

Bayesian Disturbance Injection: Robust Imitation Learning of Flexible Policies for Robot Manipulation

Hanbit Oh^a, Hikaru Sasaki^a, Brendan Michael^a, Takamitsu Matsubara^a

^a*Division of Information Science, Graduate School of Science and Technology,
NAIST, 8916-5, Takayama-cho, Ikoma-city, 630-0192, Nara, Japan*

Abstract

Humans demonstrate a variety of interesting behavioral characteristics when performing tasks, such as selecting between seemingly equivalent optimal actions, performing recovery actions when deviating from the optimal trajectory, or moderating actions in response to sensed risks. However, imitation learning, which attempts to teach robots to perform these same tasks from observations of human demonstrations, often fails to capture such behavior. Specifically, commonly used learning algorithms embody inherent contradictions between the learning assumptions (*e.g.*, single optimal action) and actual human behavior (*e.g.*, multiple optimal actions), thereby limiting robot generalizability, applicability, and demonstration feasibility. To address this, this paper proposes designing imitation learning algorithms with a focus on utilizing human behavioral characteristics, thereby embodying principles for capturing and exploiting actual demonstrator behavioral characteristics. This paper presents the first imitation learning framework, Bayesian Disturbance Injection (BDI), that typifies human behavioral characteristics by incorporating model flexibility, robustification, and risk sensitivity. Bayesian

inference is used to learn flexible non-parametric multi-action policies, while simultaneously robustifying policies by injecting risk-sensitive disturbances to induce human recovery action and ensuring demonstration feasibility. Our method is evaluated through risk-sensitive simulations and real-robot experiments (*e.g.*, table-sweep task, shaft-reach task and shaft-insertion task) using the UR5e 6-DOF robotic arm, to demonstrate the improved characterisation of behavior. Results show significant improvement in task performance, through improved flexibility, robustness as well as demonstration feasibility.

Keywords: Imitation learning, Disturbance injection, Human behavior characteristics, Robotic manipulation

PACS: 0000, 1111

2000 MSC: 0000, 1111

1. Introduction

Creating robot controllers via machine learning has found widespread usage in both research (Kuindersma et al., 2016; Levine et al., 2018) and commercial (Levine and Abbeel, 2014; Zhang et al., 2018; Wang et al., 2021; Dadhich et al., 2016; Bojarski et al., 2016) applications. Controller learning often utilizes large-scale exploration (Levine et al., 2018), reward mechanisms (*e.g.*, an optimal control or reinforcement learning), and with highly accurate dynamics models (Kuindersma et al., 2016) to learn autonomous control. However, in scenarios with exploration or dynamics model sparsity, imitation learning (Billard et al., 2008; Argall et al., 2009; Osa et al., 2018) is an intuitive method for learning skills via observations of an expert demonstrator, avoiding complex explicit programming, reward design, or large-scale

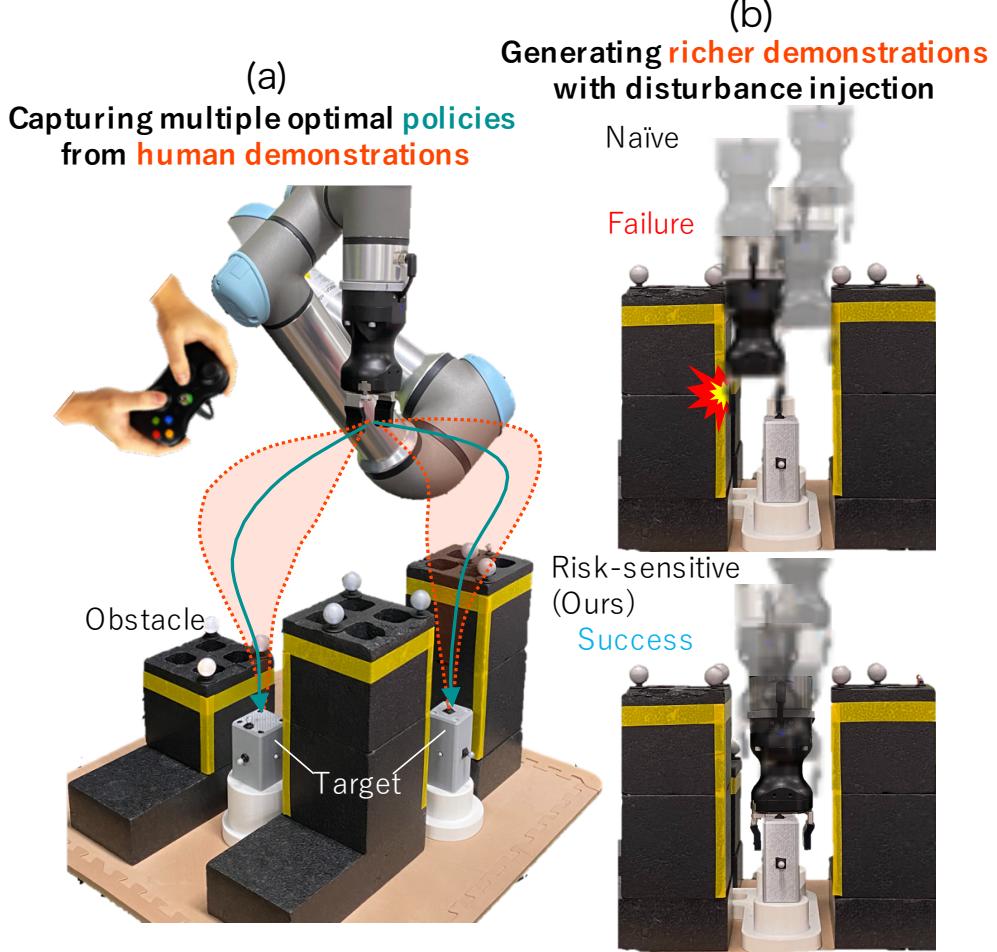


Figure 1: Illustration of the critical functions of the proposed method. (a): Multiple optimal policies are captured from complex human demonstrations, which may involve multiple optimal actions. (b): Generating richer (more exploratory) demonstrations by injecting disturbances into expert’s actions. Risk-sensitive disturbance models, which regulates its level response to risks of states, is employed to ensure demonstration feasibility.

exploration.

Although many conventional imitation learning methods have been proposed, they are often inherently flawed when learning realistic robot manipulation tasks from humans. Specifically, methods often have fundamental algorithmic contradictions between the assumed characteristics of demonstration data, and the actual characteristics embodied by human demonstrators. For example, methods often algorithmically presuppose unique optimal actions for any given state (Pomerleau, 1989; Levine and Abbeel, 2014; Giusti et al., 2016), or that sufficient exploratory actions can be performed in task space (Bojarski et al., 2016; Dyrstad et al., 2018). However, in many practical scenarios, human demonstrators may act contrary to these assumptions, exhibiting behavior such as multiple optimal actions or performing actions in idiosyncratic patterns that lack exploratory movements.

To emphasise the significance of the problem, consider the task of learning robotic grasping (Cutkosky and Howe, 1990; Napier, 1956) with multiple targets or obstacles, whereby demonstrators provide demonstrations controlling a robot to grasp an object. Standard imitation learning algorithms presuppose traits such as unique optimal grasp configurations, and sufficient diversity of demonstrations of this unique configuration to ensure generalizability. However, human behavioral characteristics introduce uncertainty or probabilistic behavior, which challenges these assumptions. For example, demonstrators may arbitrarily or idiosyncratically select between various equivalently optimal actions to determine which path to take, or may be biased to specific regions of the demonstration space. This fundamental contradiction limits the generalizability of the policy by introducing additional

modelling complexities, such as the covariate shift (Ross and Bagnell, 2010).

Therefore, robotic manipulation in real-world scenarios necessitates the design of algorithms that embody principles for capturing actual *demonstrator behavioral characteristics*. Specifically, this paper focuses on three key characteristics that are not typified by standard imitation learning algorithmic assumptions: (i) the ability to *flexibly* adapt to a wide range of spatial scenarios, (ii) the capability to *robustly* overcome deviation from optimal trajectories, and (iii) the ability to *risk-sensitively* respond to ensure feasibility. As an illustrative example of these characteristics, Figure 1 demonstrates a robot reaching and grasping shafts while avoiding obstacles; where there are multiple seemingly equivalent optimal actions (green arrows). As such, this necessitates *flexible policy models* capable of learning multiple optimal policies. Concurrently, robustification is applied using *disturbance injection approaches* (Laskey et al., 2017; Oh et al., 2021) to expand demonstration coverage (shaded region) and induce the learning of recovery behavior for retaining the optimal trajectory. However, as shown in Figure 1-(b), naïve disturbance injection may result in demonstration infeasibility (*e.g.*, environmental collisions or confusion in decision making) thereby necessitating *risk-sensitive disturbance models*, which regulate the disturbance level in response to state riskiness, thereby ensuring demonstration feasibility.

To the authors' knowledge, there is no unified framework for imitation learning that can simultaneously consider all of the above requirements. Our insight into this stems from the difficulty of formulating all three elements as a single framework. For example, in flexible policy learning, using non-parametric probabilistic policy models (*e.g.*, (Sasaki and Matsubara, 2019))

effectively captures multiple optimal actions from real human demonstrations where the number of optimal actions in each state cannot be specified a priori. However, the previously proposed disturbance injection method (Laskey et al., 2017) optimizes the disturbance level by minimizing the covariate shift, which corresponds to the maximum likelihood estimation based on the assumption of a deterministic policy model and a fixed disturbance level parameter. Thus, such flexible policy learning cannot be directly integrated into the previous framework. To address this difficulty, we propose to reformulate it as a non-parametric Bayesian inference problem, which employs the objective function of robustification as the likelihood and other non-parametric flexible policy and risk-sensitive disturbance models as the prior distribution. As such, this paper presents a novel Bayesian imitation learning framework that learns a probabilistic policy model capable of being both flexible to variations in demonstrations and robust to sources of error in policy application by injecting risk-sensitive disturbances, referred to as Bayesian Disturbance Injection (BDI).

Specifically, this paper establishes Multi-modal Heteroscedastic Gaussian Process BDI (MHGP-BDI), in which robust multi-modal probabilistic policy learning uses flexible regression models (Ross and Dy, 2013) as non-parametric mixture policies (Figure 2-(b)). To learn robustification, the demonstrator is induced to provide recovery behavior, via disturbances injected into their actions (Figure 2-(a)). To model risk-sensitive behavior, *state-dependent disturbances* are learnt, which are approximated during policy learning via a heteroscedastic variance regression model (Lazaro-Gredilla and Titsias, 2011) (Figure 2-(b)). Given this conditional relationship be-

tween policy learning and disturbance optimization, this approach unifies learning within a single probabilistic framework. As such, inference of the policy and injection disturbance is performed simultaneously by variational Bayesian inference, thereby presenting a more authentic characterisation of the experts' behavior for imitation learning.

To evaluate the effectiveness of the proposed framework, experiments in learning probabilistic behavior from risk-sensitive simulation (*e.g.*, wall-avoidance task) and real robot experiments (*e.g.*, table-sweep task, shaft-reach task, shaft-insertion task) using the UR5e 6-DOF robotic arm are performed. Results show improved flexibility and robustness with increased learning performance and demonstration feasibility relative to comparison methods, giving a novel viewpoint of human behavioral characteristic learning.

Advancing from our preliminary publication (Oh et al., 2021), the key contributions of this paper are as follows:

1. provides a novel perspective on imitation learning that captures integrated human behavior characteristics;
2. provides a formulation that incorporates imitation learning models of flexibility, robustification, and risk sensitivity via a non-parametric Bayesian approach;
3. provides a novel Bayesian imitation learning framework, Bayesian Disturbance Injection (BDI), to learn flexible non-parametric multi-action policies, while simultaneously robustifying policies by injecting risk-sensitive disturbances to induce human recovery action and ensuring

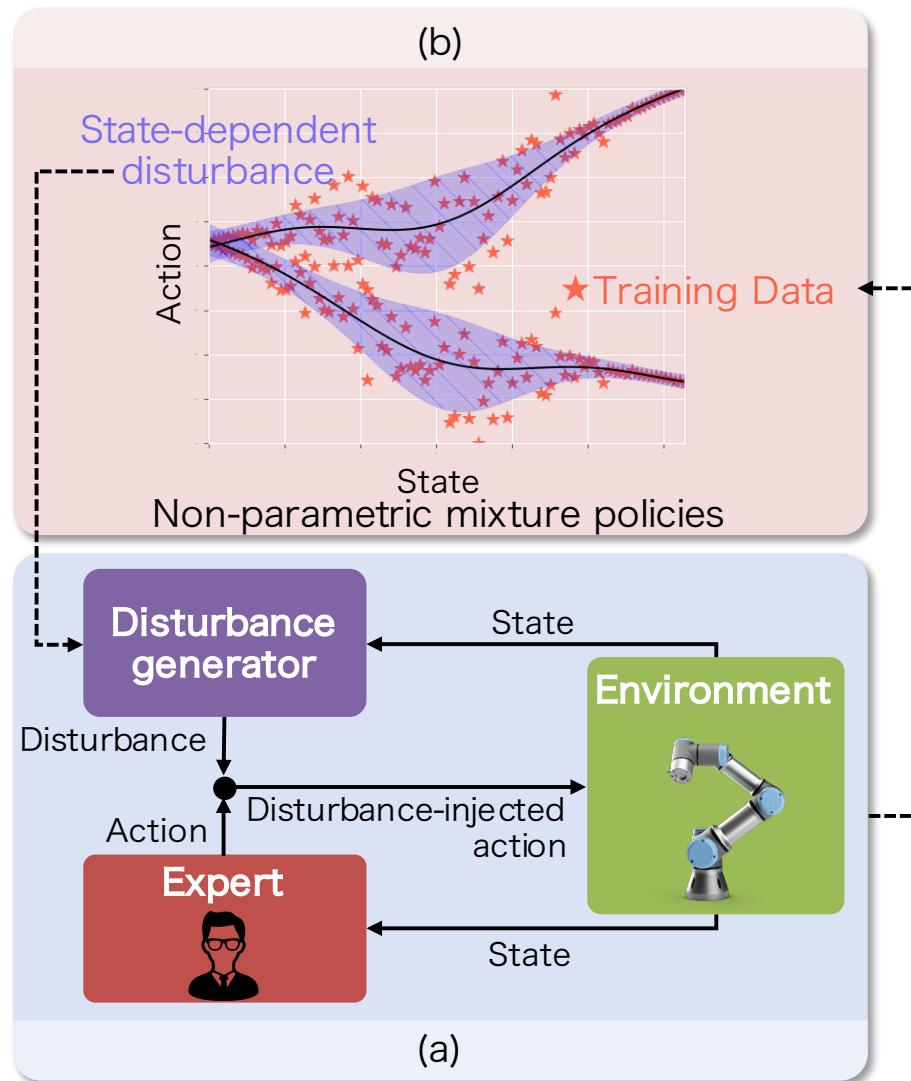


Figure 2: Overview of MHGP-BDI, learning robust multi-modal policy with state-dependent disturbance injection. (a): Collect and accumulate training datasets by injecting disturbance into the expert's demonstration actions. (b): Optimize the disturbance at which level can be regulated in a state-dependent manner. These processes (a) and (b) are repeated to obtain a robust multi-modal policy finally.

demonstration feasibility;

4. validates the effectiveness of the proposed approach by comparing it with state-of-the-art methods on simulations (*e.g.*, wall-avoidance task) and real robot experiments of general assembly tasks (*e.g.*, table-sweep task, shaft-reach task, shaft-insertion task);

The remainder of this paper is organized as follows. Section 2 summarizes previous research on imitation learning. Section 3 introduces preliminaries of imitation learning from human demonstrations. Section 4 presents our proposed methods. Section 5 presents the simulation setup and results. Section 6 presents the experimental results in real robot experiments on assembly tasks. Finally, discussion and conclusion are described in Sections 7 and 8, respectively.

