Adaptive Incentive Design in Dynamic Principal-Agent Problem via Kernelized Bandits

Arghya Mallick, Anuj S. Vora, Sergio Grammatico, and Peyman Mohajerin Esfahani

Abstract. We consider the dynamic principal-agent problem under asymmetric information, wherein a principal sequentially designs contracts to incentivize an agent with unknown preferences and hidden actions. A fundamental bottleneck in the existing literature is the assumption of deterministic agent utility, which renders the principal's expected utility discontinuous and forces computationally intractable discretizations of the contract space. In this paper, we address this limitation by introducing a stochastic counterpart into the agent's utility model, capturing the inherent physical and behavioral variations in realistic subsystems. We formally prove that this stochastic formulation restores the continuity of the principal's expected utility. Leveraging this continuous geometric structure, we formulate the interaction as a structured multi-armed bandit problem subject to heteroscedastic noise. We propose a `Heteroscedastic GP-UCB` algorithm that utilizes a Neural Network (Arcsin) kernel, chosen to capture the non-stationary, sigmoidal geometry of the utility landscape. For an m-dimensional compact contract space, we establish a high-probability cumulative regret bound of $\mathcal{O}\left(\sqrt{T}(\log T)^{m+1}\right)$. Finally, we demonstrate the practical efficacy of our theoretical framework by formulating the Vehicle-to-Grid (V2G) incentive design problem, proving its equivalence to a dynamic principal-agent problem, and showing superior economic performance for grid aggregators.

1. Introduction

We study the classical principal-agent problem [21] in a dynamic setting. The broader significance of this problem—underscored by the 2016 Nobel Memorial Prize in Economic Sciences [37]—lies in its ability to model asymmetric information. Specifically, a principal sequentially designs contracts to incentivize a strategic agent whose actions are hidden (unobservable to the principal) but stochastically govern the task outcome and, consequently, the principal's payoff. Within the control community, such incentive design has historically been investigated through the lens of Stackelberg games [17, 16, 24], addressing applications ranging from demand response [40] to network congestion [5]. While recent works have made rich theoretical strides in adaptive incentive design [33, 29, 24], they frequently rely on restrictive assumptions that deviate from the classical setting. For instance, the approaches in [33] and [24] propose no-regret learning algorithms for dynamic Stackelberg games,

Date: August 18, 2026.

This work was partly funded by EU project DriVe2X (grant no. 101056934), and the ERC project TRUST (grant no. 949796). Arghya Mallick, and Sergio Grammatico are with Delft Center for Systems and Control, Delft University of Technology, Delft, The Netherlands. Email: {A.Mallick, s.grammatico}@tudelft.nl; Anuj S. Vora is with Jio Institute, Navi Mumbai, India. Email: anuj.vora@jioinstitute.edu.in; Peyman Mohajerin Esfahani is with the University of Toronto, Toronto, Canada. Email: P.MohajerinEsfahani@utoronto.ca.

but fundamentally assume that the leader (principal) can directly observe the follower's (agent's) actions at each round, or that the follower's utility is linearly parameterized.

In contrast to these idealized assumptions, we preserve the structural integrity of the original principal-agent problem, specifically the core challenge of the agent's unobservable actions (known as *moral hazard*). To navigate this, we cast the dynamic contract design problem as a structured multi-armed bandit problem [31, 9]. Bandit frameworks naturally accommodate sequential decision-making under partial feedback among the principal, the agent, and the environment [2, 42, 39, 27]. Recently, bandit-based algorithms have made foundational contributions to automated contract design on modern crowdsourcing platforms, such as *Amazon Mechanical Turk* [15, 14]. Beyond digital markets, this adaptive framework holds immense potential for decentralized cyber-physical systems, most notably Vehicle-to-Grid (V2G) technology [26]. V2G enables electric vehicles (EVs) to act as distributed energy storage, providing critical balancing services to the power grid. However, while the physical infrastructure is increasingly mature [7], widespread adoption is severely bottlenecked by misaligned economic incentives and unobservable user costs [36]. Motivated by these research gaps, we focus on developing computationally efficient algorithms while modeling a principal-agent-compatible incentive framework for the end users of 'V2G' technology.

Formally, in each round t, the principal posts a contract $x_t \in \mathcal{X} := [0,1]^m$, that specifies an incentive payment $(x_t)_i$ for the i-th outcome of the task, with $i \in [m] := \{1, \dots m\}$. An agent observes the contract, takes an action $a_t^\star \in \mathcal{A}$, and the outcome of the task is realized. The uncoupled utility (or payoff) function of the principal is $U^{\mathrm{P}} : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ and the utility function of the agent is $U^{\mathrm{A}} : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$. We assume the agent is rational and always takes an action that maximizes its utility U^{A}. For simplicity, dropping the time t as a subscript, we compactly formulate the problem as

$$a^\star(x) := \arg\max_{a \in \mathcal{A}} U^{\mathrm{A}}(x, a), \tag{1a}$$

$$x^\star := \arg\max_{x \in \mathcal{X}} \left\{ U^{\mathrm{PA}}(x) := U^{\mathrm{P}}\big(x, a^\star(x)\big) \right\}, \tag{1b}$$

where $a^\star(x)$ is the best response of the agent against the contract x, $U^{\mathrm{PA}} : \mathcal{X} \to \mathbb{R}$ is the principal's utility with embedded best response of the agent, and $x^\star$ denotes the optimal contract. The main technical challenge is that the principal does not explicitly know U^{PA} because of the agent's hidden action $a^\star$. Our goal is to learn $x^\star$ via online random feedback from the principal's utility U^{PA}. To quantify the sub-optimality of the principal's strategy in this sequential interaction, we use the traditional notion of cumulative regret as

$$R_T := T \cdot U^{\mathrm{PA}}(x^\star) - \sum_{t=1}^{T} U^{\mathrm{PA}}(x_t), \tag{2}$$

where $x^\star$ is as defined in (1b). We are interested in designing an algorithm to propose a sequence of $\{x_t\}_{t \le T}$ using the available feedback information from the environment so that we ensure the upper bound of R_T grows sub-linearly in T.

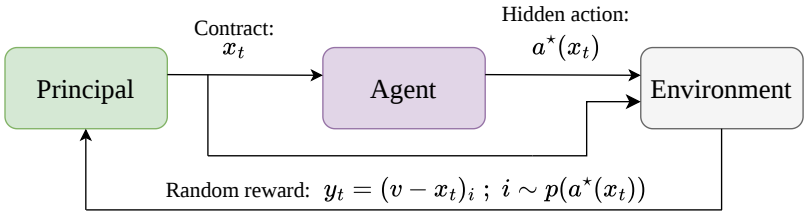


FIGURE 1. Schematic of principal-agent feedback model (1).

1.1. Related literature

Computational Contract Design. While the principal-agent problem is rooted in economic theory [19], recent literature has focused on its computational aspects. Authors in [3] establish that computing optimal contracts in combinatorial settings is NP-hard in general. Hence, research shifted towards identifying robust, simple contract structures, such as linear contracts [11], which provide approximation guarantees in static environments. However, these works assume that the principal has full knowledge of the agent's type and payoff function, an assumption that rarely holds in practice.

Learning Contracts with Bandit Feedback. In the repeated principal-agent setting, the problem is modeled as a multi-armed bandit with a finite number of arms. Authors in [15] pioneer this direction with the `Agnostic Zooming` algorithm. Their approach relies on adaptively discretizing the contract space; however, the analysis necessitates strong regularity assumptions, specifically that contracts are monotone and outcomes satisfy first-order stochastic dominance. Authors in [41] relaxed these conditions, offering guarantees for general utility functions. More recently, authors in [4] proposed the `Discover and Cover` algorithm to address the problem with finite agent actions. While this approach avoids the geometric assumptions of earlier works, it suffers from the *curse of dimensionality*, as the complexity scales exponentially with the number of agent actions.

Continuity of principal's utility. The principal's expected utility U^{PA} is inherently discontinuous, formally characterized by [37] as a piecewise-affine function with sharp transitions. Prior dynamic algorithms [15, 41] treat this discontinuity as an obstacle that requires inefficient contract-space discretization. In contrast, our work introduces a stochastic utility framework that smooths out these discontinuities, enabling the application of kernelized bandits [8] directly in the continuous domain.

1.2. Contributions

(i) **Modeling: Resolving discontinuity of U^{PA}.** We identify that the discontinuity of the principal's utility U^{PA}, a persistent barrier in dynamic contract design, stems from the idealized assumption of deterministic agent utilities. By incorporating a stochastic counterpart into the agent's utility function (capturing unmodeled physical transients and behavioral fluctuations), we prove that the principal's expected utility $\tilde{U}^{\mathrm{PA}}$ becomes continuous over its domain (Theorem 3.4). This modeling shift provides a more realistic representation of the

problem and fundamentally resolves the need for the intractable discretization relied upon by existing state-of-the-art methods [4, 15, 41].

(ii) **Algorithm: Sub-linear regret via tailored kernel.** Leveraging the continuity result of $\tilde{U}^{\mathrm{PA}}$, we propose the `Heteroscedastic GP-UCB` algorithm paired with a Neural Network kernel, specifically chosen to capture the non-stationary, sigmoidal geometry of $\tilde{U}^{\mathrm{PA}}$. By bounding the eigenvalue tail sum of this specific kernel (Proposition 4.4), we establish a formal high-probability regret bound of: $\mathbb{P}\left[\tilde{R}_T \leq \mathcal{O}\left(\sqrt{T}(\log T)^{m+1}\right)\right] \geq 1-\delta$ (Theorem 4.5), where m denotes the dimension of contract space $\mathcal{X}$, and $\delta > 0$ is the confidence level.

(iii) **Application: V2G incentive design.** Finally, we demonstrate the practical relevance of our framework by addressing the Vehicle-to-Grid (V2G) incentive design problem. We formulate a smart-charging scheme where aggregators incentivize electric vehicle owners to provide ancillary services. We prove (Proposition 5.2) that this complex engineering problem, governed by stochastic battery dynamics and user preferences, formally reduces to a dynamic principal-agent formulation.

It is noteworthy to mention that our proposed algorithm overcomes the best-known regret bound of the prior work [4] (i.e., $R_T \leq \mathcal{O}(m^n T^{4/5} \log T)$) by improving the original principal-agent problem formulation (i.e., from U^{PA} to $\tilde{U}^{\mathrm{PA}}$). We do not claim any improvement on R_T for the original (deterministic) problem formulation. Moreover, while R_T of [4] scales exponentially with the number of actions of the agent (i.e., m^n), the regret bound of our work (i.e., $\tilde{R}_T$) entirely eliminates this exponential dependence on n, where n and m denote the number of actions of the agent and the number of outcomes, respectively. In a nutshell, our methodology establishes that adopting a more realistic system model enables a faster learning rate.

