Preference-Based Bayesian Optimization in High Dimensions with Human Feedback

Myra Cheng MYRACHENG@CALTECH.EDU
Ellen Novoseller ENOVOSELLER@CALTECH.EDU
Maegan Tucker MTUCKER@CALTECH.EDU
Richard Cheng RCHENG@CALTECH.EDU
Joel Burdick JBURDICK@CALTECH.EDU
Yisong Yue YYUE@CALTECH.EDU
California Institute of Technology, Pasadena, CA 91125

Abstract

Existing preference-based learning methods are restricted to low-dimensional spaces due to computational limitations. However, many applications of preference-based learning, such as optimizing parameters of robotic systems, are high-dimensional problems. To address this issue, we present LineCoSpar, a human-in-the-loop framework that enables preference-based optimization in high dimensions by iteratively exploring one-dimensional subspaces. After verifying its performance and sample-efficiency in simulation, we apply the algorithm to two studies with real human feedback. We use LineCoSpar to learn a stable controller for the cartpole problem and optimize walking gaits in real-world exoskeleton experiments.

1. Introduction

Human-in-the-loop learning techniques have demonstrated significant potential in human-robot interaction tasks (Bajcsy et al., 2017; Christen et al., 2019; Cremer et al., 2019; Zhang et al., 2017), such as improving robotic assistive devices and tailoring them to individual users. To learn optimal parameters in these settings with subjective human feedback, we rely on users' relative preferences, which are more reliable than numerical scores (Basu et al., 2017; Joachims et al., 2005; Chapelle et al., 2012). Existing online preference-based learning methods are unfortunately limited to low-dimensional spaces due to computational constraints. For example, previous real-world demonstrations of preference-based learning are in two dimensions (Tucker et al., 2020b) or rely on domain knowledge to narrow the search space before performing online learning (Thatte et al., 2018). However, many practical human-robot interaction problems are in high-dimensional optimization spaces, while domain knowledge is difficult to obtain.

To bridge the gap, we present LineCoSpar, a human preference-based learning approach that integrates existing techniques for preference learning and high-dimensional optimization into a unified framework. LineCoSpar relies on preference feedback to iteratively explore one-dimensional subspaces. We demonstrate in simulation that LineCoSpar exhibits sample-efficient convergence to user-preferred actions in high-dimensional spaces. We deploy this algorithm in studies with human users, first demonstrating that it learns stable

Algorithm 1 LineCoSpar

```
1: procedure LineCoSpar(Utility prior parameters; $m$ = granularity of discretization; $n$ = number of actions per iteration)
2:     $\mathcal{D} = \emptyset$, $\mathcal{W} = \emptyset$                ▷ $\mathcal{D}$: preference data, $\mathcal{W}$: actions in $\mathcal{D}$
3:     Set $\boldsymbol{p}_1$ to uniformly-random action
4:     for $t = 1, 2, \ldots, T$ do
5:         $\mathcal{L}_t$ = random line through $\boldsymbol{p}_t$, discretized via $m$
6:         $\mathcal{V}_t = \mathcal{L}_t \cup \mathcal{W}$                ▷ Points over which to update posterior
7:         $(\boldsymbol{\mu}_t, \Sigma_t)$ = posterior over $\mathcal{V}_t$ given $\mathcal{D}$
8:         for $j = 1, 2, \ldots, n$ do
9:             Sample utility function $f_{t_j} \sim \mathcal{N}(\boldsymbol{\mu}_t, \Sigma_t)$
10:            Execute action $\boldsymbol{a}_{t_j} = \text{argmax}_{\boldsymbol{a} \in \mathcal{V}_t} f_{t_j}(\boldsymbol{a})$
11:        Add $\binom{n}{2}$ pairwise preferences among $\boldsymbol{a} \in \{\boldsymbol{a}_{t_j}\}_{1 \leq j \leq n}$ to $\mathcal{D}$
12:        Add coactive feedback $\{\boldsymbol{a}'_t\}$ to $\mathcal{D}$
13:        Set $\mathcal{W} = \mathcal{W} \cup \{\boldsymbol{a}_{t_j}\}_{1 \leq j \leq n} \cup \{\boldsymbol{a}'_t\}$                ▷ Update actions in $\mathcal{D}$
14:        Set $\boldsymbol{p}_{t+1} = \text{argmax}_{\boldsymbol{a} \in \mathcal{V}_t} \mu_t(\boldsymbol{a})$
```

controllers in the four-parameter cartpole environment via human preferences. Then, we apply LineCoSpar to the real-world problem of exoskeleton gait optimization with six parameters to learn individualized walking gaits that maximize the user's comfort.

