IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Data-Efficient Constrained Robot Learning with Probabilistic Lagrangian Control

Shiming He and Yuzhe Ding

***Abstract*—We propose a novel framework for data-efficient black-box robot learning under constraints. Our approach integrates probabilistic inference with Lagrangian optimization. With the guide of a learned Gaussian process model, the Lagrange multiplier is controlled by the probability of whether the constraints would be satisfied. This reduces the typical oscillations seen in primal-dual updates and therefore improves both data efficiency and safety during learning. Both synthetic results and robot experiments demonstrate that our method is a scalable and effective solution for constrained robot learning problems.**

***Index Terms*—Probabilistic inference, reinforcement learning, compliance and impedance control**

I. Introduction

ROBOT learning often involves optimizing black-box objectives through costly interactions with the environment. In many real-world applications, the complexity increases due to safety or task constraints that are also black-box and difficult to evaluate. These challenges necessitate methods that both data-efficient and safety-aware.

A commonly used approach to these challenges is Bayesian Optimization (BO), which builds a probabilistic model of the objective and selects query points that are expected to be most informative. One effective and straightforward way to extend BO to solve constrained problems is to jointly model the objective and constraints [1], and then exploit this model to develop a query strategy that only explores regions likely to satisfy the constraints [2]. However, such methods inherit a fundamental limitation in that they do not scale well with respect to the dimensionality of the policy space [3].

To address this limitation, recent works have explored the use of gradient information within the BO framework, such as Gradient-Information Bayesian Optimization (GIBO) [4], which selects query points that improve gradient estimates to facilitate more effective search in high-dimensional policy spaces. In constrained settings, the use of gradient information naturally motivates incorporating Lagrangian relaxation techniques. However, in reinforcement learning (RL), where such techniques are used to solve constrained Markov Decision Processes (CMDPs), the primal-dual updates often exhibit oscillatory behavior [5].

This instability causes two main issues in the robot learning. First, the severity of physical hazards to robotic systems, such as excessive force, collisions, and actuator stress, often increases with the magnitude of the constraint violation. Large oscillations during learning can therefore significantly elevate the risk of damage. Second, although Lagrangian methods are known to almost surely converge when applied to CMDPs [6], the primary goal in the robot learning is not to achieve optimality, but rather to improve performance within a limited budget of iterations [7]. Unfortunately, oscillatory updates can slow down the exploration of the high reward region near the constraint boundary. This issue arises because, in many cases such as maximizing velocity under energy constraints, optimal policies often lie close to the boundary of the feasible region. Oscillations in learning dynamics can consistently perturb the policy away from this boundary.

We propose to reduce such oscillations arise in primal-dual updates while preserving the data-efficiency and the scalability of GIBO. Our key insight is that, while prior work has attempted to address this instability with heuristic control-based update rules for multiplier, they require careful manual parameter tuning and may still suffer from oscillatory behavior. By contrast, we leverage a probabilistic model of the objective and the constraints in the form of a multi-task Gaussian process (GP), which enables joint inference over both the objective and constraints, as well as their gradients with a tractable posterior. From this posterior, we derive a probabilistic criterion that quantifies the likelihood of constraint satisfaction before updating the policy. This facilitates an active control method for the multiplier, keeping it as small as possible to preserve performance, while ensuring a high probability of satisfying the constraints for each policy update.

Concretely, we propose GIBO-Lag, a safety-aware extension of GIBO for constrained black-box policy search. The proposed algorithm incorporates an active multiplier control method that reduces oscillations in the learning dynamics. We validate our approach on a large-scale synthetic problem and on multiple robot learning problems. Results demonstrate that GIBO-Lag consistently achieves feasible solutions in a data-efficient manner, exhibiting reduced oscillations compared to Lagrangian primal-dual subgradient methods.

II. Related Work

Constrained robot learning is an active field of research. Brunke et al. [8] provide a comprehensive survey that categorizes safety constraints into hard, probabilistic, and soft types,

Manuscript received: May 20, 2025; Revised October 21, 2025; Accepted December 7, 2025.

This paper was recommended for publication by Editor Clément Gosselin upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by Scientific Research Foundation of Hangzhou City University (X-202401). (Corresponding author: Shiming He)

Shiming He is with the School of Information and Electrical Engineering, Hangzhou City University, Hangzhou 310015, China (e-mail: hsm@hzcu.edu.cn).

Yuzhe Ding is with the College of Electrical Engineering, Zhejiang University, Hangzhou 310027, China (e-mail: yuzheding@zju.edu.cn).

Digital Object Identifier (DOI): see top of this page.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

and distinguishes between safety during and after learning. Depending on the specific problem setting, BO approaches vary across these categories. Early work on constrained BO extends the Expected Improvement (EI) acquisition function by incorporating constraint uncertainty [2]. Its performance is further improved by addressing the vanishing acquisition value problem through the logEI acquisition function [9]. However, both EI with constriants (EIC) and logEIC do not explicitly enforce safety during learning. Their acquisition functions may still favor exploration even at points with a low probability of satisfying the constraints when the expected improvement is high. Sui et al. [10] proposed SafeOpt, which guarantees safety with high probability during learning and has become a popular framework with many follow-up variants [11–14]. The core idea of SafeOpt is to determine two sets: one contains parameters that could optimize the unknown objective, and the other one with candidates to enlarge the safe set using a Lipschitz constant. While in [14] particle swarms have been utilized for an adaptive discretization the parameter space to scale up BO, the resulting optimizer does not come with the potential to extend its learning to policies that go beyond the scope of control parameter tuning.

An effective approach for scaling BO to higher-dimensional parameter spaces is to confine samples locally. Representative approaches include Trust Region BO (TuRBO) [15] and Gradient Information BO (GIBO) [4]. Scalable Constrained Bayesian Optimization (SCBO) proposed in [16] extends TuRBO to black-box constraints. SCBO employs Thompson sampling to select points of minimum total violation during learning, eventually identifying feasible policies with high performance. Thus, SCBO is a method that considers soft constraints. Our method adopts a similar problem setting, and builds upon GIBO to address constraint violations. Such violations are characterized as oscillatory learning dynamics. We introduce a probabilistic criterion that estimates how likely constraint violations are to occur after each policy update.

