Adaptive Discretization for Probabilistic Safety Cost Functions Evaluation

Christoph Zimmer[1] Danny Driess[2] Mona Meister[1] Duy Nguyen-Tuong[1]
[1]Bosch Center for Artificial Intelligence [2]University of Stuttgart, MLR Laboratory

Abstract

In many real-world planning applications, e.g. dynamic design of experiments, autonomous driving and robot manipulation, it is necessary to evaluate candidate movement paths with respect to a safety cost function. Here, the continuous candidate paths need to be discretized first and, subsequently, evaluated on the discretization points. The resulting quality of planned paths, thus, highly depends on the definition of the safety cost functions, and the resolution of the discretization. In this paper, we propose an approach for evaluating continuous candidate paths by employing an *adaptive discretization* scheme, with a *probabilistic cost function* learned from observations. The obtained path is then guaranteed to be ϵ-safe, i.e. the remaining risk of still finding an unsafe point on the trajectory is smaller than ϵ. The proposed approach is investigated theoretically, as well as empirically validated on several robotic path planning scenarios.

1 Introduction

Safe planning of continuous paths within high-dimensional spaces is crucial for many real-world applications, e.g. robot manipulation and safe design of experiments (DoE) [Dolgov et al., 2008, Zimmer et al., 2018]. The underlying problem is to find a continuous path connecting a starting point with a target point, while fulfilling given safety requirements, such as obstacle avoidance. Usually, these requirements are incorporated into a cost function used for evaluation of candidate paths. In robot task planning, for example, the safety cost function exploits the knowledge about the environment with known obstacles and their configurations [Jaillet et al., 2008]. In safe DoE, on the other hand, the cost function incorporates the safety condition necessary for dynamic exploration in a high-dimensional state-space [Zimmer et al., 2018]. Given a safety cost function, a candidate path is usually discretized first and, subsequently, evaluated on these discretization points for finding the most safe path [Zimmer et al., 2018, Jaillet et al., 2008]. As the discretization represents an approximation of the continuous path, the discretization resolution is a trade-off between the evaluation accuracy and computational effort. Identifying an appropriate discretization step-size is a tedious task in practice, as it highly depends on the landscape of the respective safety cost function.

In this paper, we approach the problem of *safety evaluation* by learning a probabilistic safety cost function, while taking into account uncertainties in the environment and observations. Here, the cost function represents the probability of safety values, e.g. collision with obstacles during the robot planning task, or entering an unsafe region of the state-space during the dynamic exploration for design of experiments. As the evaluation effort highly depends on the number of discretization points, we develop an algorithm for *automatically inferring* the number and locations of necessary discretization points to be evaluated. The required number and locations of discretizations will depend on the considered safety cost function, as well as the continuous candidate paths. Intuitively, more discretization points are necessary, if the cost function is "complicated" for a given path. Thus, given a desired safety-level, a learned probabilistic safety cost function and a candidate path, the proposed algorithm will infer the required number and locations of the discretization points, such that the evaluation of the safety has the desired precision for the given continuous path. Technically, we employ Gaussian Processes (GPs) to learn the safety cost function given observations. The adaptive discretization of a given path with respect to the learned probabilistic cost function is performed *incrementally* by solving an optimization problem. The

Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108.

evaluation of the discretized path is then guaranteed to be ϵ-safe, i.e. the remaining risk of still finding an unsafe point on the trajectory is smaller than ϵ.

In Section 2, we provide a brief overview on related work. In Section 3, essential background on Gaussian Processes and learning probabilistic safety cost functions is introduced. Section 4 describes the proposed algorithm. In Section 5, we provide some results on the theoretical investigation of the proposed algorithm. For application in robotic path planning as an exemplary use case, we combine our adaptive discretization method with the well-known probabilistic roadmap (PRM) [Kavraki et al., 1996]. Thus, Section 6 provides a description of our approach when combined with the PRM for robotic path planning. The resulting safe planning approach is evaluated on several path planning tasks in Section 7. A conclusion is given in Section 8. The Appendix provides further details on proofs and experiments.

2 Related Work

Our approach shares some similarities with the work on the estimation of excursion sets [Azzimonti et al., 2016, Azzimonti et al., 2019]. However, the considered problem is different. Estimation of excursion sets focuses on the probability of the set of safe points, while we are interested in the probability to be safe at all incrementally found discretization points, resulting in different definitions of safety functions, see Appendix (Section B) for more explanations. In the field of safe active learning with Gaussian Processes [Schreiter et al., 2015, Turchetta et al., 2016, Zimmer et al., 2018], the proposed approach is mostly related to the work by [Zimmer et al., 2018] on safe exploration. While [Zimmer et al., 2018] employs an equidistant discretization and, thus, being inefficient for complicated cost functions, our approach attempts to overcome this limitation by introducing the adaptive discretization scheme. Our approach will be compared with the equidistant discretization as employed by [Zimmer et al., 2018] in Section 7.

The focus of this work is on the evaluation of the safety cost function, which can be complementarily combined with path planning approaches as done for evaluation in Section 7. In general, planning problems have been thoroughly investigated in the robotics community, enabling autonomous navigation and task execution [Latombe, 1990, Choset, 2005, LaValle, 2006]. There is a large body of work on deterministic planning [Canny, 1985, Jan et al., 2013, Elbanhawi et al., 2013], and probabilistic planning approaches [Elbanhawi and Simic, 2014, Chakravorty and Kumar, 2011, Luders et al., 2013]. Most of the path planning approaches consider the environment, e.g. the position and shape of the obstacles, to be deterministically known. For example, collision detection can typically be performed by employing assumptions on the geometry of obstacles and robot [Gottschalk et al., 1996, Reggiani et al., 2002]. In constrast to those approaches, we attempt to *learn* the environment directly from observations using Gaussian Processes. This idea has been exploited recently by several work, e.g. [Dragiev et al., 2011, Björkman et al., 2013, Driess et al., 2017]. The resulting environment map is subsequently employed as a probabilistic safety cost function to evaluate candidate paths. To the best of our knowledge, the proposed adaptive discretization technique for evaluating probabilistic cost functions in combination with a planning method is novel, while enabling efficient planning with safety guarantees.