2. Related Work

A key objective of imitation learning is to ensure that models can capture the variation and stochasticity inherent in human motion while ensuring application of the learned policy can restrain deviation from optimal trajectories and ensure humans can accomplish demonstrations. To address these, prior approaches explore modelling *flexibility*, and *robustification* methods retain generalizability by mitigating compounding errors in policy application. However, naïve robustification may influence *demonstration feasibility*, and as such, methods for addressing this trade-off are explored.

2.1. Flexibility

Learning generalized optimal action policies from human demonstrations, which often contain complex behaviors (*e.g.*, multiple optimal actions for a task), requires elaborate policy models with non-linearity and stochasticity.

Classical approaches to modelling uses dynamical frameworks for learning trajectories from demonstrations, *e.g.*, Dynamic Movement Primitives (DMPs), which can generalize the learned trajectories to new situations (*i.e.*, goal location or speed). However, this generalization depends on heuristics (*e.g.*, the appropriate number of basis functions regarding the complexity of trajectories), and is thus unsuitable for learning state-dependent feedback policies ([Schaal et al., 2005](#); [Ijspeert et al., 2013](#); [Khansari-Zadeh and Billard, 2011](#)).

To avoid imposing a priori structure, Gaussian Mixture Regression (GMR) is a non-parametric, intuitive means to learn trajectories or policies from demonstrators in the state-action-space. In this, Gaussian Mixture Modelling (GMM) ([Calinon, 2016](#)) is used as a basis function to capture non-linearities during learning, and has been utilized in imitation learning that deals with human demonstrations ([Kyrrarini et al., 2019](#)). However, GMR requires that basis functions be engineered by hand to deal with high-dimensional systems ([Huang et al., 2019](#)).

In data-driven manner, nonlinearities can also be captured flexibly using Variational Auto-Encoders (VAE), which is a generative model that can embed high-dimensional features in latent variables ([Kingma and Welling, 2013](#)). Furthermore, Conditional VAE (CVAE) can learn multi-modality by conditioning latent variables on a decoder ([Sohn et al., 2015](#)), and has

been applied to capture multiple optimal actions from human demonstrations (Rahmatizadeh et al., 2018; Ren et al., 2020). However, such CVAE-based methods typically require large amounts of data to capture multi-modality, and even though such multi-modality is obtained using latent variables, learned policies may be sub-optimal for the high precision task; since latent variables are randomly sampled from a standard Gaussian prior distribution (Hsiao et al., 2019).

As an alternative, Gaussian Process Regression (GPR) deals with implicit (high-dimensional) feature spaces with kernel functions. It thus can directly deal with high-dimensional observations without explicitly learning in this high-dimensional space (Rasmussen, 2003). In particular, Overlapping Mixtures of Gaussian Processes (OMGP) (Lázaro-Gredilla et al., 2012) learns a multi-modal distribution by overlapping multiple GPs, and has been employed as a policy model with multiple optimal actions on flexible task learning of robotic policies (Sasaki and Matsubara, 2019). To further reduce a priori tuning, Infinite Overlapping Mixtures of Gaussian Processes (IOMGP) (Ross and Dy, 2013) requires only an upper bound of the number of GPs to be estimated. As such, IOMGP is an intuitive means of learning flexible multi-modal policies from unlabeled human demonstration data and is employed in this paper.

2.2. Robustness and Demonstration Feasibility

While flexibility is key to capturing demonstrator motion, a major issue limiting application of learned policies is the problem of *covariate shift* (Ross and Bagnell, 2010). Specifically, environment variations (*e.g.*, manipulator starting position) induces differences between the policy distribution

as learned by the manipulator and the actual task distribution during application.

A more general approach to minimizing the covariate shift in imitation learning is Dataset Aggregation (DAgger) (Ross et al., 2011), whereby when the robot moves to a state not included in the training data, the expert augments the model by teaching the optimal recovery. However, this approach has limited applicability in practice due to the risk of exploring unknown states during policy application and the high overhead cost of human experts continuing to teach the robot the optimal actions.

An intuitive approach to robustifying learned policies against sources of error, without needing to a priori specify task-relevant learning parameters, is to exploit phenomenon similar to persistence excitation (Sastry and Bodson, 2011). In this, disturbances are injected into the expert’s demonstrated actions, and the recovery behavior of the expert is learned given this perturbation. In an imitation learning context, Disturbances for Augmenting Robot Trajectories (DART) (Laskey et al., 2017) exploits this phenomenon for learning a deterministic policy model with a single optimal action. Additionally, DART is well suited to creating a richer dataset, by concurrently determining the optimal disturbance level to be injected into the demonstrated actions during policy learning.

However, the applicability of algorithms proposed to implement DART (Laskey et al., 2017) is limited, since DART employs a naïve disturbance model which cannot regulate the level of disturbance regarding given states (*i.e.*, state-independent disturbance). For robotics tasks with local precision involving small clearance, such as a Figure 1, applying a uniform level of dis-

turbance regardless of the state may lead to the risk of unintended collisions. On the other hand, simply limiting a level of disturbance to a small level does not sufficiently reduce the covariate shift.

A natural approach to addressing this in an imitation learning context, is to explore how human demonstrators approach this problem. Demonstrators, when aware of environmental risks, decrease movement velocity to increase action accuracy (Nagengast et al., 2011), based on a *speed-accuracy trade-off* (Wickelgren, 1977). Inspired by such risk-sensitive behavior, this paper proposes a state-dependent disturbance model, which regulates the disturbance level to be small at risky states (*e.g.*, close to obstacles). As such, our disturbance injection robustifies policies, while maintaining demonstration feasibility. Specifically, a Heteroscedastic Gaussian Process (HGP) (Lazaro-Gredilla and Titsias, 2011), which can accurately infer probabilistic regression models with input-dependent variance, and is employed as a state-dependent disturbance model in this paper.

3. Preliminaries

3.1. Imitation Learning from Expert’s Demonstration

The objective of imitation learning is to learn a control policy by imitating the action from the expert’s demonstration data. A dynamics model is denoted as Markovian with a state $\mathbf{s}_t \in \mathbb{R}^Q$, an action $a_t \in \mathbb{R}$, an initial state probability $p(\mathbf{s}_0)$ and a state transition distribution $p(\mathbf{s}_{t+1} | \mathbf{s}_t, a_t)$. For simplicity but without loss of generality, the following derivation involves on one-dimensional action. In this, a policy $\pi(a_t | \mathbf{s}_t)$ decides an action from a state, while a trajectory $\tau = (\mathbf{s}_0, a_0, \mathbf{s}_1, a_1 \dots a_{T-1}, \mathbf{s}_T)$ is a sequence of

state-action pairs of T steps. The trajectory distribution is defined as:

$$p(\boldsymbol{\tau} \mid \pi) = p(\mathbf{s}_0) \prod_{t=0}^{T-1} \pi(a_t \mid \mathbf{s}_t) p(\mathbf{s}_{t+1} \mid \mathbf{s}_t, a_t). \quad (1)$$

A significant aspect of imitation learning is to reproduce the expert's behavior, thus the function to compute the expected similarity between two policies with regard to trajectories is defined as:

$$J(\pi, \pi' \mid \boldsymbol{\tau}) = - \sum_{t=0}^{T-1} \mathbb{E}_{\pi(a|\mathbf{s}_t), \pi'(a'|\mathbf{s}_t)} [||a - a'||_2^2]. \quad (2)$$

A learned policy π^R is obtained by solving the following optimization problem using a trajectory collected by an expert's policy π^* :

$$\pi^R = \arg \max_{\pi} \mathbb{E}_{p(\boldsymbol{\tau} \mid \pi^*)} [J(\pi, \pi^* \mid \boldsymbol{\tau})]. \quad (3)$$

As discussed in Section 2.2, such imitation learning may suffer from the problem of covariate shift, where the agent applying the learned policy drifts away from the demonstrated states due to compounding errors. This drift issue is delineated as the distributive difference between the trajectory during training data collection and learned policy application:

$$|\mathbb{E}_{p(\boldsymbol{\tau} \mid \pi^*)} [J(\pi^R, \pi^* \mid \boldsymbol{\tau})] - \mathbb{E}_{p(\boldsymbol{\tau} \mid \pi^R)} [J(\pi^R, \pi^* \mid \boldsymbol{\tau})]|. \quad (4)$$

3.2. Robust Imitation Learning by Injecting Disturbance into Expert

To learn robust policies from compounding errors, DART has been previously proposed (Laskey et al., 2017). In this approach, disturbances are injected into the expert demonstrations to generate a richer training data set. The level of injecting disturbances is optimized, to reduce the covariate

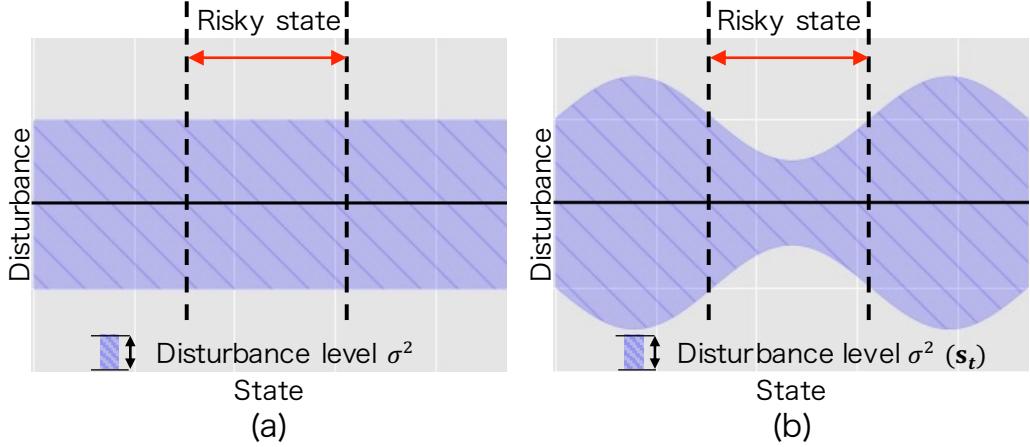


Figure 3: Illustration of the comparison between (a) state-independent and (b) state-dependent disturbance models. (a): a constant disturbance level regardless of the risk of the state, may be dangerous in risky states. (b): the disturbance level can be modified according to the state; *e.g.*, in risky states the disturbance level is reduced.

shift between the collected demonstration data and predicted trajectories. The disturbance distribution is optimized iteratively in the data collection process. Finally, the robust policy is learned using the collected data.

This injection disturbance is assumed that sampled from a Gaussian distribution as $\epsilon_t \sim \mathcal{N}(0, \sigma_k^2)$, where k is the number of optimization iterations. The injection disturbance ϵ_t is added into the expert's action a_t^* . The distribution of trajectories from a disturbance injected expert, is denoted as $p(\tau | \pi^*, \sigma_k^2)$ and the distribution of trajectories from a learned policy is $p(\tau | \pi_k^R)$. To reduce the covariate shift, DART proposes to use the upper

bound of covariate shift by Pinsker's inequality as:

$$\begin{aligned} & \left| \mathbb{E}_{p(\boldsymbol{\tau} | \pi^*, \sigma_k^2)} [J(\pi^R, \pi^* | \boldsymbol{\tau})] - \mathbb{E}_{p(\boldsymbol{\tau} | \pi_k^R)} [J(\pi^R, \pi^* | \boldsymbol{\tau})] \right| \\ & \leq T \sqrt{\frac{1}{2} \text{KL}(p(\boldsymbol{\tau} | \pi_k^R) || p(\boldsymbol{\tau} | \pi^*, \sigma_k^2))}, \end{aligned} \quad (5)$$

where, $\text{KL}(\cdot || \cdot)$ is Kullback-Leibler divergence. However, the upper bound (5) is analytically intractable to compute since the trajectory distribution of learned policy $p(\boldsymbol{\tau} | \pi_k^R)$ is unknown. Therefore, DART solves the upper bound by replacing the trajectory distribution of the learned policy with the trajectory distribution of the disturbance-injected expert. As such, data are collected over several iterations and a disturbance distribution is optimized at each iteration as:

$$\begin{aligned} \sigma_{k+1}^2 &= \arg \max_{\sigma^2} \mathbb{E}_{p(\boldsymbol{\tau} | \pi^*, \sigma_k^2)} \\ & \left[\sum_{t=0}^{T-1} \mathbb{E}_{\pi_k^R(a'_t | s_t)} [\log \mathcal{N}(a'_t | a_t, \sigma^2)] \right], \end{aligned} \quad (6)$$

where, a learned policy at k th iteration π_k^R is obtained in the similar form as (3) by following:

$$\pi_k^R = \arg \max_{\pi} \sum_{i=1}^{k-1} \mathbb{E}_{p(\boldsymbol{\tau} | \pi^*, \sigma_i^2)} [J(\pi, \pi^* | \boldsymbol{\tau})]. \quad (7)$$

Although DART can reduce the covariate shift by injecting disturbances into expert demonstrations, its applicability still suffers from the following issues. The applied policy model is deterministic, which means that it cannot recognize complex human behavior (*e.g.*, multiple optimal actions) from the training data. Additionally, as shown in Figure 3-(a), the disturbance is injected uniformly regardless of the current state of the robot, which may

induce dangerous situations (*e.g.*, physical contacts as in Figure 1-right). Furthermore, the disturbance level optimization (6) corresponds to the maximum likelihood estimation based on the assumption of a deterministic policy model and a fixed disturbance level parameter; thus, non-parametric policy learning (*e.g.*, (Sasaki and Matsubara, 2019)) in which effectively captures multiple optimal actions without requiring the specified number of optimal actions in each state, cannot be directly integrated into the DART framework. Therefore, a scheme to resolve these issues simultaneously via non-parametric Bayesian inference is derived in the next section.