Apart from BO approaches, there are alternatives that demonstrate promising data efficiency in constrained robot learning by leveraging either the dynamical model or human demonstration. SafePen [17] and SAMBA [18] are extensions of the model-based PILCO [19] framework for safety-critical systems. They use a GP approximation to the transition dynamics. While model-based approaches can usually achieve high data-efficiency, they are limited to relative small state-action space [3]. Recently, Padalkar et al. [20] proposed a Kernelized Guided RL framework, which efficiently learns in-contact tasks from demonstrations (LfD) while enforcing safety during learning. Built on model-free RL, it avoids the limitations of model-based methods. Other safety-aware RL approaches, such as trust region methods [21] and Lyapunov methods [22, 23], are generally less data-efficient than LfD. RL typically uses neural network policy with numerous parameters. In contrast, we focus on learning policies with a few dozen parameters, such as weights for certain neural network layers or weights for movement primitives. Exploiting this structured policy representation reduces problem complexity and improves data efficiency. Moreover, our approach requires only priors on expected return, and can learn from scratch. It is also well-suited for fine-tuning policies learned by aforementioned safety-aware RL approaches, as demonstrated in our experiments.

III. Preliminaries

In this work, we formulate the robot learning as an episodic and constrained policy search problem. The agent interacts with the environment where each episode yields a trajectory $\tau = (s_0, a_0, \ldots, s_{K-1}, a_{K-1})$ of K states s and actions a. Each trajectory τ_θ is parameterized by a $d-$dimensional vector $\theta \in \Theta$ that lies in a bounded and closed set $\Theta \subset \mathbb{R}^d$. θ encodes configurable property of the robotic system. Specifically, θ represents the weights of movement primitives used to generate time-varying stiffness in a variable impedance controller in the experimental section. The overall task performance of a policy π_θ is quantified through a reward function $f_r : (\mathcal{S}\times\mathcal{A})^K \mapsto \mathbb{R}$.

In addition to maximizing task performance, the *feasible* policy must satisfy a set of constraints, which are encoded by a cost function $f_c : (\mathcal{S} \times \mathcal{A})^K \mapsto \mathbb{R}$ that evaluates whether a trajectory meets task-specific or safety-related requirements. Both the reward f_r and the cost f_c are treated as black-box functions, i.e., they are not available in closed form and can only be assessed through episodic roll-outs.

For estimating them, we assume both the reward function f_r and the cost function f_c are samples from a known GP prior so that correlations can be induced between tasks. This is realized by intrinsic model of coregionalization (ICM) [1] that defines a covariance function between input-task pairs

$$K_f((\theta, t), (\theta', t')) = K_t(t, t') \otimes K(\theta, \theta') \tag{1}$$

where $\otimes$ is the Kronecker product operator, K_t and K are covariance matrix, and they specify the similarities between tasks (t, t') and between inputs (θ, θ'), respectively. In addition, we assume the kernel functions $K : \Theta \times \Theta \mapsto \mathbb{R}$ are at least twice differentiable [4, 7]. This allows us to further derive the Jacobian of ICM and incorporate gradient information.

We define the set of feasible policies as $\Pi_c := \{\pi \in \Pi : f_c(\theta) \leq 0\}$. The constrained optimization objective is then given by

$$\theta^* = \arg\max_{\pi\in\Pi_c} f_r(\theta). \tag{2}$$

We approach the policy search problem by sampling trajectories τ with different policy parameterizations. For each roll-out, the specific choice of θ yields a noisy observation of both the reward and the cost function

$$y = f(\theta) + \epsilon \tag{3}$$

where $f = [f_r, f_c]^\mathsf{T}$, and we assume $\epsilon \sim \mathcal{N}(\mathbf{0}, \Sigma_n)$.

Lagrange relaxation is often used to solve CMDP, and it approaches the problem by converting it into an equivalent unconstrained problem

$$\min_{\lambda>=0} \max_{\theta} \mathcal{L}(\lambda, \theta) = \min_{\lambda>=0} \max_{\theta} (f_r - \lambda f_c) \tag{4}$$

where $\mathcal{L}$ is the Lagrangian and λ is the Lagrange multiplier. By optimizing the dual problem $g(\lambda) = \max_\theta \mathcal{L}(\lambda, \theta)$ over λ, the optimizer finds a feasible saddle point $(\theta_*(\lambda_*), \lambda_*)$ of (4). It leads to a solution that may be of use to the primary

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

CMDP problem (see weak duality [24]). In the context of RL, objectives and constraints are analytically intractable, and therefore, a two-timescale stochastic approximation approach is adopted to update θ and λ with their gradients [6]

$$\theta_{i+1} \leftarrow \Gamma_\theta \left[\theta_i + \eta_1(i)\nabla_\theta \mathcal{L}(\lambda_i, \theta_i)\right], \tag{5}$$

$$\lambda_{i+1} \leftarrow \Gamma_\lambda \left[\lambda_i - \eta_2(i)\nabla_\lambda \mathcal{L}(\lambda_i, \theta_i)\right] \tag{6}$$

where Γ_θ and Γ_λ are projection operators. Γ_θ projects θ onto a compact and convex set to ensure the iterate θ stable. Γ_λ keeps the multiplier λ within the interval $[0, \lambda_{\max}]$.

IV. Constrained Local Policy Search

A. Intrinsic Jacobian GP Model of Coregionalization

We start from deriving a Jacobian GP to model the reward and cost functions. The model is used to jointly infer functions and their gradients, and the resulting gradient estimates are then utilized to update the policy. With the Jacobian GP model in [4] and the ICM (1), we derive the joint distribution between noisy observations from different tasks and the derivative at policy $\hat{\theta}$

$$\begin{aligned}\begin{bmatrix}\overline{y}\\ \mathrm{vec}J(\hat{\theta})\end{bmatrix} \sim \mathcal{N}\Bigg(&\begin{bmatrix}\mu(\overline{\theta})\\ \nabla\mu(\hat{\theta})\end{bmatrix},\\ &\begin{bmatrix}K(\overline{\theta},\overline{\theta})\otimes K_t & \nabla K(\overline{\theta},\hat{\theta})\otimes K_t\\ \nabla K(\hat{\theta},\overline{\theta})\otimes K_t & \nabla^2 K(\hat{\theta},\hat{\theta})\otimes K_t\end{bmatrix}\Bigg)\end{aligned} \tag{7}$$