The goal of our paper for safe path planning is to derive an adaptive discretization algorithm that bounds the risk of a remaining point being unsafe. We note that another quantity of interest could be the probability for the continuous path being safe, see e.g. [Adler and Taylor, 2007].

3 Learning and Evaluation of Safety Cost Function with GPs

In this paper, we employ a learned probabilistic safety cost function. Here, we use a Gaussian Process model (GP) $g : X \subset \mathbb{R}^d \to Z \subset \mathbb{R}$, mapping an input point $\boldsymbol{x}$ to a safety cost value z. In general, a GP is specified by its mean function $\mu(\cdot)$ and kernel function $k(\cdot,\cdot)$, i.e. $g(\boldsymbol{x}_i) \sim \mathcal{GP}(\mu(\boldsymbol{x}_i), k(\boldsymbol{x}_i, \boldsymbol{x}_j))$ [Rasmussen and Williams, 2006]. Given n noisy observations and a candidate path $\boldsymbol{\tau}$ discretized by m discretization points, i.e. $\boldsymbol{\tau} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m) \in \mathbb{R}^{d\times m}$, the predictive distribution for the safety values according to the GP prior is given as

$$p(\boldsymbol{\zeta}|\boldsymbol{\tau}, \boldsymbol{T}_n, \boldsymbol{Z}_n) \;=\; \mathcal{N}\left(\boldsymbol{\zeta}|\boldsymbol{\mu}(\boldsymbol{\tau}), \boldsymbol{\Sigma}(\boldsymbol{\tau})\right), \qquad (1)$$

where $\boldsymbol{\zeta} = (z_1, \ldots, z_m) \in \mathbb{R}^m$ contains m corresponding safety predictions for the discretization points. $\boldsymbol{Z}_n \in \mathbb{R}^n$ is a vector concatenating evaluated safety cost values z, and $\boldsymbol{T}_n \in \mathbb{R}^{n\times d}$ is a matrix containing observed input points $\boldsymbol{x}$. The mean vector and covariance matrix of the GP are defined by

$$\begin{aligned} \boldsymbol{\mu}(\boldsymbol{\tau}) &= \boldsymbol{k}(\boldsymbol{T}_n, \boldsymbol{\tau})^T (\boldsymbol{K}_n + \lambda^2 \boldsymbol{I})^{-1} \boldsymbol{Z}_n\,, \qquad (2)\\ \boldsymbol{\Sigma}(\boldsymbol{\tau}) &= \boldsymbol{k}^{**}(\boldsymbol{\tau}, \boldsymbol{\tau}) - \boldsymbol{k}(\boldsymbol{T}_n, \boldsymbol{\tau})^T (\boldsymbol{K}_n + \lambda^2 \boldsymbol{I})^{-1} \boldsymbol{k}(\boldsymbol{T}_n, \boldsymbol{\tau})\,, \end{aligned}$$

where $\boldsymbol{K}_n \in \mathbb{R}^{n\times n}$ represents the covariance matrix. As covariance function, a Gaussian kernel can be employed, i.e. $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \lambda_g^2 \exp(-\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{x}_j)^T \boldsymbol{\Lambda}_g^2 (\boldsymbol{x}_i - \boldsymbol{x}_j))$, which is parametrized by $\boldsymbol{\theta}_g = (\lambda_g^2, \boldsymbol{\Lambda}_g^2)$. Furthermore, we have an n-dimensional identity matrix $\boldsymbol{I}$, the output noise variance λ^2, and $\boldsymbol{k}^{**}(\cdot,\cdot) \in \mathbb{R}^{m\times m}$ as a matrix with $k_{ij}^{**}(\cdot,\cdot) = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. The matrix $\boldsymbol{k}(\cdot,\cdot) \in \mathbb{R}^{n\times m}$ contains kernel evaluations relating $\boldsymbol{\tau}$ to the previous n observed input points [Rasmussen and Williams, 2006, Zimmer et al., 2018]. Thus, for learning the *safety cost function* g by optimizing the hyper-parameters $\boldsymbol{\theta}_g$ and λ^2 the safety cost is evaluated for n input points, giving rise to the supervised training data set $\boldsymbol{T}_n$ as inputs and $\boldsymbol{Z}_n$ as outputs. Without loss of generality, the *evaluation of the cost*, i.e. z values, is designed such that all values greater than 0 are considered safe for the corresponding input $\boldsymbol{x}$. In safe dynamic DoE [Zimmer et al., 2018], for example, z is computed using feedback from the system, while indicating the distance of a given point $\boldsymbol{x}$ from the unknown safety boundary in the input space.

Having learned the probabilistic cost function g, we can evaluate the level of safety $\xi_m(\boldsymbol{\tau})$ for a candidate path as

$$\xi_m(\boldsymbol{\tau}) := P(z_1 > 0, \dots, z_m > 0) \tag{3}$$
$$= \int_{z_1>0,\dots,z_m>0} \mathcal{N}\left(z_1,\dots,z_m | \boldsymbol{\mu}(\boldsymbol{\tau}), \boldsymbol{\Sigma}(\boldsymbol{\tau})\right) dz_1,\dots,z_m$$

Thus, a path is considered to be safe, if $\xi_m(\boldsymbol{\tau})$ is sufficiently large. In general, the computation of $\xi_m(\boldsymbol{\tau})$ is analytically intractable and, thus, needs to rely on some approximation, such as Monte-Carlo sampling, expectation propagation [Minka, 2001] or Genz's approximation method [Genz, 1992].

In the experiments, we focus on modeling obstacles in a path planning scenario with GP implicit surfaces. Here, the cost function can be interpreted as a signed-distance function to the obstacles. Our method is not limited to obstacle modeling with implicit surfaces. For example, in an engine model setup, we can derive a cost function follows: Assume that pressure is a safety critical quantity and we know from domain experts that a pressure of 80 is a safety threshold. Then, we can define a cost function such that all pressures above 80 are mapped to negative values and all below to positive values. This cost function reflects our setting. While using a GP classifier is possible and has been used in the literature [Schreiter et al., 2015], GP regression allows us to gain more insights in how strongly safety is violated (little scratch versus destroying the whole system) like the distance to the obstacle or the distance to the pressure safety threshold and is, therefore, relevant information.