4. Proposed Method

In this section, a novel Bayesian imitation learning framework is proposed (Figure 2) to learn a probabilistic policy via expert demonstrations with disturbance injection. Specifically, flexibility, robustness, and risk-sensitivity are incorporated as a single formulation in a Bayesian manner; thus, it is referred to as Bayesian Disturbance Injection (BDI). The general form of BDI is derived in Section 4.1. As an overview, a non-parametric mixture model is utilized as a policy prior for capturing multiple optimal actions from human demonstration. A heteroscedastic model is employed as a disturbance prior for regulating disturbance level regarding states as shown in Figure 3-(b). The disturbance optimization term (6) is employed as a likelihood for minimising the covariate shift. This combination derives an imitation learning method, which learns a multi-modal policy and an injection disturbance distribution by Bayesian inference. Given this model, the predictive distribution is induced in a Bayesian form. A specific implementation of BDI, which em-

ploy IOMGP ([Ross and Dy, 2013](#)) as a policy prior and HGP ([Lazaro-Gredilla and Titsias, 2011](#)) as a disturbance prior, is derived from Section [4.2](#).

4.1. Bayesian Disturbance Injection (BDI)

Bayesian treatment is employed to learn probabilistic policies and disturbances in a single incorporated framework. As such, each goal function of the learning a policy [\(7\)](#) and disturbances [\(6\)](#) are formulated as a single likelihood. In addition, prior distributions of policy and disturbances are defined, and their respective posterior distributions are obtained via Bayesian inference.

To capture complex human behaviors as involving uncertainties, the probabilistic policy model which output action a_t from the state \mathbf{s}_t with Gaussian disturbance $\epsilon_t \sim \mathcal{N}(0, \sigma^2)$ is defined as: $a_t = f(\mathbf{s}_t) + \epsilon_t$, where $f(\cdot)$ is an output of a latent non-linear function. By applying this policy model to the objective function of policy learning [\(7\)](#), a log-likelihood function that integrates policy and disturbances is derived as follows:

$$J(\pi^*, \mathbf{f}, \sigma^2 | \boldsymbol{\tau}) = \sum_{t=0}^{T-1} \log p(a_t^* | f(\mathbf{s}_t), \sigma^2), \quad (8)$$

where, $\mathbf{f} = [f(\mathbf{s}_t)]_{t=0}^{T-1}$ is a set of a latent function outputs. Note that this log likelihood function [\(8\)](#) is equal to the objective function of disturbance learning [\(6\)](#) if the mean and variables are swapped in a Gaussian distribution (the value of the distribution remains the same).

In addition, to infer a policy and disturbances in a non-parametric way from iteratively accumulated state-action pairs ($\{\mathbf{a}^*, \mathbf{S}\} = \{a_n^*, \mathbf{s}_n\}_{n=1}^N$, where $N = \sum_{j=1}^k N_j$, N_j is a size of the dataset that collected at j -th iteration),

Algorithm 1 BDI

Input: σ_1^2 **Output:** $p(\mathbf{f}, \sigma^2 | \mathbf{a}^*, \mathbf{S})$

- 1: **for** $k = 1$ to K **do**
 - 2: Get dataset through the disturbance injected expert:
 $\{a_t^*, \mathbf{s}_t\}_{t=1}^{N_k} \sim p(\boldsymbol{\tau} | \pi^*, \sigma_k^2)$
 - 3: Aggregate datasets :
 $\mathbf{a}^* \leftarrow \mathbf{a}^* \cup \{a_t^*\}_{t=1}^{N_k}, \mathbf{S} \leftarrow \mathbf{S} \cup \{\mathbf{s}_t\}_{t=1}^{N_k}$
 - 4: Update $p(\mathbf{f}, \sigma^2 | \mathbf{a}^*, \mathbf{S})$
 - 5: **end for**
-

the prior distribution of a policy and disturbances are defined as $p(\mathbf{f} | \mathbf{S})$ and $p(\sigma^2)$, respectively. Accordingly, posterior distributions of a policy and disturbances are simultaneously inferred by Bayesian inference as:

$$p(\mathbf{f}, \sigma^2 | \mathbf{a}^*, \mathbf{S}) = \frac{p(\mathbf{a}^* | \mathbf{f}, \sigma^2)p(\mathbf{f} | \mathbf{S})p(\sigma^2)}{\pi^*(\mathbf{a}^* | \mathbf{S})}. \quad (9)$$

A summary of the BDI is shown in Algorithm 1.

4.2. Multi-modal Heteroscedastic Gaussian Process BDI (MHGP-BDI)

4.2.1. Formulation:

To learn a multi-modal policy, the policy prior is considered as the product of infinite GPs, inspired by IOMGP. In addition, to learn state-dependent disturbances that can regulate its level respond to states, the prior of disturbances is considered as a state-dependent variance GP prior, inspired by HGP. Intuitively, Figure 4 shows a probabilistic policy model in which expert's actions \mathbf{a}^* are estimated by $\mathbf{f}^{(m)}, \mathbf{Z}, \mathbf{g}$. The latent function $\mathbf{f}^{(m)}$ is the output of m -th GP given state \mathbf{S} . To allocate the expert's n -th action a_n^* to

the m -th latent function $\mathbf{f}^{(m)}$, the indicator matrix $\mathbf{Z} \in \mathbb{R}^{N \times \infty}$ is defined. To estimate the optimal number of GPs, a random variable v_m quantifies the uncertainty assigned to $\mathbf{f}^{(m)}$. In addition, to learn an injection disturbance which can regulate its level in a state-dependent way, a state-dependent disturbance level $\sigma^2(\mathbf{s}_n) = e^{g(\mathbf{s}_n)}$ is introduced, where $g(\cdot)$ is an output of GP given a state \mathbf{s}_n .

Policy prior: the set of latent functions is denoted as $\{\mathbf{f}^{(m)}\} = \{\mathbf{f}^{(m)}\}_{m=1}^\infty$ and a GP prior is given by :

$$p(\{\mathbf{f}^{(m)}\} \mid \mathbf{S}, \{\boldsymbol{\omega}^{(m)}\}) = \prod_{m=1}^{\infty} \mathcal{N}(\mathbf{f}^{(m)} \mid \mathbf{0}, \mathbf{K}_{\mathbf{f}}^{(m)}; \omega^{(m)}), \quad (10)$$

where $\mathbf{K}_{\mathbf{f}}^{(m)} = k_{\mathbf{f}}^{(m)}(\mathbf{S}, \mathbf{S})$ is the m -th kernel Gram matrix with the kernel function $k_{\mathbf{f}}^{(m)}(\cdot, \cdot)$ and a kernel hyperparameter $\omega_{\mathbf{f}}^{(m)}$. Let $\{\boldsymbol{\omega}_{\mathbf{f}}^{(m)}\} = \{\omega_{\mathbf{f}}^{(m)}\}_{m=1}^\infty$ be the set of hyperparameters of infinite number of kernel functions.

To infer the optimal number of GPs from the above GP mixtures (10), the Stick Breaking Process (SBP) ([Sethuraman, 1994](#)) is used as a prior of \mathbf{Z} , which can be interpreted as an infinite mixture model as follows:

$$p(\mathbf{Z} \mid \mathbf{v}) = \prod_{n=1}^N \prod_{m=1}^{\infty} \left(v_m \prod_{j=1}^{m-1} (1 - v_j) \right)^{\mathbf{Z}_{nm}}, \quad (11)$$

$$p(\mathbf{v} \mid \beta) = \prod_{m=1}^{\infty} \text{Beta}(v_m \mid 1, \beta). \quad (12)$$

Note that the implementation of variational Bayesian learning approximates infinite-dimensional inference with a predefined upper bound of M . In this process, v_m is a random variable indicating the probability that the data corresponds to the m -th GP. Thus, it is possible to estimate the optimal number of GPs with a high probability of allocation starting from an infinite number

of GPs. β is a hyperparameter of SBP denoting the level of concentration of the data in the cluster.

Disturbance prior: the above policy model differs from the IOMGP model for regression (Ross and Dy, 2013); our model employs a state-dependent disturbance level $e^{g(s_n)}$ where the values are determined in response to the state. To learn a state-dependent disturbance, the disturbance prior is considered as a heteroscedastic Gaussian disturbance, inspired by HGP (Lazaro-Gredilla and Titsias, 2011). Accordingly, a GP prior is placed on a latent function $\mathbf{g} = \{g(\mathbf{s}_n)\}_{n=1}^N$, which represent a level of disturbance as:

$$p(\mathbf{g} | \mathbf{S}; \omega_{\mathbf{g}}) = \mathcal{N}(\mathbf{g} | \mu_0 \mathbf{1}_N, \mathbf{K}_{\mathbf{g}}; \omega_{\mathbf{g}}), \quad (13)$$

where, μ_0 is mean of disturbance distribution, $\mathbf{1}_{N_i}$ is a vector whose size is N_i and all components are one, and $\mathbf{K}_{\mathbf{g}}$ is kernel Gram matrix with a kernel hyperparameter $\omega_{\mathbf{g}}$.

Likelihood: the likelihood function, as in (8), for the variables $(\{\mathbf{f}^{(m)}\}, \mathbf{g}, \mathbf{Z})$ in the policy and disturbance models, is derived as follows:

$$\begin{aligned} p(\mathbf{a}^* | \mathbf{g}, \{\mathbf{f}^{(m)}\}, \mathbf{Z}) \\ = \prod_{n=1}^N \prod_{m=1}^{\infty} \mathcal{N}(a_n^* | \mathbf{f}_n^{(m)}, e^{\mathbf{g}_n})^{\mathbf{Z}_{nm}}. \end{aligned} \quad (14)$$

This formulation is described in a graphical model that defines the relationship between the variables as shown in Figure 4, and the joint distribution of the model as :

$$\begin{aligned} p(\mathbf{a}^*, \mathbf{g}, \{\mathbf{f}^{(m)}\}, \mathbf{Z}, \mathbf{v} | \mathbf{S}; \Omega) \\ = p(\mathbf{a}^* | \mathbf{g}, \{\mathbf{f}^{(m)}\}, \mathbf{Z}) p(\mathbf{g} | \mathbf{S}; \omega_{\mathbf{g}}) \cdot \\ p(\{\mathbf{f}^{(m)}\} | \mathbf{S}; \{\omega_{\mathbf{f}}^{(m)}\}) p(\mathbf{Z} | \mathbf{v}) p(\mathbf{v} | \beta), \end{aligned} \quad (15)$$

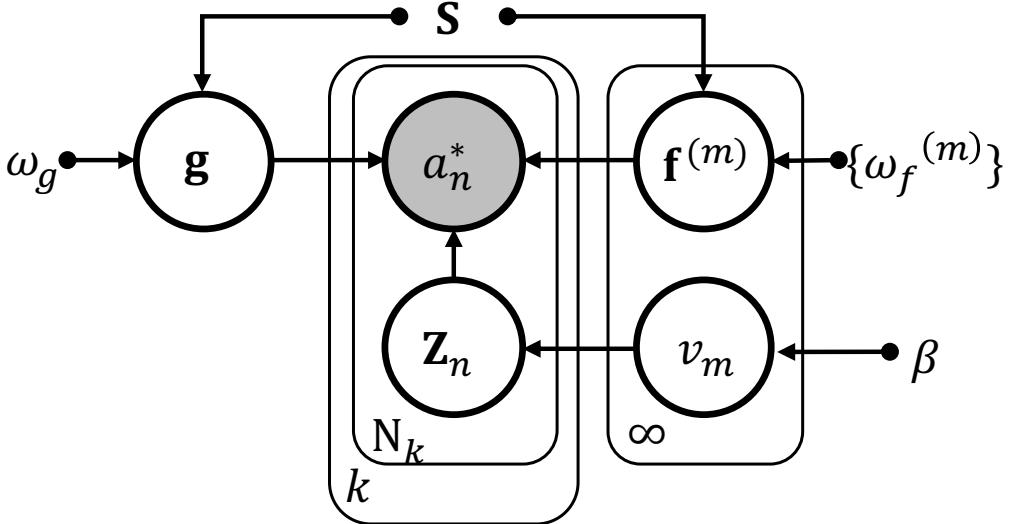


Figure 4: Graphical model of policy with state-dependent injection disturbance.

where $\Omega = (\{\omega_f^{(m)}\}, \omega_g, \mu_0, \beta)$ represents a set of hyperparameters.

4.2.2. Optimization of Policies and Injection Disturbance via Variational Bayesian Inference:

Bayesian inference is a framework that estimates the posterior distributions of the policies and their predictive distributions for new input data rather than point estimates of the policy parameters. To obtain the posterior and the predictive distributions, the marginal likelihood is calculated as :

$$\begin{aligned} p(\mathbf{a}^* | \mathbf{S}; \Omega) \\ = \int p(\mathbf{a}^*, \mathbf{g}, \{\mathbf{f}^{(m)}\}, \mathbf{Z}, \mathbf{v} | \mathbf{S}; \Omega) d\mathbf{g} d\{\mathbf{f}^{(m)}\} d\mathbf{Z} d\mathbf{v}. \end{aligned} \quad (16)$$

However, it is intractable to calculate the log marginal likelihood of (16)

analytically. Therefore, the variational lower bound is derived as the objective function of variational learning. The true posterior distribution is approximated by the variational posterior distribution, which maximizes the variational lower bound. Such the variational lower bound $\mathcal{L}(q, \Omega)$ is derived by applying the Jensen inequality to the log marginal likelihood, as:

$$\begin{aligned} & \log p(\mathbf{a}^* \mid \mathbf{S}; \Omega) \\ & \geq \int q \log \frac{p(\mathbf{a}^*, \mathbf{g}, \{\mathbf{f}^{(m)}\}, \mathbf{Z}, \mathbf{v} \mid \mathbf{S}; \Omega)}{q} d\mathbf{g} d\{\mathbf{f}^{(m)}\} d\mathbf{Z} d\mathbf{v} \\ & = \mathcal{L}(q, \Omega), \end{aligned} \quad (17)$$

where, $q = q(\mathbf{g}, \{\mathbf{f}^{(m)}\}, \mathbf{Z}, \mathbf{v})$ represents a set of variational posteriors.

As a common fashion of variational inference, the variational posterior distribution is assumed to be factorized among all latent variables (known as the *mean-field approximation* ([Parisi, 1988](#))) as follows:

$$q(\mathbf{g}, \{\mathbf{f}^{(m)}\}, \mathbf{Z}, \mathbf{v}) = q(\mathbf{g}) q(\{\mathbf{f}^{(m)}\}) q(\mathbf{Z}) \prod_{m=1}^{\infty} q(v_m). \quad (18)$$

In addition, to compute the variational lower bound in closed form, the posterior of \mathbf{g} is restricted to a multivariate Gaussian distribution. Furthermore, to reduce the computational complexity and facilitate the optimization problem, similar to Gaussian approximation ([Opper and Archambeau, 2009](#)), a positive variational parameter $\boldsymbol{\Lambda} = \text{diag}\{\lambda_n\}_{n=1}^N$ is employed as :

$$q(\mathbf{g}) = \mathcal{N}(\mathbf{g} \mid \boldsymbol{\mu}_{\mathbf{g}}, \boldsymbol{\Sigma}_{\mathbf{g}}), \quad (19)$$

$$\boldsymbol{\mu}_{\mathbf{g}} = \mathbf{K}_{\mathbf{g}} \left(\boldsymbol{\Lambda} - \frac{1}{2} \mathbf{I} \right) \mathbf{1}_N + \mu_0 \mathbf{1}_N, \quad (20)$$

$$\boldsymbol{\Sigma}_{\mathbf{g}}^{-1} = \mathbf{K}_{\mathbf{g}}^{-1} + \boldsymbol{\Lambda}, \quad (21)$$

Table 1: Computational complexity of each optimization in MHGP-BDI: N and M are number of training data sets and upper bound of mixtures, respectively.