2. The Learning Algorithm

The LineCoSpar algorithm (Alg. 1) learns a Bayesian model over the user's preferences in a high-dimensional space. To learn from preferences, we adopt the dueling bandit setting (Sui et al., 2017, 2018; Yue et al., 2012), in which the algorithm selects actions and receives relative preferences between them. The procedure, which is based on Thompson sampling, iterates through: 1) updating a Bayesian posterior over the actions' utilities given the data, 2) sampling utility functions from the posterior, 3) executing the actions that maximize the sampled utility functions, and 4) observing preferences among the executed actions.

Drawing inspiration from the LineBO algorithm (Kirschner et al., 2019), LineCoSpar exploits low-dimensional structure in the search space by sequentially considering one-dimensional subspaces from which to sample actions. This allows the algorithm to maintain its Bayesian preference relation function over a subset of the action space in each iteration.

A related previous work is CoSpar, which finds user-preferred parameters across one and two dimensions (Tucker et al., 2020b). Compared to CoSpar, LineCoSpar learns the model posterior much more efficiently, and it can be scaled to high dimensions.

2.1 Modeling Utilities Using Pairwise Preference Data

LineCoSpar uses pairwise comparisons to learn a Bayesian posterior over the utilities of different actions to the user. This is based on the Gaussian process preference model in Chu and Ghahramani (2005). Let $\mathcal{A} \subset \mathbb{R}^d$ be the set of all possible actions. We assume that each action $\boldsymbol{a} \in \mathcal{A}$ has a latent utility $f(\boldsymbol{a})$ to the user. Throughout the learning process, LineCoSpar maintains a dataset of all feedback from the user, $\mathcal{D} = \{\boldsymbol{a}_{k_1} \succ \boldsymbol{a}_{k_2} \mid k = 1, \ldots, N\}$, consisting of N preferences, where $\boldsymbol{a}_{k_1} \succ \boldsymbol{a}_{k_2}$ indicates that the user prefers action

$\boldsymbol{a}_{k_1}$ to action $\boldsymbol{a}_{k_2}$. In iteration t of the algorithm, we consider a subset of the actions $\mathcal{V}_t \subset \mathcal{A}$ with cardinality $V_t := |\mathcal{V}_t|$ and use the preference data $\mathcal{D}$ to update the posterior utilities of the actions in $\mathcal{V}_t$ (we define $\mathcal{V}_t$ in 2.2). Defining $\boldsymbol{f} = [f(\boldsymbol{a}_{t_1}), f(\boldsymbol{a}_{t_2}), \ldots, f(\boldsymbol{a}_{t_{V_t}})]^T \in \mathbb{R}^{V_t}$, where $\boldsymbol{a}_{t_i}$ is the i^{th} action in $\mathcal{V}_t$, the utilities $\boldsymbol{f}$ have posterior: $\mathcal{P}(\boldsymbol{f}|\mathcal{D}) \propto \mathcal{P}(\mathcal{D}|\boldsymbol{f})\mathcal{P}(\boldsymbol{f})$.