4 Adaptive Discretization for Evaluation of Continuous Path

As indicated by Eq. (3), the evaluation of the safety depends on the number of discretization points. The higher m, the more accurate is the total safety evaluation for the continuous path, which also increases the computational effort. Thus, we aim to develop an approach for evaluation of Eq. (3), where the discretization points are sequentially chosen, such that the evaluation of the safety has the desired precision for a given continuous path.

4.1 Problem Statement

Let our path $\boldsymbol{\tau} = (\boldsymbol{x}_t)_{t\in T}$ be parametrized by t, such that each t in an interval $T \subset \mathbb{R}$ is mapped to a point in the input space $\boldsymbol{x}_t$ by a mapping h. We assume that the mapping $h : T \to X$ is continuously differentiable or at least continuous and piecewise continuously differentiable with finitely many pieces. We discretize $\boldsymbol{\tau}$ with m discretization points at locations t_i, with $i = 1, ..., m$ and $m \geq 2$. Let $\boldsymbol{x}_{t_1} = h(t_1)$ denote the start point and $\boldsymbol{x}_{t_2} = h(t_2)$ the end point on the path, i.e. $\boldsymbol{\tau} = (\boldsymbol{x}_{t_1}, ..., \boldsymbol{x}_{t_2})$. For a candidate path connecting the start point with the end point, we are interested whether it is safe or, more generally, how much cost we accumulate along that path. To this end, we map each point on the discretized path $\boldsymbol{x}_{t_i}$ to a safety or cost value given a learned cost function, i.e. $z_i \sim g(\boldsymbol{x}_{t_i})$, where g is a GP. We assume that $z_i > 0$ indicates safe performance or a sufficient cost. Thus, we want to approximate the probability for a safe operation as given in Eq. (3) with m discrete points. The problem is to find the number of discretization points m and their corresponding locations t_i, such that $P(z_t \leq 0 | z_1 > 0, \dots, z_m > 0) \cdot P(z_1 > 0, \dots, z_m > 0) < \epsilon$ for all $t \in T = [t_1, t_2]$ with ϵ being the required precision.

4.2 The Algorithm

The basic steps of the procedure are summarized in Algorithm 1. The main goal is to estimate $\xi_m(\boldsymbol{\tau})$ with few discretizations m, while ensuring a required precision ϵ. Given m locations $t_1, \dots, t_m$ with positive safety ξ_m, the next position t^* to be evaluated is determined by

$$\begin{aligned} t^* &= \operatorname{argmax}_{t\in T} P(z_t \leq 0 | z_1 > 0, \dots, z_m > 0) \cdot \xi_m \\ &= \operatorname{argmax}_{t\in T} P(z_t \leq 0, z_1 > 0, \dots, z_m > 0) \end{aligned} \tag{4}$$

Thus, t^* is found by the worst-case probability for observing a negative value at another location $t \in T$. When this probability becomes small, it is reasonable to terminate the algorithm (see Step 5 in Algorithm 1). Practically, the probability $P(z_t | z_1 > 0, \dots, z_m > 0)$ can be computed, for example, by Monte Carlo sampling.

Algorithm 1 Adaptive Discretization for Evaluation of Probabilistic Cost Function

1: **Input**: $\epsilon > 0$, GP cost function g, a mapping h, candidate path $\boldsymbol{\tau}$.
2: **Initialization**: Set $\boldsymbol{x}_{t_1}$ and $\boldsymbol{x}_{t_2}$ as start point and end point of $\boldsymbol{\tau}$, i.e. $m=2$
3: Compute ξ_m according to Eq. (3) using g
4: Compute $t^* = \text{argmax}_{t \in T} P(z_t \leq 0 | z_1 > 0, \ldots, z_m > 0) \cdot \xi_m$
5: **if** $P(z_{t^*} \leq 0 | z_1 > 0, \ldots, z_m > 0) \cdot \xi_m < \epsilon$ **then**
6: Return ξ_m
7: **else**
8: Set $t_{m+1} = t^*$ and evaluate $\boldsymbol{x}_{t_{m+1}} = h(t_{m+1})$
9: Add $\boldsymbol{x}_{t_{m+1}}$ to the discretization set for $\boldsymbol{\tau}$
10: **go to** Step 3 with $m = m + 1$
11: **end if**

As $z_t, z_1, \ldots, z_m$ follows a normal distribution, one can easily draw samples from this distribution and hereby approximate the desired probability.

Furthermore, note that the mapping h has to be defined based on the parametrization of the path $\boldsymbol{\tau}$. For example, if $\boldsymbol{\tau}$ is parametrized as a hyperplane with $\boldsymbol{x}_{t_1}$ and $\boldsymbol{x}_{t_2}$ as start point and end point, then h can be given as: $h(t) = \boldsymbol{x}_{t_1} + t(\boldsymbol{x}_{t_2} - \boldsymbol{x}_{t_1})$; $t \in T = [0, 1]$. Thus, locations of discretization points can be obtained by optimizing t as provided in Eq. (4).

4.3 Computational Requirements

Assume that the dataset from which the GP is trained contains n data points. Training the GP on this dataset has a complexity of $\mathcal{O}(n^3)$, which, if the dataset does not change, has to be performed only once. By saving the Cholesky decomposition of the kernel matrix $\boldsymbol{K}_n + \lambda^2 \boldsymbol{I}$, computing the safety probability of a path for m discretization points with n_{MC} many Monte-Carlo samples has a complexity of $\mathcal{O}(mn^2 + m^3 + n_{\text{MC}} m^2)$ (mn^2 to compute the covariance prediction at the m discretization points, m^3 to perform a Cholesky decomposition of this covariance prediction and $n_{\text{MC}} m^2$ to generate the n_{MC} Monte-Carlo samples).

This shows that also from a computational point of view it is desired to need as little discretization points as possible to calculate the safety probability for a desired ϵ.

The number of Monte Carlo samples needs to be carefully (potentially adaptively) chosen in order to make sure that the MC error is small enough. Alternatively to MC, techniques like expectation propagation [Minka, 2001] could be applied. Sparse GP techniques [Quiñonero-Candela and Rasmussen, 2005, Snelson and Ghahramani, 2006] together with variational inference [Titsias, 2009] could further be utilized to reduce the computational complexity.