where $\overline{y} = \begin{bmatrix} y_1^{1:m} \\ \cdots \\ y_n^{1:m}\end{bmatrix} \subset \mathbb{R}^{mn}$ with m tasks and n distinct d-dimensional inputs θ for each. $\overline{\theta} = \begin{bmatrix}\theta_1 \\ \cdots \\ \theta_n\end{bmatrix} \subset \mathbb{R}^{n\times d}$. J is a Jacobian matrix belief over the latent function $f : \mathbb{R}^d \mapsto \mathbb{R}^m$. $\mu(\overline{\theta})$ and $\nabla\mu(\hat{\theta})$ are priors that are usually assumed to be 0. By conditioning the prior joint distribution on the observation, we obtain the posterior predictive distribution

$$\begin{aligned}&\mathrm{vec}J(\hat{\theta})|\overline{\theta},\overline{y} \sim \mathcal{N}\left(\mu'(\hat{\theta}), \Sigma'(\hat{\theta})\right),\\ &\mu'(\hat{\theta}) = \nabla\mu(\hat{\theta})\\ &+\left(\nabla K(\hat{\theta},\overline{\theta})\otimes K_t\right)\left(K\left(\overline{\theta},\overline{\theta}\right)\otimes K_t\right)^{-1}\left(\overline{y}-\mu(\overline{\theta})\right),\\ &\Sigma'(\hat{\theta}) = \nabla^2 K(\hat{\theta},\hat{\theta})\otimes K_t\\ &-\left(\nabla K(\hat{\theta},\overline{\theta})\otimes K_t\right)\left(K(\overline{\theta},\overline{\theta})\otimes K_t\right)^{-1}\left(\nabla K(\overline{\theta},\hat{\theta})\otimes K_t\right).\end{aligned} \tag{8}$$

In the Jacobian ICM we use a "task priority" permutation for $\overline{y}$, whereas the ICM in [1] uses a "parameter priority" permutation, i.e., $\breve{y} = \begin{bmatrix} y_{1:n}^1 \\ \cdots \\ y_{1:n}^m\end{bmatrix} \subset \mathbb{R}^{nm}$. Thus, K_t is on the right side of K in (7) that contrasts with that of the original ICM form in (1). This adjustment on permutation is necessary for deriving the posterior distribution of the derivative of the objective function, which is not considered in previous work. The vectorized Jacobian matrix $\mathrm{vec}J(\hat{\theta})$ is thus permuted using task priority. However, this is not suitable for constructing a Lagrangian that requires $\mathrm{vec}J^\mathsf{T}(\hat{\theta})$. To circumvent this problem, we use a commutation matrix P such that

$$\begin{aligned}&\mathrm{vec}J^\mathsf{T}(\hat{\theta}) \sim \mathcal{N}\left(\breve{\mu}, \breve{\Sigma}\right)\\ &\breve{\mu} = P\mu'\\ &\breve{\Sigma} = P\Sigma' P^\mathsf{T}\end{aligned} \tag{9}$$

where P can be constructed using slices of an identity matrix.

B. Probabilistic Primal-Dual Optimization

The Jacobian ICM derived in (7) provides a posterior joint distribution of derivatives of objectives f_r and f_c. This has the advantage that the closed-form predictive distribution on the Lagrangian can be obtained by exploiting the linear combination property of the normal distribution. Thus, one can take the posterior predictive mean as the descent direction for the policy update. Further, an issue of gradient Lagrangian approach (5), (6) is that the cost and the multiplier λ suffer oscillation throughout learning, in particular when the learning rate of λ is ill-determined [5]. We overcome this issue by updating λ with an active approach that guided by the probability of whether the safety constraint is satisfied. Akin to the standard Lagrangian (5), (6), we use a two-timescale approach introduced below:

Update policy. Here we propose an update rule that improves the Lagrangian. Since we have $\nabla_\theta\mathcal{L} = \nabla f_r - \lambda\nabla f_c$, with the Jacobian ICM model (7) and its posterior (8), the derivative of the Lagrangian $\mathcal{L}$ w.r.t. $\hat{\theta}$ is normally distributed

$$\begin{aligned}&\nabla_\theta\mathcal{L}(\lambda,\hat{\theta}) \sim \mathcal{N}(\mu_l, \Sigma_l)\\ &\mu_l = \begin{bmatrix} I & -\lambda I\end{bmatrix}\breve{\mu}\\ &\Sigma_l = \begin{bmatrix} I & -\lambda I\end{bmatrix}\breve{\Sigma}\begin{bmatrix} I & -\lambda I\end{bmatrix}^\mathsf{T}\end{aligned} \tag{10}$$

where I is an identity matrix of size d. Similar to (5), the policy is thus updated with the mean of $\nabla_\theta\mathcal{L}$

$$\hat{\theta} \leftarrow \hat{\theta} + \eta\mu_l. \tag{11}$$

Update multiplier. The multiplier is normally considered as a penalty coefficient. The update law (6) integrates the "distance" that f_c exceeds the constraint, and it suggests that farther f_c away from the constraints, the more pronounced the impacts of these tasks to the optimization problem become; however how could we ascertain whether these impacts exerted are adequate and appropriate? To answer this question, we advance the understanding of the Lagrangian multiplier by translating it to weight that helps to identify a descent direction common to all tasks.

Once f_c would probably breach safety constraint, a desirable descent direction would not only guarantee that f_c will stay in constraints but also do not degrade the objective function f_r as much as possible after a gradient step. It motivates us to find an optimal descent direction controlled by λ for adaptively trade-off the reward and cost. Recall that by twice continuously differentiable assumption on the reward and cost functions, we can therefore apply a linear approximation at $\hat{\theta}$

$$f(\theta^+) \approx f(\hat{\theta}) + J(\hat{\theta})(\theta^+ - \hat{\theta}) \tag{12}$$

where $J = \begin{bmatrix}\nabla f_r^\mathsf{T} \\ \nabla f_c^\mathsf{T}\end{bmatrix}$ is a Jacobian matrix. We apply the policy update (11) to (12), resulting in

$$f(\theta^+) \approx f(\hat{\theta}) + \eta J(\hat{\theta})\mu_l \tag{13}$$

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

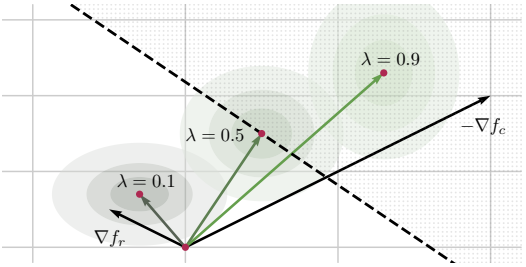

Fig. 1: A geometric illustration of the active control for the Lagrange multiplier: The constraints are denoted by the dotted line, with the dotted area being the feasible region. The current policy and the next policy are marked with the red circle. Using Eq. (14), we obtain the probability distribution of the new cost after the policy update with different λ. In this case, $\lambda = 0.9$ can guarantee the high probability safe update.