We place a Gaussian process prior over the utilities $\boldsymbol{f}$ of actions in $\mathcal{V}_t$, $\mathcal{P}(\boldsymbol{f}) = \frac{1}{(2\pi)^{V_t/2}|\Sigma_t^{\text{pr}}|^{1/2}} \exp\left(-\frac{1}{2}\boldsymbol{f}^T[\Sigma_t^{\text{pr}}]^{-1}\boldsymbol{f}\right)$, where $\Sigma_t^{\text{pr}} \in \mathbb{R}^{V_t \times V_t}$, $[\Sigma_t^{\text{pr}}]_{ij} = \mathcal{K}(\boldsymbol{a}_{t_i}, \boldsymbol{a}_{t_j})$, and $\mathcal{K}$ is a kernel. In practice, we use the squared exponential kernel. To compute the likelihood $\mathcal{P}(\mathcal{D}|\boldsymbol{f})$, we assume that the preferences may be corrupted by noise, such that: $\mathcal{P}(\boldsymbol{a}_{k_1} \succ \boldsymbol{a}_{k_2}|\boldsymbol{f}) = g\left(\frac{f(\boldsymbol{a}_{k_1}) - f(\boldsymbol{a}_{k_2})}{c}\right)$, where $g(\cdot) \in [0,1]$ is a monotonically-increasing link function, and $c > 0$ is a hyperparameter quantifying the amount of noise. While previous work uses the standard normal cumulative distribution function for g (Tucker et al., 2020b; Chu and Ghahramani, 2005), we empirically found that using the heavier-tailed sigmoid distribution, $g_{\text{sig}}(x) := \sigma(x) = \frac{1}{1+e^{-x}}$, as the link function improves performance. Thus, the full likelihood expression is: $\mathcal{P}(\mathcal{D}|\boldsymbol{f}) = \prod_{k=1}^{N} g_{\text{sig}}\left(\frac{f(\boldsymbol{a}_{k_1}) - f(\boldsymbol{a}_{k_2})}{c}\right)$. The posterior is estimated via the Laplace approximation (Chu and Ghahramani, 2005), yielding a multivariate Gaussian distribution, $\mathcal{N}(\boldsymbol{\mu}_t, \Sigma_t)$.

2.2 High-dimensional Optimization via Low-dimensional Subspaces

Existing preference-based approaches, including CoSpar, optimize over the action space $\mathcal{A}$ by discretizing the entire space before beginning the learning process. This results in m^d combinations from m uniformly-spaced points in each of the d dimensions of $\mathcal{A}$. Thus, the cardinality of this set is $A := |\mathcal{A}| = m^d$; larger m enables finer-grained search at a higher computational cost. The Bayesian preference model is jointly maintained and updated over all A points during each iteration. This is intractable for higher d since computing the posterior over A points involves expensive matrix operations, such as inverting $\Sigma_t^{\text{pr}}, \Sigma_t \in \mathbb{R}^{A \times A}$.

Inspired by Kirschner et al. (2019), LineCoSpar overcomes this intractibility by iteratively considering one-dimensional subspaces (lines), rather than the full discretized action space. In each iteration t, LineCoSpar selects uniformly-spaced points along a new random line $\mathcal{L}_t$, which is determined by a uniformly-random direction and the action $\boldsymbol{p}_t$ that maximizes the posterior mean. Including $\boldsymbol{p}_t$ in the subspace encourages exploration of higher-utility areas. The posterior $\mathcal{P}(\mathcal{D}|\boldsymbol{f})$ is calculated over $\mathcal{V}_t := \mathcal{L}_t \cup \mathcal{W}$, where $\mathcal{W}$ is the set of actions for which $\mathcal{D}$ contains preference feedback. This approach reduces the model's covariance matrices $\Sigma_t^{\text{pr}}, \Sigma_t$ from size $A \times A$ to $V_t \times V_t$. Rather than growing exponentially in d, which is impractical for online learning, LineCoSpar's complexity is constant in the dimension d and linear in the number of iterations T. Since queries are expensive in many human-in-the-loop robotics settings, T is typically low.