5 Theoretical Results

In this section, we provide some results on the theoretical investigation of the proposed algorithm. Detailed proofs and further results are given in the Appendix.

5.1 Convergence of ξ_m and justification of the algorithm's termination criterion

This subsection will show that the safety indicator ξ_m converges for $m \to \infty$ (Theorem 1). This indicates that it is reasonable to step-wise increase m, while evaluating the safety along the path. In addition, $\max_{t \in T} P(z_t \leq 0 | z_1 > 0, \ldots, z_m > 0) \cdot P(z_1 > 0, \ldots, z_m > 0)$ is a decreasing sequence in m. This represents the worst-case probability for each step. Therefore, the term is a useful termination criterion (Remark 1). This is an advantage over heuristically choosing m.

Theorem 1 (Convergence)**.** *Let $\boldsymbol{x}_{t_1}, \boldsymbol{x}_{t_2}, \ldots$ be a sequence of intermediate points. Let us denote the corresponding sequence of safety indicators with $z_1, z_2, \ldots$ Then, ξ_m defined in Eq.* (3) *converges for $m \to \infty$.*

Theorem 1 puts a theoretical justification to using ξ_m as an approximation for the cost along the path. Note that ξ_m is a decreasing sequence due to the definition as an intersection of sets of $z_i > 0$. However, it does not yet justify our termination criterion. Therefore, we also need that $\max_{t \in T} P(z_t \leq 0 | z_1 > 0, \ldots, z_m > 0) \cdot P(z_1 > 0, \ldots, z_m > 0)$ is a decreasing sequence, as only that guarantees us that the benefit of an additional iteration stays small once it got small, i.e.

Lemma 1 (Decreasing sequence)**.** *Using the same notation as in Theorem 1, it holds for each $t \in T = [t_1, t_2]$: $P(z_t \leq 0 | z_1 > 0, \ldots, z_m > 0) P(z_1 > 0, \ldots, z_m > 0) \geq P(z_t \leq 0 | z_1 > 0, \ldots, z_{m+1} > 0) \quad P(z_1 > 0, \ldots, z_{m+1} > 0)$.*

Remark 1. *Lemma 1 especially holds for the maximum over t. Therefore, the sequence $\max_{t \in T} P(z_t \leq 0 | z_1 > 0, \ldots, z_m > 0) \cdot P(z_1 > 0, \ldots, z_m > 0)$ is a decreasing sequence in m.*

Remark 1 means that the stopping criterion in Step 5 of Algorithm 1 is reasonable in the sense that if $\max_{t \in T} P(z_t \leq 0 | z_1 > 0, \ldots, z_m > 0) \cdot P(z_1 > 0, \ldots, z_m > 0)$ is small for a certain m, we know that it will only become smaller and can therefore terminate the Algorithm 1. Note that the determination criterion is guaranteed to be reached, as we will show in Theorem 3.

5.2 A different proof that leads us towards a convergence rate

We can also show that $\max_{t\in T} P(z_t \leq 0|z_1 > 0,\ldots,z_m > 0)\cdot P(z_1 > 0,\ldots,z_m > 0)$ becomes small with a different proof. This second proof is constructive in the way that it allows us to choose an ϵ and determine a minimum m, for which the term $\max_{t\in T} P(z_t \leq 0|z_1 > 0,\ldots,z_m > 0)\cdot P(z_1 > 0,\ldots,z_m > 0) < \epsilon$. This relation between $\epsilon > 0$ and m then allows us to derive a convergence rate. In addition, this also allows us to a-priori determine a number of intermediate points needed to reach a certain accuracy.

Definition 1. *For $t_1 \leq t \leq t_2$, define $t_+^m = \min\{t_i, i = 1,\ldots,m | t_i \geq t\}$ and $t_-^m = \max\{t_i, i = 1,\ldots,m | t_i \leq t\}$. Hence, $\boldsymbol{x}_{t_-^m}$ is the preceding point of $\boldsymbol{x}_t$ on the path and $\boldsymbol{x}_{t_+^m}$ is the following point.*

Within the GP framework with Gaussian kernels, mean and variance of any predictive distribution (conditional or unconditional) are continuously differentiable functions in the input variable. As $\boldsymbol{x}_t$ is a continuously differentiable function in t, mean and variance are also continuously differentiable in t. The interval T is compact. Therefore, there exist Lipschitz constants for the mean and variance as a function in t. Specifically, let us denote the mean and variance of $z_t|z_{t_+^m}$ for a $z_{t_+^m}$ as μ_{t,t_+^m} and σ_{t,t_+^m}. Then, it holds for all $\tilde{t} \in T$ that $|\mu_{t,\tilde{t}} - \mu_{t',\tilde{t}}| \leq L_1(\tilde{t})|t-t'|$ $\forall t,t' \in T$ and similar $|\sigma_{t,\tilde{t}} - \sigma_{t',\tilde{t}}| \leq L_2(\tilde{t})|t-t'|$. $L_1(\tilde{t})$ and $L_2(\tilde{t})$ are continuous functions in $\tilde{t}$. So, they have a maximum on the compact interval T which we will denote with $L_1 = \max_{\tilde{t}\in T} L_1(\tilde{t})$ and $L_2 = \max_{\tilde{t}\in T} L_2(\tilde{t})$. If the path $\boldsymbol{\tau}$ is only piecewise continuously differentiable, a Lipschitz constant can be derived for each piece and then an overall Lipschitz constant can be calculated with the triangle inequality.

Note that the Lipschitz constant L_1 depends on the values of $\boldsymbol{Z}_n$ and $z_{t_+^m}$, while L_2 does not, as the predictive variance of a GP does not depend on the outputs. While $\boldsymbol{Z}_n$ has been measured before the start of our algorithm (and n is finite), $z_{t_+^m}$ can vary and it might, therefore, be important to state the dependency of L_1 on $z_{t_+^m}$. The maximum of the first derivative of $\mu_{t,\tilde{t}}$ (see Eq. (2) for definition of mean) depends linearly on $z_{t_+^m}$ and so does L_1. Let $\bar{\mu} = \max_{t\in T} \mu_t$ be the maximum of μ_t and $\bar{\sigma} = \max_{t\in T} \sigma_t$ be the maximum of the standard deviation.