which is a linear transformation, and thanks to that, the cost estimates $F_c(\theta^+)$ after the policy update is normally distributed, and can be written as

$$F_c(\theta^+) \sim \mathcal{N}\left(f_c(\hat{\theta}) + \eta\mu_l^\mathsf{T}\begin{bmatrix}0 & I\end{bmatrix}\breve{\mu}, \; \eta\mu_l^\mathsf{T}\begin{bmatrix}0 & I\end{bmatrix}\breve{\Sigma}\begin{bmatrix}0 & I\end{bmatrix}^\mathsf{T}\mu_l\right). \tag{14}$$

Given λ, the likelihood of f_c not violating the constraint after the policy update is a cumulative probability

$$p_c(\lambda) = \Pr[F_c(\theta^+) \leq 0 \mid \lambda]. \tag{15}$$

We introduce a threshold β, and $p_c(\lambda) \geq \beta$ reveals satisfaction of the safety constraint with β probability. Intuitively, if the multiplier λ increases, the gradient of f_c becomes more dominant in the descent direction of Lagrangian, and therefore it leads to faster decrease of the cost f_c; however a large multiplier may compromise the improvement of the reward f_r. Thus, a straightforward idea is to find a minimal λ in a set $\{\lambda \mid p_c(\lambda) \geq \beta, \lambda \in [0, \lambda_{\max}]\}$ (see Fig. 1 for an illustration).

C. Acquisition of Gradient Information

To efficiently estimate the descent direction of the Lagrangian, we extend the query defined in GIBO [4] to

$$\textsc{Query}(\mathcal{D}, \hat{\theta}, \lambda) \coloneqq \arg\max_{\theta \in \Theta} \alpha_{\text{GI}}(\theta \mid \mathcal{D}, \hat{\theta}, \lambda) \tag{16}$$

where $\mathcal{D}$ is the dataset $(\overline{\theta}, \overline{y})$ and α_{GI} is the Gradient Information (GI) acquisition function

$$\alpha_{\text{GI}}(\theta \mid \mathcal{D}, \hat{\theta}, \lambda) = \mathbb{E}\left[\text{Tr}(\Sigma_l \mid \mathcal{D}) - \text{Tr}(\Sigma_l \mid \mathcal{D} \cup (\theta, y))\right] \tag{17}$$

with Σ_l being the gradient variance of the Lagrangian at $\hat{\theta}$ *before* and *after* observing a new point (θ, y).

The GI acquisition function for Lagrangian inherits an important property from the original GI, i.e., it can actively select the most informative points for estimating the descent direction of the Lagrangian. In particular the multiplier translates the uncertainty in both tasks f_r, f_c, naturally to a joint form. As an extreme case, if $\lambda \to \infty$, we attribute the variance in Lagrangian gradient to the variance in cost function f_c. On the other hand, if $\lambda = 0$, optimizer would not take the uncertainty in the gradient estimates of cost function into account.

D. GIBO with Lagrangian

We summarize the gradient information Bayesian optimization with Lagrangian (GIBO-Lag) in Algorithm 1. GIBO-Lag uses the general GIBO framework which consists of two nested loops to optimize the Lagrangian function via Lagrangian relaxation. The Jacobian ICM (7) provides the mean μ_l for descent direction (line 12), and covariance Σ_l for maximizing the gradient information (line 5). Both μ_l and Σ_l are used to compute the $p_c(\lambda)$ (line 10). GIBO-Lag actively queries the most informative points in the query loop (line 4-11). When M such queries have been made, it leaves the query loop and use gradient ascent with learned direction. To further improve data efficiency, the query loop can be early-stopped based on the improvement confidence criterion [7]. GIBO-Lag incorporates a heuristic approach (line 8-10) to find a multiplier that balances the reward and the cost, guided by a cumulative probability.

Algorithm 1 GIBO-Lag

1: **Hyperparameters**: hyperpriors for GP hyperparameters, threshold for multiplier update β, upper bound for the multiplier, batch size M, step-size η
2: $\hat{\theta} \leftarrow \theta_0$, $\mathcal{D} \leftarrow \mathcal{D}_0$
3: **repeat** ▷ Policy updates
4: **repeat** ▷ Collect data to estimate the gradient
5: $\theta \leftarrow \arg\max_\theta \alpha_{\text{GI}}(\theta | D, \hat{\theta}_i, \lambda)$ ▷ Eq. (16)
6: $y \leftarrow f(\theta) + \epsilon_i$ ▷ Policy evaluation
7: $\mathcal{D} \leftarrow \mathcal{D} \cup (\theta, y)$, $M \leftarrow M - 1$, $\lambda \leftarrow 0$
8: **repeat** ▷ Multiplier updates
9: increase λ
10: **until** $p_c(\lambda) \geq \beta$ or λ reaches its upper bound
11: **until** $M <= 1$
12: $\hat{\theta}_{i+1} \leftarrow \hat{\theta}_i + \eta\mu_l$
13: **until** required solution accuracy achieved

V. Experiments

We evaluate our approach on two problem settings. One is with large-scale synthetic objectives, and the other is with simulated and real robot learning environments. This design of experiment allows us to first validate the method under ideal conditions and then assess its performance in more realistic, challenging scenarios.

A. Synthetic Experiments

In this experiment, we adopt a commonly used benchmark for BO, known as the within-model comparison setting [25]. In this setup, synthetic objective functions are sampled from a known Gaussian Process (GP) prior, which ensures that the optimizer operates under the correct model assumptions. This also allows us to correctly set the prior in the optimizer.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

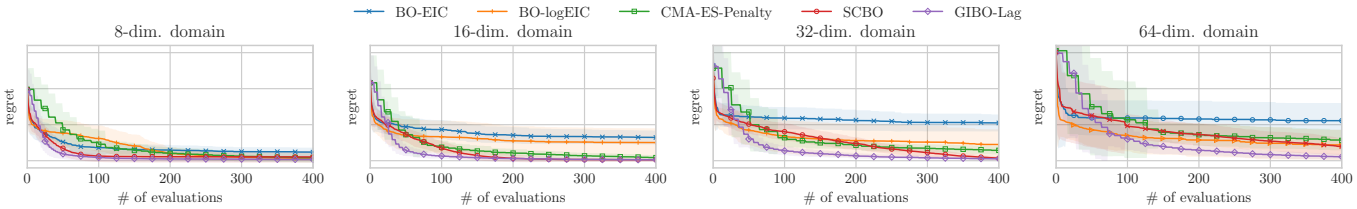

Fig. 2: Results on synthetic problems with four different dimensional function domains. Curves show average performance over 100 constrained optimization problems per domain. GIBO-Lag achieves lower regret with fewer samples, especially in high-dimensional domain.