2.3 Sampling from and Updating the Posterior

Utilities are learned using the SelfSparring (Sui et al., 2017) approach to Thompson sampling. In each iteration, given the preferences in $\mathcal{D}$, we calculate the posterior $\mathcal{N}(\boldsymbol{\mu}_t, \Sigma_t)$ of the utilities $\boldsymbol{f}$ over the points in $\mathcal{V}_t = \mathcal{L}_t \cup \mathcal{W}$. The algorithm samples n utility functions $\{f_{t_j}\}_{1 \leq j \leq n}$ from the posterior, each of which assigns utilities to the actions in $\mathcal{V}_t$, and then executes the n actions $\{\boldsymbol{a}_{t_j}\}_{1 \leq j \leq n}$ that maximize each f_{t_j}. The user provides $\binom{n}{2}$ pairwise preferences (or indicates "no preference" between two actions) among the n actions, which are added to $\mathcal{D}$. (For $n = 1$, a preference is obtained between the actions $\boldsymbol{a}_t$ and $\boldsymbol{a}_{t-1}$.) In addition to pairwise preferences, the algorithm also uses coactive feedback (Shivaswamy and Joachims, 2012, 2015), where after each time the algorithm selects an action, the user can suggest an improved action. Specifically, for each action $\boldsymbol{a}_t$, the user can suggest the dimension, direction (higher or lower), and degree in which to change $\boldsymbol{a}_t$. The resulting suggested action $\boldsymbol{a}'_t$ is added to $\mathcal{W}$, and the feedback is added to $\mathcal{D}$ as $\boldsymbol{a}'_t \succ \boldsymbol{a}_t$. The new feedback data in $\mathcal{D}$ is used to update the posterior over $\mathcal{V}_t$ in subsequent iterations.

3. Empirical Evaluation of LineCoSpar

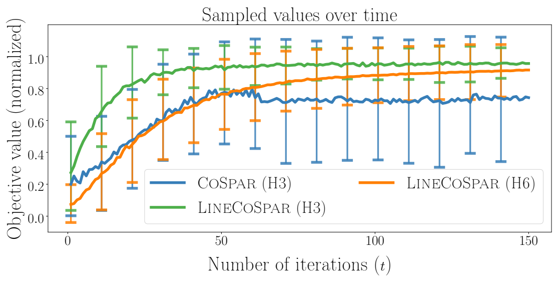

Figure 1: Standard benchmarks. Mean objective value ± SD of sampled actions using H3 and H6, averaged over 100 runs with $n = 3$. The sampled actions quickly converge to higher objective values with LineCoSpar. CoSpar is intractable in a 6-dimensional space.

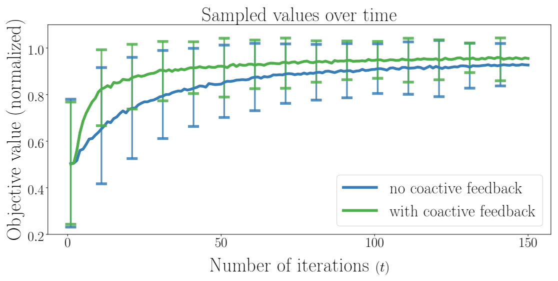

Figure 2: Coactive feedback. Mean objective value ± SD of sampled actions using random six-dimensional functions, averaged over 1000 runs with $n = 1$. Sampled actions reach high objective values in relatively few iterations, and coactive feedback accelerates this process.

3.1 Synthetic Functions

We validate the performance of LineCoSpar in simulation using both standard Bayesian optimization benchmarks and randomly-generated polynomials.[1] Preferences are generated as $\boldsymbol{a}_{k_1} \succ \boldsymbol{a}_{k_2}$ if $f(\boldsymbol{a}_{k_1}) > f(\boldsymbol{a}_{k_2})$, where f is the objective function. The true objective values f are invisible to the algorithm, which observes only the preference dataset $\mathcal{D}$. We show that LineCoSpar is sample-efficient, converges to sampling high-valued actions, and learns from both preference and coactive feedback.

First, we evaluate the performance of LineCoSpar on the standard Hartmann3 (H3) and Hartmann6 (H6) benchmarks (three and six dimensions respectively). Compared to CoSpar, LineCoSpar converges to sampling actions with higher objective values at a faster rate (Fig. 1). Thus, LineCoSpar not only enables higher-dimensional optimization, but also improves speed and accuracy of learning.