	$q(\mathbf{g})$	$q(\mathbf{v})$	$q(\{\mathbf{f}^{(m)}\}), q(\mathbf{Z}), \mathcal{L}$
MHGP-BDI	$\mathcal{O}(N^3)$	$\mathcal{O}(M^2N)$	$\mathcal{O}(MN^3)$

where, \mathbf{I} is an identity matrix.

Therefore, the optimization formulation is derived using the *Expectation-Maximization* (EM)-like algorithm. The variational posterior distributions q are optimized with fixed hyperparameters Ω' in E-step, and the hyperparameters Ω' are optimized with fixed variational posterior distributions q in M-step with:

$$\hat{q}, \hat{\Omega}' = \arg \max_{q, \Omega'} \mathcal{L}(q, \Omega'), \quad (22)$$

where, $\Omega' = (\Omega, \Lambda)$ represents a set of variational hyperparameters. See [Appendix A](#) for details of q update laws and [Appendix B](#) for details of lower bound of marginal likelihood. In addition, a summary of the proposed method is shown in [Algorithm 2](#); and [Table 1](#) shows the computational complexity of each optimization.

Algorithm 2 MHGP-BDI

Input: M, σ_1^2
Output: $\hat{q}, \hat{\Omega}'$

- 1: **for** $k = 1$ to K **do**
 - 2: Get dataset through the disturbance injected expert:
 $\{a_t^*, \mathbf{s}_t\}_{t=1}^{N_k} \sim p(\boldsymbol{\tau} \mid \pi^*, \sigma_k^2)$
 - 3: Aggregate datasets : $\mathcal{D} \leftarrow \mathcal{D} \cup \{a_t^*, \mathbf{s}_t\}_{t=1}^{N_k}$
 - 4: **while** $\mathcal{L}(q, \Omega')$ is not converged **do**
 - 5: **while** $\mathcal{L}(q, \Omega')$ is not converged **do**
 - 6: Update $q(\mathbf{f}^{(m)})$, $q(\mathbf{Z})$, and $q(v_m)$ alternately
 - 7: **end while**
 - 8: Optimize Ω' with fixed q :
 $\hat{\Omega}' \leftarrow \arg \max_{\Omega'} \mathcal{L}(q, \Omega')$
 - 9: **end while**
 - 10: **end for**
-

4.2.3. Predictive Distribution:

Using variational parameter $\boldsymbol{\Lambda}$ optimized by maximizing (22), the predictive disturbance $q(g_*)$ on a new state \mathbf{s}_* can be obtained as:

$$\begin{aligned} q(g_*) &= \int p(g_* \mid \mathbf{s}_*, \mathbf{S}, \mathbf{g}) q(\mathbf{g}) d\mathbf{g} \\ &= \mathcal{N}(g_* \mid \mu_{g*}, \sigma_{g*}^2), \end{aligned} \tag{23}$$

$$\mu_{g*} = \mathbf{k}_{g*}^\top (\boldsymbol{\Lambda} - \mathbf{I}/2) \mathbf{1}_N + \mu_0, \tag{24}$$

$$\sigma_{g*}^2 = k_{g**} - \mathbf{k}_{g*}^\top (\mathbf{K}_g + \boldsymbol{\Lambda}^{-1})^{-1} \mathbf{k}_{g*}, \tag{25}$$

where $\mathbf{k}_{g*} = \mathbf{k}_g(\mathbf{s}^*, \mathbf{S})$, and $\mathbf{k}_{g**} = \mathbf{k}_g(\mathbf{s}^*, \mathbf{s}^*)$. As such, a level of disturbance injected at the next iteration $k + 1$ is calculated as: $\sigma_{k+1}^2(\mathbf{s}_*) = e^{\mu_{g*}}$.

Table 2: Computational complexity of each prediction in MHGP-BDI: N is number of training data sets.

	$q(g_*)$	$p(a_*^{(m)} \mathbf{s}_*, \mathbf{S}, \mathbf{a}^*)$
MHGP-BDI	$\mathcal{O}(N^3)$	$\mathcal{O}(N^3)$

In addition, using the hyperparameters Ω' and the variational posterior distributions q optimized by variational Bayesian learning, the predictive distribution of the m -th action $a_*^{(m)}$ on a current state \mathbf{s}_* is derived as:

$$\begin{aligned}
& p(a_*^{(m)} | \mathbf{s}_*, \mathbf{S}, \mathbf{a}^*) \\
& \approx \int p(a_*^* | \mathbf{f}^{(m)}, g_*, \mathbf{s}_*) q(\mathbf{f}^{(m)}) q(g_*) d\mathbf{f}^{(m)} dg_* \\
& = \int \mathcal{N}(a_*^* | \mu_*^{(m)}, c_*^{2(m)} + \exp(g_*)) \mathcal{N}(g_* | \mu_{g*}, \sigma_{g*}^2) dg_*; \quad (26)
\end{aligned}$$

however, it is analytically intractable to compute. Alternatively, using a Gauss-Hermite quadrature rule (Liu and Pierce, 1994), mean $\mu_*^{(m)}$ and variance $\sigma_*^{2(m)}$ of the predictive distribution (26) can be approximated as:

$$\mu_*^{(m)} = \mathbf{k}_{f*}^{(m)\top} (\mathbf{K}_f^{(m)} + \mathbf{R}^{-1})^{-1} \mathbf{a}^*, \quad (27)$$

$$\sigma_*^{2(m)} = c_*^{2(m)} + \exp(\mu_{g*} + \sigma_{g*}^2 / 2), \quad (28)$$

$$c_*^{2(m)} = k_{f**}^{(m)} - \mathbf{k}_{f*}^{(m)\top} (\mathbf{K}_f^{(m)} + \mathbf{R}^{-1})^{-1} \mathbf{k}_{f*}^{(m)}, \quad (29)$$

where $\mathbf{k}_{f*}^{(m)} = \mathbf{k}_f^{(m)}(\mathbf{s}^*, \mathbf{S})$, and $\mathbf{k}_{f**}^{(m)} = \mathbf{k}_f^{(m)}(\mathbf{s}^*, \mathbf{s}^*)$; and Table 2 shows the computational complexity of each prediction. Additionally, m is chosen as the value that maximizes the inverse of the predicted variance $\sigma_*^{2(m)}$ as:

$$\hat{m} = \arg \max_m \frac{1}{\sigma_*^{2(m)}}, \quad (30)$$

as such, meaning the \hat{m} -th GP is selected, due to its minimal uncertainty.

5. Simulation

In this section, the proposed methodology (MHGP-BDI) is evaluated in regards to the following questions, to examine key objectives of capturing human behavior characteristics in a simulated precision wall-avoidance task: (i) flexibility: how does capturing multiple optimal human actions affect imitation learning of robotic tasks?, (ii) robustness: how does injecting disturbances into human demonstrations affect the applicability of learned policies?, and (iii) risk-sensitivity: how does injecting disturbance into human action command affect human demonstrations' feasibility?

Evaluation Metrics: Performance of MHGP-BDI is considered during the training phase and execution phase. For the former, *demonstration feasibility*, or the success rate of collecting training data with a human expert in the loop, is evaluated. On the latter, *execution performance*, or the success rate of deploying the learned policy after training, is evaluated. These metrics are reported in the wall-avoidance simulation study (Section 5.1) and the real robot assembly study (Section 6). By comparing both performances across different algorithms, each algorithm is evaluated for how effectively it obtains policy performance while ensuring the demonstration feasibility.

Comparison Methods: To evaluate the proposed method (MHGP-BDI), comparisons are made between 8 baselines. Each baseline's features (flexibility, robustness, and demonstration feasibility) are represented in Table 3. Specifically, these algorithms are implemented as:

- **Behavior Cloning (BC)** ([Bain and Sammut, 1995](#)): Conventional supervised imitation learning as described in Section 3.1 using a neural network policy model,

Table 3: Comparison models in terms of flexibility, robustness, and demonstration feasibility.

Learning Models	Flexibility	Robustness	Demonstration Feasibility
BC (Bain and Sammut, 1995)	✗	✗	✓
DART (Laskey et al., 2017)	✗	✓	✗
CVAE-BC (Ren et al., 2020)	✓	✗	✓
UGP-BC	✗	✗	✓
UGP-BDI	✗	✓	✗
UHGP-BDI	✗	✓	✓
MGP-BC	✓	✗	✓
MGP-BDI (Oh et al., 2021)	✓	✓	✗
MHGP-BDI (Proposed)	✓	✓	✓

- **Disturbances for Augmenting Robot Trajectories (DART)** (Laskey et al., 2017): Robust imitation learning by injecting disturbance into expert as described in Section 3.2 using a neural network policy model,
- **Conditional Variational AutoEncoders BC (CVAE-BC)** (Ren et al., 2020): Multi-modal imitation learning based on BC algorithm using a CVAE policy model,
- **Uni-modal GP Behavior Cloning (UGP-BC)**: BC using standard uni-modal GPs (Rasmussen, 2003),
- **Multi-modal GP BC (MGP-BC)**: BC using infinite overlapping mixtures of GPs (IOMGP),

- **UGP-BDI**: BDI using standard uni-modal GPs and state-independent disturbance model with a constant disturbance level of σ^2 ,
- **Uni-modal Heteroscedastic GP BDI (UHGP-BDI)**: BDI using standard uni-modal GPs and Heteroscedastic Gaussian Processes (HGP) as state-dependent disturbance model $\sigma^2(\mathbf{s}_t)$,
- **MGP-BDI** (Oh et al., 2021): BDI using IOMGP policy model and state-independent disturbance model which level parameter as σ^2 .

See Appendix D for how the hyperparameters of each method are set. Note, in all experiments, demonstrations are performed without injecting disturbances in the first iteration (*i.e.*, $\sigma_1^2 = 0$); since initially, there is no available evidence of which level of disturbance is suitable.

5.1. Wall-avoidance Task

Initially, a wall-avoidance task involving multiple apertures is presented (Figure 5-(a)). In this experiment, demonstrations are conducted in an environment involving states in which physical contact (*e.g.*, collisions of an agent and walls) is likely to occur, and the demonstration feasibility (*e.g.*, avoiding collision) will be evaluated. The learned policy is evaluated through test execution episodes to evaluate its flexibility capturing multiple optimal actions from demonstrations (*e.g.*, multiple paths through an aperture to reach the goal), and robustness against environmental variations (*e.g.*, starting positions of the agent or inertial of the agent) that may induces the covariate shift.

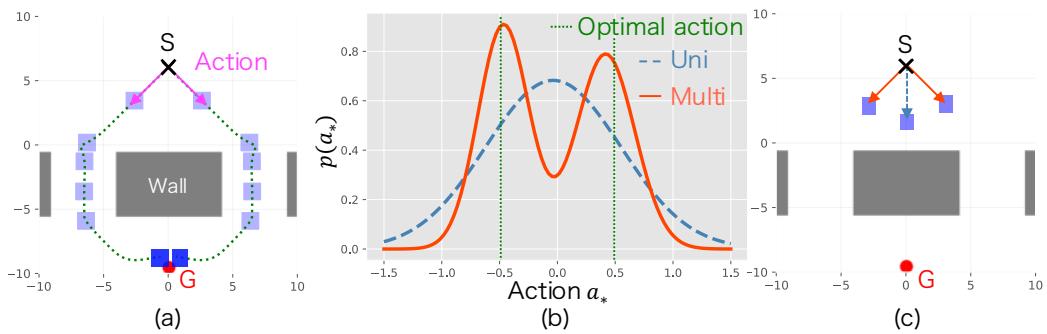


Figure 5: Wall-avoidance Task (Wide). (a): Environment of passing through a multiple aperture. S and G represent a starting and a goal position, respectively. An algorithmic supervisor’s demonstrated movement, which includes the cautious phase (*e.g.*, move slow when a robot is close to an aperture), is captured as multiple frames with a 0.025 frame rate. (b), (c): Comparing flexibility between multi-modal approaches and uni-modal approaches. (b) The predictive distribution of x-axis action a_* in a given starting position state. (c) Movements of multi-modal approaches and uni-modal approaches at policy application phase.

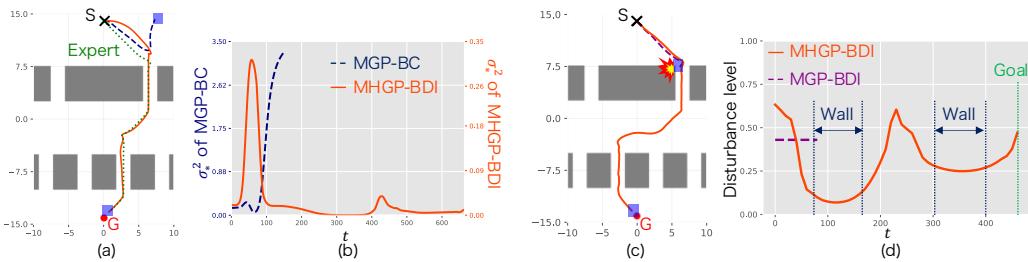


Figure 6: Wall-avoidance Task (Complex). (a), (b): Comparing robustness between MHGP-BDI and MGP-BC. (a) Generated trajectory from policy learned by MHGP-BDI and MGP-BC, and (b) sequentially depicts the predictive action variance σ_*^2 (*i.e.*, norm of the XY-axis σ_*^2) of both policy at each step. This result shows that as the agent deviates from the expert’s trajectory towards the perpendicular distance, the confidence decreases as the data becomes more sparse. (c), (d) : Demonstration feasibility comparison of MHGP-BDI and MGP-BDI. (c) Demonstration trajectory with injecting a state-dependent disturbance (MHGP-BDI) and a state-independent disturbance (MGP-BDI), and (d) sequentially depicts the level of disturbance injected at each step.

