambda}I)) - 2\log(\delta)}$$

with $\bar{p}_c(y)$ as in Lemma 1, $\varrho(y)$ from Definition 2, and $\bar{\lambda} := \max\{1, n\lambda\}$.

Proof: The proof idea is similar to that of [32, Th. 1], which is possible since the measurement uncertainty is σ-sub Gaussian with $\sigma \leq \frac{1}{4}$ by Lemma 2. Details are provided in Appendix A3.□

Lastly, we address the uncertainty (iii) inherited through context identification.

Lemma 4: Given the setting in Proposition 1, we have, with probability at least $1 - \delta_{\text{MMD}}$

$$\begin{aligned}
&|\check{p}_c(y) - \hat{p}_c(y)| \\
&\leq \frac{\varrho(y)(1 - 2\delta_{\text{MMD}})\sqrt{\log(\det(K + \bar{\lambda}I)) - 2\log(\delta_{\text{MMD}})}}{2(\ln(1 - \delta_{\text{MMD}}) - \ln(\delta_{\text{MMD}}))\sqrt{n\lambda}} \\
&\quad + (1 - \delta_{\text{MMD}})\mathbf{1}(K + n\lambda I)^{-1}K_y
\end{aligned}$$

with $\check{p}_c(y)$ as in Lemma 3, $\varrho(y)$ from Definition 2, and $\bar{\lambda} = \max\{1, n\lambda\}$.

Proof: The two quantities are identical except for a potential mismatch of actual context c and estimated context $\hat{c}$. We, thus, analyze the error $|c - \hat{c}|$, which we can rewrite as $|c - \hat{c} - (1 - \delta_{\text{MMD}}) + (1 - \delta_{\text{MMD}})| \leq |(c - \hat{c} - (1 - \delta_{\text{MMD}})| + |1 - \delta_{\text{MMD}}|$. Following [34, Lem. 2.1], the first term is a sub-Gaussian random variable with $\sigma = \frac{1 - 2\delta_{\text{MMD}}}{2(\ln(1-\delta_{\text{MMD}}) - \ln(\delta_{\text{MMD}}))}$. Thus, we can bound the error term in the same way as shown in Lemma 3. □

Algorithm 1: Pseudocode of the Safe Reinforcement Learner.

```
Input: Measurements y, safety threshold p_safe
for c ∈ C do
    Compute p̂_c(y) using (5)
    Estimate Δp_c(y) = |p̂_c(y) − p_c(y)| with Cor. 1
    if p̂_c(y) − Δp_c(y) > p_safe then
        Return: Context c
    else
        Perform experiment, measure X trajectory
        for c ∈ C do
            if MMD²(X, X_c) below threshold from Prop. 1 then
                Return: Context c
Return: Context c ∉ C
```

Combining the lemmas, we arrive at the desired statement.

Corollary 1: Under Assumption 5 and given the setting in Proposition 1, we have, with probability at least $(1 - \delta_{\text{MMD}})(1 - \delta_{\text{class}})$

$$\begin{aligned}
&|p_c(y) - \hat{p}_c(y)| \\
&\leq \varrho(y)\Bigg(\sqrt{\Gamma} + \frac{1}{4\sqrt{n\lambda}}\sqrt{\log(\det(K + \bar{\lambda}I)) - 2\log(\delta_{\text{class}})} \\
&\quad + \frac{(1 - 2\delta_{\text{MMD}})\sqrt{\log(\det(K + \bar{\lambda}I)) - 2\log(\delta_{\text{MMD}})}}{2(\ln(1 - \delta_{\text{MMD}}) - \ln(\delta_{\text{MMD}}))\sqrt{n\lambda}}\Bigg) \\
&\quad + (1 - \delta_{\text{MMD}})\mathbf{1}(K + n\lambda I)^{-1}K_y
\end{aligned}$$

with $\varrho(y)$ from Definition 2 and $\bar{\lambda} = \max\{1, n\lambda\}$, holds for any $n > 0$ and all $y \in \mathcal{Y}$.

Remark 2: So far, we assumed that the number of contexts $|\mathcal{C}| = m$ is given a priori. Accounting for potential unknown contexts is straightforward. In that case, we consider the number of contexts m_n as a variable that can change as we gather more data (more (y, c) tuples). Whenever we receive a previously unseen context, we increase $m_{n+1} = m_n + 1$. Since we have not seen this context before, we can create its measurement vector as a vector of all zeros except for a one in the last entry.

C. Safe Learning

Finally, we show how all previous results can be merged into a safe learning algorithm (see Algorithm 1).

We need one more assumption before analyzing the safety of Algorithm 1. We assume that all relevant safety information that y can provide is encoded in the context.

Assumption 6: Let $P(\text{safe})$ denote the probability of an experiment being safe. We have $P(\text{safe} \mid c, y) = P(\text{safe} \mid c)$.

Then, the safety of Algorithm 1 can be formalized as follows.

Theorem 2: Given Assumptions 1–6. Then, following Algorithm 1, for any $n \geq 0$, the experiment is safe with probability at least $(1-\delta_{\text{safe}})(p_{\text{safe}})(1-\delta_{\text{class}})(1-\delta_{\text{MMD}})$.

Proof: We distinguish two cases. In the first case, we need to identify the context. Then, we have

$$\begin{aligned} P(\text{safe} \mid c) &= P(\text{safe} \mid \hat{c} = c) P(\hat{c} = c) \\ &\geq (1-\delta_{\text{safe}})(1-\delta_{\text{MMD}}) \end{aligned}$$

by [2, Th. 1] and Proposition 1. In the second case, we are certain enough about the current context to choose a policy directly. By Corollary 1, we can bound the uncertainty of our context inference. Thus, we have

$$\begin{aligned} P(\text{safe} \mid c) &= P(\text{safe} \mid \hat{c} = c) P(c = \hat{c} \mid y) \\ &\geq (1-\delta_{\text{safe}}) P(c = \hat{c} \mid y) \\ &\geq (1-\delta_{\text{safe}}) p_{\text{safe}} (1-\delta_{\text{class}})(1-\delta_{\text{MMD}}). \end{aligned}$$

□

VI. Evaluation

We evaluate our algorithm using the scenario shown in Fig. 1. After evaluating the algorithm as a whole, we provide a comparison with the standard SafeOpt algorithm without our contributions and then specifically investigate the performance of the classification bounds.