2. Principal-Agent Feedback Model

Building on the introduction, we now discuss the principal-agent model's formulation in detail. After the agent executes the best response following (1a), an outcome $i \in [m]$ of the task is realized. The value for any outcome $i \in [m]$ for the principal is denoted by $(v)_i \in [0, 1]$ and is known to the principal. Moreover, the outcome of a task is realized following a conditional probability distribution defined by the production function $p : \mathcal{A} \to \Delta_m$, where Δ_m is the probability simplex over m outcomes. We define agents' and principals' utilities, respectively,

$$U^{\mathrm{A}}(x, a) = \sum_{i \in [m]} (x)_i \, p_i(a) - c(a), \tag{3a}$$

$$U^{\mathrm{P}}(x, a) = \sum_{i \in [m]} (v - x)_i \, p_i(a), \tag{3b}$$

where $p_i(a)$ is the probability of the i-th outcome conditional on action a, and $c : \mathcal{A} \to \mathbb{R}$ denotes the cost function of taking actions by the agent. As stated previously, $a^\star(x)$ is not observable to the principal; however, the principal observes the following random reward conditioned on $a^\star(x)$:

$$y = (v - x)_i, \text{ with the random index's law } i \sim p(a^\star(x)). \tag{4}$$

A schematic of the principal-agent model described in (1)-(4) is shown in Figure 1. We now discuss state-of-the-art methods for addressing the principal-agent problem. Furthermore, we analyze the key technical challenges faced by existing solution approaches.

2.1. **State-of-the-art results**

Authors in [15] propose an adaptive algorithm called `Agnostic Zooming` for this problem. Their algorithm provides the following upper bound on regret: $R_T(X_{\text{cand}}) \leq \mathcal{O}(\alpha \log T) T^{\frac{m+1}{m+2}}$, where $X_{\text{cand}} \subset [0,1]^m$ represents the set of monotone contracts. In addition, they also assume *first order stochastic dominance* of contracts, which makes their regret bound to be less attractive.

Authors in [41] recently proposed an algorithm where they lifted the monotonicity and *first order stochastic dominance* assumptions of [15]. They provide the upper bound of the regret in expectation as $R_T \leq \mathcal{O}(\sqrt{m}\log(T) T^{\frac{2m}{2m+1}})$. However, their proposed algorithm cannot be implemented in reality. In fact, one of the steps in their algorithm requires $\mathcal{X}$ to be discretized using certain methods from spherical coding in information theory, which remains an open problem to date (Section 3.1 of [41]).

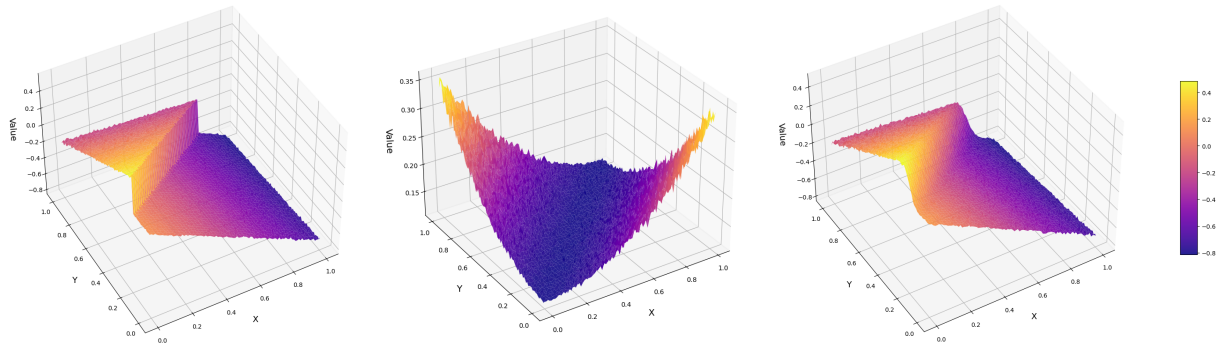
Recently, Authors in [4] propose an algorithm considering $\mathcal{A}$ is finite. They provide a probabilistic regret bound $\mathbb{P}\left[R_T \leq \mathcal{O}(m^n \mathcal{I} \log(T) T^{4/5})\right] \geq 1-\delta$, where $\mathcal{I}$ depends on m, n polynomially, and n is the number of actions available to agent. Here, the constants deteriorate performance significantly as m and n increase. This algorithm requires the private parameter n. Apart from the main results above, the survey [10] presents other weaker results on the computational aspects of contracts.

2.2. **Main technical challenges**

Assuming $\mathcal{A}$ is finite, the authors in [37] establish that the principal's utility function U^{PA} in (1b) is a piecewise affine function; it can be discontinuous on the boundary of linear pieces, and the global optimal contract is on the boundary of a linear piece. Let us first understand why U^{PA} is discontinuous on its domain with the help of a simple example. Our understanding of the root cause of this technical challenge will eventually help us to resolve it in later sections.

Assuming $\mathcal{A}$ is finite, the utility for an agent is $U^{\text{A}}(x,a) = \langle x, p(a)\rangle - c(a)$, where $a \in \mathcal{A} := \{a_1, a_2, \ldots, a_n\}$. For a fixed contract x, the objective function U^{A} admits multiple optimizers only if x lies on an indifference hyperplane between distinct actions a_i and a_j, defined as $H_{i,j} := \left\{x \in \mathcal{X} \mid U^{\text{A}}(x,a_i) = U^{\text{A}}(x,a_j)\right\}$. Consider a contract $x \in H_{i,j}$ such that the optimal action set is restricted to the pair $\{a_i, a_j\}$, meaning $U^{\text{A}}(x,a_i) = U^{\text{A}}(x,a_j) > U^{\text{A}}(x,a_k)$ for all $k \notin \{i,j\}$. If $p(a_i) \neq p(a_j)$, then the principal's utility U^{PA} exhibits a discontinuity across the hyperplane $H_{i,j}$. Letting $h_{i,j}(x) := U^{\text{A}}(x,a_i) - U^{\text{A}}(x,a_j)$ denote the affine function defining the boundary, the function U^{PA} takes the following piecewise form in the local neighborhood of x:

$$U^{\text{PA}}(x) = \langle (v-x), p(a^\star(x))\rangle + \begin{cases} \langle (v-x), p(a_j)\rangle, & \text{if } h_{i,j}(x) < 0, \\ \langle (v-x), p(a_i)\rangle, & \text{if } h_{i,j}(x) > 0. \end{cases} \tag{5}$$



(a) (b) (c)

FIGURE 2. *(High-low example 2.1)* (a) Visualization of U^{PA}. (b) Visualization of the variance of *heteroscedastic* noise σ_ε^2. (c) Geometry of $\tilde{U}^{\mathrm{PA}}$ under the Assumption 3.1.

Note that $\mathcal{X}$ is partitioned by at most $\frac{n(n-1)}{2}$ such discontinuous hyperplanes, corresponding to n actions of the agent. Let us now construct an instance of the so-called '*High-low example*' [15] to empirically visualize the landscape of U^{PA}.

Example 2.1 (High-low case study)**.** *The principal posts a contract $x \in [0,1]^2$ with two outcomes (low and high), and the agent takes action $a \in \{a^l(\text{low effort}), a^h(\text{high effort})\}$ with corresponding private costs $c(a^l), c(a^h)$, respectively. Low and high outcomes bring respective values $v = [v^l, v^h]^\top$ to the principal. To empirically construct U^{PA}, we consider $p(a^l) = [0.9, 0.1]$, $p(a^h) = [0.1, 0.9]$, $c(a^l) = 0$, $c(a^h) = 0.3$, and $v = [0.1, 0.8]^\top$. Then we discretize $[0,1]^2$ with (1000×1000) data points and sample each data point 1000 times to empirically approximate U^{PA}. Figure 2a illustrates the geometry of the empirically constructed U^{PA}. The x-axis represents contract $(x)_1$ for low outcome, and the y-axis denotes $(x)_2$ for high outcome. Unsurprisingly, we find it to be discontinuous between two affine pieces.*

The second technical challenge is the decision-variable-dependent randomness in the principal's reward structure. At round t, we can rewrite the principal's reward in (4) in the following form:

$$\begin{aligned} y_t &= (v - x_t)_i, \quad i \sim p(a^\star(x_t)), \\ &= \sum_{j \in [m]} (v - x_t)_j p_j(a^\star(x_t)) + \varepsilon(x_t) = U^{\mathrm{PA}}(x_t) + \varepsilon(x_t), \end{aligned} \tag{6}$$

where $\varepsilon(x_t)$ is the zero-mean *heteroscedastic* noise with variance $\mathbb{V}ar[\varepsilon(x_t)] = \sum_{j\in[m]} p_j(a^\star(x_t))[(v - x_t)_j - U^{\mathrm{PA}}(x_t)]^2$. Unlike homoscedastic models, the noise variance here depends on x_t and is unknown to the decision maker. Figure 2b illustrates this input-dependent variance, σ_ε^2, for Example 2.1.

3. Main Result I: Resolving Discontinuity of U^{PA}

Due to the aforementioned challenges, the authors in [4, 15, 41] prefer to discretize the contract space $\mathcal{X}$ and deal with a finite number of contracts. Instead, our approach is not to discretize $\mathcal{X}$,

but to design algorithms that can work in a continuous contract space, thereby freeing us from the technicality of upper-bounding the discretization error. In this regard, we first make the following improvement to the agent's utility.

Consider an embedding $\delta : \mathcal{A} \to \mathbb{R}^n$, and an $\mathbb{R}^n$-valued random variable w. We define the agent's best response by adding a stochastic counterpart to the agent's utility function, and update the optimal contract, respectively,

$$\tilde{a}^\star(x, w) = \arg\max_{a \in \mathcal{A}} \left\{ \tilde{U}^{\mathrm{A}}(x, a) = U^{\mathrm{A}}(x, a) + \langle w, \delta(a) \rangle \right\}, \tag{7a}$$

$$\tilde{x}^\star = \arg\max_{x \in \mathcal{X}} \left\{ \tilde{U}^{\mathrm{PA}}(x) = \mathbb{E}_w[U^{\mathrm{P}}(x, \tilde{a}^\star(x, w))] \right\}, \tag{7b}$$

while the random feedback observed by the principal

$$\tilde{y} = (v - x)_i, \quad i \sim q(x), \tag{8}$$
$$\text{where} \quad q_i(x) = \mathbb{E}_w \left[p_i(\tilde{a}^\star(x, w)) \right].$$

Note, the updated production function $q(x)$ encodes the probability of any realization $i \in [m]$ by marginalizing over randomness related to w. Moreover, the regret in (2) is updated as

$$\tilde{R}_T = T\tilde{U}^{\mathrm{P}*} - \sum_{t=1}^{T} \tilde{U}^{\mathrm{PA}}(x_t), \tag{9}$$

where $\tilde{U}^{\mathrm{P}*} = \max_{x \in \mathcal{X}} \ \tilde{U}^{\mathrm{PA}}(x)$.