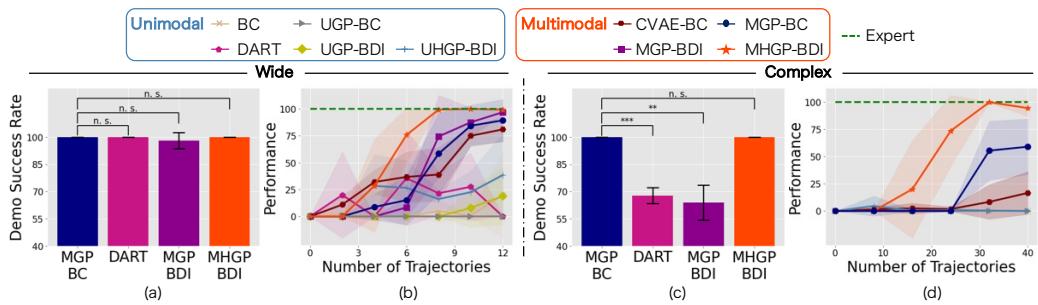


Figure 7: Wall-avoidance Task Results (Wide & Complex). (a), (c): Comparing the demonstration success rate for representative learning methods (MGP-BC, DART, MGP-BDI, and MHGP-BDI) of each robustification method. The demonstration success rate of each comparison method is measured as the mean and standard deviation of the demonstration success probability for the entire trials of the final learning iteration. Significant differences by t-test were observed between the proposed method and baselines (** : $p < 0.005$, *** : $p < 0.0005$). Note, uni-modal methods exhibit similar results in these experiments. It is seen that demo success rate is more related to robustifying than flexibility; thus, these results focus on comparing between robustifying approaches. (b), (d): Comparing task performance with the number of trajectories. The task performance of policy application is measured as the mean and standard deviation of the task success probability by conducting five learning trials and testing each learned policy 100 times.

5.1.1. Setup

In the wall-avoidance task environment (Figure 5-(a)), the aim is for the agent (blue square with width and height 1.4 cm and 1.5 cm respectively) to move from the starting position (black cross) through one of the two apertures to the goal position (red circle) without colliding with the wall (grey square). The system state is the agent’s position (*e.g.*, x , y -axis coordinates), and the action is the agent’s velocity (*e.g.*, x , y -axis).

Expert demonstrations are provided by an algorithmic supervisor, specifically human-like cautious behavior (Nagengast et al., 2011) is generated by a classical PID controller. This behavior is simulated by adjusting the agent’s velocity during task execution: high velocity (high p-gain) in open regions far from apertures, and low velocity during aperture traversal, as shown in Figure 5-(a).

Wide: under these experimental parameters, demonstrations of passing through each aperture (aperture width is 5.0 cm) is provided in sequence. If the agent collides with a wall or fails to reach the goal position within the time limit (400 steps), it is considered as a failure, and data is discarded, and the demonstration is restarted. After collecting 2 demonstration trajectories, the data are used to optimize the policy and the disturbance until the optimization equation (22) converges. In contrast, learning methods that fail to collect demonstrations more than 5 times are considered a learning failure and are not included in the task performance comparison. This process is defined as one iteration of k in Algorithm 2, and is repeated K times, adding the successful demonstrations to the training dataset and continuously updating the policy and disturbance until the fixed number of iterations is reached.

In this experiment, K is empirically chosen to stop learning when the average of injected disturbance level is sufficiently small (*i.e.*, learned policy from each comparison is achieved at $K = 6$). During the test execution stage, each element of the initial state is deviated by an additive uniform noise $\epsilon_{\mathbf{s}_0} \sim \mathcal{U}(-0.05 \text{ cm}, 0.05 \text{ cm})$, and positions of the walls and goal remain constant.

Complex: to evaluate the proposed method’s scalability, a second experiment is also presented for a more complex task, as shown in Figure 6-(a),(c). In this, apertures with a smaller width (2.0 cm) is placed in the environment, and a secondary wall with four apertures is additionally placed below the first wall of the previous experiment. The clearance for moving the agent is smaller in the both layer apertures (0.5 cm), requiring more precise control to avoid collision. Additionally, this secondary layer creates new traversal branches, inducing additional multiple optimal actions and requiring longer steps to accomplish the task. Due to the increased task complexity, the time limitation is increased to 1500 step and the maximum number of demonstration trajectories for updating the policy and disturbance estimates is increased to 8 and the maximum number of iterations is $K = 5$ (total 40 trajectories). Additionally, during the test execution stage, each element of the initial state is deviated by the wider additive uniform noise $\epsilon_{\mathbf{s}_0} \sim \mathcal{U}(-0.1 \text{ cm}, 0.1 \text{ cm})$.

5.1.2. Results

This section presents the qualitative and quantitative analysis of this simulation. The qualitative analysis is presented in terms of (i) flexibility, (ii) robustness, (iii) demonstration feasibility. In addition, the quantitative anal-

ysis is presented with previously described evaluation metrics. The results of this simulation are shown in Figure 5, 6, 7.

(i) Flexibility: Initially, to evaluate the ability of the agent to flexibly learn in scenarios with multiple-optimal actions (Figure 5-(a)), policies are learned for each of the comparison methods, and generated action distributions are shown in Figure 5-(b). In this, it is seen that the uni-modal policy learned by UHGP-BDI fails to capture multiple optimal actions at the starting position (S) of the task. Note, all other uni-modal GP-based methods (UGP-BC, UGP-BDI) exhibit very similar Gaussian distributions. Specifically, as seen in Figure 5-(c), uni-modal approaches learn a mean-centered policy from the demonstrations, resulting in an incorrect average direction and inability to reach any aperture. However, policies learned by MHGP-BDI can correctly capture the multi-modal distribution (Figure 5-(b)) and learn the two optimal actions (Figure 5-(c)). Note, all other multi-modal GP-based methods (MGP-BC, MGP-BDI) exhibit very similar Gaussian mixture distributions.

(ii) Robustness: To evaluate the effect of demonstrations on policy learning and application (*i.e.*, the test execution phase), initially the successful demonstrations from the MGP-BC method are used for policy learning. The results for applying policy learning is seen in (Figure 6-(a)), where immediately the agent poorly performs the task by veering away from trained trajectory, and does not recover back to the optimal trajectory. This demonstrates the error compounding problem, whereby the lack of robustness in the learned model causes the agent to visit unexplored and unrecoverable states. This effect can be seen in (Figure 6-(b)), whereby the action variance

of MGP-BC is dramatically increased during policy learning, in a failed attempt to mitigate the problem. As such, the confidence of the policy learned by MGP-BC decreases monotonically after the 60 th time step and fails the task (time-limitation). Note that the other multi-modal neural network-based approach (CVAE-BC) exhibits a similar phenomenon.

In contrast, in the MHGP-BDI method, error compounding is minimised by injecting disturbances into demonstrations, thereby collecting recovery actions under conditions that drift from an optimal trajectory. Accordingly, when applying the policies learned in the MHGP-BDI method, even though the agent similarly immediately drifts, it can recover to an optimal trajectory and complete the task (Figure 6-(a)). Even if there is a momentary decrease in confidence due to environmental variations, the policy exhibits a high confidence (Figure 6-(b)). Note that confidence is relatively lower when passing through the first aperture (60 th time step) than the second aperture (440 time step), since the perpendicular distance from the expert’s trajectory to the agent is larger, induced by the environmental variations (*e.g.*, random starting position and inertial effects).

(iii) Demonstration Feasibility: Given this demonstration of flexibility and robustification, the disturbance injection approaches are then evaluated in terms of their ability to limit collisions. Specifically, the ability of methods which utilizes either a state-independent (MGP-BDI) or a state-dependent (MHGP-BDI) disturbance, is evaluated in aperture traversal. In Figure 6-(c), it is seen that state-independent methods, which do not regulate disturbance, collide with the walls, due to its constant level of disturbance (as seen in Figure 6-(d)). As such, state-independent robustification (MGP-

BDI) injects disturbances that are unsafe, and render this method unable to collect supervisor demonstrations, and the learning process cannot proceed any further. Note that the other state-independent approach (DART) exhibit similar phenomenon. In contrast, MHGP-BDI equipped with a state-dependent disturbances, successfully navigates the tasks-space, by reducing the level of disturbance to about 23% of that of the MGP-BDI when it comes close to aperture ($t = 86$) (Figure 6-(d)). This cautious-like behavior enables the agent to pass through the aperture safely, and complete the demonstrations.

Quantitative Evaluation: To evaluate the stability of these approaches, these experiments were repeated five times. The averaged demo success probabilities for representative learning methods (MGP-BC, DART, MGP-BDI, and MHGP-BDI) of each robustification method are shown in Figure 7-(a), (c). In addition, the averaged task execution performance of each learned policy is shown in Figure 7-(b), (d).

In the wide aperture experiments, (Figure 7-(a)), the demonstration feasibility is not significantly different from MGP-BC even with the disturbance injection learning approaches (DART, MGP-BDI and MHGP-BDI), since the aperture size is sufficiently large. However, (Figure 7-(b)), the uni-modal policy schemes (BC, DART, UGP-BC, UGP-BDI and UHGP-BDI) all fail to learn the multi-modal task and as expected produce low performance (under 50%) results, due to lack of flexibility (as discussed in Figure 5-(b), (c)). Note, DART, UGP-BDI, and UHGP-BDI gain additional robustness over the standard uni-modal approaches; since some deviated states, induced by control errors due to failure to capture multiple optimal actions, may be cov-

ered by disturbance injection. Thus its performance increases monotonically in the early stages; however it eventually cannot exceed 50% due to the limitation of learning flexibility. In comparison, the multi-modal policy schemes (CVAE-BC, MGP-BC, MGP-BDI and MHGP-BDI) improve the learning performance by nearly 100% with increasing number of trained trajectories.

In the complex aperture experiments, (Figure 7-(d)), even multi-modal BC approaches (CVAE-BC, MGP-BC) using a flexible multi-modal policy, learned policies' task performance cannot exceed 60%, due to the lack of robustness (as discussed in Figure 6-(a), (b)). However, if disturbances are injected into demonstrations in a state-independent manner (DART, MGP-BDI), this perturbation may cause physical contact at narrow apertures, and lead to demonstration failure (as discussed in Figure 6-(c), (d)). This failure is seen in DART and MGP-BDI; both have a low demonstration success rate in the complex simulation (Figure 7-(c)), with demonstration success decreased by 32% compared to the wide-version. Accordingly, DART and MGP-BDI are removed from the comparison of learning performance in the complex aperture experiments (Figure 7-(d)), since they failed 5 times demonstrations during learning iteration. In contrast, MHGP-BDI, which can learn a state-dependent disturbances, has a 100% demonstration success rate for both simulations, and consistently shows superior learning efficiency and obtain policies with high task performance (nearly 100%, only very small failures due to some specific starting position or given environmental noise).

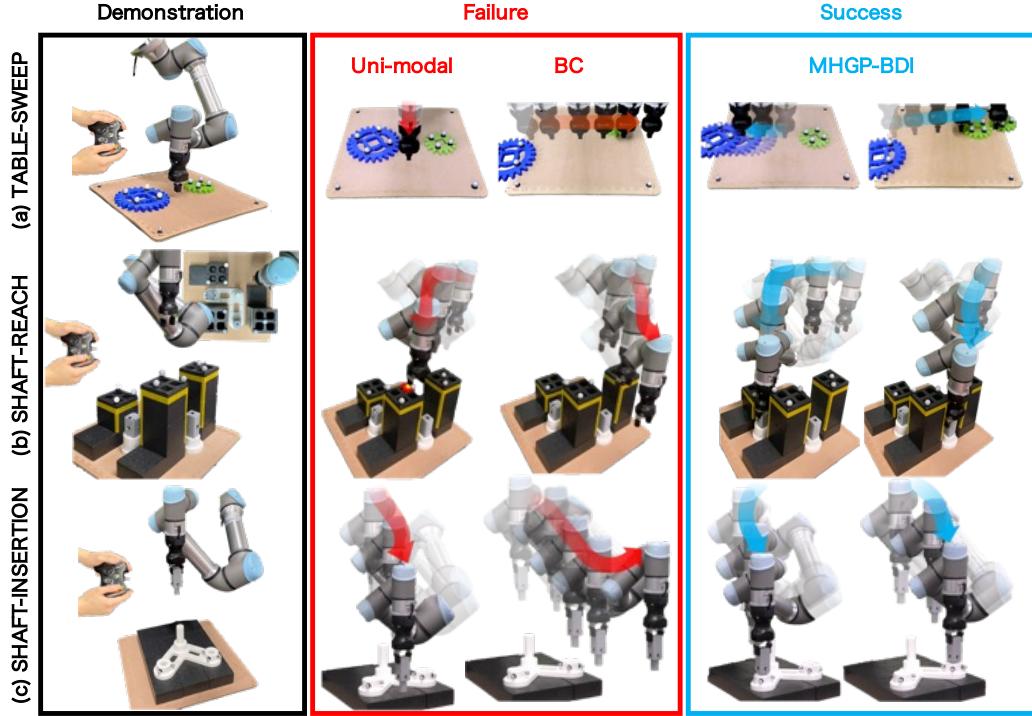


Figure 8: Real Robot Experiments Setup. Experimental environments for 6-DOF robotic arm (UR5e) assembly tasks with human expert are conducted as: (a) sweeping gears on the table, (b) reaching to a shaft with avoiding obstacles, (c) inserting a shaft into a hole. Test execution scenes of learned policies: **Failure:** (Uni-modal) Due to the inability to capture the multiple optimal actions, these approaches learn mean-centred policy, resulting in (a) sweeping a centre of the gears, (b) colliding to an obstacle between shafts, (c) putting a shaft onto the centre of the holes. (BC) Even though the approach can capture multiple optimal actions, without disturbance injection in demonstrations, policies are vulnerable to environmental variations, resulting in a robot departure from the demonstrated states; thus, the robot (a) cannot sweep gears completely or ((b), (c)) go out of the task space. **Success:** Our proposed method (MHGP-BDI) provides policies that are learned by capturing optimal actions or initiating recovery actions by injecting optimized disturbances, which allow the robot to successfully (a) sweep the whole gears, (b) reach to both shafts and (c) insert a shaft into the holes, in a given any starting position. Our supplementary video can be seen at: <https://youtu.be/NeJy8pfkrC4>.