A. Safe Reinforcement Learning in Uncertain Contexts

In Fig. 1, we have a Furuta pendulum [10] to whose pole we can add weights. We also have a camera that can take a picture of the current weight, allowing us to infer which weight was added before an experiment. We consider learning a balancing controller for the Furuta pendulum using the SafeOpt [2] algorithm. In particular, we consider linear state-feedback control, i.e., we multiply the four-dimensional state, consisting of angle α and angular velocity $\dot{\alpha}$ of the rotatory arm, and angle θ and angular velocity $\dot{\theta}$ of the pole, with a feedback matrix $F \in \mathbb{R}^{1\times 4}$. We then focus on letting SafeOpt find an optimal value for the feedback gain multiplied by the pole angle while keeping the other entries of F fixed. During the search, SafeOpt shall avoid failures with high probability. Here, we define failure as the pole dropping. For interfacing the pendulum system, we leverage code provided with [35].

For SafeOpt, we use a Matern kernel for the parameter optimization with a length scale of 0.1 and a Gaussian kernel with a length scale of 1 for the contexts. Apart from that, we leave the hyperparameters provided in the official code [2] untouched. For classification, we choose, inspired by the classification example in [36, Ch. 3], a Gaussian kernel with a log length scale of 7.5 and log magnitude of 1.5. We further set $\lambda = 1 \times 10^{-4}$ and $\Gamma = 2$. We keep those two parameters for all experiments. In the following section, we discuss and numerically estimate the value of Γ for a different setting. From this discussion, we conclude that $\Gamma = 2$ is a sensible choice for making sure to stay safe and not allow for failures while at the same time not being too conservative. For the context identification, we also choose a Gaussian kernel and compute the length scale based on the data samples as suggested in [5].

At the beginning of each experiment, we let a random number generator determine the current context and, accordingly, add one of the two weights or no weight to the tip of the pole. We then take an image of the weight using a standard smartphone camera as is shown in Fig. 1. We convert each image to grayscale and scale it to a size of 32×32 pixels. For this rescaled image, we then compute the classification bounds from Corollary 1. Nevertheless, as we assume no prior knowledge, the uncertainty bounds are high during the first iterations. Thus, we must always identify the current context in the first iterations. In these cases, we use the initially given safe controller and excite the system by adding a chirp signal. We show example trajectories for the two weights and one without any weight in Fig. 2.

In case we need to identify the context, we seek to compute the MMD between the current trajectory and all stored trajectories. As trajectory data is naturally correlated through time, we follow the approach from [6] and subsample the data such that the subsampled trajectories satisfy Assumption 4. Therefore, we collect two independent trajectories per context and compute the MMD for increasing values of a, which we show in Fig. 3. We see that for $a > 50$, the MMD is reliably kept at a low level. Thus, during context identification experiments, we subsample by choosing only every 50th sample and then compute the MMD between the current trajectory and all stored trajectories to identify the context. During our experiments, we identified the context correctly in every iteration in which context identification was required.

Over time, we build a dataset of weight images and contexts that allows us to make more confident classification decisions. In Fig. 4, we show the classification probability estimates and uncertainty bounds of ten randomly selected images given a dataset consisting of the following:

1) the first ten images in our dataset (top row);
2) the first half of the dataset (middle row);
3) the entire dataset (312 images, bottom row).

We can see that the uncertainty steadily decreases. After accessing the entire dataset, we can make classification decisions with confidence above 70% in the case of weight two. We consider this to be a relatively small dataset for image classification. We further see a few misclassifications, marked by red crosses in Fig. 4, especially for the small datasets. Those are accompanied by large uncertainty intervals, i.e., our algorithm correctly identifies that the output of the classifier should not be trusted in those cases.

During all experiments, the pole of the Furuta pendulum never dropped, i.e., we successfully retained safety.

B. Comparison

Having shown the general applicability of our algorithm, we next compare the resulting algorithm to SafeOpt without our bounds and context identification. On the one hand, identifying the context in cases where the classifier is too uncertain comes

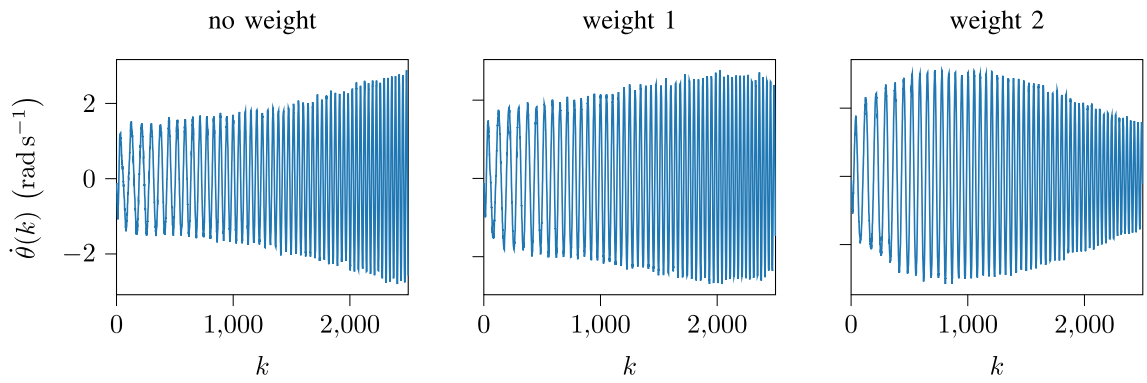

Fig. 2. Trajectories of context identification experiments.

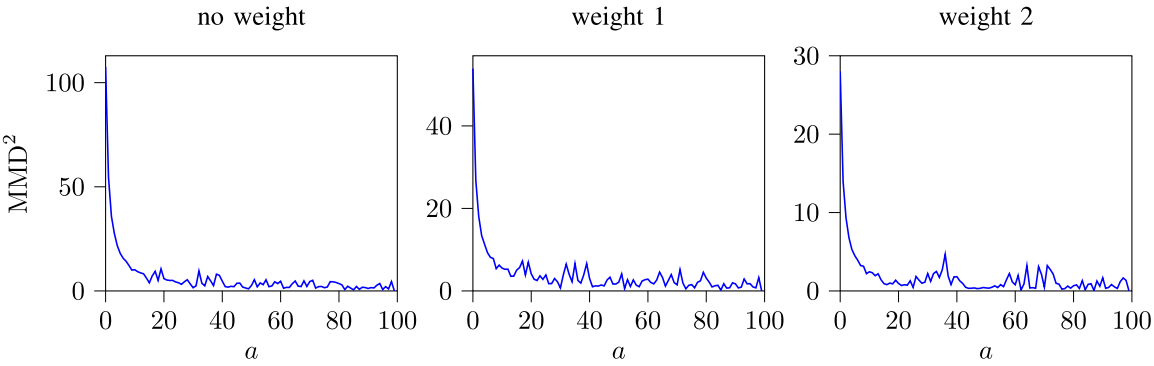

Fig. 3. MMD of context identification experiments for different weights for varying a. For $a > 50$, we see that the MMD is at a low level, i.e., trajectories are approximately independent.