Definition 2. *For $\epsilon_1 > 0$, we define $\alpha(\epsilon_1)$ such that $P_{N(\mu,\sigma)}([\mu - \alpha(\epsilon_1)\sigma, \mu + \alpha(\epsilon_1)\sigma]) = P_{N(0,1)}([-\alpha(\epsilon_1), \alpha(\epsilon_1)]) = 1 - \epsilon_1$ (e.g. $\alpha(\epsilon_1) = 2$ for $\epsilon_1 = 0.05$, 2σ interval captures 95% of the probability mass).*

Assumption 1. *We denote the variance of the GP as σ_t and the mean as μ_t. We assume that either the variance is uniformly lower bounded or the absolute value of the mean. Formally, $\exists \alpha_1 > 0, \alpha_2 > 0$ such that $\forall t \in T$ it holds $|\mu_t| > \alpha_1$ or $\sigma_t > \alpha_2$.*

Assumption 1 is not restrictive, as it only requires that our GP is not degenerate (almost zero variance) and has almost zero mean at the same time. We define $\beta = \min(\alpha_1, \alpha_2)$.

Theorem 2. *As in Theorem 1, let $\boldsymbol{x}_{t_1}, \boldsymbol{x}_{t_2}, \ldots$ be a sequence of intermediate points for which it holds $t_1 \leq t_i \leq t_2, i = 1,2,\ldots$. Let us denote the corresponding sequence of safety indicators with $z_1, z_2, \ldots$. When introducing an $m+1$-st point into a set of m points, we can bound the probability*

$$\max_{t\in T} P(z_t \leq 0|z_1 > 0,\ldots,z_m > 0)\ \cdot\ \xi_m < \epsilon \qquad (5)$$

for any $\epsilon > 0$ if m is large enough such that

$$|t_+^m - t_-^m| < \frac{\delta}{L_1\gamma + \alpha(\epsilon_1) L_2} \qquad (6)$$

with $\delta = \frac{\epsilon_2}{2}\cdot\sqrt{2\pi}\beta$, $\gamma = \sqrt{2}\bar{\sigma}\log(\frac{1}{\epsilon_4}) + \bar{\mu}$ (see Lemma 5), t_+^m and t_-^m as in Definition 1, $\alpha(\epsilon_1)$ as in Definition 2, and β as in Assumption 1. ϵ_1, ϵ_2 and ϵ_4 can be chosen independently but need to fulfill $\epsilon_1 + \epsilon_2 + 2\epsilon_4 \leq \epsilon$.

In case of equidistant design, the condition on m can be simplified as we state in Corollary 1.

Corollary 1. *Using the same notation as in Theorem 2, it holds for an equidistant design with $t_i - t_{i-1} = \Delta, i = 2,\ldots,m$. We have the following bound $\Delta < \frac{\delta}{L_1\gamma + \alpha(\epsilon_1)L_2}$, for $m > \frac{L_1\gamma + \alpha(\epsilon_1)L_2}{\delta} + 1$.*

5.3 A convergence rate

As we have established a relation between ϵ and m, we can now state a convergence rate.

Theorem 3. *Under the assumption of Theorem 2, we achieve a convergence rate of $\frac{1}{1+\kappa}$ for an arbitrary $\kappa > 0$, i.e.*

$$P(z_t \leq 0|z_1 > 0,\ldots,z_m > 0)\cdot\xi_m = \mathcal{O}\left(m^{-\frac{1}{1+\kappa}}\right).$$

This convergence rate is build on the condition (6) stating that the set of intermediate points will become dense if m increases. In order to ensure this condition, we can enrich the adaptive set of points, for example, by choosing every m_{mix}-th point according to an equidistant placement. This approach would have the *same* convergence rate, however, with a constant worse by a factor of m_{mix}. Here, m_{mix} reflects the user's prefered balance between theoretical guaranty and amount of adaptiveness.

6 Gaussian Process Probabilistic Roadmap

We now describe how the proposed approach can be embedded into sampling based path planning algorithms, in particular probabilistic roadmaps [Kavraki et al., 1996], for planning safe paths in not perfectly known environments with obstacles. The resulting framework, which we call GP-PRM, provides provably safe graph construction, as well as path planning using the established graph. Note that the main focus of this work is the adaptive evaluation of the safety cost function. Therefore, many extensions to the resulting planning algorithm are possible.

6.1 Obstacles as GP Implicit Surfaces

For representing the obstacles and their uncertainty, we employ a GP implicit surface model, which has been used in robotics for grasping and tactile exploration [Dragiev et al., 2011, Björkman et al., 2013, Driess et al., 2017]. The underlying assumption is that objects in the input space are encoded in the zero-level set of a signed-distance function $z : X \to \mathbb{R}$, where $z(\boldsymbol{x}) > 0$ indicates *no* objects at $\boldsymbol{x}$, $z(\boldsymbol{x}) = 0$ and $z(\boldsymbol{x}) < 0$ imply that $\boldsymbol{x}$ is directly *on* the surface of an object and *inside* of an object, respectively. The surface is approximated as the zero-level set of the GP mean function $\mathcal{S} = \left\{\boldsymbol{x} \in \mathbb{R}^d : \mu(\boldsymbol{x}) = 0\right\}$. The advantage of this approach is that obstacles can be learned directly from raw sensor observations, i.e. no pre-processing to extract polyhedral shapes (or even convex decompositions) are necessary as required by related work, e.g. [Gottschalk et al., 1996]. Furthermore, such a model naturally fits into our safety evaluation with adaptive discretization.