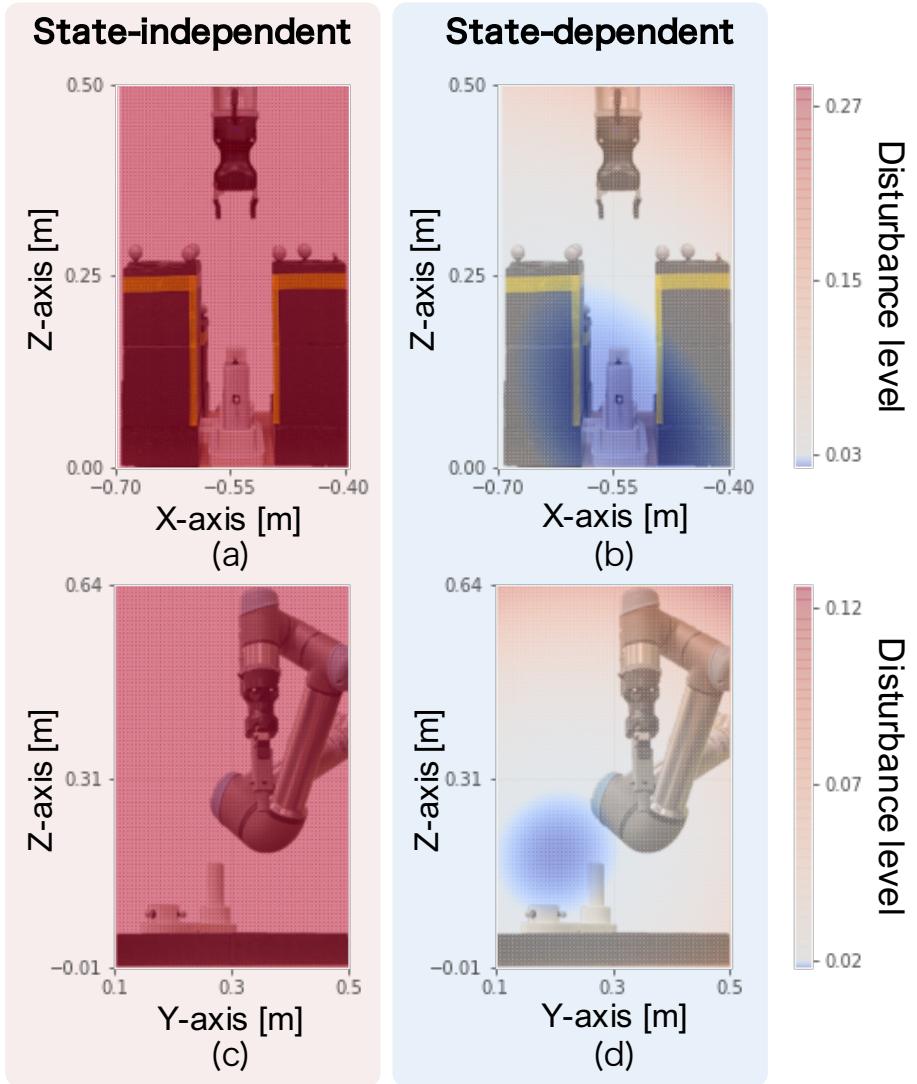


Figure 9: Comparison of Two Types of Disturbances. Both disturbances (state-independent and state-dependent) are injected into a human demonstration during a shaft-reach task ((a), (b)) and a shaft-insertion task ((c), (d)). Graphs showing the disturbance level with regards to the end-effector position with fixed y-coordinate ((a), (b)) fixed Y-axis = 0.23 m) and the grasped shaft position with fixed X-coordinate ((c), (d)) fixed X-axis = -0.7 m) : state-independent disturbances have a uniform level in any state (left), and state-dependent disturbances, obtained by MHGP-BDI, have a spectrum of level depending on the state (right). Colors of disturbance level are normalized by the amount of clearances for each task.

Table 4: Real Robot Experiments Results. Each learning model’s demonstration success is measured at the last iteration of learning each task (10 demonstration attempts). If a human fails demonstration (*e.g.*, robot crashes with obstacles or fails to complete the task within the time limit) over 5 times at a single iteration then finish the learning process, failing to obtain a policy. Such learning models are not able to measure the task execution performance of learned policy; thus it is annotated as N/A. The test execution performance of policies learned by each learning model has been measured over 10 test executions.

Learning Models	Demonstration Success			Test Execution Performance		
	Table-sweep	Shaft-reach	Shaft-insertion	Table-sweep	Shaft-reach	Shaft-insertion
BC	10/10	10/10	10/10	0/20	0/10	0/10
DART	10/10	4/10	4/10	0/20	N/A	N/A
CVAE-BC	10/10	10/10	10/10	16/20	4/10	5/10
UGP-BC	10/10	10/10	10/10	0/20	0/10	0/10
UGP-BDI	10/10	0/10	0/10	0/20	N/A	N/A
UHGP-BDI	10/10	10/10	10/10	0/20	0/10	0/10
MGP-BC	10/10	10/10	10/10	10/20	7/10	5/10
MGP-BDI	10/10	2/10	1/10	20/20	N/A	N/A
MHGP-BDI (Proposed)	10/10	10/10	10/10	20/20	10/10	10/10

6. Real Robot Experiments

In this section, three experiments are conducted to demonstrate the proposed method’s applicability on various scenarios as shown in Figure 8. MHGP-BDI is applied to a 6-DOF UR5e (Universal Robotics) robot to learn

three assembly tasks:

- **Table-sweep task:** the robot’s ability to reach multiple objects and sweep them out of the table, is evaluated. Demonstrations of sweeping two gears on the table are provided by a human, as shown in Figure 8-(a). The state of the system is defined as the relative 2D coordinate from the robotic arm to two gears ($Q = 4$); an action is defined as the velocity of the robotic arm in the x and y axis.
- **Shaft-reach task:** the robot’s ability to avoid fixed obstacles and reach a shaft to grasp it, is evaluated. Demonstrations of reaching one of the assembly supplies (*e.g.*, shaft) without colliding with fixed obstacles are provided by a human, as shown in Figure 8-(b). The state of the system is defined as the relative 3D coordinate between the robot arm and two shafts ($Q = 6$), an action is defined as the velocity of the robot arm in the x , y and z axis. This is a more difficult task than the table-sweep task, as: (1) the state-action space is larger to deal with a more general setting, (2) the environment is prone to physical contact (*e.g.*, collision with obstacles).
- **Shaft-insertion task:** the robot’s ability for inserting a shaft into a hole for assembly, is evaluated. Demonstrations of inserting the assembly supplies (*e.g.*, shaft) into one of the holes (on both side of white “L” shaped base) are provided by a human, as shown in Figure 8-(c). The state of the system is defined as the relative 3D coordinate between the robot arm and two holes ($Q = 6$), an action is defined as the velocity of the robot arm in the x , y and z axis. This scenario is

more complicated than the shaft-reach task as: (1) Physical contacts are involved, requiring more sensitive behavior, (2) the clearance for inserting shaft is smaller (only 1 mm), requiring more precise control.

In the following experiments, learned policies are evaluated in terms of ability to flexibly learn tasks with multiple optimal actions (*e.g.*, the order in which to interact with the objects), and well as robustness to environmental covariance shift inducing disturbances (*e.g.*, friction between the objects and environment, inducing variations in movement). Here, the test execution performance of the learned policies is measured by 10 deployment tests of the final learned policy for each learning method. In addition, human demonstrations are evaluated in terms of feasibility for completing a task. For example, if the robot collides with obstacles or fails to complete a task during the demonstration stage within the time limit (400 steps), it is considered a failure, and the demonstration is instead repeated. Suppose a human fails demonstration over 5 times at a single learning iteration. In that case, it is considered learning failure (terminate the learning process), and such learning methods are removed from the task performance comparison. Here, the demonstration success is measured by conducting 10 demonstration attempts with same conditions (*e.g.*, disturbance model) used at the last learning iteration.

To measure the state of the system, markers are attached to each object (gear, shaft, hole) and tracked through a motion capture system (OptiTrack Flex13). In addition, to validate the robustness of the policy to deviations from optimal trajectories, each element of the initial state is deviated with additive uniform noise: (1) table-sweep task: $\epsilon_{\mathbf{s}_0} \sim \mathcal{U}(-0.05 \text{ m}, 0.05 \text{ m})$

(2) shaft-reach/insertion task: $\epsilon_{s_0} \sim \mathcal{U}(-0.005 \text{ m}, 0.005 \text{ m})$ The assembly model used ([Siemens, 2017](#)) (*e.g.*, gears, shafts, base) is a standardized benchmark task for robotic assembly.

6.1. Table-sweep Task

6.1.1. Setup

Initially, two gears and the robot arm are placed at fixed coordinates on a table. The human expert performs demonstrations in which the objects are swept off the table. Two demonstrations from these initial conditions are then performed, capturing both variations in the order of which the objects are swept from the table. The method optimizes a policy and disturbances until (22) is converged. This process is repeated $K = 4$ times (8 trajectories).

The learned policies' performance is evaluated according to the number of gears swept out of the table at the end of the test execution episode.

6.1.2. Result

The results of this experiment are seen in Table 4. In the table-sweep task, the expert can successfully perform demonstrations using any of the proposed methods, even when disturbances (*i.e.*, state-dependent or state-independent) are injected; since the environment does not involve any obstacles in which disturbances may induce risks (*e.g.*, collisions or confusion in decision making).

Given these successful demonstrations, task performance is then evaluated in Table 4. In this, it is seen that the uni-modal policy methods (BC, DART, UGP-BC, UGP-BDI, UHGP-BDI) all fail. Specifically, in terms of flexibility, it is seen that instead of capturing multiple optimal actions at

the start of sweeping, instead a mean-centered policy is learned that fails to reach either gears (Figure 8-(a) Uni-modal failure). As such, they have a zero task execution performance, and demonstrate a lack of flexibility. In comparison, the multi-modal policy methods (CVAE-BC, MGP-BC, MGP-BDI and MHGP-BDI) correctly learn that there are multiple optimal actions (*e.g.*, move to blue or green gear), and outputs actions to sweep the two gears accordingly (Figure 8-(a) Success). However, while CVAE-BC and MGP-BC incorporate flexibility, it has a low task performance (80% and 50%, respectively). This is due the dynamic behavior of gears varying between the test execution and training due to environmental variations (*e.g.*, friction between gears and the table), thereby introducing error compounding and resulting in the robot being unable to sweep the remaining gear after the first sweep (Figure 8-(a) BC failure). In contrast, while the proposed disturbance-injected methods also experiences some uncertainty, it recovers and successfully sweep gears (Figure 8-(a) Success); thus MGP-BDI and MHGP-BDI show greatly improved performance (both are 100%).

6.2. Shaft-reach Task

6.2.1. Setup

Prior to the start of a demonstration, two shafts and robot arm are placed at fixed positions between the obstacles (black blocks) on the table. Following the same procedure as outlined in Section 5.1.1, the human expert performs demonstrations in which the robot arm reach to each shaft alternatively. When the robot arm collides with an obstacle, it is considered a failure. After collecting two demonstrations, a policy and disturbances are optimized until (22) is converged. This process is repeated $K = 4$ times (8 trajectories).

The learned policies' performance is evaluated according to the success of the test execution episode, determined by whether the robot arm grasped the shaft at the end of the episode.

6.2.2. Result

The results of this experiment are seen in Table 4. In this, it is seen that the state-independent disturbance injection methods (DART, UGP-BDI and MGP-BDI) have a poor demonstration success rate, 40%, 0% and 20% respectively. Specifically, to examine this result, the learned disturbance is visualised in the state-space (Figure 9-(a), (b)). In this, it is seen that state-independent methods generate disturbances with a uniform level, and as such inducing physical contacts (*e.g.*, collide with obstacle) at the demonstration. In contrast, a state-dependent disturbance injection methods (UHGP-BDI and MHGP-BDI) can regulate disturbance level small when robot arm close to obstacles (Figure 9-(b)), both have a 100% demonstrations success rate.

At the policy execution phase, it is seen that the uni-modal policy methods (BC, UGP-BC, UHGP-BDI) both fail to correctly learn policies to account for multiple optimal actions in the environment; thus robot collide with obstacle between the two shafts as shown in Figure 8(b)-Uni-modal failure. As such, they have a 0% success rate, and demonstrate a lack of flexibility. In contrast, the multi-modal policy methods (CVAE-BC, MGP-BC and MHGP-BDI) show improved performance (40%, 70% and 100%, respectively). However, it is clear that even when incorporating flexibility, the success rate for BC is poor; since environmental variation (*e.g.*, starting position), the robot may deviate from trained states (Figure 8(b)-BC failure), demonstrating a lack of robustness.

Table 5: Shaft-insertion Task Results (Multiple Subjects). Experimental results of robotic shaft insertion varied by four expert subjects. To obtain sufficient expert demonstrations, test subjects are practiced velocity control and make a smooth trajectory with simple instructions (*e.g.*, move the shaft from the starting point to the hole while sequentially decelerating the robotic arm), before performing the demonstrations. Each multi-modal approach (MGP-BC, MGP-BDI, and MHGP-BDI) has been validated during the demonstration and test execution phases. The success rate of each learning model is the mean and standard deviation of the results from four subjects.

Subjects	Demonstration Success			Test Execution Performance		
	MGP	MGP	MHGP	MGP	MGP	MHGP
	BC	BDI	BDI	BC	BDI	BDI
#1	10/10	1/10	10/10	5/10	N/A	10/10
#2	10/10	0/10	10/10	3/10	N/A	9/10
#3	10/10	1/10	10/10	6/10	N/A	10/10
#4	10/10	0/10	10/10	2/10	N/A	9/10
Success Rate (%)	100 \pm 0	5.0 \pm 5.0	100 \pm 0	40.0 \pm 15.8	N/A	95.0 \pm 5.0

6.3. Shaft-insertion Task

6.3.1. Setup

Before the start of a demonstration, the “L” shaped base and shaft grasped robot arm are placed at fixed starting positions in the environment. This task involves physical contact (*e.g.*, between shaft and base) and requires a scheme to protect the experimental environment, including a robot and objects. As such, an impedance control ([Duchaine and Gosselin, 2007](#)) is implemented, that cancels the force by adding reverse direction velocity

when the shaft collides with the base.

Following the same procedure as outlined in Section 5.1.1, the human expert performs demonstrations in which the robot arm inserts the shaft into each hole alternatively. After collecting four demonstration, a policy and disturbances are optimized until (22) is converged. This process is repeated $K = 3$ times (12 trajectories).

The learned policies' performance is evaluated according to the success of the test execution episode, determined by whether the shaft is in the hole at the end of the episode.

6.3.2. Result

The results of this experiment are seen in Table 4. In the demonstration phase, methods that employ state-independent disturbance injections (DART, UGP-BDI and MGP-BDI) have a uniform strong level of disturbance in any state (seen Figure 9-(c)). This disturbances make it challenging to insert the shaft; thus leading to a poor demonstration success rate (40%, 0% and 10%, respectively). In contrast, a state-dependent disturbance injection methods (UHGP-BDI and MHGP-BDI) regulates the disturbance level when the shaft is close to the hole (Figure 9-(d)), and as such has a superior demonstrations success rate (both are 100%). This allows for both enriching the demonstrations in clear open spaces, and allowing for precision manipulation in tasks that require fine control, such as physical contact.

At the test execution of learned policies, it is seen that, as expected, the uni-modal policy methods (BC, UGP-BC, UHGP-BDI) learned mean-centered policies that generate movements between the two holes and fail the task (Figure 8(c)-Uni-modal failure); they have a 0% success rate, demon-

strating a lack of flexibility. Furthermore, similar to Section 6.2.2, incorporating flexibility without robustification (CVAE-BC and MGP-BC), causes the robot to deviate from trained states (Figure 8(c)-BC failure), giving a poor success rate (50% and 50%, respectively). In contrast, policies learned by MHGP-BDI can output multiple optimal actions while robust to sources of error as shown in Figure 8. In particular, despite the small clearance in the hole’s vicinity, it is seen that the robot can overcome with precise control, resulting in improved performance (100%).