While the modification in (7a) serves to restore the continuity of the principal's utility, it fundamentally reflects the inherent uncertainty of real-world agent behavior. In [4, 37], agent utilities are assumed to be static for a fixed x and a. However, in practical settings like a crowdsourcing market, a worker's cost (effort) fluctuates due to unobserved latent factors such as fatigue, attention shifts, or external interruptions. Therefore, for the same contract and action pair (x, a), the agent's utility could take different values for different instances of time. By introducing stochasticity into the agent's utility function, we effectively treat the principal's problem as an optimization over the *robust expected behavior* of the agent, rather than overfitting to a deterministic model that inherently fails to capture such physical transients.

Assumption 3.1 (Stochastic regularity)**.** *For the stochastic addition in the best response* (7a)*, we assume the following:*

(i) One-to-one embedding: For all $a_1, a_2 \in \mathcal{A}$, there exists a continuous embedding $\delta : \mathcal{A} \to \mathbb{R}^n$ such that $\delta(a_1) \neq \delta(a_2)$ if $a_1 \neq a_2$.

(ii) Full-dimensional support: The random variable $w \in \mathbb{R}^n$ admits a probability density distribution that is absolutely continuous with respect to Lebesgue measure. In particular, for all $v_1 \in \mathbb{R}^n, v_2 \in \mathbb{R}$, the probability of any hyperplane is $\mathbb{P}[w \in \mathbb{R}^n \mid \langle v_1, w \rangle = v_2] = 0$ (e.g., uniform distribution supported on $[0, 1]^n$).

Example 3.2 (Embedding). *If $\mathcal{A}$ is finite, then it suffices to choose the embedding image (i.e., $n = 1$) as $\delta(a_i) = (i-1)/|\mathcal{A}|$. If $\mathcal{A}$ is a continuous set, a straightforward embedding is the identity map: $\delta(a) = a$. To visualize an example of the proposed modification, consider $U^{\mathrm{A}}(x, a) = x - (a^4 - 2a^2)$ for $x = 0$, and Figure 3 depicts how the optimizer $a^\star$ shifts to $\tilde{a}^\star$ as we add the identity embedding.*

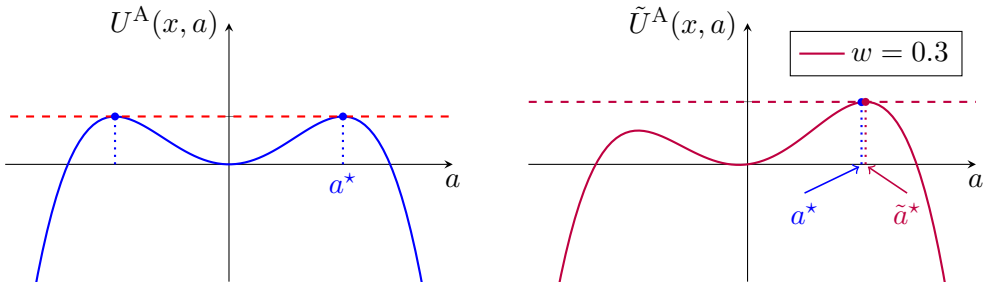


FIGURE 3. Left: The agent's utility $U^{\mathrm{A}}(x, a)$ with optimizer marked at $a^\star$. Right: The modified utility $\tilde{U}^{\mathrm{A}}(x, a) := U^{\mathrm{A}}(x, a) + \langle w, a\rangle$ with a realization $w = 0.3$. Angled pointers clearly distinguish the original optimizer $a^\star$ from the newly shifted optimizer $\tilde{a}^\star$.

We also assume the following standard property for U^{A} and U^{P}.

Assumption 3.3 (Compact and continuous domain). *The agent's utility $U^{\mathrm{A}} : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ and the uncoupled principal's utility $U^{\mathrm{P}} : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ are continuous on their compact domains $\mathcal{X}$ and $\mathcal{A}$.*

The above assumptions allows us to characterize the following property of the correspondence $\tilde{a}^\star(x, w)$ in (7a) and the principal's utility $\tilde{U}^{\mathrm{PA}}$.

Theorem 3.4 (Continuous principal's utility). *Under Assumptions 3.1 and 3.3, the following holds:*

(i) The agent's best response $\tilde{a}^\star(x, w)$ in (7a) is a singleton and continuous at any $x \in \mathcal{X}$ almost surely, i.e., for any $x \in \mathcal{X}$, we have

$$\mathbb{P}\Big[w \in \Omega \mid \ \tilde{a}^\star(x,w) \text{ is singleton and } \forall a_n \in \tilde{a}^\star(x_n, w), \lim_{n\to\infty} a_n = \tilde{a}^\star(x,w)\Big] = 1. \tag{10}$$

(ii) The principal's utility $\tilde{U}^{\mathrm{PA}}$ defined in (7b) is continuous on its domain.

Proof. The proof is decomposed into two parts:

Part (i). Define the value function

$$V(x, w) := \max_{a\in\mathcal{A}} \left\{U^{\mathrm{A}}(x, a) + \langle w, \delta(a)\rangle\right\}.$$

Since $\mathcal{A}$ is compact and U^{A} is continuous on its domain, V is finite everywhere. Moreover, for a fixed x, V is convex in w as it is the maximum of a family of affine functions. By an extension result from Danskin's theorem [6], we have

$$\partial_w V(x, w) = \mathrm{co}\big\{\delta(a) \mid a \in \tilde{a}^\star(x, w)\big\},$$

where $\partial_w V(x,w)$ denotes the convex subdifferential of V with respect to w, and $\text{co}\{\cdot\}$ denotes the convex hull of its input. Consequently, we have

$$V \text{ is differentiable at } x \text{ and } w \iff \tilde{a}^\star(x,w) \text{ is a singleton.}$$

From Rademacher's theorem [12, Section 3.1], we know that if the function V is a finite convex function on $\mathbb{R}^n$, which is the case here, it is differentiable almost everywhere with respect to Lebesgue measure. Therefore, the set

$$N := \{w \in \mathbb{R}^n : \tilde{a}^\star(x,w) \text{ is not a singleton}\}$$

has Lebesgue measure zero. Because the distribution of w is absolutely continuous with respect to Lebesgue measure (following Assumption 3.1), then

$$\mathbb{P}[w \in N] = 0 \iff \mathbb{P}\big[\tilde{a}^\star(x,w) \text{ is a singleton}\big] = 1 \tag{11}$$

By Assumptions 3.3 and 3.1, $(U^{\text{A}}(x,a) + \langle w, \delta(a)\rangle)$ is jointly continuous in x and a, which leads us to apply the celebrated Berge's maximum theorem [1, Theorem 17.31] on (7a). And we say that the correspondence $\tilde{a}^\star(x,w)$ is upper-hemicontinuous at x. At the same time, we just proved in (11) that $\tilde{a}^\star(x,w)$ is unique. Thus, $\tilde{a}^\star(x,w)$ is continuous on x.

Part (ii). For any $x \in \mathcal{X}$, and any convergent sequence of $\{x_n\} \to x$, we have

$$\begin{aligned}\lim_{x_n \to x} \tilde{U}^{\text{PA}}(x_n) = \lim_{x_n \to x} \mathbb{E}_w[U^{\text{P}}(x_n, \tilde{a}^\star(x_n,w))] &= \mathbb{E}_w\left[\lim_{x_n \to x} U^{\text{P}}(x_n, \tilde{a}^\star(x_n,w))\right] \\ &= \mathbb{E}_w\left[U^{\text{P}}(x, \tilde{a}^\star(x,w))\right] = \tilde{U}^{\text{PA}}(x)\end{aligned}$$

where the second equality follows from the Dominated Convergence Theorem [12, Theorem 1.19], as the function U^{P} is continuous (Assumption 3.3), and hence uniformly bounded over any compact set. The third equality holds true because of the continuity of $\tilde{a}^\star(x,w)$ in x as shown in the proof of *part* (i). This concludes the proof. □

It is noteworthy to mention that Theorem 3.4 is built using the abstract formulation in (1), which implies that our result is also applicable to the class of repeated Stackelberg games [24]. Any other utility structure besides the principal-agent model in (3) will follow Theorem 3.4 as long as the necessary assumptions are met. To address the technical challenge of input-dependent randomness in (6), we propose a dedicated algorithm in Section 4.2.

4. Main Result II: Tailored Kernel and Sub-linear Regret

In this section, we focus on designing an algorithm for the principal-agent model given by (3) and (4). Since the previous work [37] has already established geometric characterizations of the principal's utility landscape with finite agent actions, we aim to exploit this to achieve a better regret rate. To this end, we restrict the complexity of our problem with the following assumption.

Assumption 4.1 (Finite action space)**.** *The agent has a finite number of actions, i.e.,* $|\mathcal{A}| < \infty$.

By Assumption 4.1, a candidate embedding can be $\delta(a_j) = (j-1)/|\mathcal{A}|$ for $a_j \in \mathcal{A}$, which leads to the agent's best response $\tilde{a}^\star(x,w) = \arg\max_{a_j \in \mathcal{A}} \left\{ \sum_{i\in[m]}(x)_i \; p_i(a_j) - c(a_j) + \frac{w(j-1)}{|\mathcal{A}|} \right\}$, and the principal's utility, $\tilde{U}^{\mathrm{PA}}(x) = \mathbb{E}_w \Big[\sum_{i\in[m]}(v-x)_i \; p_i(\tilde{a}^\star(x,w)) \Big]$. We now visualize $\tilde{U}^{\mathrm{PA}}$ using Example 2.1 where $|\mathcal{A}| = 2$. Assuming that $w \sim \mathcal{N}(0, \sigma_w^2)$, Figure 2c shows a continuous $\tilde{U}^{\mathrm{PA}}$ as stated in Theorem 3.4. In this regard, we identify two key observations: (1) the surface of $\tilde{U}^{\mathrm{PA}}$ consists of flat plateaus and steep transitions (see Figure 2c), and (2) the variance around $\tilde{U}^{\mathrm{PA}}$ is still decision-variable-dependent as given in (6). To address both observations, we consider stochastic bandits [23] over a continuous domain. While parametric bandits [34] assume linear reward structures, akin to the simpler theory of linear contracts, we adopt a nonparametric approach using Gaussian Processes (GPs) to capture the complex structure of the principal's utility.