6.2 GP-PRM

The proposed GP-PRM for safe path planning consists of two phases, the construction and the planning phase. In the first phase, a roadmap of safe paths is created which is represented by a graph (V, E) of configuration vertices $V \subset X$ and edges $E \subset \{(\boldsymbol{x}, \boldsymbol{x}') : \boldsymbol{x}, \boldsymbol{x}' \in V\}$. In the second phase, the actual path from a start $\boldsymbol{x}_s \in X$ to a goal $\boldsymbol{x}_g \in X$ configuration is planned. The graph (V, E) defines continuous paths $\boldsymbol{\tau}$ as linear splines via a sequence of adjacent vertices as support points, such that $h(t_1) = \boldsymbol{x}_{t_1} = \boldsymbol{x}_s$, $h(t_2) = \boldsymbol{x}_{t_2} = \boldsymbol{x}_g$. The resulting algorithm (cf. Sec. A.3 Algo. 2) for the construction phase follows in large parts the standard PRM procedure modified to incorporate the safety by our proposed adaptive discretization procedure. First, $n_V \in \mathbb{N}$ configurations from the space likely not containing an obstacle $X_{\text{free}} \subset X$ are sampled and added to V. The free space for a required safety probability $p_{\min}$ is estimated based on the safety evaluation Eq. (3) for a single point only, i.e. $X_{\text{free}} = \{\boldsymbol{x} \in X : P(z(\boldsymbol{x}) > 0) > p_{\min}\}$, which can be computed by the 1D Gaussian cumulative density function. To connect the vertices, first the $k_{\text{NN}} \in \mathbb{N}$ nearest neighbors $N_{\text{near}}(\boldsymbol{x}) \subset V$ of each $\boldsymbol{x} \in V$ are obtained. Then the safety probabilities of the paths connecting $\boldsymbol{x}$ to $\boldsymbol{x}' \in N_{\text{near}}$ are determined with our adaptive evaluation algorithm. If the safety probability is larger than $p_{\min}$, the edge $(\boldsymbol{x}, \boldsymbol{x}')$ is added to the graph. After the roadmap has been constructed, planning a path results in a graph search problem. In contrast to a standard PRM, where one would query the shortest path from the start to the goal, we additionally want to satisfy safety probability requirements on the *complete* path. In this work, we consider two approaches, one where a variety of candidate paths is generated by sampling a waypoint. The other approach is to solve a weighted shortest path problem, where the weights are determined by the distance and the safety probability on each edge. For details refer to the appendix Sec. A.3. Due to the fact that we discard edges with safety probability smaller than $p_{\min}$, the creation of the roadmap already pre-selects reasonable candidate paths. This is motivated by the fact that a longer path has less or equal safety probability than a sub-path.

	Adaptive	**Equi. coarse**	**Equi. fine**
PRM creation	27 ± 1	56 ± 4	162 ± 4
Planning	76 ± 3	309 ± 6	1309 ± 12
Path safe?	✓	✗	✓

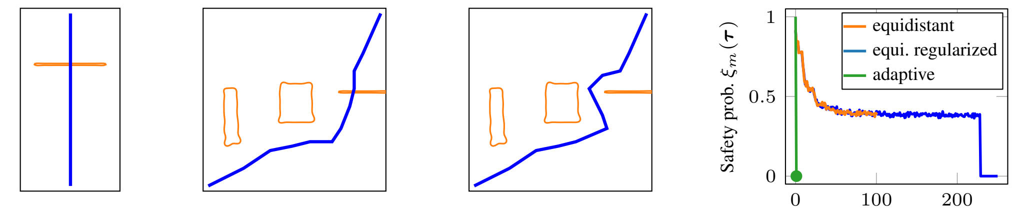
Table 1: Runtime in seconds for creation and planning of scenario in Fig. 1 with thin obstacle. We compare our adaptive discretization with equidistant discretization (coarse and fine). Planning is performed with the waypoint approach. The red cross indicates that the roadmap falsely contains paths that are unsafe.

7 Evaluation

We demonstrate the advantages of our adaptive evaluation algorithm compared to an equidistant evaluation as employed by [Zimmer et al., 2018] and show the behavior of the resulting GP-PRM planner for various scenarios. The appendix additionally contains another experiment and states all hyperparameters.

7.1 Thin Obstacle

In this experiment, we show that an equidistant evaluation could lead to highly overestimated safety probabilities. Fig. 1a shows the reconstructed object surface of a thin obstacle (orange). The true obstacle has a thickness of 0.01. The candidate path shown in blue with length 1.0 passes directly through the obstacle. For the position of the obstacle shown in Fig. 1a, the



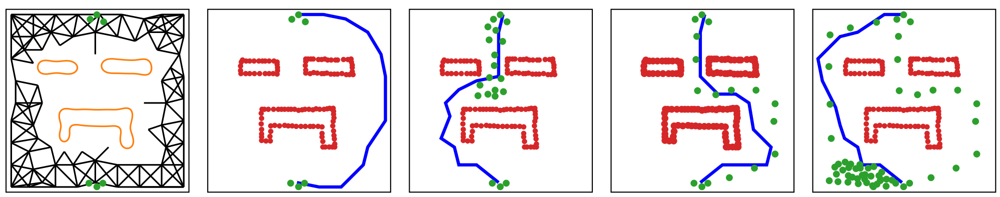
(a) Across thin obstacle

(b) Path from equidistant evaluation: $p = 0.39$

(c) Path from adaptive evaluation: $p = 0.46$

(d) Safety prob. for path of (b) over number of discretization points m

Figure 1: Path safety evaluation through thin obstacle. Orange is the learned obstacle surface from the GP model, blue denotes the path to be evaluated. The path in (b) is evaluated with the equidistant method, which falsely classifies the path as safe. (d) Evolution of the safety probability for path of (b) over the number of discretization points. The green line shows the adaptive evaluation, which after only 3 points evaluates the safety to zero.