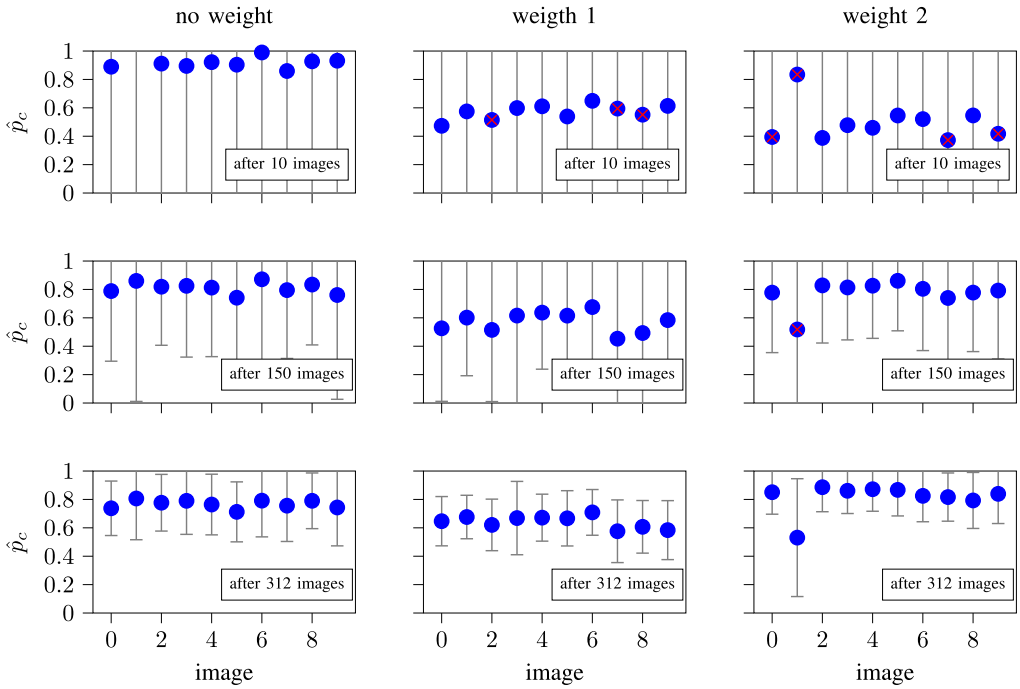

Fig. 4. Prediction of weights based on camera images. We show the prediction $\hat{p}_c$ and uncertainty intervals from Corollary 1 for ten images without weight (left), ten with weight one (middle), and ten with weight two (right). Wrong predictions are marked with red crosses. From top to bottom, we see how the uncertainty intervals decrease from a dataset of ten images, over one with around 150 images (middle), to the full dataset of 312 images.

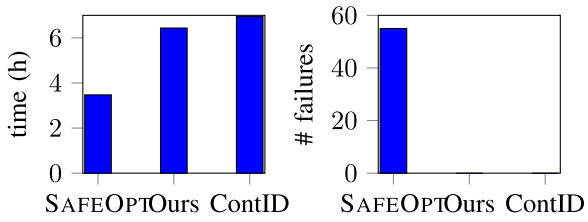

Fig. 5. Training time and number of failures for pure SAFEOPT, SAFEOPT with context identification *and* classification (our), and SAFEOPT with context identification before the start of every experiment (ContID). While our extensions require more samples for the additional context identification and, therefore, more training time, we do not incur any failures while we have several when only using SAFEOPT.

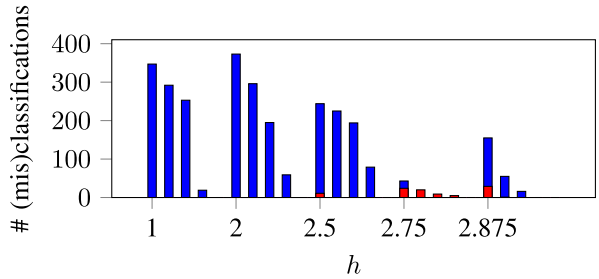

Fig. 6. Correct and incorrect classifications for the simulated Furuta pendulum with noisy measurements and different values of p_{safe}. For each context, we see the number of correct classifications in blue and misclassifications in red, from left to right, for increasing p_{safe}. The lower p_{safe}, the more contexts we can classify, but the larger also our errors, especially for the two contexts that are close together.

at the expense of requiring additional experimentation. On the other hand, considering contexts may improve performance and even be necessary to ensure safety.

For the comparison, we again consider the Furuta pendulum, but this time in simulation, again based on the code from [35], and change pole mass and length in the simulation code. We then let SAFEOPT optimize the feedback gain multiplied by the pole angle while not providing any information about the current context. After, we run the algorithm proposed in this work. Instead of camera images, in the simulated case, we assume that we receive noisy measurements, where the noise variance is normally distributed with a standard deviation of 0.1, of the height of the weight at the beginning of an experiment. Given the noisy height measurement, we evaluate the classifier and classification bounds and accept the outcome if the lower bound is above $p_{\text{safe}} = 0.8$. Otherwise, we perform an identification experiment as before. The parameter p_{safe} is a tuning parameter and mainly depends on the task at hand, i.e., the consequences of misclassification and, hence, a potential violation of a safety constraint. We discuss the choice of p_{safe} in more detail in Section VI-C. Having certainty about the context, we perform a SAFEOPT experiment. We adopt the kernel parameters for SAFEOPT, but reduce the length scales of the kernel for classification to 0.1.

We report results in Fig. 5. The left plot shows the experimentation time required for pure SAFEOPT, SAFEOPT with both context identification and classification, and SAFEOPT where we run a context identification at the beginning of each experiment without attempting any classification. Each context identification and also each SAFEOPT experiment lasts 2500 samples with an underlying sampling frequency of 200 Hz, and we run the entire loop for 1 000 iterations. Clearly, when identifying the context at each time step, the overall required experimentation time is double the time the pure SAFEOPT algorithm needs. Nevertheless, we see that when leveraging the classification, we can already, at this stage, save some time. Considering that it initially requires some training time until the classification starts to be effective, we can expect even more significant relative savings when running the algorithm for a longer time. But, certainly, running SAFEOPT while ignoring the unknown contexts will require the least time. However, in the right plot of Fig. 5, we also see the downside of this approach: while both our scheme and the one that identifies the context at the beginning of each experiment have zero failures after 1 000 iterations, SAFEOPT without considering contexts accumulated 55. While this is uncritical in a simulation experiment, in real experiments with costly and fragile hardware, this can be problematic. The other extreme case would be to consider SAFEOPT with contexts, assuming contexts to be known. In essence, that would result in equivalent performance in terms of failures as the runs we did with context identification and the runs we did with both context identification *and* classification, as both always recovered the true context. However, it would also result in the same training time as SAFEOPT, as we would assume that the context is known, rendering both context identification and classification unnecessary. Thus, under this assumption, the contributions of this article would not be required, and pure SAFEOPT, or a more advanced version of the algorithm, would be the more suitable choice.