We also test LineCoSpar using randomly-generated d-dimensional polynomials (for $d = 6$) as objective functions: $p(\boldsymbol{a}) = \sum_{i=1}^{d} \alpha_i \sum_{j=1}^{d} \beta_j a_j$, where a_j denotes the j^{th} element of $\boldsymbol{a}$, and all α_i, β_i are sampled independently from the uniform distribution $\mathcal{U}(-1, 1)$.

Coactive feedback is simulated for each sampled action $\boldsymbol{a}_t$ by finding an action $\boldsymbol{a}'_t$, differing from $\boldsymbol{a}_t$ along only one dimension, such that $f(\boldsymbol{a}'_t) > f(\boldsymbol{a}_t)$. The action $\boldsymbol{a}'_t$ is determined by randomly choosing a dimension i ($1 \leq i \leq d$) and direction (positive or negative), and

1. The code can be found at `github.com/myracheng/linecospar`. All experiments use a kernel with lengthscale 0.15 in every dimension, signal variance 1e−4, noise variance 1e−5, and preference noise 0.005.

taking a step from $\boldsymbol{a}_t$ along this vector. If the resulting action $\boldsymbol{a}'_t$ has a higher objective value, it is added to the dataset $\mathcal{D}$ as $\boldsymbol{a}'_t \succ \boldsymbol{a}_t$. We demonstrate high sample efficiency, which is critical for human-in-the-loop learning, in which each query can be expensive. In particular, LineCoSpar consistently identifies high-valued actions in ≈ 20 iterations (Fig. 2).

3.2 User Studies

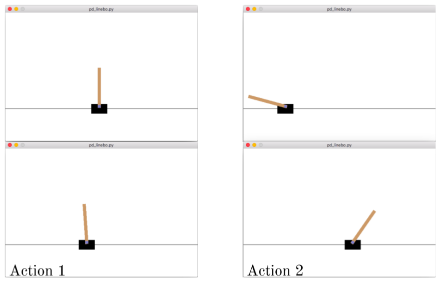

Figure 3: Cartpole setup. In each iteration, the subject indicates a preference between the cartpole behaviors resulting from different actions (controller parameters). Above, Action 1 (left) has higher reward than Action 2 (right).

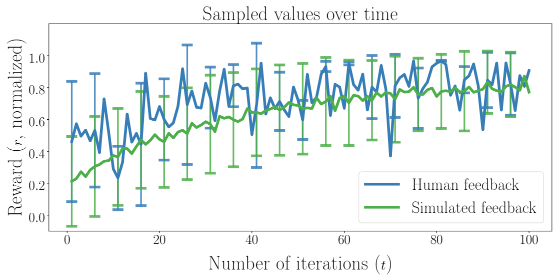

Figure 4: Simulated versus human feedback in the cartpole environment. Mean reward ± SD, averaged over 100 runs in simulation (green) and over three human subjects (blue). The sampled actions converge to similar reward values despite inconsistencies in the preferences.

3.2.1 Cartpole Environment

The cartpole environment, an instance of the classic inverted pendulum problem, features a pole attached to a cart moving along a horizontal axis (Brockman et al., 2016). The system can be stabilized by a proportional-derivative (PD) controller with four tunable parameters, which are the proportional and derivative gains for position and velocity. To learn an optimal controller, LineCoSpar searches over the action space consisting of these four parameters. We define the reward r_i as the number of timesteps that the cartpole stays upright within a certain angle and position with the controller parameters $\boldsymbol{a}_i$. In simulation, at iteration t of LineCoSpar, the algorithm receives preference $\boldsymbol{a}_{t_1} \succ \boldsymbol{a}_{t_2}$ if $r_{t_1} > r_{t_2}$.