A GP over the feasibility set $\mathcal{X} = [0,1]^m$, denoted by $GP_{\mathcal{X}}(\mu_t(\cdot), k(\cdot,\cdot))$, is a collection of random variables $(\tilde{U}^{\mathrm{PA}}(x))_{x\in\mathcal{X}}$, such that every finite sub-collection of random variables $(\tilde{U}^{\mathrm{PA}}(x_i))_{i=1}^n$ is jointly Gaussian with mean $\mu_t(x_i) = \mathbb{E}[\tilde{U}^{\mathrm{PA}}(x_i)]$ and covariance $k(x_i,x_j) = \mathbb{E}[(\tilde{U}^{\mathrm{PA}}(x_i) - \mu_t(x_i))(\tilde{U}^{\mathrm{PA}}(x_j) - \mu_t(x_j))]$, $1 \le i,j \le n, n \in \mathbb{N}$. Algorithms use $GP_{\mathcal{X}}(0, k(\cdot,\cdot))$ as an initial prior distribution for the unknown function $\tilde{U}^{\mathrm{PA}}$ over $\mathcal{X}$, where $k(\cdot,\cdot)$ is the kernel associated with a suitable Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}_k$. Algorithms such as GP-UCB [35], GP-PI [18], and IGP-UCB [8] leverage GPs by imposing regularity assumptions on the objective function $\tilde{U}^{\mathrm{PA}}$: either (1) $\tilde{U}^{\mathrm{PA}}$ is a sample from a GP prior, or (2) $\tilde{U}^{\mathrm{PA}}$ resides in the RKHS $\mathcal{H}_k$ of a kernel k. Given the structured geometry of $\tilde{U}^{\mathrm{PA}}$, the agnostic RKHS assumption is theoretically more robust than assuming the function is a random draw from a GP. Consequently, the focus is to identify a kernel k whose RKHS $\mathcal{H}_k$ naturally models flat plateaus and steep transitions.

4.1. **Kernel design**

Standard stationary kernels, such as the Gaussian kernel, are ill-suited for this problem as their reliance on a single, global lengthscale fails to adapt to the heterogeneous geometry of $\tilde{U}^{\mathrm{PA}}$. While the underlying deterministic utility (U^{PA}) exhibits distinct affine segments, our stochastic smoothing transforms them into a non-stationary landscape characterized by flat plateaus and steep transitions (Figure 2c). To mimic this structure, we employ the *Arcsin kernel* (aka Neural Network kernel) [38, Section 3], which arises as the limit of a single-hidden-layer Bayesian Neural Network with error function (erf) activation. Unlike stationary kernels, this kernel induces a non-stationary prior inherently designed to model sigmoidal superpositions. This inductive bias allows it to represent the soft thresholding behavior and varying local smoothness of the principal's expected utility.

Consider the input-output mapping of a single-hidden-layer neural network $f_N : \mathbb{R}^n \to \mathbb{R}$ with an activation function $\phi : \mathbb{R} \to \mathbb{R}$, defined as

$$\phi(z) = \frac{2}{\sqrt{\pi}} \int_0^z \exp\left(-t^2\right) dt, \tag{12a}$$

$$f_N(x) = \sum_{i=1}^{N} \varsigma_i \phi(\langle r_i, x \rangle + b_i), \tag{12b}$$

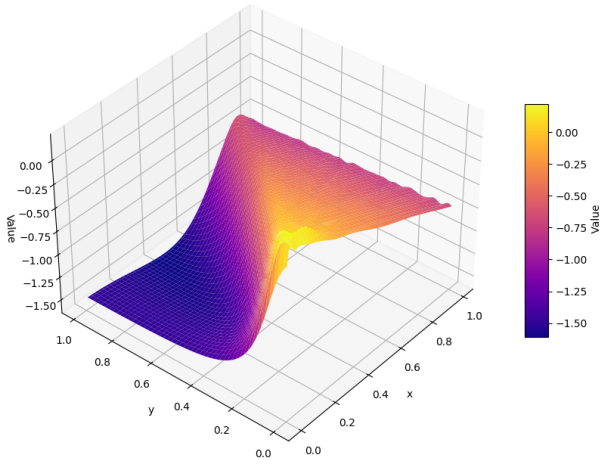


FIGURE 4. Representative function of $\tilde{U}^{\mathrm{PA}}$, which is built with Neural Network kernel using kernel regression where the empirical samples are generated from an instance of 'High-low' example 2.1.

where the independent Gaussian priors $\varsigma_i \sim \mathcal{N}(0, \sigma_\varsigma^2/N)$, $r_i \sim \mathcal{N}(0, \sigma_v^2 \mathbf{I})$, and $b_i \sim \mathcal{N}(0, \sigma_b^2)$. The kernel function $k(x, y)$ is derived as the limiting covariance

$$k(x, y) := \lim_{N \to \infty} \mathbb{E}[f_N(x) f_N(y)] = \sigma_\varsigma^2 \mathbb{E}_{z_x, z_y}[\phi(z_x)\phi(z_y)],$$

where the last equality follows by applying the central limit theorem, $z_x = \langle r, x \rangle + b$ and $z_y = \langle r, y \rangle + b$ are jointly Gaussian variables. The expectation in the above equation admits a closed-form solution known as the Neural Network (aka 'arcsin') kernel [38], described by

$$k(x, y) = \frac{2\sigma_\varsigma^2}{\pi} \arcsin\left(\frac{\sigma_v^2 \langle x, y \rangle + \sigma_b^2}{\sqrt{(1 + \sigma_v^2 \langle x, x \rangle + \sigma_b^2)(1 + \sigma_v^2 \langle y, y \rangle + \sigma_b^2)}} \right), \tag{13}$$

where σ_v^2 and σ_b^2 are treated as hyper-parameters. Apart from being symmetric, the positive definiteness of this kernel is guaranteed by its construction as an expectation of squares, $\sum_{i,j} c_i k(x_i, x_j) c_j = \sigma_w^2 \mathbb{E}\left[\left(\sum_i c_i \phi(z_{x_i}) \right)^2 \right] \geq 0$. By the Moore-Aronszajn theorem, any symmetric and positive definite kernel on $\mathcal{X}$ induces a unique RKHS $\mathcal{H}_k$ defined as

$$\mathcal{H}_k := \left\{ h \in \mathbb{R}^{\mathcal{X}} \mid \exists \alpha_i \in \mathbb{R}, x_i \in \mathcal{X}\ \forall i \in \mathbb{N} \text{ with } h(x) = \sum_{i=1}^{\infty} \alpha_i k(x_i, x) \text{ and } \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} \alpha_i k(x_i, x_j) \alpha_j < \infty \right\}.$$

To empirically validate the correct hypothesis space of $\tilde{U}^{\mathrm{PA}}$, we run a kernel regression on an instance of 'High-low' example (similar to Example 2.1) with $k(x, y)$ in (13). We plot the learned function in Figure 4, which clearly shows the flat plateaus with sigmoidal transitions we were looking for. Therefore, the $\mathcal{H}_k$ induced by $k(x, y)$ in (13) is a candidate hypothesis space for $\tilde{U}^{\mathrm{PA}}$.

4.2. **Description of the algorithm**

First, we develop methods to address the *heteroscedastic* noise in the principal's utility, which is denoted as a technical challenge in (6). In particular, we learn the unknown variance of this noise using feedback from repeated sampling of the principal's utility at the same input. A good estimate of the variance helps us obtain a more accurate upper confidence bound for the unknown $\tilde{U}^{\mathrm{PA}}$. Then we detail our main algorithm, based on a upper confidence bound-based acquisition function constructed using the Neural Network kernel in (13). To start with, similar to (6), we represent the feedback as a noisy version of $\tilde{U}^{\mathrm{PA}}$, and write

$$\tilde{y}_t = \tilde{U}^{\mathrm{PA}}(x_t) + \tilde{\varepsilon}(x_t), \tag{14}$$

where $\tilde{\varepsilon}(x_t)$ is assumed to be a zero-mean *heteroscedastic* noise with the variance $\mathbb{V}ar(\tilde{\varepsilon}(x_t)) = \sum_{i\in[m]} q_i(x_t)[(v - x_t)_i - \tilde{U}^{\mathrm{PA}}(x_t)]^2$, and the production function $q(x_t)$ is defined in (8). It is straightforward to follow that $\mathbb{V}ar(\tilde{\varepsilon}(x_t))$ is continuous in x_t since the continuity of $\tilde{a}^\star(x_t, w)$ in x_t (Theorem 3.4) makes $q_i(x_t)$ to be continuous on its argument. Here, we first impose the following assumption on $\tilde{\varepsilon}(x_t)$ to restrict it within a well-defined function class.

Assumption 4.2 (Noise function class)**.** *The noise variance $\mathbb{V}ar(\tilde{\varepsilon}(x_t))$ belongs to an RKHS induced by some kernel κ^v, i.e., $\mathbb{V}ar(\tilde{\varepsilon}(\cdot)) \in \mathcal{H}_{\kappa^v}$ with its RKHS norm is upper bounded with $B_v > 0$.*

Such assumptions are quite common when dealing with *heteroscedastic* noise [20, 30]. To learn $\mathbb{V}ar(\tilde{\varepsilon}(\cdot))$, we construct a repeated setting, where we collect $\ell > 1$ samples for each x_t and form the feedback collection $\{\tilde{y}_t^i\}_{i=1}^{\ell}$. Then, we evaluate the sample mean and variance of $\tilde{\varepsilon}(x_t)$ as

$$\hat{y}_t = \frac{1}{\ell}\sum_{i=1}^{\ell} \tilde{y}_t^i = \frac{1}{\ell}\sum_{i=1}^{\ell} (\tilde{U}^{\mathrm{PA}}(x_t) + \tilde{\varepsilon}_i(x_t)), \tag{15a}$$

$$\hat{v}_t = \frac{1}{\ell - 1}\sum_{i=1}^{\ell} (\tilde{y}_t^i - \hat{y}_t)^2. \tag{15b}$$

While we have random feedback samples of $\tilde{y}_t$, we represent the randomness of sample variance using a zero-mean noise $\eta(x_t)$. Since $(v)_i, (x)_i \in [0, 1]$, it is straightforward to check that the noise $\tilde{\varepsilon}(x_t)$ is uniformly bounded as $|\tilde{\varepsilon}(x_t)| < \sqrt{\chi_u}$, which leads to its variance function $\mathbb{V}ar(\tilde{\varepsilon}(x_t)) < \chi_u$ with $\chi_u > 0$. And this allows us to further state $\hat{v}_t = \mathbb{V}ar(\tilde{\varepsilon}(x_t)) + \eta(x_t)$. Moreover, the noise $\eta(x_t)$ adheres to the following standard assumption.