C. Sensitivity Analysis and Limitations

The two examples have shown the benefit of the classification bounds proposed in this article. Nevertheless, when to accept a classified context without context identification depends on p_{safe}, which is a tuning parameter that we set to 0.8 in the previous examples. In this section, we show how the choice of this parameter influences the classification results. Furthermore, we discuss a case with more and gradually changing contexts.

We still consider the same setup as before, with the simulated Furuta pendulum and noisy height measurements providing information about the current contexts. This time, we consider five contexts, where the heights are $h \in \{1, 2, 2.5, 2.75, 2.875\}$, disturbed with normally distributed noise with a standard deviation of 0.1. Thus, in this case, at least the last two contexts are hard to distinguish for the classifier. We then run five times 2000 experiments picking one of $p_{\text{safe}} \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$ for each of the five runs, and report in Fig. 6, which contexts were classified correctly (in blue) and incorrectly (in red) for the different p_{safe} values. Clearly, as we increase the safety threshold, the classifier less often exceeds it. On the other hand, also the number of misclassifications decreases as we increase p_{safe}. We can further see that for the contexts that are much farther apart from each other than the noise level, there are no classification errors even for $p_{\text{safe}} = 0.5$. However, especially for

$h = 2.75$, we have relatively many misclassifications and, even for $p_{\text{safe}} = 0.5$, overall only few cases in which the classifier exceeds the threshold. Thus, when contexts are hard to distinguish, our classifier will often report low certainty and render the context classification necessary.

D. Classification Bounds

Next, we evaluate the performance of the classification bounds. We first qualitatively compare the nature of our uncertainty bounds with more standard, expected risk bounds from [9] and the recently proposed deterministic bounds from [28] in a simple, synthetic example. Then, we demonstrate the applicability of our bounds in two standard classification benchmarks: the modified National Institute of Standards and Technology (MNIST) dataset [37] and the German traffic sign recognition benchmark (GTSRB) [38]. In this part of the evaluation, we disregard the uncertainty from context identification but investigate directly the error $|p_c(y) - \check{p}_c(y)|$.

Qualitative comparison: For a qualitative comparison, we choose a probability function $p_0(y) = (1 + \exp(-y + 1))^{-1}$ and $p_1(y) = 1 - p_0(y)$, where y is a scalar parameter. The training set consists of 50 y values in the range -6 to -4.7, 50 y values between 0.5 and 1.78, and 50 x values between 5.7 and 7. We sample a context for each x value. The context is either zero or one with probability $p_0(y)$ and $p_1(y)$, respectively.

For all three approaches, we then compute the estimate $\hat{p}_c$ using CMEs as presented in Section IV-B, and their respective bounds for 1 00 y values sampled uniformly in the interval $[-6, 7]$. For all approaches, we use a Gaussian kernel with length scale 1. In [39, Appendix], the authors present a method to empirically estimate a bound on the RKHS norm of a given function. For the function p_0, we get $\Gamma = 2$ as a conservative estimate in the region where we evaluate the function. Thus, we choose $\Gamma = 2$.

The bounds in [9, Th. 4] are expected risk bounds. As such, they do not directly allow us to bound the error $|p_c(y) - \check{p}_c(y)|$. Instead, we can infer the likelihood of misclassifying a sample for any input y. The bounds are extremely conservative if used in such a way and report a misclassification risk above 99% for the chosen hyperparameters. However, making such predictions is not the purpose for which these bounds were developed. In [9], they were used to tune the hyperparameters of a CME-based classifier. The authors showed that the bounds are very useful in providing a tradeoff between accuracy on the training data and model complexity that leads to good generalization properties. Nevertheless, even with the optimized hyperparameters in [9], the bounds are too conservative to apply to our problem setting.

The bounds from [28] are closer to the ones proposed in this article. Similar to ours, they are input-dependent and, thus, give, for any specific input x, an upper bound on the deviation $|p_c(y) - \check{p}_c(y)|$. However, while our bound is a high probability bound, the bound from [28] is deterministic. Their central assumption is that training data is corrupted by bounded noise with a known bound. In our setting, we aim at estimating $p_c(y)$ while only receiving binary labels. We can interpret this as measuring $p_c(y)$ with a maximum error of one. Then, we can use the bounds from [28]. In particular, we use the simplified version of the bounds provided in [28, Th. 1] that do not require computing the RKHS norm of $\bar{p}_c$.

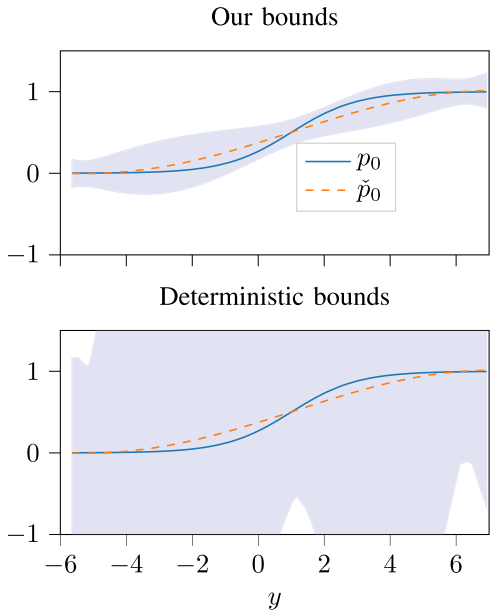

Fig. 7. Our bounds compared with those of [28]. The bounds are illustrated through the blue shaded areas. Compared to our bounds (top), the bounds from [28] (bottom) are, due to their deterministic nature, way more conservative and, therefore,noninformative for classification.

In Fig. 7, we compare our bounds (bottom) with those from [28] (top). We show the true probability function $p_0(y)$, the estimates $\check{p}_0(y)$, and the bounds in both plots. The data-dependent nature can be seen in both figures. However, the bounds from [28] are way more conservative than ours. This is natural since the bounds from [28] are deterministic, i.e., they need to hold for any ground truth function compatible with the data and noise model, while ours are high probability statements. Thus, while the bounds from [28] are essential results for general function estimation and applications in, for instance, robust control, the bounds are noninformative for classification.