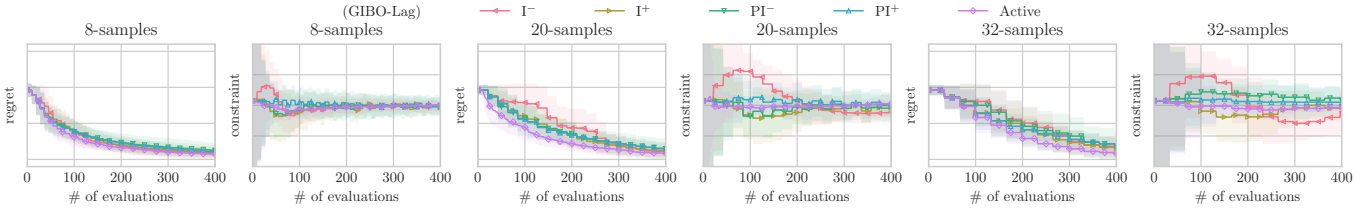

Fig. 3: Assessing the robustness of GIBO-Lag with active multiplier control method. Experimentsare conducted on a 64-dimensional domain with different batch sizes. For I- and PI-controlled multiplier, the superscripts $+$ and $-$ indicate relatively larger and smaller control parameters, respectively. Active multiplier control method consistently demonstrates reduced oscillations and overshoot across all batch sizes, as evidenced in the constraint plots.

Such a setting enables a fair comparison of performance and scalability across different BO methods under ideal conditions.

We divide the within-model comparison into two parts. In the first part, we demonstrate GIBO-Lag can achieve higher data efficiency and lower regret while satisfying constraints. To this end, we compare it against several state-of-the-art baselines: BO with Expected Improvement With Constraints (BO-EIC) [2], BO with logEIC (BO-logEIC) [9], Scalable Constrained BO (SCBO) [16], and Covariance Matrix Adaption Evolution strategy with Penalty (CMA-ES-Penalty). In the second part, we aim to validate the effectiveness of the proposed active multiplier control method in reducing the oscillations that arise in primal-dual methods. Within the framework of GIBO-Lag, we compare our active approach with heuristic control-based *integral* (I) and *proportional-integral* (PI) multiplier updates.

Experiments are conducted over a d-dimensional unit domain $I = [0, 1]^d$ with $d = 8, 16, 32, 64$. For each domain, we sample 1000 function values from a multi-task GP prior with a Squared Exponential (SE) kernel and signal variance $\sigma = 0.01$. The resulting correlated posterior mean serves as the reward and cost function. This process is repeated to generate 100 synthetic constrained optimization problems per domain.

We assume a zero mean function for both reward and cost in the GP prior. The covariance matrix is set to $K_t = \left[\begin{smallmatrix} 1 & 0.2 \\ 0.2 & 1 \end{smallmatrix}\right]$, indicating that the reward and cost functions are positively correlated. This on one hand reflects realistic scenarios where increasing the reward often comes with higher cost, and on the other hand demonstrates algorithm's efficiency, in particular in navigating trade-offs. To maintain comparable optimization difficulty across dimensions, we increase lengthscales of GP as dimensions increase. Lengthscale distributions are available in Appendix A.5 of [4]. In synthetic experiments, all algorithms have a budget of 400 noisy evaluations of the reward and cost functions, and are started in the middle of the domain $[0.5]^d$. The GP prior is provided to all BO methods, and to ensure a fair comparison, we fine-tune space-dependent hyperparameters for the CMA-ES algorithm on the cluster to optimize its average performance.

Fig. 2 shows the normalized difference between the constrained global optimum and the function value of the optimizer's best guesses within the feasible region. If the initial policy exceeds the constraint $f_c(\theta_0) > 0$, we set the initial best guesses $f_r(\hat{\theta}_0) = 0$. When comparing points that violate constraints, we define the best guess as the one with the least total violation. Under this evaluation criterion, GIBO-Lag consistently achieves lower regret compared to the baseline methods. Furthermore, it significantly reduces the variance of the obtained regret, indicating more stable performance across different dimensional domain. Compared to other local optimizers, GIBO-Lag converges faster toward the constrained optimum, exhibiting higher data efficiency. Note that global optimizers (BO-EIC and BO-logEIC) converge much faster than GIBO-Lag in particular in high dimensional domain. This may be because global optimizers can easily locate the feasible region in our synthetic setting, whereas local optimizers must continue to probe an unfeasible region for a more promising direction, when the initial policy exceed the constraints.

In the second part of the synthetic experiment, we evaluate the effectiveness of the proposed active multiplier control method in stabilizing the optimization process. Specifically, we conduct an ablation study on a 64-dimensional domain with batch sizes of 8, 20, and 32. Following the control concept from [5], we compare our active multiplier update method against I and PI update method. Notably, all three

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

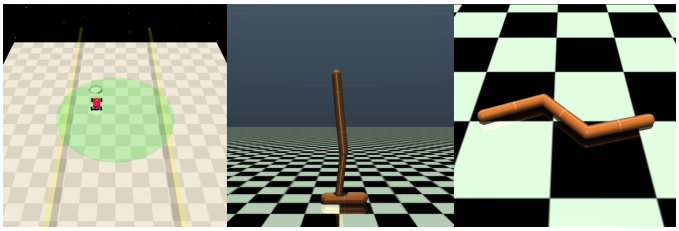

Fig. 4: Safety-Gymnasium environments. Left: Circle1, middle: Hopper, and right: Swimmer.