Assumption 4.3 (Noise process regularity)**.** *The noise $\eta(x_t)$ is $\rho_\eta(x_t)$-sub-Gaussian with known $\rho_\eta^2(x_t)$. Moreover, the process $\{\eta(x_t)\}_{t\geq 1}$ is generated independently in time.*

The sub-Gaussianity assumption of $\eta(x_t)$ naturally follows from uniform boundedness of $\tilde{\varepsilon}(x_t)$. Following a brief introduction of GP models at the start of Section 4, the posterior GP mean $(\mu_t(\cdot))$ and variance $(\sigma_t^2(\cdot))$ of $\tilde{U}^{\mathrm{PA}}$ are respectively calculated based on the previous measurements $\hat{y}_{1:t} = [\hat{y}_1, \dots \hat{y}_t]^\top$

$$\begin{aligned} \mu_t(x) &= k_t(x)^\top (K_t + \lambda\Sigma_t)^{-1}\hat{y}_{1:t}, \\ \sigma_t^2(x) &= \frac{1}{\lambda}(k(x,x) - k_t(x)^\top (K_t + \lambda\Sigma_t)^{-1}k_t(x)), \end{aligned} \tag{16}$$

where $\Sigma_t := \text{diag}(\mathbb{V}ar(\tilde{\varepsilon}(x_1)), \ldots, \mathbb{V}ar(\tilde{\varepsilon}(x_t)))$, $(K_t)_{i,j} = k(x_i, x_j)$ is an element of the covariance matrix K_t, $\lambda > 0$ is a parameter trading off the magnitude of prior variance, and $k_t(x) := [k(x_1, x), \ldots, k(x_t, x)]^\top$. Here, $k(x_i, x_j)$ refers to our proposed Neural Network kernel in (13). As Σ_t depends on the unknown variance $\mathbb{V}ar(\tilde{\varepsilon}(x_t))$, we construct a separate GP model to learn it. Since we could not decode any specialized geometry for $\tilde{\varepsilon}(\cdot)$, we consider a universal kernel, such as a Gaussian kernel, to define the GP. The corresponding posterior mean μ_t^v and variance σ_t^v are calculated based on (16) by replacing Σ_t with $\Sigma_t^v := \text{diag}[\rho_\eta^2(x_1), \ldots, \rho_\eta^2(x_t)]$, $\hat{y}_{1:t}$ with $\hat{v}_{1:t} = [\hat{v}_1, \ldots, \hat{v}_t]^\top$, and using Gaussian kernel $k^v(\cdot, \cdot)$. Then, we build the upper confidence bound $\text{ucb}_t(\cdot)$ as

$$\text{ucb}_t(x) := \mu_{t-1}^v(x) + \beta_t^v \sigma_{t-1}^v(x), \tag{17}$$

where the exploration-exploitation trade-off parameter $\beta_t^v = \sqrt{2 \log \left(\frac{\det(\lambda \Sigma_t^v + K_t^v)^{0.5}}{\delta \det(\lambda \Sigma_t^v)^{0.5}} \right)} + \sqrt{\lambda} B_v$, taken from [20, Lemma 7]. We can approximate Σ_t with the upper confidence bound of $\mathbb{V}ar(\tilde{\varepsilon}(x_t))$ as

$$\hat{\Sigma}_t = \text{diag}(\min\{\text{ucb}_t(x_1), \chi_u\}, \ldots, \min\{\text{ucb}_t(x_t), \chi_u\})/\ell. \tag{18}$$

Now we are in a position to define the upper confidence bound-based acquisition function for $\tilde{U}^{\text{PA}}$ using the current posterior mean μ_{t-1} and variance σ_{t-1}^2 as

$$x_t = \arg\max_{x \in \mathcal{X}} \underbrace{\mu_{t-1}(x) + \beta_t \sigma_{t-1}(x)}_{\text{Acquisition function}}, \tag{19}$$

where x_t is the action of the algorithm to maximize the acquisition function, and the exploitation-exploration trade-off parameter

$$\beta_t = \sqrt{2 \log \left(\frac{\det(\lambda \hat{\Sigma}_t + K_t)^{0.5}}{\delta \det(\lambda \hat{\Sigma}_t)^{0.5}} \right)} + \sqrt{\lambda} B, \tag{20}$$

which follows from [20, Lemma 7]. In (20), δ is the confidence interval, and B is the upper-bound of the RKHS-norm of its kernel. Intuitively, β_t determines how much we trust the accuracy of the current posterior mean (μ_{t-1}), which is an approximation of $\tilde{U}^{\text{PA}}$ on the fly, by suitably adding the posterior variance (σ_{t-1}), since the posterior variance quantifies the uncertainty of μ_{t-1} being the true unknown $\tilde{U}^{\text{PA}}$. Note, the role of $\hat{\Sigma}_t$ in (20) is to better estimate the unknown variance matrix (i.e., Σ_t), which essentially improves the exploitation-exploration trade-off. Needless to say, the posterior mean and variance of the acquisition function in (19) is built with our proposed Neural Network kernel in (13). In Algorithm 1, we explain the operation of our proposed methodology. Note that the initial dataset $\mathcal{X}_0$ is created by sampling any fixed $x_0 \in \mathcal{X}$ for ℓ times.

4.3. **Regret bound**

The asymptotic performance of `Heteroscedastic GP-UCB` is based on the regret bound of [30, Theorem 1] where the authors provide

$$\mathbb{P}\left[\tilde{R}_T \leq \beta_T \ell \sqrt{\frac{2 \hat{\gamma}_T T}{\log(1 + \frac{\ell}{\chi_u^2})}} \right] \geq 1 - \delta, \tag{21}$$

Algorithm 1: `Heteroscedastic GP-UCB`

Initialization: B, B_v, λ, ℓ, Kernel functions $k(\cdot,\cdot)$, $k^v(\cdot,\cdot)$, Confidence level δ, Initial dataset $\mathcal{X}_0 = \{(x_0^i, \tilde{y}_0^i)_{i\in[l]}\}$, and Prior $\mu_0 = \mu_0^v = 0$.
for $t = 1, 2, \ldots$ **do**
  Compute $\hat{y}_t$ and $\hat{v}_t$ following (15) (use $\mathcal{X}_0$ if $t = 1$).
  Update $\text{ucb}_t(\cdot)$ following (17).
  Update $\hat{\Sigma}_t$ following (18).
  Update the posterior $\mu_t(\cdot)$ and $\sigma_t^2(\cdot)$ following (16).
  Select $x_t = \arg\max_{x\in\mathcal{X}} \ \mu_{t-1}(x) + \beta_t \sigma_{t-1}(x)$
  for $i \in \{1, \ldots, \ell\}$ **do**
    Collect $\{\tilde{y}_t^i\}$, where $\tilde{y}_t^i = \tilde{U}^{\text{PA}}(x_t) + \tilde{\varepsilon}_i(x_t)$
  end
end

where δ is the confidence level, β_T is defined in (20), and $\hat{\gamma}_T$ denotes the maximum information gain [35]. At time T, $\hat{\gamma}_T$ is defined using mutual information $I(y_{1:T}, \tilde{U}_{1:T}^{\text{PA}})$ [28, Equation 2.28] between the feedback $y_{1:T} = [y_1, \ldots y_T]^\top$ and $\tilde{U}_{1:T}^{\text{PA}} = [\tilde{U}^{\text{PA}}(x_1), \ldots \tilde{U}^{\text{PA}}(x_T)]^\top$ as

$$\hat{\gamma}_T := \max_{A\subset\mathcal{X}, |A|=T} \quad I(y_{1:T}, \tilde{U}_{1:T}^{\text{PA}}). \tag{22}$$

In other words, maximum information gain is essentially the maximum uncertainty reduction of the unknown function $\tilde{U}^{\text{PA}}$ for a set of T sampled data (say, A) in $\mathcal{X}$. The main technical challenge in this setup is to find an upper bound for $\hat{\gamma}_T$. $\hat{\gamma}_T$ is kernel-specific, and there exist standard upper bounds for $\hat{\gamma}_T$ where the kernel k is Gaussian, Matern, or Linear [35]. However, to our best knowledge, no such upper bound exists for our proposed neural-network kernel in (13). Following [35, Theorem 8], the upper bound of $\hat{\gamma}_T$ is a function of the eigenvalue tail sum $B_k(T_*)$ of the respective kernel, where $B_k(T_*) := \sum_{s=T_*+1}^{\infty} \lambda_s$, and λ_s is the eigenvalue corresponding to the integral operator of any kernel $k(x, y)$. For our proposed neural network kernel in (13), the following result provides an upper bound for $B_k(T_*)$.

Proposition 4.4 (Bound on eigenvalue tail sum)**.** *Given the neural network kernel $k(x, y)$ in (13), its eigenvalues λ_s, corresponding to the integral operator, decay exponentially. Consequently, the eigenvalue tail sum $B_k(T_*) = \sum_{s=T_*+1}^{\infty} \lambda_s$ is bounded as*

$$B_k(T_*) \leq \mathcal{O}\big(\exp\left(-\beta T_*^{1/m}\right)\big) \tag{23}$$

for some constant $\beta > 0$.

The proof is given in Appendix A.2. The proof of the Proposition 4.4 hinges on the analytic property of any kernel. The analytic property refers to a certain level of smoothness of any kernel. It turns out that our proposed kernel enjoys this property (Lemma 1) which leads to its exponentially decaying eigenvalues. With the help of Proposition 4.4, we manage to derive an explicit upper bound for $\hat{\gamma}_T$, which helps us establish the main regret bound of this study.

Theorem 4.5 (Sub-linear regret). *Consider $\tilde{U}^{\mathrm{PA}} \in \mathcal{H}_k$ with bounded RKHS-norm (i.e., $\|\tilde{U}^{\mathrm{PA}}\|_{\mathcal{H}_k} \leq B$) and sampling model in (14) with unknown variance $\mathbb{V}ar(\tilde{\varepsilon}(x_t))$ that satisfies Assumptions 4.2 and 4.3. Let $\{x_t\}_{t=1}^T$ denote the set of actions chosen by Algorithm 1. Setting $\lambda = 1$ in (20), the cumulative regret $\tilde{R}_T$ defined in (9) is bounded as*

$$\mathbb{P}\left[\tilde{R}_T \leq \mathcal{O}\left(\sqrt{T}(\log T)^{m+1}\right)\right] \geq 1 - \delta \tag{24}$$

for a chosen confidence level $\delta > 0$.