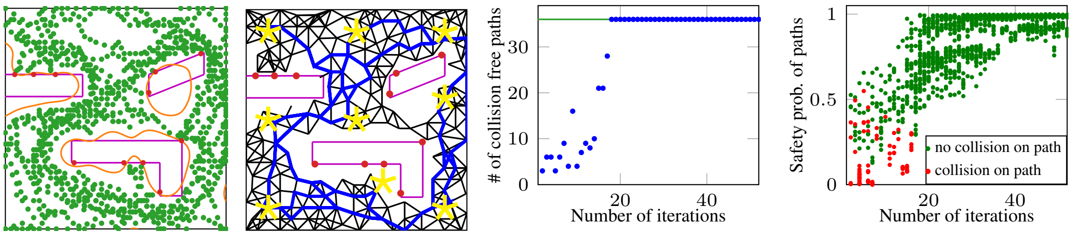
(a) Roadmap of (b) (b) $p = 0.35$ (c) $p = 0.44$ (d) $p = 0.69$ (e) $p = 0.80$

Figure 2: Adaptation to data. Orange in (a) show the learned obstacles from the GP. The blue path in (b)-(e) is the safest found path. Red dots are obstacle observations ($z(\boldsymbol{x}) = 0$), green dots free space ($z(\boldsymbol{x}) = 1$).

equidistant evaluation yields a safety probability of $p{=}0.53{\pm}0.008$ (100 different random seeds). To show that this not only happens for this specific configuration, we move the object in y-direction to 100 different positions and repeat the experiment. For the equidistant evaluation, in 59% of the obstacle positions, the estimated safety probability is greater than zero. The mean of those probabilities greater than zero is 0.51. In contrast, the adaptive evaluation in all these cases directly evaluates the safety probability to $p{=}0$ with only 3 discretization points. Two of those points arise from the initialization of the algorithm (start and end of the path). The point chosen by the algorithm is always inside the obstacle in this experiment. Next, we consider a planning task in a workspace that contains such a thin obstacle. Figure 1b and 1c show the reconstructed surfaces of the obstacles. The path passing through the thin obstacle in Figure 1b is planned with the equidistant evaluation and falsely predicts a safety of $p{=}0.39$. Again, with our proposed adaptive evaluation, such an unsafe path is never considered and the safest path with $p{=}0.46$ shown in Figure 1c avoids the thin obstacle. Note that here, the equidistant evaluation iterates until the covariance matrix $\boldsymbol{\Sigma}(\boldsymbol{\tau})$ becomes numerically indefinite, which happens before the resolution of the equidistant algorithm is fine enough to capture the thin obstacle. If the covariance is regularized, $\boldsymbol{\Sigma}(\boldsymbol{\tau}) + \sigma_p^2 \boldsymbol{I}$ for $\sigma_p = 10^{-5}$, the equidistant evaluation can be computed for a much finer resolution. In Figure 1d the evolution of safety probability for the path shown in 1b is plotted over the number of discretization points. The adaptive evaluation again only requires 3 points (green curve) to correctly evaluate the path as unsafe. The unregularized equidistant evaluation terminates after $m{=}101$ discretization points when $\boldsymbol{\Sigma}(\boldsymbol{\tau})$ becomes numerically indefinite, predicting a safety of $p{=}0.39$. The blue curve shows the regularized equidistant evaluation. Only after $m = 229$ points, the path is considered to be unsafe. Looking at these results shows the difficulty of determining the number of discretization points a priori, since a heuristically chosen convergence criteria would even in the regularized equidistant case terminate before the resolution is fine enough to correctly classify the path as unsafe. Furthermore, as can be seen in Table 1, the adaptive evaluation algorithm is significantly faster in both creating the PRM and the subsequent planning. Even when spending twice the amount of time for a coarse equidistant evaluation (resolution 0.012) compared to the adaptive algorithm, the equidistant variant still falsely plans an unsafe path.

7.2 Adaptation to Data

In Figure 2, we demonstrate how the knowledge about the environment influences the planned paths. In each of the subfigures 2b - 2e, the data representing the obstacles (red points) is the same, while the amount of observed free space data (green) varies. The path shown in blue is the safest path found for each. Figure 2a



(a) collected data (b) planning roadmap (c) Planning success (d) Safety prob. of planned paths

Figure 3: Workspace exploration result after 51 iterations. Violet are the true obstacles, orange in (a) the obstacle estimations based on the GP data. Green points in (a), (b) are collision free observations, red obstacle observations. The yellow stars in (b) are the 9 desired targets with the planned paths in blue. (c) shows the number of planned collision free paths between the 9 targets (36 are possible in total), (d) the safety probabilities of these paths.

shows the roadmap created of data in Figure 2b and the GP obstacle reconstruction (orange line). Depending on the available free space data, the safest planned paths follow this data. If very little data is available (Figure 2b), the safest path avoids the obstacles by large margin. When the information is available that there are no additional obstacles in the middle between the obstacles as for example in Figure 2c, the safest path passes through a more complex region of the workspace.

7.3 Exploration and Safe Path Planning for Autonomous Robots

In this experiment, we consider a 2D robot that explores the environment. The robot can only observe whether there is an obstacle or not at its current location, e.g. through direct contact in case of a vacuum cleaner robot. Starting with only a single observation, one iteration consists of first sampling a target position in the workspace. Then a path towards this target is planned based on the roadmap constructed with the currently available data with our proposed algorithm. The robot then moves either until it has reached the target or until it has collided with an obstacle. During the execution, data is collected. Afterwards, the GP model is updated with the new data and the procedure starts again. Figure 3a shows the true obstacles, the collected data and the obstacle GP model learned after 51 iterations. To evaluate the exploration progress, after each iteration, paths are planned between 9 target points (yellow stars in Figure 3b). In Figure 3c the number of successfully planned paths, i.e. collision free, between the targets is shown. After 18 iterations, all 36 paths between the targets are collision free. Figure 3d visualizes the corresponding safety probabilities. Since the exploration itself is performed with our proposed algorithm, very little collision data is required to safely plan paths in the whole workspace.

The true obstacles of this scenario have sharp corners. As can be seen in Figure 3a and also 1, 2, representing sharp objects with a homogeneous squared exponential kernel is difficult (which has been mentioned in the literature repeatedly in the context of implicit surface modeling [Driess et al., 2017, Driess et al., 2019]). However, in our case this is not crucial problem, since for solving the safe path planning task, one does not have to precisely reconstruct the exact surfaces of all obstacles. As shown in Figure 3a, only 11 obstacle observations are necessary to complete the task, while being far away from precisely modeling the exact obstacle shape. Furthermore, representing sharp corners with a GP implicit surface model can also lead to numerical instabilities. Here, our proposed adaptive discretization scheme is also advantageous, since it is numerically more stable than an equidistant variant.