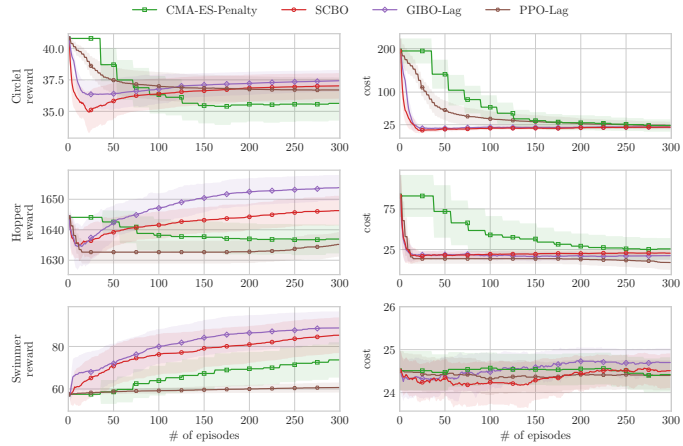

Fig. 5: Average fine-tuning performance of constrained RL agents over 100 runs. GIBO-Lag achieves higher rewards while satisfying constraints (cost < 25) across multiple environments.

multiplier update methods are implemented within the GIBO-Lag framework. This ensures that any observed differences in performance can be attributed solely to the multiplier update. For the I and PI methods, we hand-tune their parameters to examine how different settings affect the oscillatory behavior of the constraint violations. As shown in Fig. 3, our active control approach not only achieves lower regrets but also improves the learning dynamics, exhibiting less oscillations and overshoot. In contrast, the I and PI methods often require manual parameter tuning, and the results clearly show that they may still suffer from instability and oscillatory behavior. These improvements are observed across all batch sizes, underlining the robustness of the active multiplier control method.

B. Robot Learning Experiments

To assess the effectiveness of our method in more realistic settings, we consider the following two tasks:

Adapting RL Agents to New Constraints. In this experiment, we consider a scenario where an agent, pretrained using proximal policy optimization (PPO-Lag), have to adapt to a stricter constraints. We follow the setting from [26]. The goal is to fine-tune its strategy to meet new safety requirements in real-world applications while continuing to maximize rewards. The agent uses 2-layer NNs with $n_h = 64$ hidden units each, but only the weights of the linear mapping from the last hidden layer to the actions are optimized. This leads to $(n_h + 1) \times \dim(\mathcal{A})-$dimensional parameters.

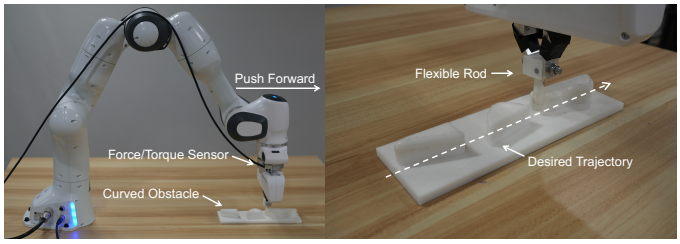

Fig. 6: Experimental setup for variable impedance control using a flexible rod. The robot follows a desired pushing trajectory (dotted line), making contact with the curved obstacle in each episode. The objective is to minimize trajectory tracking error while satisfying contact force constraints.

We compare our algorithm to its competitors (SCBO, CMA-ES-Penalty) in the following three environments from the Safety-Gymnasium [27] (see Fig. 4): SafetyPointCircle1-v0 (130 dim.), HopperVelocity-v1 (195 dim.), and SwimmerVelocity-v1 (130 dim.). Detailed task descriptions and the definitions of reward and cost functions can be found in the Safety-Gymnasium documentation.For the navigation task (Circle1), we modify the cost signal by changing the wall location. For the velocity task (Hopper and Swimmer), we change the cost signal by decreasing the velocity threshold. We also include PPO-Lag in our comparison, and reduce its training batch size to encourage fast adaption. We do not compare again global optimizers (BO-EIC and BO-logEIC) in this and the following experiment, as they explore the entire policy domain, which may easily generate unstable policies that severely violate safety constraints, in particular at the start of optimization. Alongside PPO-Lag, GIBO-Lag, SCBO, and CMA-ES-Penalty are evaluated within an episodic setting where the number of steps may differ between environments but remains consistent across all algorithms. We show performance over the episode in this experiment. All algorithms start from the same checkpoint with a limited budget of 300 episodes.

The results in Fig. 5 are based on averages from 100 independent runs. In the Circle1 and Hopper environments, the policy trained using PPO-Lag no longer meets the constraint $f_c \leq 25$ due to changes in the cost signal. In Fig. 5 (a) and (b) we observe that all the algorithms aim to satisfy the constraints, which leads to a decrease in rewards. When the new cost aligns with the constraints, we see a slight increase in rewards. In both cases, GIBO-Lag and SCBO significantly outperform PPO-Lag and CMA-ES-Penalty, and GIBO-Lag achieves higher rewards than SCBO while their ability to handle constraints are similar. This finding is consistent with Fig. 2 in that GIBO-Lag outperforms SCBO. To further validate our results, we tune the cost signal in the Swimmer environment so that constrained optimization is initialized almost on the boundary. Again, GIBO-Lag demonstrates effectiveness in improving rewards when the constraint is active.

Learning the Precision-Compliance Tradeoff. Impedance control is a widely used approach for robot to ensure safe physical interaction with the environment, and the impedance can vary during the task [28]. This is known as variable

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

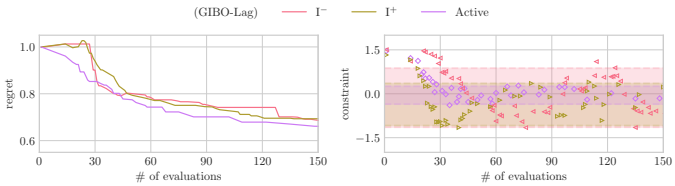

Fig. 7: Training curves of GIBO-Lag with different multiplier update method. Markers indicate policy update iterations, and shaded areas represent the 95% quantile interval of cost function after it first satisfies the constraint.

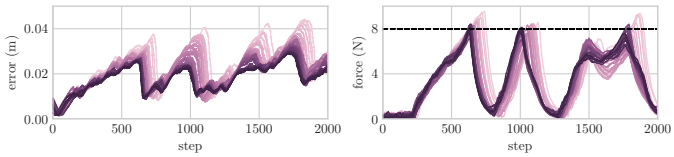

Fig. 8: Tracking error and contact force after each policy update iteration. Darker lines correspond to later iterations. The learned policy reduces tracking error while maintaining contact force below 8 N per episode.

impedance control. Our goal is to help robots learn to adapt impedance to specific tasks and environments. In this real-world experiment, the robot arm aims to track the trajectory as accurate as possible while keeping contact force within safety limits. This goal is ubiquitous in robotic applications such as polishing and peg-in-hole tasks, which require a sophisticated trade-off between precision and compliance. A high-precision robot controller may lead to high contact forces that damage the robot or object. A fully compliant controller, on the other hand, might cause the robot to get stuck due to environmental friction and resistance.