In addition, to evaluate intersubject robustness of the methods, four human experts with experience in robotics are used to compare multi-modal approaches (MGP-BC, MGP-BDI, MHGP-BDI); with results shown in Table 5. Note, for the sake of simplicity and fairness of analysis, this experiment is conducted between GP-based multi-modal imitation learning approaches. In this, injecting state-independent disturbances into demonstrations results in demonstration-infeasibility for all subjects; thus, MGP-BDI has a poor demonstration success rate ($5.0 \pm 5.0\%$). Test execution performance of MGP-BC similarly demonstrates poor average success rate ($40 \pm 15.8\%$), due to error compounding similar to previous experiments. However, these results show a higher intrasubject variance, due to the inherent differences between human-specific strategies. In contrast, MHGP-BDI consistently obtains a superior success rate on both demonstrations and test executions ($100 \pm 0\%$ and $95 \pm 5.0\%$, respectively) with multiple subjects.

7. Discussion

As demonstrated in the experimental results, our proposed method of combining flexibility, robustification, and risk-sensitivity is effective for learning robust multi-action policies. By introducing a state-dependent disturbance, our proposed method automatically adjusts the level of disturbance to be appropriate depending on the state and can collect richer demonstration datasets, including recovery actions under challenging situations without losing demonstration feasibility. Furthermore, several possibilities for extending the proposed method to address other significant robot learning challenges are discussed in this section.

7.1. Exploiting Other Human Characteristics

In regards to the overarching conceptual idea of learning *human behavioral characteristics* as a fundamental part of imitation learning of robotic tasks, the proposed framework is suitable for modelling inherent behaviors, and appropriately utilizes them. BDI is a specific implementation of our proposal which models this by injecting disturbance into an expert’s demonstration to learn robust multi-optimal policies within a Bayesian framework. Experimental results show that BDI significantly outperforms comparative methods, mitigating the contradictions between the assumptions of standard imitation learning algorithms and actual demonstrator behavior.

In the future, BDI can be extended and applied to mitigate not only the contradictions presented in this paper but also the following contradictions: While the standard imitation learning assumes that the demonstrator is capable of outputting the optimal behavior in any given state, in robotic tasks

that require high precision, such as a needle threading task (Jourdan et al., 2004), even a human demonstrator rarely succeeds at one time without any mistakes. Applying imitation learning in such tasks requires a lot of time and cost for collecting demonstration data. To alleviate this contradiction, human demonstrations can be parameterized with weighted values of task achievement and enabling to learn from demonstration data that contains mistakes (Brown et al., 2019; Chen et al., 2020; Tahara et al., 2022).

In addition, while standard imitation learning only deals with low-level control abilities, such as determining velocity from a given robot’s position, many tasks in daily human life have long-term processes and are divided into symbolic sub-tasks, which require high-level planning ability to determine the sequence of sub-tasks (Fikes and Nilsson, 1971). In applying conventional imitation learning to such tasks, even changing one sub-task requires re-learning the entire task from the beginning, which is inefficient and imposes a heavy burden on a human demonstrator. To address this contradiction, the hierarchical policy model (Fox et al., 2019; Xu et al., 2019), which can simultaneously execute high-level planning in symbolic task space and low-level control in geometric task space, can be employed to learn a human high-level planning ability.

7.2. Improving Computational Efficiency

Since the main purpose of our experiment is to investigate the effect of our method on several tasks with underlying contradictions in the demonstration data, only the Gaussian Processes (GPs) are employed as an inference tool for generating policies or disturbance of BDI. While GP allows several advantages to our method, it still suffers from computational complexity, which

significantly increases with the number of data points N as Table 1. As such, BDI as applied to long-term tasks with real-time control systems is limited by this underlying policy generation method. To address this, GPs' kernels can be approximated with randomized Fourier features from the fastfood algorithm (Le et al., 2013), which lowers the computational time of computing the inverse kernel matrix from $O(N^3)$ to $O(NW^3)$, where W is a dimension of feature space.

7.3. Dealing with Environmental Uncertainty

While environmental uncertainty is not directly addressed in this paper, uncertainty and fuzziness of environment are another important challenges in applications to real systems. In the field of adaptive control, to cope with various uncertainties on environment, attempts to achieve more accurate control (Xin et al., 2022; Zhuang et al., 2022) or safe-conscious engineering (Cheng et al., 2021) commonly propose a probabilistic model to capture complex system dynamics' uncertainty or adapt a dynamics model to an unknown environment through iterative online learning. In light of this, BDI can be extended for employing the Model Predictive Control (MPC)-type policy model (Pereira et al., 2018), in which a sequence of states and actions are estimated with a stochastic dynamics model at each time step; thereby, enabling BDI to account for environmental uncertainty.

8. Conclusion

This paper presents a novel paradigm on imitation learning, by focusing on learning human behavioral characteristics, and demonstrating its importance and usage. Our proposal Bayesian imitation learning framework

injects risk-sensitive disturbances into an expert’s demonstration to learn robust multi-action policies. This framework captures intrinsic human behavioral characteristics and allows for learning reduced covariate shift policies by collecting training data on an optimal set of states without losing demonstration feasibility. The effectiveness of the proposed method is verified on several simulations and real robotic tasks with human demonstrations.

Acknowledgments

This paper is based on results obtained from a project, JPNP20006, commissioned by the New Energy and Industrial Technology Development Organization (NEDO).

Appendix A. Update laws of variational posteriors q

The analytical solution of variational posterior $q^*(\mathbf{f}^{(m)})$ is given by the following derivation:

$$\begin{aligned} \log q^*(\mathbf{f}) &= \int \left\{ \log p(\mathbf{a}^*, \mathbf{g}, \{\mathbf{f}^{(m)}\}, \mathbf{Z}, \mathbf{v} \mid \mathbf{S}; \boldsymbol{\theta}) \right\} \\ &\quad q(\mathbf{g})q(\mathbf{Z})q(\mathbf{v})d\mathbf{g}d\mathbf{Z}d\mathbf{v} + \text{Const}, \end{aligned} \tag{A.1}$$

where, Const is a constant term for normalizing distributions. Accordingly, $q^*(\mathbf{f}^{(m)})$ is obtained by solving (A.1) as:

$$q^*(\mathbf{f}^{(m)}) = \mathcal{N}(\mathbf{f}^{(m)} \mid \boldsymbol{\mu}_{\mathbf{f}}^{(m)}, \mathbf{C}^{(m)}), \quad (\text{A.2})$$

$$\boldsymbol{\mu}_{\mathbf{f}}^{(m)} = \mathbf{C}^{(m)} \mathbf{B}^{(m)} \mathbf{a}^*, \quad (\text{A.3})$$

$$\mathbf{C}^{(m)} = (\mathbf{K}_{\mathbf{f}}^{-1(m)} + \mathbf{B}^{(m)})^{-1}, \quad (\text{A.4})$$

$$\mathbf{B}^{(m)} = \text{diag}\{r_{nm}/\mathbf{H}_{nn}\}, \quad (\text{A.5})$$

$$\mathbf{H} = \text{diag}\{\exp([\boldsymbol{\mu}_{\mathbf{g}}]_n - [\Sigma_{\mathbf{g}}]_{nn}/2)\}. \quad (\text{A.6})$$

As similar to (A.1), the analytical solution of variational posterior $q^*(\mathbf{Z})$ is given by the following derivation:

$$\begin{aligned} \log q^*(\mathbf{Z}) &= \int \left\{ \log p(\mathbf{a}^*, \mathbf{g}, \{\mathbf{f}^{(m)}\}, \mathbf{Z}, \mathbf{v}) \right\} \\ &\quad q(\mathbf{f})q(\mathbf{g})q(\mathbf{v})d\mathbf{g}d\mathbf{f}d\mathbf{v} + \text{Const.} \end{aligned} \quad (\text{A.7})$$

Therefore, $q^*(\mathbf{Z})$ is obtained by solving (A.7) as:

$$q^*(\mathbf{Z}) = \prod_{n=1}^N \prod_{m=1}^{\infty} r_{nm}^{\mathbf{Z}_{nm}}, \quad (\text{A.8})$$

$$r_{nm} = \frac{\rho_{nm}}{\sum_{m=1}^{\infty} \rho_{nm}}, \quad (\text{A.9})$$

$$\begin{aligned} \log \rho_{nm} &= -\frac{1}{2\mathbf{H}_{nn}} \{(a_n^* - [\boldsymbol{\mu}_{\mathbf{f}}^{(m)}]_n)^2 + [\mathbf{C}^{(m)}]_{nn}\} \\ &\quad - \frac{1}{2} \log (2\pi \mathbf{H}_{nn}) - \psi(\alpha_m + \beta_m) \\ &\quad + \psi(\alpha_m) + \sum_{j=1}^{m-1} \{\psi(\beta_j) - \psi(\alpha_j + \beta_j)\}, \end{aligned} \quad (\text{A.10})$$

where, $\psi(\cdot)$ is the digamma function.

As well as, the analytical solution of variational posterior $q^*(\mathbf{v})$ is given by the following derivation:

$$\begin{aligned} \log q^*(\mathbf{v}) &= \int \left\{ \log p(\mathbf{a}^*, \mathbf{g}, \{\mathbf{f}^{(m)}\}, \mathbf{Z}, \mathbf{v}) \right\} \\ &\quad q(\mathbf{f})q(\mathbf{g})q(\mathbf{Z})d\mathbf{g}d\mathbf{f}d\mathbf{Z} + \text{Const.} \end{aligned} \quad (\text{A.11})$$

As such, $q^*(v_m)$ is obtained by solving (A.11) as:

$$q^*(v_m) = \text{Beta}(v_m \mid \alpha_m, \beta_m), \quad (\text{A.12})$$

$$\alpha_m = 1 + \sum_{n=1}^N r_{nm}, \quad (\text{A.13})$$

$$\beta_m = \beta + \sum_{j=m+1}^{\infty} \sum_{n=1}^N r_{nj}, \quad (\text{A.14})$$

where, Beta is the beta function.

Appendix B. Lower bound of marginal likelihood $\mathcal{L}(q, \Omega')$

The lower bound of the marginal likelihood $\mathcal{L}(q, \Omega')$ is analytically obtained by the following derivation:

$$\begin{aligned}
& \mathcal{L}(q, \Omega') \\
&= \sum_{m=1}^{\infty} \log \mathcal{N}(\mathbf{a}^* \mid \mathbf{0}, \mathbf{K}_f^{(m)} + \mathbf{B}^{-1(m)}) \\
&\quad + \sum_{n=1}^N \sum_{m=1}^{\infty} \left[r_{nm} \left\{ \psi(\alpha_m) - \psi(\alpha_m + \beta_m) - \frac{1}{2} [\boldsymbol{\mu}_g]_n \right. \right. \\
&\quad \left. \left. + \sum_{j=1}^{m-1} \{ \psi(\beta_j) - \psi(\alpha_j + \beta_j) \} - \frac{1}{2} \log 2\pi - \log r_{nm} \right\} \right. \\
&\quad \left. - \frac{1}{2} \log \{ [\mathbf{B}^{(m)}]_{nn} / 2\pi \} \right] - \sum_{m=1}^{\infty} \text{KL}(q(v_m) \parallel p(v_m)) \\
&\quad - \text{KL}(\mathcal{N}(\mathbf{g} \mid \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g) \parallel \mathcal{N}(\mathbf{g} \mid \mu_0 \mathbf{1}, \mathbf{K}_g)), \tag{B.1}
\end{aligned}$$

where,

$$\begin{aligned}
& \text{KL}(q(v_m) \parallel p(v_m)) \\
&= \log \{ \text{Beta}(v_m \mid 1, \beta) / \text{Beta}(v_m \mid \alpha_m, \beta_m) \} \\
&\quad + (\alpha_m - 1) \psi(\alpha_m) + (\beta_m - \alpha) \psi(\beta_m) \\
&\quad + (1 - \alpha_m + \alpha - \beta_m) \psi(\alpha_m + \beta_m), \tag{B.2}
\end{aligned}$$

and

$$\begin{aligned}
& \text{KL}(\mathcal{N}(\mathbf{g} \mid \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g) \parallel \mathcal{N}(\mathbf{g} \mid \mu_0 \mathbf{1}, \mathbf{K}_g)) \\
&= \frac{1}{2} \left[\log \{ |\mathbf{I} + \mathbf{K}_g \boldsymbol{\Lambda}| \} - 1 + \text{tr} \{ (\mathbf{I} + \boldsymbol{\Lambda} \mathbf{K}_g)^{-1} \} \right. \\
&\quad \left. + \mathbf{1}^\top \left(\boldsymbol{\Lambda} - \frac{1}{2} \mathbf{I} \right)^\top \mathbf{K}_g \left(\boldsymbol{\Lambda} - \frac{1}{2} \mathbf{I} \right) \mathbf{1} \right]. \tag{B.3}
\end{aligned}$$

Table C.6: Computational complexity analysis results: optimization time and prediction time in MHGP-BDI

N	M	Optimization time [sec]	Prediction time [sec]
713	2	2.3	0.0055
711	5	5.2	0.013
719	10	9.9	0.022
1409	2	12.3	0.022
1407	5	27.0	0.053
1415	10	49.2	0.10

Appendix C. Computational complexity analysis of MHGP-BDI

As described in Table 1 and Table 2, the computational complexity of MHGP-BDI is mainly related to the number of training data sets (N) and the upper bound of mixtures (M). To analyze the impact of N and M on computational complexity of MHGP-BDI in practice, *optimization time*, duration for one optimization loop, and *prediction time*, average time for 10-step prediction, were measured in a wide version of wall avoidance simulation task; the results show that the computational complexity of MHGP-BDI is increased as N and M increase, as described in Table C.6. All experiments were ran on an Intel CPU Core i9-9900 K.

Appendix D. Hyperparameters

Details of hyperparameters of MHGP-BDI and all comparison models are provided as below.

Appendix D.1. MHGP-BDI

The hyperparameters of MHGP-BDI are as follow: $M, \omega_f, \omega_g, \mu_0, \Lambda, \beta$. These hyperparameters are empirically chosen with certain heuristic methodologies in this paper. Such hyperparameters' selection and sensitivity analysis are described as below.

The maximum number of mixtures GPs (M) and the concentration level parameter of SBP (β) are both related to the flexibility of the policy model. To spread out data to multiple GPs, β is initialized as $\beta = 100$ in all experiments. In addition, M is initialized based on the computational complexity of MHGP-BDI. Such as, a larger M makes it better for capturing multiple optimal behaviors from human demonstrations; however, the computational complexity of the algorithm increases as M grows, as shown in Table C.6. Therefore, to ensure convergence within a reasonable time frame period, $M = 5$ in all experiments.