Proof. The broad idea of the proof is to derive a kernel-specific upper bound of the information gain for an already existing regret bound in [30]. To bound the information gain, we follow key ideas from [35] to bound the eigenvalue tail sum of our proposed kernel in (13), and Proposition 4.4 does this job. To prove Proposition 4.4, we require the analytic property of our kernel, which is given by Lemma 1. The rest of the proof uses a few known algebraic manipulations to simplify the regret bound.

We first write from Theorem 1 of [30] for the coefficient of risk $\alpha = 0$

$$\mathbb{P}\left[\tilde{R}_T \leq \beta_T \ell \sqrt{\frac{2\hat{\gamma}_T T}{\log(1 + \frac{\ell}{\chi_u^2})}}\right] \geq 1 - \delta, \tag{25}$$

where δ is the confidence level, and the maximum information gain [30, Equation 42] is defined as

$$\hat{\gamma}_T = \max_{A \subset D, |A| = T} \sum_{t=1}^{T} 0.5 \log\left(1 + \frac{\sigma_{t-1}^2(x_t | \mathrm{diag}(\chi_u^2/\ell))}{\chi_u^2/\ell}\right). \tag{26}$$

One could compare (26) with the information gain of homoscedastic noise (γ_T in [35, Lemma 5.3]) and state that $\hat{\gamma}_T$ is simply same as γ_T with a constant noise variance $\sigma^2 = (\chi_u^2/\ell)$. Therefore, making the required modification and replacing the upper bound of β_T from equation (25) in [30]

$$\tilde{R}_T \leq \frac{\ell\sqrt{T}}{\sqrt{\log(1 + \frac{\ell}{\chi_u^2})}} \left(\sqrt{4\gamma_T \log(1/\delta) + 2\gamma_T^2} + B\sqrt{2\lambda\gamma_T}\right). \tag{27}$$

Now, the key part is to upper bound γ_T for our proposed kernel $k(x, y)$ in (13). For this, we use the result in [35, Theorem 8] as

$$\gamma_T \leq \frac{0.5}{1 - e^{-1}} \max_{r \leq T} \left(T_* \log(r n_T/\sigma^2) + C_4 \sigma^{-2}(1 - r/T)(\log T)(T^{\tau+1} B_k(T_*) + 1)\right) + \mathcal{O}(T^{1-\frac{\tau}{m}}) \tag{28}$$

for any $T_* \in \{1, \ldots, n_T\}$ and $\tau > 0$. Here $B_k(T_*) = \sum_{s > T_*} \lambda_s$ with $\{\lambda_s\}$ being the operator spectrum of the kernel $k(x, y)$ with respect to uniform distribution over D, $n_T = C_4 T^\tau \log T$ with $C_4 = 2\mathcal{V}(D)(2\tau + 1)$, and $\mathcal{V}(D) = \int_{x \in D} dx$.

Absorbing (28) inside 'Big-O' notation, we write

$$\gamma_T \leq \mathcal{O}\Big(\max_{r=1,\ldots,T} (T_* \log(r n_T)) + (T - r) T^\tau B_k(T_*)\Big) + \mathcal{O}(T^{1-\tau/m}).$$

Now setting $\tau = m$ converts $\mathcal{O}(T^{1-m/m}) = \mathcal{O}(1)$. As $T \to \infty$, we can further compress the upper bound and write

$$\begin{aligned}
\gamma_T &\leq \mathcal{O}\Big(\max_{r=1,\dots,T} (T_* \log(r n_T)) + T^{m+1} B_k(T_*) \Big), \\
&= \mathcal{O}\Big(\max_{r=1,\dots,T} (T_* \log(r T^m)) + T^{m+1} B_k(T_*) \Big), \quad \text{(by setting } n_T = \mathcal{O}(T^m \log T)\text{)}, \\
&= \mathcal{O}\Big(T_* \log(T^{m+1}) + T^{m+1} B_k(T_*) \Big), \quad \text{(max is attained at } r = T\text{)}, \\
&= \mathcal{O}\Big(T_* \log(T) + T^{m+1} B_k(T_*) \Big). \qquad (29)
\end{aligned}$$

From Proposition 4.4, we know that $B_k(T_*) \leq \mathcal{O}(e^{-\beta T_*^{1/m}})$. Replacing it in the above equation

$$\begin{aligned}
\gamma_T &\leq \mathcal{O}\Big(T_* \log T + T^{m+1} e^{-\beta T_*^{1/m}} \Big), \\
&= \mathcal{O}\Big((\log T)^m \log T \Big), \ \text{(by setting } T_* = \Big(\frac{m+1}{\beta} \log T \Big)^m \text{, the second term is } \mathcal{O}(1)\text{)} \\
&= \mathcal{O}\Big((\log T)^{m+1} \Big). \qquad (30)
\end{aligned}$$

Now, simplifying (27) in terms of γ_T

$$\begin{aligned}
\tilde{R}_T &\leq \mathcal{O}\left(\sqrt{T} (\sqrt{\gamma_T + \gamma_T^2} + \sqrt{\gamma_T}) \right), \\
&= \mathcal{O}\left(\sqrt{T} (\gamma_T + \sqrt{\gamma_T}) \right) \text{(as } \gamma_T^2 \text{ is dominant over } \gamma_T\text{)}, \\
&= \mathcal{O}\left(\sqrt{T} (\gamma_T) \right) \text{(as } \gamma_T \text{ is dominant over } \sqrt{\gamma_T}\text{)}, \\
&= \mathcal{O}\Big(\sqrt{T} (\log T)^{m+1} \Big) \ \text{(replacing equation (30))}. \qquad (31)
\end{aligned}$$

Therefore, we have $\mathbb{P}\left[\tilde{R}_T \leq \mathcal{O}\left(\sqrt{T} (\log T)^{m+1} \right) \right] \geq 1 - \delta$, where $\delta > 0$ is the user-defined confidence level. This concludes the proof. □

5. **Application: V2G Incentive Design**

In a standard V2G ecosystem, an Aggregator coordinates a fleet of electric vehicle (EV) owners to trade energy with the Distribution System Operator (DSO). In particular, Aggregators provide ancillary services to local DSOs in response to time-varying tariffs (λ^g), while procuring them from EV users. One key conflict arises because discharging power to the grid accelerates battery degradation of the EVs. This degradation imposes a private, stochastic cost on the EV owner, driven by the unobservable decline of battery health. In addition, the EV users impose a subjective cost for "range anxiety" while participating in V2G. Because the Aggregator cannot directly observe these individual users' costs or forcefully command the EVs to discharge, they must rely on incentives.

5.1. **Mapping V2G to principal-agent model**

To model the interaction between the Aggregator and EV users, we first clarify the research goal to be addressed in this application. To this end, we ask the following research question.

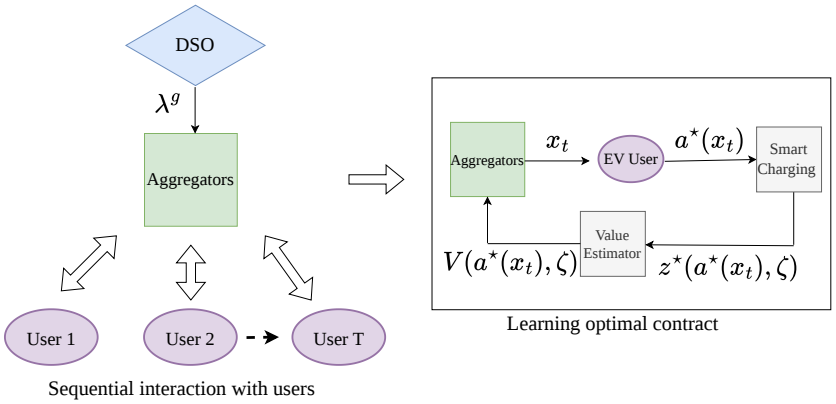


FIGURE 5. V2G incentive design schematic showing sequential interaction between aggregators and EV users.

Problem. *How can an aggregator dynamically learn an adaptive incentive scheme (contract) that elicits optimal EV participation to maximize the Aggregator's revenue, despite the unobservable and stochastic nature of user costs?*

The solution approach is to map the V2G ecosystem into our dynamic principal-agent framework depicted in Figure 1. In particular, we want to mathematically model the interaction between the Aggregator and EV user in the form of (3), where the Aggregator is the principal and the EV user is the agent. As depicted in Figure 5, we model the incentive design problem as a sequential interaction between the Aggregator and an EV user, where at time t, the aggregator offers a contract $x_t := [x^{\mathrm{l}}, x^{\mathrm{h}}] \in [0,1]^2$, specifying monetary incentives for low (l) and high (h) V2G contribution outcome set $\{\mathrm{l}, \mathrm{h}\}$. The EV user responds with a decision $a(x_t) \in \mathcal{A} := \{0,1\}$, where 0 denotes participation and 1 denotes non-participation in the V2G program. Assume there exists a utility function $U^{\mathrm{A}}(x_t, a_t) := \sum_{i\in[2]}(x_t)_i \; p_i(a_t) - c(a_t)$ for the EV user, where $c : \mathcal{A} \to \mathbb{R}$ denotes the user's cost function, and $p : \{0,1\} \to \Delta_2$ is the production function. The EV user optimizes U^{A} to retrieve the best response $a^\star(x_t)$. Based on $a^\star(x_t)$, an outcome $\{l, h\} \ni i_t \sim p(a^\star(x_t))$ is realized. However, the process from the EV user's optimal action $(a^\star(x_t))$ to the realization of an outcome (i_t) includes several steps within the context of V2G operation. In particular, the non-trivial part is to characterize a production function $p(\cdot)$ that links EV users' best response $a^\star(x_t)$ to a stochastic outcome (i_t) for the Aggregator.

5.2. **Characterization of** $p(\cdot)$

To formalize the characterization, we split the V2G operation into two parts: (1) *smart charging*, and (2) *value estimator*. The *smart charging* problem, parameterized in $a^\star(x_t)$, provides a stochastic charging profile, which is passed through a deterministic *value estimator* to obtain the outcome and value for the aggregator.

Smart charging: Consider the decision vector $z = [P^{\mathrm{c}}, P^{\mathrm{d}}, \delta, E]^\top$ where $P^{\mathrm{c}} \in \mathbb{R}^N$ is the charging power, $P^{\mathrm{d}} \in \mathbb{R}^N$ is the discharging power for N intervals, $E \in \mathbb{R}^N$ is the EV battery energy profile,

and $\delta_i \in \{0,1\}, \forall i \in [N]$ refers the charging ($\delta_i = 1$) or the discharging ($\delta_i = 0$) mode of the EV. Given the user action $a^\star(x_t)$, the smart charging problem is compactly given as

$$\max_z J(z,\zeta) \quad \text{s.t.} \quad z \in \mathcal{Z}(a^\star(x_t),\zeta), \tag{32}$$

where J encodes net charging benefit for the EV user, $\mathcal{Z}$ encodes the operational constraints, and the uncertain vector is $\zeta = [N, \eta^{\mathrm{c}}, \eta^{\mathrm{d}}, E_{\text{initial}}, E_{\text{desired}}]$. A detailed formulation of smart charging is given in Appendix B, along with an explanation of each parameter in ζ. As the uncertain vector ζ is chosen by the user, we impose the following assumptions.