Inspired by [29], the experiment setup is shown in Fig. 6. During each roll-out with a fixed duration T, the robot arm holds a flexible 3D-printed rod rigidly attached to its end-effector and moves it along a straight line in the horizontal plane. A fixed curved obstacle lies in the rod's path, resulting in inevitable contact during motion. As the rod contacts the obstacle, friction and resistance lead to intermittent sticking and jerky movement.

Constrained optimization problem: we want the robot to move the rod at a constant speed along the centerline of the obstacle while maintaining safe contact forces during the entire roll-out. Thus, the optimal policy π^* is obtained by solving

$$\pi^* = \arg\max_{\pi \in \Pi_c} \sum_{t=0}^{T} - \left\| \begin{bmatrix} x(t) - vt \\ y(t) \end{bmatrix} \right\|_2 \tag{18}$$

where $x(t)$ and $y(t)$ denote the position of the end-effector in the horizontal plane, and vt represents the desired forward movement along the x-axis at constant speed v. The constraint set Π_c includes all policies π such that the maximum measured contact force during a roll-out remains within safety limits:

$$\Pi_c = \{\pi \in \Pi : \max_{t \in [0,T]} \|[f_x(t), f_y(t)]\|_2 \leq f_{\max}\} \tag{19}$$

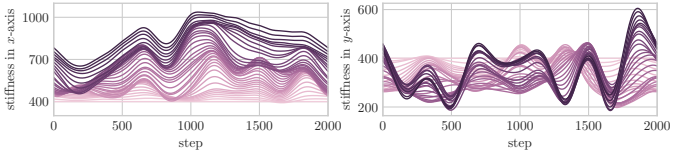

Fig. 9: Learned variable stiffness after each policy update iteration. Learned policy adaptively reduces stiffness in regions with high contact force, improving compliance to meet the force constraint while compromising on tracking accuracy.

where $f_x(t)$ and $f_y(t)$ denote contact force measured by a six-axis force/torque sensor in the $x-$axis and $y-$axis.

Policy parameterization: the policy π is parameterized by a set of weights $\theta = \{w_i\}_i^{24} = 1$, which modulate the stiffness of a Cartesian impedance controller. Specifically, the robot uses the following torque command τ

$$\tau = J^{\dagger}(q)\left[-K(\eta - \eta_d) - DJ(q)\dot{q}\right] \tag{20}$$

where $J(q) \in \mathbb{R}^{6\times 7}$ is the Jacobian matrix related to the task frame, and $\eta - \eta_d \in \mathbb{R}^6$ represents the translational and rotational tracking error in the task space. K and D are the virtual Cartesian stiffness and damping matrices, respectively. To reduce the dimensionality of the policy space, we assume K and D are diagonal and only modulate the stiffness in $x-$axis and $y-$axis. The remaining translational stiffness (z-axis) and all rotational stiffness values are untouched. The time-varying stiffness values $k_x(t)$ and $k_y(t)$, corresponding to the first two diagonal entries of K, are generated using weighted combinations of movement primitives

$$k_x(t), k_y(t) = 400 + \frac{\sum_{i=1}^{N} \Phi_i(t) w_i}{\sum_{i=1}^{N} \Phi_i(t)} \tag{21}$$

where $\Phi_i(t) = \exp(\cos(\omega t - c_i) - 1)$ are von Mises basis functions centered at fixed locations c_i evenly spaced over the roll-out duration (0-20 seconds), and $\omega = 2\pi/20$. Each axis uses 12 basis functions, resulting in a total of 24 weights $\theta = \{w_i\}_i^{24} = 1$, which are optimized by the GIBO-Lag.

We evaluate GIBO-Lag on this task with a budget of 150 roll-outs. Each roll-out lasts for T =20 s, and all measures are updated at 100 Hz, resulting in 2000 time steps per roll-out. The optimizer is initialized with zero weights so that the stiffness on both $x-$axis and $y-$axis are fixed to 400. To further validate the effectiveness of the active multiplier control method in a more realistic setting, we compare it again with the standard primal-dual subgradient algorithm. As in the ablation study from the synthetic experiment, we use I^+ and I^- to denote large and small controller gains, respectively, for the multiplier update. Fig. 7 shows the training curves of GIBO-Lag under different multiplier update strategy. The left plot shows the best performance achieved so far, i.e., the tracking error corresponding to the lowest constraint violation. GIBO-Lag with the active multiplier control method outperforms the baseline method, achieving both faster convergence and lower tracking error. The right plot illustrates the amplitude of constraint oscillations during training. The active multiplier control method reduces the oscillation amplitude to 42.4% and

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

30.2% of that observed with I^+ and I^-, respectively. Note that we adopt the improvement confidence criterion proposed in [7] to early stop the query loop, allowing more policy update iterations to be included in the plot. Fig. 8 and Fig. 9 show the evolution of tracking error, contact force, and learned stiffness over policy updates. Intuitively, when the flexible rod enters more curved regions, both the tracking error and contact force tend to increase, while the stiffness decreases to allow greater flexibility. This trend is evidenced by two noticeable drops in the learned stiffness around steps 500 and 1700, which become more pronounced with more policy update iterations.

VI. Conclusion

In this letter, we propose a novel data-efficient method, GIBO-Lag, for constrained robot learning by combining the strengths of local policy search and Gaussian Process modeling. GIBO-Lag extends the GIBO framework to constrained settings by introducing a probabilistic Lagrangian formulation, where policy updates are guided by the posterior mean of the Lagrangian. Further, the Lagrange multiplier is actively controlled by the probability of whether the constraints would be satisfied. This enables the algorithm to effectively trade off the reward and cost, and therefore reduces the oscillations and overshoot behaviors during training. We demonstrate the effectiveness of our approach on both simulated and real-world robot learning tasks.