This comparison shows that state-of-the-art bounds for multiclass classification do not meet the requirements for our problem setting. We require bounds that are both input-dependent and probabilistic such that they yield practically usable worst-case bounds. In contrast to existing bounds, the bounds we derived in Section V-B meet both requirements.

MNIST: To challenge the scalability of the bounds, we next consider the popular MNIST dataset. The MNIST dataset consists of images of handwritten digits. Thus, the task for our classifier is to predict, which digit can be seen in a specific image. At the same time, we seek to infer how certain we are about the classification through our bounds.[1]

For image classification, we again choose a Gaussian kernel with a log length scale of 7.5 and log magnitude of 2.6. We normalize the images such that the pixels take values in $[-1, 1]$.

[1]The code for this example is available at https://github.com/baumanndominik/cme_based_classification_bounds.

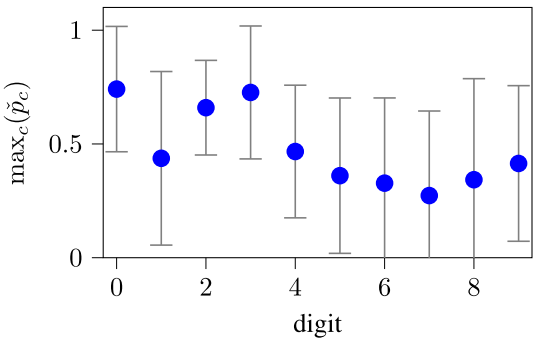

Fig. 8. Our classifier and bounds applied to the MNIST dataset. After having seen 10 000 training images, we can already make confident decisions for some digits.

The MNIST dataset is split into a training and test set. We use the first 10 000 training images to train our algorithm and then evaluate it on the first occurrence of each digit in the test dataset. For the test examples, we show both the probability of the most likely digit and the uncertainty bounds in Fig. 8. After having seen 10 000 training examples, the uncertainty in many test cases is still very high and would not enable us to make confident predictions. While this represents a limitation, it is a natural one. A single observed outcome for a specific parameter setting can already yield significant insights in a regression task. This is not the case in classification. If we throw a coin once and it comes up tails, we cannot infer whether or not the coin is likely to be fair. Especially in this light, the bounds developed herein are an essential asset in the classification setting. Popular classifiers have been reported to be overconfident [21]. Hence, it is crucial to add reliable bounds, particularly if the classifier's output is used in safety-critical environments.

However, Fig. 8 also shows that in some instances, e.g., for zero, one, and seven, we are confident that our estimator classifies correctly. For comparison, we computed the expected risk bounds from [9, Th. 4]. Those reveal a significant misclassification risk over the entire input domain. Thus, they would not let us make any confident classification decision, rendering the entire classification useless. This underlines the importance of input-dependent uncertainty bounds for safe learning. If we either accept all or none of the predictions, it may take us too long until we are confident enough. With the input-dependent bounds developed in this article, we judge the prediction uncertainty locally at the current input and can make confident classification decisions in specific parts of the input space even if, overall, the uncertainty is still high.

German traffic sign recognition benchmark: While the MNIST dataset is widely used, it may not be obvious how misclassifying a digit could be fatal. Therefore, we next consider the GTSRB [38]. The GTSRB contains images of different traffic signs that should be classified. Suppose an algorithm that classifies traffic signs is used, for instance, within a self-driving car that chooses its driving policy based on this classification. In that case, we must be sure about our predictions.

Also here, we consider a Gaussian kernel, this time with a log length scale of 7 and log magnitude of 1.5. Besides normalizing the pixel values, we rescale the images, which a priori have varying sizes, to 32×32 pixels and convert them to grayscale images.

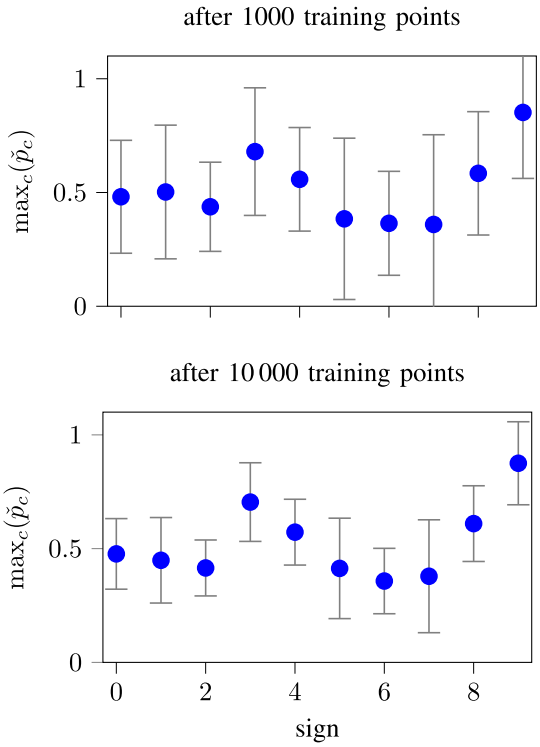

Fig. 9. Results of the GTSRB dataset. Also in this case, we can return practically useful bounds on classification probabilities.

Similar to MNIST, the GTSRB is divided into training and test set. We randomly select ten traffic signs. Then, we train the classifier on the first 1 000 images that contain those signs and use the first occurrence of those signs in the test set for evaluation. As shown in Fig. 9 (top), the uncertainty is still very high. In particular, since the lower bound of the classification probability barely reaches 50%, at this stage, the classifier should, if at all, only be trusted to recognize sign number nine. We then re-evaluate the same signs after providing 10 000 images for training. This significantly reduces the uncertainty (see Fig. 9, bottom). This shows that the bounds can easily be used in online learning settings and tightened as we receive more data. After having seen 10 000 training examples, we can also confidently (and correctly) identify signs three and eight. Meanwhile, for sign two, for instance, the bounds tell us that the algorithm cannot provide a reliable classification probability. Since misclassification in this example may cause accidents, knowing for which signs we cannot rely on the classifier is precious information. This shows that the classification bounds are informative in cases with relatively few training data, as they can clearly indicate when we can start to trust the classifier.

VII. Conclusion

This article presents a safe learning algorithm for context-conditional dynamical systems with unknown contexts. We show how contexts can be identified from data or classified using CMEs, providing high-probability guarantees in both cases. Subsequently, we show how they can be combined with a popular safe learning algorithm. We demonstrate that given measurements that allow us to distinguish contexts clearly, the classification bounds can save training time. Otherwise, if the

measurements do not allow us to distinguish contexts clearly, we need to identify the contexts through experiments, which increases training time.

A further contribution of this work is the derivation of frequentist uncertainty intervals for a multiclass classifier. This result has applications beyond the area of safe learning. Clearly, a limiting factor of the bounds is that we require an upper bound on the RKHS norm. The bounds share this problem with SAFEOPT and related safe learning approaches, which also require an upper bound on this norm. Thus, estimating this norm from data is subject to ongoing research.