We also use LineCoSpar to learn the cartpole controller from preference feedback given by humans. In this setting, the subject is instructed that the cartpole should be stable, upright, and centered. In iteration t, after observing the different behaviors of the cartpole resulting from $\boldsymbol{a}_{t_1}$ and $\boldsymbol{a}_{t_2}$, the subject indicates a pairwise preference between the two actions (Fig. 3). Across the three subjects evaluated, 25% of their preferences are inconsistent with the synthetic reward function defined above (i.e., the human's preference is $\boldsymbol{a}_{t_1} \succ \boldsymbol{a}_{t_2}$ but $r_{t_1} < r_{t_2}$). The inconsistency rates are 32%, 24%, and 17% for each subject respectively. One possible reason for these deviations is that it is challenging to distinguish between actions with similar utilities; the normalized average difference in reward $|r_{t_1} - r_{t_2}|$ is 0.3 across all presented pairs and 0.1 across the inconsistent pairs. Also, the reward function may not fully capture the human's notion of stability. Despite these inconsistencies between human and simulated feedback, LineCoSpar learns to sample high-reward actions at comparable rates (Fig. 4). This suggests that the algorithm is robust to noisy user preferences.

The cartpole environment enables us to quantitatively validate the performance of the algorithm as it learns from human feedback, using the reward r as a ground-truth metric.

This evidence that LineCoSpar learns to sample high-utility actions based on human preferences motivates the next section of human trials with an exoskeleton.

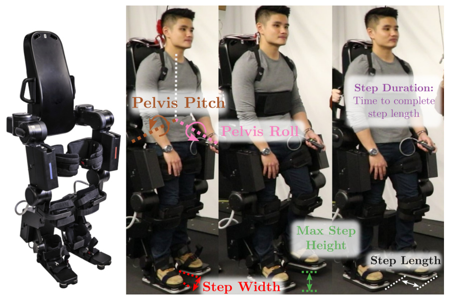

Figure 5: Lower-body exoskeleton. We use LineCoSpar to optimize over the six parameters (labeled above) of exoskeleton walking gaits.

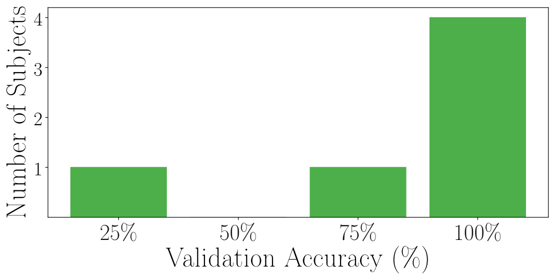

Figure 6: Gait validation. The x-axis is the percent of validation trials in which the subject prefers $\boldsymbol{a}_{\max}$ over a random action. Five of six subjects predominantly ($\geq 75\%$) prefer $\boldsymbol{a}_{\max}$.

3.2.2 Lower-Body Exoskeleton

We deploy LineCoSpar with humans wearing a lower-body exoskeleton. The exoskeleton is placed around existing limbs and helps those with mobility impairments regain their ability to walk. Knowledge about user-preferred walking gaits is limited due to the high cost of clinical trials, a barrier which this algorithm helps to lower by improving learning efficiency. Toward the goal of maximizing user comfort, LineCoSpar learns an individualized walking gait by optimizing six parameters (Fig. 5). Each iteration takes only a few seconds of computational time. At each iteration, the algorithm prompts the exoskeleton user to test a gait, indicate a preference compared to the previously-tested gait, and give coactive feedback. See Tucker et al. (2020a) for details on the experimental setup.[2]

Let $\boldsymbol{a}_{\max}$ be the action maximizing the final posterior mean after T iterations, i.e., $\boldsymbol{a}_{\max} := \text{argmax}_{\boldsymbol{a} \in \mathcal{V}_T} \mu_T(\boldsymbol{a})$. As there is no objective function to quantify gait comfort for each individual, we validate that the subjects predominantly prefer the predicted optimal gait $\boldsymbol{a}_{\max}$ over four other randomly-selected gaits (Fig. 6). For four of the six subjects, all validation preferences match the posterior, while the other subjects match three and one of the four preferences, respectively.

4. Future Directions

We demonstrate that LineCoSpar performs sample-efficient optimization in a variety of high-dimensional settings. This enables finding optima personalized to individual users, and more broadly, understanding the underlying human utility functions. For example, in the exoskeleton setting, LineCoSpar can give insight into the similarities and differences among different users' gait preferences; such knowledge paves the way for developing universally-preferred gaits.