A limitation of our approach is the assumption of additive Gaussian noise for analytical tractability, which may not fully capture complex dynamics. However, in our robot learning experiments, the main source of noise in the reward and cost function comes from sensors, rather than from the system dynamics. Also, episodic returns show low variance across repeated evaluations, supporting the use of a simplified additive noise model. Further, there is an implicit assumption that the primal maximizer for the optimal multiplier is feasible. This assumption may not always hold in practice, and GIBO-Lag could get stuck in an infeasible local optimum. Thus, we recommend warm-starting optimizer with a policy learned from RL or human demonstration. Future work may further address this limitation by incorporating trust-region methods.

References

[1] E. V. Bonilla, K. Chai, and C. Williams, "Multi-task gaussian process prediction," *Advances in Neural Information Processing Systems*, vol. 20, 2007.

[2] M. A. Gelbart, J. Snoek, and R. P. Adams, "Bayesian optimization with unknown constraints," in *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, 2014, pp. 250–259.

[3] K. Chatzilygeroudis, V. Vassiliades, F. Stulp, S. Calinon, and J.-B. Mouret, "A survey on policy search algorithms for learning robot controllers in a handful of trials," *IEEE Transactions on Robotics*, vol. 36, no. 2, pp. 328–347, 2019.

[4] S. Müller, A. von Rohr, and S. Trimpe, "Local policy search with Bayesian optimization," in *Advances in Neural Information Processing Systems*, 2021, pp. 20 708–20 720.

[5] A. Stooke, J. Achiam, and P. Abbeel, "Responsive safety in reinforcement learning by pid lagrangian methods," in *International Conference on Machine Learning*. PMLR, 2020, pp. 9133–9143.

[6] C. Tessler, D. J. Mankowitz, and S. Mannor, "Reward constrained policy optimization," in *International Conference on Learning Representations*, 2019.

[7] S. He, A. von Rohr, D. Baumann, J. Xiang, and S. Trimpe, "Simulation-aided policy tuning for black-box robot learning," *IEEE Transactions on Robotics*, vol. 41, pp. 2533–2548, 2025.

[8] L. Brunke, M. Greeff, A. W. Hall, Z. Yuan, S. Zhou, J. Panerati, and A. P. Schoellig, "Safe learning in robotics: From learning-based control to safe reinforcement learning," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, pp. 411–444, 2022.

[9] S. Ament, S. Daulton, D. Eriksson, M. Balandat, and E. Bakshy, "Unexpected improvements to expected improvement for bayesian optimization," *Advances in Neural Information Processing Systems*, vol. 36, pp. 20 577–20 612, 2023.

[10] Y. Sui, A. Gotovos, J. Burdick, and A. Krause, "Safe exploration for optimization with gaussian processes," in *International Conference on Machine Learning*. PMLR, 2015, pp. 997–1005.

[11] A. Wachi, Y. Sui, Y. Yue, and M. Ono, "Safe exploration and optimization of constrained mdps using gaussian processes," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[12] F. Berkenkamp, A. Krause, and A. P. Schoellig, "Bayesian optimization with safety constraints: safe and automatic parameter tuning in robotics," *Machine Learning*, vol. 112, no. 10, pp. 3713–3747, 2023.

[13] D. Baumann, A. Marco, M. Turchetta, and S. Trimpe, "Gosafe: Globally optimal safe robot learning," in *IEEE International Conference on Robotics and Automation*. IEEE, 2021, pp. 4452–4458.

[14] R. R. Duivenvoorden, F. Berkenkamp, N. Carion, A. Krause, and A. P. Schoellig, "Constrained bayesian optimization with particle swarms for safe adaptive controller tuning," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 11 800–11 807, 2017.

[15] D. Eriksson, M. Pearce, J. Gardner, R. D. Turner, and M. Poloczek, "Scalable global optimization via local bayesian optimization," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[16] D. Eriksson and M. Poloczek, "Scalable constrained bayesian optimization," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2021, pp. 730–738.

[17] K. Polymenakos, A. Abate, and S. Roberts, "Safe policy search using gaussian process models," in *Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems*, 2019, pp. 1565–1573.

[18] A. I. Cowen-Rivers, D. Palenicek, V. Moens, M. A. Abdullah, A. Sootla, J. Wang, and H. Bou-Ammar, "Samba: Safe model-based & active reinforcement learning," *Machine Learning*, vol. 111, no. 1, pp. 173–203, 2022.

[19] M. Deisenroth and C. E. Rasmussen, "Pilco: A model-based and data-efficient approach to policy search," in *International Conference on Machine Learning*, 2011, pp. 465–472.

[20] A. Padalkar, F. Stulp, G. Neumann, and J. Silvério, "Towards safe and efficient learning in the wild: guiding rl with constrained uncertainty-aware movement primitives," *IEEE Robotics and Automation Letters*, vol. 10, pp. 6880–6887, 2025.

[21] T.-Y. Yang, J. Rosca, K. Narasimhan, and P. J. Ramadge, "Projection-based constrained policy optimization," in *International Conference on Learning Representations*, 2019.

[22] J. Choi, F. Castañeda, C. J. Tomlin, and K. Sreenath, "Reinforcement learning for safety-critical control under model uncertainty, using control lyapunov functions and control barrier functions," in *Robotics: Science and Systems (RSS)*, 2020.

[23] Y. Chow, O. Nachum, E. Duenez-Guzman, and M. Ghavamzadeh, "A lyapunov-based approach to safe reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[24] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[25] P. Hennig and C. J. Schuler, "Entropy search for information-efficient global optimization," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 1809–1837, 2012.

[26] L. P. Fröhlich, M. N. Zeilinger, and E. D. Klenske, "Cautious bayesian optimization for efficient and scalable policy search," in *Learning for Dynamics and Control*. PMLR, 2021, pp. 227–240.

[27] J. Ji, B. Zhang, J. Zhou, X. Pan, W. Huang, R. Sun, Y. Geng, Y. Zhong, J. Dai, and Y. Yang, "Safety gymnasium: A unified safe reinforcement learning benchmark," *Advances in Neural Information Processing Systems*, vol. 36, pp. 18 964–18 993, 2023.

[28] K. Kronander and A. Billard, "Stability considerations for variable impedance control," *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1298–1305, 2016.

[29] Z. Jin, D. Qin, A. Liu, W.-a. Zhang, and L. Yu, "Model predictive variable impedance control of manipulators for adaptive precision-compliance tradeoff," *IEEE/ASME Transactions on Mechatronics*, vol. 28, no. 2, pp. 1174–1186, 2022.