APPENDIX

A. Proofs

Here, we present extended proofs for Lemmas 1 and 3. Before stating the proofs, we introduce some useful derivations used throughout.

1) Useful Derivations: In this section, we restate and adapt some derivations from [33, Proof of Theorem 2], which we will use in the proofs of our main results. We first define the feature map $\varphi(y) := k(y, \cdot)$ that maps any point from $\mathcal{Y}$ to the RKHS $\mathcal{H}_k$. Since we define the inputs y to take values in the real numbers, we can further define the inner product in the RKHS as $\langle g, h\rangle_k = g^{\mathrm{T}}h$ and the RKHS norm $\|g\|_k = \sqrt{g^{\mathrm{T}}g}$ for any two functions f and g in $\mathcal{H}_k$. Defining $\Phi := (\varphi(y_1)^{\mathrm{T}}, \dots, \varphi(y_n)^{\mathrm{T}})^{\mathrm{T}}$, we can write $K = \Phi\Phi^{\mathrm{T}}$, $K_y = \Phi\varphi(y)$, and $(p_c(y_1), \dots, p_c(y_n)) = \Phi p_c$.

The matrix $(\Phi^{\mathrm{T}}\Phi + n\lambda I)$ is strictly positive definite. Thus, we have

$$\Phi^{\mathrm{T}}(\Phi\Phi^{\mathrm{T}} + n\lambda I)^{-1} = (\Phi^{\mathrm{T}}\Phi + n\lambda I)^{-1}\Phi^{\mathrm{T}}. \tag{6}$$

Using (6), we can further conclude

$$\varphi(x) = \Phi^{\mathrm{T}}(\Phi\Phi^{\mathrm{T}} + n\lambda I)^{-1} + n\lambda(\Phi^{\mathrm{T}}\Phi + n\lambda I)^{-1}\varphi(y) \tag{7}$$

which leads us to

$$\varphi(y)^{\mathrm{T}}\varphi(x) = K_y^{\mathrm{T}}(\Phi\Phi^{\mathrm{T}} + n\lambda I)^{-1}K_y + n\lambda\varphi(y)^{\mathrm{T}}(\Phi^{\mathrm{T}}\Phi + n\lambda I)^{-1}\varphi(y). \tag{8}$$

Finally, we have

$$n\lambda\varphi(y)^{\mathrm{T}}(\Phi^{\mathrm{T}}\Phi + n\lambda I)^{-1}\varphi(x) = k(y, y) - K_y^{\mathrm{T}}(K + n\lambda I)^{-1}K_y. \tag{9}$$

2) Proof of Lemma 1: We have for all $c \in \mathcal{Y}$

$$\begin{aligned}
&|p_c(y) - \bar{p}_c(y)| = \left|p_c(y) - K_y^{\mathrm{T}}(K + n\lambda I)^{-1}\mathbf{p_y}\right| \\
&= \left|\varphi(y)^{\mathrm{T}}p_c - \varphi(y)^{\mathrm{T}}\Phi^{\mathrm{T}}(\Phi\Phi^{\mathrm{T}} + n\lambda I)^{-1}\Phi p_c\right| && \text{Definitions in Section A1} \\
&= \left|\varphi(y)^{\mathrm{T}}p_c - \varphi(y)^{\mathrm{T}}(\Phi^{\mathrm{T}}\Phi + n\lambda I)^{-1}\Phi^{\mathrm{T}}\Phi p_c\right| && (6) \\
&= \left|n\lambda\varphi(x)^{\mathrm{T}}(\Phi^{\mathrm{T}}\Phi + n\lambda I)^{-1}p_c\right| && (7) \\
&\le \left\|n\lambda\varphi(y)^{\mathrm{T}}(\Phi^{\mathrm{T}}\Phi + n\lambda I)^{-1}\right\|_k \|p_c\|_k && \text{Cauchy–Schwarz} \\
&\le \sqrt{\Gamma}\left\|n\lambda\varphi(y)^{\mathrm{T}}(\Phi^{\mathrm{T}}\Phi + n\lambda I)^{-1}\right\|_k && \text{Assumption 5} \\
&= \sqrt{\Gamma}\sqrt{n\lambda\varphi(x)^{\mathrm{T}}(\Phi^{\mathrm{T}}\Phi + n\lambda I)^{-1}n\lambda(\Phi^{\mathrm{T}}\Phi + n\lambda I)^{-1}\varphi(y)} && \text{Definitions in Section A1} \\
&\le \sqrt{\Gamma}(n\lambda\varphi(x)^{\mathrm{T}}(\Phi^{\mathrm{T}}\Phi + n\lambda I)^{-1}(\Phi^{\mathrm{T}}\Phi + n\lambda I) \\
&\quad (\Phi^{\mathrm{T}}\Phi + n\lambda I)^{-1}\varphi(y))^{\frac{1}{2}} && \text{K is pos. def.} \\
&= \sqrt{\Gamma}\sqrt{k(y, y) - K_y^{\mathrm{T}}(K + n\lambda I)^{-1}K_y} && (9)
\end{aligned}$$

from which the claim follows through Definition 2.

3) Proof of Lemma 3: We have for all $c \in \mathcal{Y}$

$$\begin{aligned}
&|\bar{p}_c(y) - \check{p}_c(y)| = \left|K_y^{\mathrm{T}}(K + n\lambda I)^{-1}(\mathbf{p_y} - \mathbf{c})\right| \\
&= \left|\varphi(y)^{\mathrm{T}}\Phi(K + n\lambda I)^{-1}(\mathbf{p_y} - \mathbf{c})\right| && \text{Definitions in Section A1} \\
&= \left|\varphi(y)^{\mathrm{T}}(K + n\lambda I)^{-1}\Phi^{\mathrm{T}}(\mathbf{p_y} - \mathbf{c})\right| && (6) \\
&\le \left\|\varphi(x)^{\mathrm{T}}(K + n\lambda I)^{-\frac{1}{2}}\right\|_k \left\|(K + n\lambda I)^{-\frac{1}{2}}\Phi^{\mathrm{T}}(\mathbf{p_y} - \mathbf{c})\right\|_k && \text{Cauchy–Schwarz} \\
&= \sqrt{\varphi(x)^{\mathrm{T}}(K + n\lambda I)^{-1}\varphi(y)} \\
&\quad \sqrt{(\Phi^{\mathrm{T}}(\mathbf{p_y} - \mathbf{c}))^{\mathrm{T}}(K + n\lambda I)^{-1}\Phi^{\mathrm{T}}(\mathbf{p_y} - \mathbf{c})} \\
&= \sqrt{\frac{1}{n\lambda}}\varrho(y)\sqrt{(\Phi^{\mathrm{T}}(\mathbf{p_y} - \mathbf{c}))^{\mathrm{T}}(K + n\lambda I)^{-1}\Phi^{\mathrm{T}}(\mathbf{p_y} - \mathbf{c})} && (9), \text{Def.2} \\
&= \sqrt{\frac{1}{n\lambda}}\varrho(y)\sqrt{(\mathbf{p_y} - \mathbf{c})^{\mathrm{T}}K(K + n\lambda I)^{-1}(\mathbf{p_y} - \mathbf{c})} && \text{Definitions in Section A1.}
\end{aligned}$$

The claim then follows from [32, Th. 1] since the random variables are σ-sub Gaussian with $\sigma \le \frac{1}{4}$ following Lemma 2.