In the future, we hope to explore the noise in human preferences and its influence on the learning process, as well as to gain a theoretical understanding of how factors like preference quantity affect the convergence rate. Modeling how users give coactive feedback is also an interesting open problem.

2. A video of the experimental results can be found at `vimeo.com/394608113`.

References

Andrea Bajcsy, Dylan P Losey, Marcia K O'Malley, and Anca D Dragan. Learning robot objectives from physical human interaction. In *Conf. on Robot Learning*, pages 217–226, 2017.

Chandrayee Basu, Qian Yang, David Hungerman, Mukesh Sinahal, and Anca D Dragan. Do you want your autonomous car to drive like you? In *Int. Conf. on Human-Robot Interaction*, pages 417–425. IEEE, 2017.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016.

Olivier Chapelle, Thorsten Joachims, Filip Radlinski, and Yisong Yue. Large-scale validation and analysis of interleaved search evaluation. *ACM Transactions on Information Systems*, 30(1):6, 2012.

Sammy Christen, Stefan Stevšić, and Otmar Hilliges. Guided deep reinforcement learning of control policies for dexterous human-robot interaction. In *IEEE Int. Conf. on Robotics and Automation*, pages 2161–2167, 2019.

Wei Chu and Zoubin Ghahramani. Preference learning with Gaussian processes. In *Int. Conf. on Machine Learning*, pages 137–144. ACM, 2005.

Sven Cremer, Sumit Kumar Das, Indika B Wijayasinghe, Dan O Popa, and Frank L Lewis. Model-free online neuroadaptive controller with intent estimation for physical human–robot interaction. *IEEE Trans. on Robotics*, 2019.

Thorsten Joachims, L.A. Granka, B. Pan, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. In *SIGIR*, volume 5, pages 154–161, 2005.

Johannes Kirschner, Mojmir Mutny, Nicole Hiller, Rasmus Ischebeck, and Andreas Krause. Adaptive and safe Bayesian optimization in high dimensions via one-dimensional subspaces. In *Int. Conf. on Machine Learning*, pages 3429–3438, 2019.

Pannaga Shivaswamy and Thorsten Joachims. Online structured prediction via coactive learning. In *Int. Conf. on Machine Learning*, pages 59–66. Omnipress, 2012.

Pannaga Shivaswamy and Thorsten Joachims. Coactive learning. *Journal of Artificial Intelligence Research*, 53:1–40, 2015.

Yanan Sui, Vincent Zhuang, Joel W Burdick, and Yisong Yue. Multi-dueling bandits with dependent arms. In *Conf. on Uncertainty in Artificial Intelligence*, 2017.

Yanan Sui, Masrour Zoghi, Katja Hofmann, and Yisong Yue. Advancements in dueling bandits. In *IJCAI*, pages 5502–5510, 2018.

Nitish Thatte, Helei Duan, and Hartmut Geyer. A method for online optimization of lower limb assistive devices with high dimensional parameter spaces. In *Int. Conf. on Robotics and Automation*, pages 1–6. IEEE, 2018.

Maegan Tucker, Myra Cheng, Ellen Novoseller, Richard Cheng, Yisong Yue, Joel W. Burdick, and Aaron D. Ames. Human preference-based learning for high-dimensional optimization of exoskeleton walking gaits, 2020a.

Maegan Tucker, Ellen Novoseller, Claudia Kann, Yanan Sui, Yisong Yue, Joel Burdick, and Aaron D Ames. Preference-based learning for exoskeleton gait optimization. In *Int. Conf. on Robotics and Automation*. IEEE, 2020b.

Yisong Yue, Josef Broder, Robert Kleinberg, and Thorsten Joachims. The k-armed dueling bandits problem. *Journal of Computer and System Sciences*, 78(5):1538–1556, 2012.

Juanjuan Zhang, Pieter Fiers, Kirby A Witte, Rachel W Jackson, Katherine L Poggensee, Christopher G Atkeson, and Steven H Collins. Human-in-the-loop optimization of exoskeleton assistance during walking. *Science*, 356(6344):1280–1284, 2017.