Assumption 5.1 (Smart charging setting)**.** *Consider problem* (32) *where the uncertain vector* ζ *is defined on the probability space* $(\Omega, \mathcal{F}, \mathbb{P})$ *with support* Ξ*.*

(i) **Measurability:** *For every* $a \in \mathcal{A}$*, the correspondence* $\zeta \mapsto \mathcal{Z}(a,\zeta)$ *has a measurable graph, while the map* $(z,\zeta) \mapsto J(z,\zeta)$ *is Borel measurable in* ζ*.*

(ii) **Feasibility:** *We assume that the feasibility set* $\mathcal{Z}(a,\zeta)$ *in* (32) *is non-empty and compact for all* $\zeta \in \Xi$ *and* $a \in \mathcal{A}$*.*

Assumption 5.1 helps construct a well-defined production function $p(\cdot)$, as we will show it later. The smart charging profile $\{\hat{P}^{\mathrm{c}}(a^\star(x_t),\zeta), \hat{P}^{\mathrm{d}}(a^\star(x_t),\zeta)\}$ is used in the next step for the value estimator (see Figure 5) where $\hat{P}^{\mathrm{c}}(a^\star(x_t),\zeta)$ and $\hat{P}^{\mathrm{d}}(a^\star(x_t),\zeta)$ are part of the solution of (32) for a realization of the uncertain parameter ζ.

Value estimator: To formalize the value estimator, we first define the net profit of the Aggregator (after transactions with DSO) stands as $J^P(a^\star(x_t),\zeta) := \Big[J(z^\star(a^\star(x_t)),\zeta) + (\hat{P}^d(a^\star(x_t),\zeta) - \hat{P}^c(a^\star(x_t),\zeta))^\top \lambda^g \Delta t\Big]$, where $z^\star(a^\star(x_t))$ is the solution of (32). Then we define the value estimator function $V : \mathcal{A} \times \Xi \to \{v^l, v^h\}$ such that, $v^{\mathrm{h}} \geq v^{\mathrm{l}}$, and

$$V(a^\star(x_t),\zeta) = \begin{cases} v^{\mathrm{h}} & \text{if, } J_t^P(a^\star(x_t),\zeta) \geq J', \\ v^{\mathrm{l}} & \text{otherwise,} \end{cases} \tag{33}$$

where the threshold profit J' and the values $v = [v^{\mathrm{l}}, v^{\mathrm{h}}]^\top$ are chosen by the aggregator. Since ζ is a well-defined random variable, we assume that v^{h} and v^{l} are realized with probabilities p^{h} and p^{l}, respectively. Essentially, this translates to defining the production function $p(a) := [p^{\mathrm{l}}, p^{\mathrm{h}}]^\top$. To validate the mathematical correctness of $p(a)$, we need to show that it is possible to define $p^{\mathrm{h}} = \mathbb{P}_\zeta[J_t^P(a,\zeta) \geq J']$ and $p^{\mathrm{l}} = 1 - p^{\mathrm{h}}$ for any $a \in \mathcal{A}$. Thus, we provide the following Proposition.

Proposition 5.2 (Equivalence to principal-agent)**.** *Under Assumption 5.1, the proposed V2G incentive scheme constitutes a Principal-Agent problem wherein the smart charging formulation in* (32) *and the value estimator in* (33) *induce a well-defined production function* $p : \mathcal{A} \to \Delta_2$*.*

The proof is given in Appendix A.3. Finally, the Aggregator aims to find the optimal contract $x^\star = \arg\max_{x \in [0,1]^2} \{U^{\mathrm{PA}}(x) = \sum_{i \in [2]} (v - x)_i p_i(a^\star(x))\}$ using the reward values (i.e., $y_t = (v - x_t)_i,\ i \sim p(a^\star(x_t))$) from the interaction with sequentially arriving EV users. To apply our proposed algorithm `Heteroscedastic GP-UCB`, we consider the stochastic modification in the EV user's utility (i.e., from U^{A} to $\tilde{U}^{\mathrm{A}}$), in addition to the required Assumptions for the algorithm.

TABLE 1. Performance comparison of different methods. Effort levels correspond to a_1 (low), a_2 (medium), and a_3 (high).

Agent	Low	Med.	High
Cost $c(a_i)$	$\mathcal{N}(0,\ 0.05)$	$\mathcal{N}(0.1,\ 0.05)$	$\mathcal{N}(0.2,\ 0.05)$
Production $p(a_i)$	[0.95,0.05]	[0.2,0.8]	[0.05,0.95]
Principal			
Outcome i	Low $(i=0)$	High $(i=1)$	
Value v	$v_0 = 0.1$	$v_1 = 0.8$	

6. Numerical Experiments

We present two numerical experiments evaluating the performance of our algorithm against state-of-the-art benchmarks. First, we simulate a complex variant of the 'High-low' example discussed in Section 2.2, where the agent's action space is expanded to $n = 3$. Second, we validate the proposed V2G incentive scheme through a realistic case study. We benchmark our performance against `Agnostic Zooming` and `Discover and Cover` algorithms.

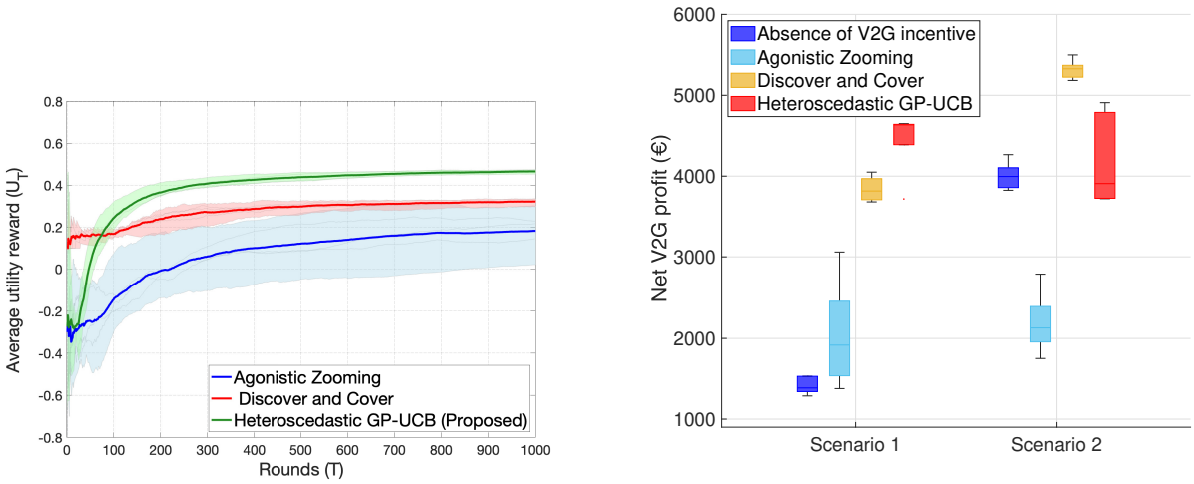
6.1. Performance comparison in 'High-low' example

We retain the contract dimension $m = 2$ while increasing the number of available actions to $n = 3$. The detailed parameters of the feedback environment are provided in Table 1. We compare the algorithms over a horizon of $T = 1000$ rounds, averaging over 5 independent episodes per algorithm. Since the exact optimal contract is intractable to compute *a priori* in complex physical environments, we follow the convention established by [15] and evaluate empirical performance using the *average utility reward*, defined as $U_T = \frac{1}{T}\sum_{t=1}^{T} \tilde{U}^{\mathrm{PA}}(x_t)$. We explicitly note that this metric must be interpreted differently from cumulative regret. While cumulative regret ideally flattens to indicate sub-linear growth, a successful average utility curve converges upward to a positive horizontal asymptote, which represents the maximum achievable per-round yield. Consequently, an algorithm whose average utility converges to a higher, stable plateau demonstrates superior identification and exploitation of the optimal contract.

As illustrated in Figure 6a, our proposed algorithm outperforms the benchmarks. Additionally, we conduct an ablation study to highlight the impact of heteroscedastic noise modeling and kernel selection in Appendix C.2. Overall, the observed performance gap validates the efficacy of our two key contributions: (1) optimization within a continuous domain, and (2) utility geometry-aware kernel choice paired with heteroscedastic noise modeling.

6.2. Results on V2G incentive design

We evaluate the economic impact of our proposed incentive scheme on the Aggregator's net profit. To evaluate the aggregator's net profit from V2G participation of consecutive T users, we define net V2G profit as $J_{\mathrm{net}} := \vartheta \sum_{t=1}^{T} \tilde{U}^{\mathrm{PA}}(x_t)$, where ϑ is the re-normalization constant for conversion



(a) (b)

FIGURE 6. Performance comparison on the synthetic benchmark and V2G profit analysis: (a) Performance comparison of `Heteroscedastic GP-UCB` on the synthetic benchmark. The y-axis displays the **Average Utility Reward** (U_T), not cumulative regret. Therefore, the convergence of our proposed algorithm (green line) to a high, positive horizontal asymptote (≈ 0.45) indicates successful exploitation of the optimal contract, outperforming the baselines, which fail to converge to this optimal yield. (b) Comparison of the net V2G profit (J_{net}) for two scenarios, highlighting the impact of V2G incentive schemes under the benchmarked algorithms.

to Euro (€) currency. Specifically, we compare the net profit under two conditions: (i) a baseline scenario without an incentive mechanism, and (ii) a scenario where the V2G incentive mechanism is managed by the benchmarked algorithms. *Experimental Setup:* We construct a realistic simulation setup for this problem; detailed implementation specifics are provided in Appendix C. We simulate two distinct user types, θ_1 and θ_2. The θ_1 user incurs a low internal cost for V2G participation and is willing to participate without incentives. Conversely, the θ_2 user is averse to participation due to high internal costs. We investigate two scenarios: **Scenario 1 (High Aversion)** We sample θ_2 with probability 0.95. This dominance of reluctance creates an ideal testing ground for the impact of incentive design. **Scenario 2 (Mixed Population):** We uniformly sample θ_1 and θ_2. This presents a challenging setting where generating profit exceeding the baseline (no-incentive) case is difficult. We conduct the interaction with $T = 1000$ sequential users, and gather results over 5 independent episodes.