B. Example for Definition 1

Defining contexts based on η might seem to be an abstract choice at first. Therefore, let us here make it more intuitive by calculating it for the Furuta pendulum example from Section VI. When considering the balancing of the Furuta pendulum, the system can be approximated as a linear, time-invariant system. If we further assume that the state measurements we receive are perturbed by Gaussian noise, the resulting trajectory data also follows a Gaussian distribution. Given that we subsample the data such that it is approximately i.i.d., we can now analytically compute the MMD. For ease of presentation, we consider a scalar state, e.g., the angular velocity of the pole that we also used in Section VI. Then, for two contexts c_{a} and c_{b} that generate Gaussian data distributions $\mathcal{N}(\mu_{\mathrm{a}}, \sigma_{\mathrm{a}}^2)$ and $\mathcal{N}(\mu_{\mathrm{b}}, \sigma_{\mathrm{b}}^2)$ with a Gaussian kernel k_{mmd} we have in the infinite sample limit

$$\mathrm{MMD}(X_{\mathrm{a}}, X_{\mathrm{b}}) = \frac{\exp\left(\frac{2|\mu_{\mathrm{a}}|^2}{2(2\sigma_{\mathrm{a}}^2 + \gamma^2)}\right)}{\sqrt{2\pi(2\sigma_{\mathrm{a}}^2 + \gamma^2)}} + \frac{\exp\left(\frac{2|\mu_{\mathrm{b}}|^2}{2(2\sigma_{\mathrm{b}}^2 + \gamma^2)}\right)}{\sqrt{2\pi(2\sigma_{\mathrm{b}}^2 + \gamma^2)}} - \frac{2\exp\left(\frac{|\mu_{\mathrm{a}} + \mu_{\mathrm{b}}|^2}{2(\sigma_{\mathrm{a}}^2 + \sigma_{\mathrm{b}}^2 + \gamma^2)}\right)}{\sqrt{2\pi(\sigma_{\mathrm{a}}^2 + \sigma_{\mathrm{b}}^2 + \gamma^2)}}$$

with γ the length scale of the Gaussian kernel. This result is based on the derivations from [40].

With the experimental data that we have, we can now approximate both mean and standard deviation for both contexts and compute the MMD. We can also compute the corresponding η from Proposition 1. When comparing both, we see that for differentiating the context "no weight" from "weight 2" and "weight 1" from "weight 2," the 2500 data samples we collected are sufficient to have an η that is below the threshold given in Proposition 1. To be able to guarantee that we can differentiate context "no weight" from context "weight 1," we would have needed around 350 000 data points. However, we see in the evaluation that also with 2500 data samples, we can reliably identify the context.

ACKNOWLEDGMENT

The authors would like to thank Alexander von Rohr, Sebastian Mair, Christian Fiedler, Friedrich Solowjow, Antonio Ribeiro, and Torbjörn Wigren for insightful discussions and comments on earlier manuscript versions, and Claas Thesing for providing the Python implementation of the MMD test.

REFERENCES

[1] L. Brunke et al., "Safe learning in robotics: From learning-based control to safe reinforcement learning," *Annu. Rev. Control, Robot., Auton. Syst.*, vol. 5, no. 1, pp. 411–444, 2022.

[2] F. Berkenkamp, A. Krause, and A. P. Schoellig, "Bayesian optimization with safety constraints: Safe and automatic parameter tuning in robotics," *Mach. Learn.*, vol. 112, no. 10, pp. 3713–3747, 2021.

[3] T. Standley, O. Sener, D. Chen, and S. Savarese, "Image2mass: Estimating the mass of an object from its image," in Proc. *Conf. Robot Learn.*, 2017, pp. 324–333.

[4] J. Achterhold and J. Stueckler, "Explore the context: Optimal data collection for context-conditional dynamics models," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2021, pp. 3529–3537.

[5] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 723–773, 2012.

[6] F. Solowjow, D. Baumann, C. Fiedler, A. Jocham, T. Seel, and S. Trimpe, "A kernel two-sample test for dynamical systems," 2020, *arXiv:2004.11098*.

[7] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, and A. Lopez, "A comprehensive survey on support vector machine classification: Applications, challenges and trends," *Neurocomputing*, vol. 408, pp. 189–215, 2020.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[9] K. Hsu, R. Nock, and F. Ramos, "Hyperparameter learning for conditional kernel mean embeddings with rademacher complexity bounds," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discov. Databases*, 2018, pp. 227–242.

[10] K. Furuta, M. Yamakita, and S. Kobayashi, "Swing-up control of inverted pendulum using pseudo-state feedback," in *Proc. Inst. Mech. Eng., Part I: J. Syst. Control Eng.*, vol. 206, no. 4, 1992, pp. 263–269.

[11] A. Krause and C. S. Ong, "Contextual Gaussian process bandit optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 2447–2455.

[12] J. Langford and T. Zhang, "The Epoch-Greedy algorithm for contextual multi-armed bandits," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 817–824.

[13] C.-C. Wang, S. R. Kulkarni, and H. V. Poor, "Bandit problems with side observations," *IEEE Trans. Autom. Control*, vol. 50, no. 3, pp. 338–355, Mar. 2005.

[14] T.-L. Lai and S. Yakowitz, "Machine learning and nonparametric bandit theory," *IEEE Trans. Autom. Control*, vol. 40, no. 7, pp. 1199–1209, Jul. 1995.

[15] Q. Feng, B. Letham, H. Mao, and E. Bakshy, "High-dimensional contextual policy search with unknown context rewards using Bayesian optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 22032–22044.

[16] G. Swamy, S. Choudhury, J. A. Bagnell, and Z. S. Wu, "Sequence model imitation learning with unobserved contexts," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 17665–17676.