In the appendix, further runs of the same experiment with different random seeds are shown as well as more details on the parameters. Please also refer to the video attachment for this experiment.

8 Conclusion

In this paper, we propose an adaptive discretization approach for safety path evaluation, while employing a probabilistic cost function. Our approach can be combined with other state-of-the-art planning techniques, e.g. probabilistic roadmap, to obtain provable safe path planning. The proposed technique is, however, general and presents a building block for safety related planning task, e.g. dynamic DoE and safe active learning.

Acknowledgement

We gratefully acknowledge proofreading and feedback from Sebastian Gerwinn.

References

[Adler and Taylor, 2007] Adler, R. J. and Taylor, J. E. (2007). *Random fields and geometry.* Springer.

[Azzimonti et al., 2016] Azzimonti, D., Bect, J., Chevalier, C., and Ginsbourger, D. (2016). Quantifying uncertainties on excursion sets under a gaussian random field prior. *SIAM/ASA J. Uncertainty Quantification*, 4:850–874.

[Azzimonti et al., 2019] Azzimonti, D., Ginsbourger, D., Rohmer, J., and Idier, D. (2019). Profile extrema for visualizing and quantifying uncertainties on excursion regions. application to coastal flooding. *Technometrics*, 0:1–26.

[Björkman et al., 2013] Björkman, M., Bekiroglu, Y., Högman, V., and Kragic, D. (2013). Enhancing visual perception of shape through tactile glances. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS).*

[Canny, 1985] Canny, J. (1985). A voronoi method for the piano-movers problem. In *IEEE Int. Conf. Robot. Autom. (ICRA).*

[Chakravorty and Kumar, 2011] Chakravorty, S. and Kumar, S. (2011). Generalized sampling-based motion planners. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics).*

[Choset, 2005] Choset, H. M. (2005). Principles of robot motion: Theory, algorithms, and implementation. Prentice-Hall.

[Dolgov et al., 2008] Dolgov, D., Thrun, S., Montemerlo, M., and Diebel, J. (2008). Practical search techniques in path planning for autonomous driving.

[Dragiev et al., 2011] Dragiev, S., Toussaint, M., and Gienger, M. (2011). Gaussian process implicit surfaces for shape estimation and grasping. In *2011 IEEE International Conference on Robotics and Automation.*

[Driess et al., 2017] Driess, D., Englert, P., and Toussaint, M. (2017). Active learning with query paths for tactile object shape exploration. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS).*

[Driess et al., 2019] Driess, D., Hennes, D., and Toussaint, M. (2019). Active multi-contact continuous tactile exploration with gaussian process differential entropy. In *Proceedings of the International Conference on Robotics and Automation (ICRA).*

[Elbanhawi and Simic, 2014] Elbanhawi, M. and Simic, M. (2014). Sampling-based robot motion planning: A review. *IEEE Access*, 2:56–77.

[Elbanhawi et al., 2013] Elbanhawi, M., Simic, M., and Jazar, R. (2013). Autonomous mobile robot path planning: A novel roadmap approach. In *Apllied Mech. Mater.*

[Genz, 1992] Genz, A. (1992). Numerical computation of multivariate normal probabilities. *Journal of Computational and Graphical Statistics*, 1:141–150.

[Gottschalk et al., 1996] Gottschalk, S., Lin, M. C., and Manocha, D. (1996). Obbtree: A hierarchical structure for rapid interference detection.

[Jaillet et al., 2008] Jaillet, L., Cortés, J., and Siméon, T. (2008). Transition-based rrt for path planning in continuous cost spaces. *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems.*

[Jan et al., 2013] Jan, G. E., Sun, C. C., Tsai, W. C., and Lin, T. H. (2013). An o(nlogn) shortest path algorithm based on delaunay triangulation. In *IEEE/ASME Trans. on Mechatronics.*

[Kavraki et al., 1996] Kavraki, L. E., Svestka, P., Latombe, J. C., and Overmars, M. H. (1996). Probabilistic roadmaps for path planning in high-dimensional confguration spaces. In *IEEE Tranaction on Robotics and Automation.*

[Latombe, 1990] Latombe, J.-C. (1990). Robot motion planning. Springer-Verlag.

[LaValle, 2006] LaValle, S. M. (2006). Planning algorithms. Cambridge Univ. Press.

[Luders et al., 2013] Luders, B. D., Karaman, S., and How, J. P. (2013). Robust sampling-based motion planning with asymptotic optimality guarantees.

[Minka, 2001] Minka, T. P. (2001). Expectation Propagation for Approximate Bayesian Inference. In *Uncertainty in Artificial Intelligence*. Morgan Kaufmann.

[Quiñonero-Candela and Rasmussen, 2005] Quiñonero-Candela, J. and Rasmussen, C. E. (2005). A Unifying View of Sparse Approximate Gaussian Process Regression. In *Journal of Machine Learning Research.*

[Rasmussen and Williams, 2006] Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning.* The MIT Press.

[Reggiani et al., 2002] Reggiani, M., Mazzoli, M., and Caselli, S. (2002). An experimental evaluation of collision detection packages for robot motion planning.

[Schreiter et al., 2015] Schreiter, J., Nguyen-Tuong, D., Eberts, M., Bischoff, B., Markert, H., and Toussaint, M. (2015). Safe Exploration for Active Learning with Gaussian Processes. In *ECML/PKDD*, volume 9286.

[Snelson and Ghahramani, 2006] Snelson, E. L. and Ghahramani, Z. (2006). Sparse Gaussian Processes using Pseudo-inputs. In *Advances in Neural Information Processing Systems*.

[Titsias, 2009] Titsias, M. K. (2009). Variational Learning of Inducing Variables in Sparse Gaussian Processes. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*.

[Turchetta et al., 2016] Turchetta, M., Berkenkamp, F., and Krause, A. (2016). Safe exploration in finite markov decision processes with gaussian processes. In *30th Conference on Neural Information Processing Systems*.

[Zimmer et al., 2018] Zimmer, C., Meister, M., and Nguyen-Tuong, D. (2018). Safe active learning for time-series modeling with gaussian processes. In *32nd Conference on Neural Information Processing Systems*.