ω_f and ω_g are parameters of kernel function to regress policy's and disturbance's latent function (f and g , respectively). In all experiments, Radial Basis Function (RBF) kernel ($k(x, y) = \exp\{-(\|x - y\|^2)/(2\omega^2)\}$), which is most commonly used kernel in GP regression (Rasmussen, 2003), is employed for all GPs. Each parameter is initialized using the maximum and the minimum of state as: $|\max(\mathbf{S}) - \min(\mathbf{S})|$.

The initial mean of disturbance level (μ_0) and the positive variational parameter (Λ) are related to action variation of human demonstrations. In all experiments, $\Lambda = \text{diag}\{\lambda_n\}_{n=1}^N$ is initialized as $\lambda_n = 1/2$. In addition, μ_0 is initialized with variance of actions as: $\text{var}(\mathbf{a}^*) \times 0.01$.

To analyze sensitivity to hyperparameters, one-at-a-time parameter sen-

Table D.7: MHGP-BDI hyperparameters sensitivity analysis results. The demonstration success probability of each learning model is measured for entire learning iteration with one learning trial. The test execution performance of policy application is measured as the task success probability by conducting one learning trial and testing each final learned policy 100 times. Success rate for all demonstrations are 100 %.

Parameters	Initial Value	Test Execution Performance
M	2	99%
	5	100%
	10	100%
ω_f	$ \max(\mathbf{S}) - \min(\mathbf{S}) \times 0.1$	100%
	$ \max(\mathbf{S}) - \min(\mathbf{S}) $	100%
	$ \max(\mathbf{S}) - \min(\mathbf{S}) \times 10$	70%
μ_0	$\text{var}(\mathbf{a}^*) \times 0.001$	100%
	$\text{var}(\mathbf{a}^*) \times 0.01$	100%
	$\text{var}(\mathbf{a}^*) \times 0.1$	100%

sitivity (Hamby, 1994) is employed, in which the demonstration success rate and the test execution performance are measured in the wide version of wall-avoidance simulation task with varying one parameter at a time while holding the others fixed. Note, to simplify analysis, β and Λ are initialized as $\beta = 100$ and $\lambda_n = 1/2$ in all experiments. As described in Table D.7, MHGP-BDI is robust to a wide range of hyperparameters.

Appendix D.2. Other GP-based comparisons

The hyperparameter of other GP-based comparisons are described in Table D.8.

Table D.8: Hyperparameters of other GP-based comparisons. Since these methods employ the GP model, parameters are selected the same as in MHGP-BDI, but if the parameters are not available in the implementation, it is annotated as N/A.

Learning Models	Hyperparameters					
	M	β	ω_f	ω_g	μ_0	λ_n
UGP-BC	1	N/A	$ \max(\mathbf{S}) - \min(\mathbf{S}) $	N/A	N/A	N/A
UGP-BDI	1	N/A	$ \max(\mathbf{S}) - \min(\mathbf{S}) $	N/A	N/A	N/A
UHGP-BDI	1	N/A	$ \max(\mathbf{S}) - \min(\mathbf{S}) $	$\text{var}(\mathbf{a}^*) \times 0.01$	1/2	
MGP-BC	5	100	$ \max(\mathbf{S}) - \min(\mathbf{S}) $	N/A	N/A	N/A
MGP-BDI	5	100	$ \max(\mathbf{S}) - \min(\mathbf{S}) $	N/A	N/A	N/A

Appendix D.3. Neural networks-based comparisons

The hyperparameter of neural networks-based comparisons are described in Table D.9 and Table D.10. These hyperparameters are set based on original papers ([Laskey et al., 2017](#); [Ren et al., 2020](#)), but some parameters are tuned to improve performance in our domain.

Table D.9: Hyperparameters of BC and DART.

Hyperparameter	Value
optimizer	Adam
learning rate	1×10^{-2}
weight decay	1×10^{-5}
number of hidden layers	2
number of hidden units per layer	64
number of sample per minibatch	128
activation function	Tanh

Table D.10: Hyperparameters of CVAE-BC.

Hyperparameter	Value
optimizer	Adam
learning rate	1×10^{-3}
weight decay	1×10^{-5}
number of hidden layers	2
number of hidden units per layer	64
number of sample per minibatch	64
activation function	ReLU
number of latent dimension	5

References

- Argall, B.D., Chernova, S., Veloso, M., Browning, B., 2009. A survey of robot learning from demonstration. *Robotics and autonomous systems* 57, 469–483.

- Bain, M., Sammut, C., 1995. A framework for behavioural cloning., in: Machine Intelligence, pp. 103–129.
- Billard, A., Calinon, S., Dillmann, R., Schaal, S., 2008. Robot programming by demonstration, in: Springer Handbook of Robotics. Springer, pp. 1371–1394.
- Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L.D., Monfort, M., Muller, U., Zhang, J., et al., 2016. End to end learning for self-driving cars, in: Neural Information Processing Systems. Deep Learning Symposium.
- Brown, D.S., Goo, W., Niekum, S., 2019. Better-than-demonstrator imitation learning via automatically-ranked demonstrations, in: Proceedings of the Conference on Robot Learning, pp. 330–359.
- Calinon, S., 2016. A tutorial on task-parameterized movement learning and retrieval. Intelligent service robotics 9, 1–29.
- Chen, L., Paleja, R., Gombolay, M., 2020. Learning from suboptimal demonstration via self-supervised reward regression, in: Proceedings of the Conference on Robot Learning, pp. 1262–1277.
- Cheng, P., Wang, H., Stojanovic, V., He, S., Shi, K., Luan, X., Liu, F., Sun, C., 2021. Asynchronous fault detection observer for 2-d Markov jump systems. IEEE Transactions on Cybernetics , 1–12.
- Cutkosky, M.R., Howe, R.D., 1990. Human grasp choice and robotic grasp analysis, in: Dextrous robot hands. Springer, pp. 5–31.

- Dadhich, S., Bodin, U., Andersson, U., 2016. Key challenges in automation of earth-moving machines. *Automation in Construction* 68, 212–222.
- Duchaine, V., Gosselin, C.M., 2007. General model of human-robot cooperation using a novel velocity based variable impedance control, in: Second Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (WHC'07), pp. 446–451.
- Dyrstad, J.S., Ruud Øye, E., Stahl, A., Reidar Mathiassen, J., 2018. Teaching a robot to grasp real fish by imitation learning from a human supervisor in virtual reality, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 7185–7192.
- Fikes, R.E., Nilsson, N.J., 1971. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial intelligence* 2, 189–208.
- Fox, R., Berenstein, R., Stoica, I., Goldberg, K., 2019. Multi-task hierarchical imitation learning for home automation, in: IEEE International Conference on Automation Science and Engineering, IEEE. pp. 1–8.
- Giusti, A., Guzzi, J., Cireşan, D.C., He, F.L., Rodríguez, J.P., Fontana, F., Faessler, M., Forster, C., Schmidhuber, J., Caro, G.D., Scaramuzza, D., Gambardella, L.M., 2016. A machine learning approach to visual perception of forest trails for mobile robots. *IEEE Robotics and Automation Letters* 1, 661–667.
- Hamby, D.M., 1994. A review of techniques for parameter sensitivity analysis of environmental models. *Environmental monitoring and assessment* 32, 135–154.

- Hsiao, F.I., Kuo, J.H., Sun, M., 2019. Learning a multi-modal policy via imitating demonstrations with mixed behaviors. arXiv preprint arXiv:1903.10304 .
- Huang, Y., Rozo, L., Silvério, J., Caldwell, D.G., 2019. Kernelized movement primitives. *The International Journal of Robotics Research* 38, 833–852.
- Ijspeert, A.J., Nakanishi, J., Hoffmann, H., Pastor, P., Schaal, S., 2013. Dynamical movement primitives: learning attractor models for motor behaviors. *Neural computation* 25, 328–373.
- Jourdan, I., Dutson, E., Garcia, A., Vleugels, T., Leroy, J., Mutter, D., Marescaux, J., 2004. Stereoscopic vision provides a significant advantage for precision robotic laparoscopy. *Journal of British Surgery* 91, 879–885.
- Khansari-Zadeh, S.M., Billard, A., 2011. Learning stable nonlinear dynamical systems with Gaussian mixture models. *IEEE Transactions on Robotics* 27, 943–957.
- Kingma, D.P., Welling, M., 2013. Auto-encoding variational Bayes. *International Conference on Learning Representations* .
- Kuindersma, S., Deits, R., Fallon, M., Valenzuela, A., Dai, H., Permenter, F., Koolen, T., Marion, P., Tedrake, R., 2016. Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot. *Autonomous robots* 40, 429–455.
- Kyrarini, M., Haseeb, M.A., Ristić-Durrant, D., Gräser, A., 2019. Robot learning of industrial assembly task via human demonstrations. *Autonomous Robots* 43, 239–257.

- Laskey, M., Lee, J., Fox, R., Dragan, A., Goldberg, K., 2017. DART: Noise injection for robust imitation learning, in: Proceedings of the Conference on Robot Learning, pp. 143–156.
- Lazaro-Gredilla, M., Titsias, M., 2011. Variational heteroscedastic Gaussian process regression, in: Proceedings of the International Conference on Machine Learning, pp. 841–848.
- Lázaro-Gredilla, M., Van Vaerenbergh, S., Lawrence, N.D., 2012. Overlapping mixtures of Gaussian processes for the data association problem. Pattern recognition 45, 1386–1395.
- Le, Q., Sarlós, T., Smola, A., 2013. Fastfood: Approximating kernel expansions in loglinear time, in: Proceedings of the International Conference on Machine Learning, pp. 244–252.
- Levine, S., Abbeel, P., 2014. Learning neural network policies with guided policy search under unknown dynamics., in: Advances in Neural Information Processing Systems, pp. 1071–1079.
- Levine, S., Pastor, P., Krizhevsky, A., Ibarz, J., Quillen, D., 2018. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. The International journal of robotics research 37, 421–436.
- Liu, Q., Pierce, D.A., 1994. A note on Gauss—Hermite quadrature. Biometrika 81, 624–629.
- Nagengast, A.J., Braun, D.A., Wolpert, D.M., 2011. Risk sensitivity in a

- motor task with speed-accuracy trade-off. *Journal of neurophysiology* 105, 2668–2674.
- Napier, J.R., 1956. The prehensile movements of the human hand. *The Journal of bone and joint surgery*. 38, 902–913.
- Oh, H., Sasaki, H., Michael, B., Matsubara, T., 2021. Bayesian Disturbance Injection: Robust imitation learning of flexible policies, in: *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 8629–8635.
- Opper, M., Archambeau, C., 2009. The variational Gaussian approximation revisited. *Neural computation* 21, 786–792.
- Osa, T., Pajarinen, J., Neumann, G., Bagnell, J.A., Abbeel, P., Peters, J., 2018. An algorithmic perspective on imitation learning. *Foundations and Trends® in Robotics* 7, 1–179.
- Parisi, G., 1988. Statistical field theory. Addison-Wesley.
- Pereira, M., Fan, D.D., An, G.N., Theodorou, E., 2018. MPC-inspired neural network policies for sequential decision making. *arXiv preprint arXiv:1802.05803* .
- Pomerleau, D.A., 1989. ALVINN: An autonomous land vehicle in a neural network, in: *Advances in Neural Information Processing Systems*, p. 305–313.
- Rahmatizadeh, R., Abolghasemi, P., Bölöni, L., Levine, S., 2018. Vision-based multi-task manipulation for inexpensive robots using end-to-end

- learning from demonstration, in: Proceedings of the IEEE International Conference on Robotics and Automation, IEEE. pp. 3758–3765.
- Rasmussen, C.E., 2003. Gaussian processes in machine learning, in: Summer School on Machine Learning, Springer. pp. 63–71.
- Ren, A.Z., Veer, S., Majumdar, A., 2020. Generalization guarantees for imitation learning, in: Proceedings of the Conference on Robot Learning, pp. 1426–1442.
- Ross, J.C., Dy, J.G., 2013. Nonparametric mixture of Gaussian processes with constraints, in: Proceedings of the International Conference on Machine Learning, pp. 1346–1354.
- Ross, S., Bagnell, D., 2010. Efficient reductions for imitation learning, in: Proceedings of the International Conference on Artificial Intelligence and Statistics, pp. 661–668.
- Ross, S., Gordon, G., Bagnell, D., 2011. A reduction of imitation learning and structured prediction to no-regret online learning, in: Proceedings of the International Conference on Artificial Intelligence and Statistics, pp. 627–635.
- Sasaki, H., Matsubara, T., 2019. Multimodal policy search using overlapping mixtures of sparse Gaussian process prior, in: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 2433–2439.
- Sastry, S., Bodson, M., 2011. Adaptive control: stability, convergence and robustness. Courier Corporation.

- Schaal, S., Peters, J., Nakanishi, J., Ijspeert, A., 2005. Learning movement primitives, in: Robotics research. The International Symposium, Springer. pp. 561–572.
- Sethuraman, J., 1994. A constructive definition of Dirichlet priors. *Statistica sinica* , 639–650.
- Siemens, 2017. Robot learning challenge. URL: <https://new.siemens.com/us/en/company/fairs-events/robot-learning.html>. last accessed 27 February 2021.
- Sohn, K., Lee, H., Yan, X., 2015. Learning structured output representation using deep conditional generative models. Advances in Neural Information Processing Systems 28.
- Tahara, H., Sasaki, H., Oh, H., Michael, B., Matsubara, T., 2022. Disturbance-injected Robust Imitation Learning with Task Achievement, in: Intl. Conf. on Robotics and Automation, pp. 2466–2472.
- Wang, Y., Beltran-Hernandez, C.C., Wan, W., Harada, K., 2021. Hybrid trajectory and force learning of complex assembly tasks: A combined learning framework. *IEEE Access* 9, 60175–60186.
- Wickelgren, W.A., 1977. Speed-accuracy tradeoff and information processing dynamics. *Acta psychologica* 41, 67–85.
- Xin, X., Tu, Y., Stojanovic, V., Wang, H., Shi, K., He, S., Pan, T., 2022. Online reinforcement learning multiplayer non-zero sum games of continuous-time Markov jump linear systems. *Applied Mathematics and Computation* 412, 126537.

- Xu, D., Martín-Martín, R., Huang, D.A., Zhu, Y., Savarese, S., Fei-Fei, L.F., 2019. Regression planning networks. *Advances in Neural Information Processing Systems* 32.
- Zhang, T., McCarthy, Z., Jow, O., Lee, D., Chen, X., Goldberg, K., Abbeel, P., 2018. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation, in: *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1–8.
- Zhuang, Z., Tao, H., Chen, Y., Stojanovic, V., Paszke, W., 2022. Iterative learning control for repetitive tasks with randomly varying trial lengths using successive projection. *International Journal of Adaptive Control and Signal Processing* 36, 1196–1215.