[17] C. König, M. Turchetta, J. Lygeros, A. Rupenyan, and A. Krause, "Safe and efficient model-free adaptive control via Bayesian optimization," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2021, pp. 9782–9788.

[18] K. Zhou, J. Doyle, and K. Glover, *Robust and Optimal Control*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1996.

[19] W.-H. Chen, J. Yang, L. Guo, and S. Li, "Disturbance-observer-based control and related methods—An overview," *IEEE Trans. Ind. Electron.*, vol. 63, no. 2, pp. 1083–1095, Feb. 2016.

[20] B.K. Sriperumbudur, K. Fukumizu, A. Gretton, B. Schölkopf, and G. R. Lanckriet, "On the empirical estimation of integral probability metrics," *Electron. J. Statist.*, vol. 6, pp. 1550–1599, 2012.

[21] Y. Bai, S. Mei, H. Wang, and C. Xiong, "Don't just blame over-parametrization for over-confidence: Theoretical analysis of calibration in binary classification," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 566–576.

[22] M. Kull, M. P. Nieto, M. Kängsepp, T. S. Filho, H. Song, and P. Flach, "Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with Dirichlet calibration," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 12316–12326.

[23] Y. Lei, Ü. Dogan, D.-X. Zhou, and M. Kloft, "Data-dependent generalization bounds for multi-class classification," *IEEE Trans. Inf. Theory*, vol. 65, no. 5, pp. 2995–3021, May 2019.

[24] M. Seeger, "PAC-Bayesian generalisation error bounds for Gaussian process classification," *J. Mach. Learn. Res.*, vol. 3, pp. 233–269, 2002.

[25] C. Gupta, A. Podkopaev, and A. Ramdas, "Distribution-free binary classification: Prediction sets, confidence intervals and calibration," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 3711–3723.

[26] J. Park and K. Muandet, "A measure-theoretic approach to kernel conditional mean embeddings," *Adv. Neural Inf. Process. Syst.*, pp. 21247–21259, 2020.

[27] S. Smale and D.-X. Zhou, "Learning theory estimates via integral operators and their approximations," *Constructive Approximation*, vol. 26, no. 2, pp. 153–172, 2007.

[28] E. T. Maddalena, P. Scharnhorst, and C. N. Jones, "Deterministic error bounds for kernel-based learning techniques under bounded noise," *Automatica*, vol. 134, 2021, Art. no. 109896.

[29] B.K. Sriperumbudur, A. Gretton, K. Fukumizu, B. Schölkopf, and G. R. Lanckriet, "Hilbert space embeddings and metrics on probability measures," *J. Mach. Learn. Res.*, vol. 11, pp. 1517–1561, 2010.

[30] L. Song, J. Huang, A. Smola, and K. Fukumizu, "Hilbert space embeddings of conditional distributions with applications to dynamical systems," in *Proc. Int. Conf. Mach. Learn.*, 2009, pp. 961–968.

[31] K. Fukumizu, F. R. Bach, and M. I. Jordan, "Dimensionality reduction for supervised learning with reproducing kernel Hilbert spaces," *J. Mach. Learn. Res.*, vol. 5, pp. 73–99, 2004.

[32] C. Fiedler, C. W. Scherer, and S. Trimpe, "Practical and rigorous uncertainty bounds for Gaussian process regression," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 7439–7447.

[33] S. R. Chowdhury and A. Gopalan, "On kernelized multi-armed bandits," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 844–853.

[34] V. V. Buldygin and K. Moskvichova, "The sub-Gaussian norm of a binary random variable," *Theory Probability Math. Statist.*, vol. 86, pp. 33–49, 2013.

[35] S. Bleher, S. Heim, and S. Trimpe, "Learning fast and precise pixel-to-torque control: A platform for reproducible research of learning on hardware," *IEEE Robot. Autom. Mag.*, vol. 29, no. 2, pp. 75–84, Jun. 2022.

[36] C. K. Williams and C. E. Rasmussen, *Gaussian Processes for Machine Learning*.Cambridge, MA, USA: MIT Press, 2006.

[37] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[38] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition," *Neural Netw.*, vol. 32, pp. 323–332, 2012.

[39] P. Scharnhorst, E. T. Maddalena, Y. Jiang, and C. N. Jones, "Robust uncertainty bounds in reproducing kernel Hilbert spaces: A convex optimization approach," *IEEE Trans. Autom. Control*, vol. 68, no. 5, pp. 2848–2861, 2023.

[40] R. M. Rustamov, "Closed-form expressions for maximum mean discrepancy with applications to Wasserstein auto-encoders," *Stat*, vol. 10, no. 1, 2021, Art. no. e329.

Dominik Baumann received the Dipl.-Ing. degree in electrical engineering from the Dresden University of Technology, Dresden, Germany, in 2016 and the Ph.D. degree in electrical engineering from KTH Stockholm, Stockholm, Sweden, in 2020.

He is currently an Assistant Professor with Aalto University, Espoo, Finland. He was a joint Ph.D. student with the Max Planck Institute for Intelligent Systems, Stuttgart/Tübingen, Germany, and KTH Stockholm. After his Ph.D., he was a Postdoctoral Researcher with RWTH Aachen University, Aachen, Germany, and Uppsala University, Uppsala, Sweden. His research interests revolve around learning and control for networked multiagent systems.

Dr. Baumann was the recipient of the best paper award at the 2019 ACM/IEEE International Conference on Cyber-Physical Systems, the best demo award at the 2019 ACM/IEEE International Conference on Information Processing in Sensor Systems, and the future award of the Ewald Marquardt Foundation.

Thomas B. Schön (Senior Member, IEEE) received the B.Sc. degree in business administration and economics from Linköping University, Linköping, Sweden, the M.Sc. degree in applied physics and electrical engineering, from Linköping University, in 2001, and the Ph.D. degree in automatic control from Linköping University, in 2006.

He is currently the Beijer Professor of Artificial Intelligence with the Department of Information Technology, Uppsala University, Uppsala, Sweden. He has held visiting positions with the University of Cambridge, Cambridge, U.K., the University of Newcastle, Newcastle, NSW, Australia, and Universidad Tácnica Federico Santa María, Valparaíso, Chile.

Dr. Schön was the recipient of the Tage Erlander prize for natural sciences and technology in 2017 and the Arnberg prize in 2016, both awarded by the Royal Swedish Academy of Sciences (KVA), Automatica Best Paper Prize in 2014, and in 2013, Best Ph.D. Thesis Award by The European Association for Signal Processing, and Best Teacher Award at the Institute of Technology, Linköping University in 2009. In 2018, he was elected to The Royal Swedish Academy of Engineering Sciences (IVA) and The Royal Society of Sciences at Uppsala.