Discussion of Results: Figure 6b illustrates box plots of the key outcomes. In **Scenario 1**, we observe the substantial impact of the incentive design, with the baseline benchmark `Agnostic Zooming` achieving a minimum **43**% increase in J_{net}. Notably, our algorithm `Heteroscedastic GP-UCB` outperforms all benchmarks by a significant margin. Even in the challenging **Scenario 2**, our algorithm yields a **5**% higher profit compared to the absence of any V2G incentive scheme. This

case study demonstrates that employing specialized learning-based algorithms can unlock diverse, economically viable business models for V2G technology.

7. Limitations and Future Work

Numerical results demonstrate that our algorithm outperforms existing benchmarks. Furthermore, the V2G application confirms its robustness in complex, real-world settings. Nevertheless, we acknowledge the following limitations that suggest clear directions for future research.

Computational Scalability. The kernel matrix inversion in (16) incurs a computational complexity of $\mathcal{O}(T^3)$. While feasible for our experimental horizon ($T = 1000$), this limits long-term deployment. Future work could integrate sparse GP approximations or variational inference to reduce computational complexity, enhancing real-time applications.

Finite Agent Action Space. While we eliminate exponential dependence on action count n, our second contribution still relies on a finite action set (Assumption 4.1). Extending this framework to continuous agent action spaces would further theoretically improve upon existing methods.

Appendix A. Technical Proofs

A.1. Auxiliary Lemma

We first present a preparatory lemma that will be used to establish the main results of this study.

Lemma 1 (Analytic kernel)**.** *Let $D = [0, 1]^m$ be the compact domain in $\mathbb{R}^m$. The kernel in (13) is an analytic function on the domain $D \times D$.*

Proof. We observe that the kernel $k(x, y)$ is a composition of some functions. The broad idea of our proof is that we will show each composition is an analytic function in the domain $D \times D$.

(i) The dot-product kernel $k_{\text{dot}}(x, y) = \sigma_v\langle x, y\rangle + \sigma_b$ is a polynomial in x and y. Therefore, it is analytic in the domain.

(ii) Say $g(x, y) = (1 + k_{\text{dot}}(x, x))(1 + k_{\text{dot}}(y, y))$. We find $k_{\text{dot}}(x, x) = \sigma_b^2 + \sigma_v^2\langle x, x\rangle > \sigma_b^2 > 0$ that implies $(1 + k_{\text{dot}}(x, x)) > 1$. Therefore, $g(x, y) > 1$ is an analytic function as it is the product of two analytic functions, and $\sqrt{g}$ is also an analytic function for all $g(x, y) > 0$.

(iii) Representing $z(x, y) = \frac{k_{\text{dot}}(x,y)}{\sqrt{(1+k_{\text{dot}}(x,x))\cdot(1+k_{\text{dot}}(y,y))}}$, it is an analytic function as the ratio of two analytic functions (k_{dot} and $\sqrt{g}$) remain analytic as long as the denominator is non-zero.

(iv) Finally, it is well known that $\arcsin(z)$ is analytic if $z \in (-1, 1)$. Therefore, we just need to check the range of $z(x, y)$.

Assume $a = k_{\text{dot}}(x, y), b = k_{\text{dot}}(x, x)$ and $c = k_{\text{dot}}(y, y)$. One could show using Cauchy-Schwarz inequality, $a^2 \leq bc$, furthermore it is easy to see that $\frac{a^2}{(1+b)(1+c)} < 1$. Hence $|z(x, y)| < 1$ implies $z \in [-h, h]$ with $h < 1$. This completes the proof. □

A.2. **Proof of Proposition 4.4**

It is known from the spectral theory of integral operators that analytic kernels have eigenvalues λ_s which decay exponentially [25, 32]. Using Lemma 1, we can confirm the same for the eigenvalues of our proposed kernel $k(x, y)$.

Following the above argument, we bound the tail sum $B_k(T_*)$ as

$$B_k(T_*) \leq \sum_{s=T_*+1}^{\infty} Ce^{-vs^{1/m}} \leq C \int_{T_*}^{\infty} e^{-vs^{1/m}} ds$$

for some constants C, and v. It can be shown with some calculation that the above integral transforms to

$$B_k(T_*) \leq C'\Gamma(m, vT_*^{1/m}),$$

where Γ is the upper incomplete gamma function and C' is some constant. Using the formulae $\Gamma(m, q) = (m-1)!e^{-q}\sum_{k=0}^{m-1}(q^k/k!)$ [13], we can show for larger $q = vT_*^{1/m}$, our bound behaves as

$$B_k(T_*) \leq \mathcal{O}((vT_*^{1/m})^{m-1}e^{-vT_*^{1/m}}). \tag{34}$$

One can see that the exponential decay of $e^{-vT_*^{1/m}}$ is much stronger than the polynomial growth of $(vT_*^{1/m})^{m-1}$ as $T_* \to \infty$. Therefore, for any small $\delta > 0$, we can find a constant C_δ such that $(vT_*^{1/m})^{m-1} \leq C_\delta e^{\delta(vT_*^{1/m})}$. Substituting this into the above equation

$$B_k(T_*) \leq \mathcal{O}\left(e^{\delta vT_*^{1/m}} \cdot e^{-vT_*^{1/m}}\right) = \mathcal{O}\left(e^{-(1-\delta)vT_*^{1/m}}\right).$$

By choosing δ such that $\beta = (1-\delta)v > 0$, we obtain

$$B_k(T_*) \leq \mathcal{O}(e^{-\beta T_*^{1/m}}).$$

This concludes the proof. □

A.3. **Proof of Proposition 5.2**

We first recall the setup of the principal-agent problem with respect to the Figure 5. The principal (aggregator) shows the contract $x_t \in [0, 1]^2$ to the agent (EV user), and the agent takes a strategic action $a^\star(x_t)$ in hindsight. As a result, an outcome is realized following the condition in (33). The main focus here is to understand the existence of a production function $p(\cdot)$ as defined in (1). If we manage to show that $J^P(a^\star(x_t), \zeta) = \left[J(z^\star(a^\star(x_t)), \zeta) + (\hat{P}^d(a^\star(x_t), \zeta) - \hat{P}^c(a^\star(x_t), \zeta))^\top \lambda^g \Delta t\right]$, is a well-defined random variable in (33), we can confirm the existence of a production function. To establish that $J^P(a^\star(x_t), \zeta)$ is a random variable, we proceed by the following steps.

(1) Since (35a) indicates that J is continuous in z, we apply Berge's maximum theorem [1, Theorem 17.31] to confirm that $J(z^\star(a^\star(x_t)), \zeta)$ is continuous in ζ.
(2) Moreover, since $J(z, \zeta)$ is a Borel measurable function of ζ (Assumption 5.1), both the optimizer $z^\star(a^\star(x_t))$ and optimal objective value $J(z^\star(a^\star(x_t),), \zeta)$ become Borel measurable in ζ, where the feasibility of the optimizer $z^\star(a^\star(x_t))$ in (32) is ensured by Assumption 5.1.

Finally, $J^P(a^\star(x_t), \zeta)$ is a linear combination of measurable functions, which makes it a measurable function of ζ. Hence, it is a well-defined random variable, which makes the definition p^{h} and p^{l} consistent. This concludes the proof. □

Appendix B. Detailed smart charging formulation

Consider $P^{\mathrm{c}} \in \mathbb{R}^N$ is the charging profile, and $P^{\mathrm{d}} \in \mathbb{R}^N$ is the discharging profile for N intervals. The aggregator defines the buying price $\lambda_i^b = c\lambda_i^g$, $\forall i \in [N]$, and the selling price $\lambda_i^s = c'\lambda_i^g$, $\forall i \in [N]$, where c, c' are the aggregator-defined constants resposible for profit margin such that $c \in (0, 1], c' \geq 1$. $E \in \mathbb{R}^N$ is the EV battery energy profile with $\overline{E}$ and $\underline{E}$ being the limits of the energy level. Furthermore, we use $\delta_i \in \{0, 1\}, \forall i \in [N]$ to refer to the charging ($\delta_i = 1$) or the discharging ($\delta_i = 0$) mode of the EV, and we use η^c/η^d to refer to the charging/discharging coefficients embedding the efficiency and battery capacity of the EV, respectively. Based on the active load profile up to time t, we also have the bounds on the charging/discharging power as $\overline{P}_t$ and $\underline{P}_t$, respectively. Given these terminologies, the smart charging algorithm solves the following optimization problem:

$$
\begin{aligned}
\max_{P^{\mathrm{c}}, P^{\mathrm{d}}, \delta, E} \quad & J := (P^{\mathrm{c}}(\lambda^s)^\top - P^{\mathrm{d}}(\lambda^b)^\top)\Delta t \\
& - \alpha(\|P^c\|_2^2 + \|P^d\|_2^2), && \text{(35a)} \\
\text{s.t.} \quad & E_{i+1} = E_i + \eta^c P^{\mathrm{c}} \Delta t - \eta^d P^{\mathrm{d}} \Delta t, \forall i \in [N], && \text{(35b)} \\
& 0 \leq P_i^{\mathrm{c}} \leq \delta_i \overline{P}_{t,i}, \ \forall i \in [N], && \text{(35c)} \\
& 0 \leq P_i^{\mathrm{d}} \leq (1 - \delta_i)\underline{P}_{t,i}, \ \forall i \in [N], && \text{(35d)} \\
& \underline{E} \leq E_i \leq \overline{E}, \ \forall i \in [N], && \text{(35e)} \\
& E_1 = E_{\text{initial}}, \ E_{N+1} = E_{\text{desired}}, && \text{(35f)} \\
& \delta_j = 1, \ \forall j \in [N], \text{ if } a^\star(x_t) = 1, && \text{(35g)}
\end{aligned}
$$

where the objective is to maximize the net profit $J \in \mathbb{R}$ for the EV user, the constraints include energy dynamics of the EV battery, limiting bounds for charging/discharging power and energy, α is a regularizer coefficient, Δt is the time interval, and finally, $N, E_{\text{initial}}, E_{\text{desired}}$ are the parameters given by the EV users.

Appendix C. Implementation details of numerical experiments and additional results

C.1. Implementation details

We have used Assumption 5.1 to construct the random vector ζ in Table 2 such that its realized values appear realistic for the application, while maintaining the feasibility of the smart charging problem in (35). The values of the fixed parameters in (35) and (33) are given in Table 2. In addition, the tariff data from DSO (λ^g) is taken from Figure 7 of [22]. Given λ^g, one can easily construct λ^b and λ^s using the profit margin constants c and c' from Table 2.

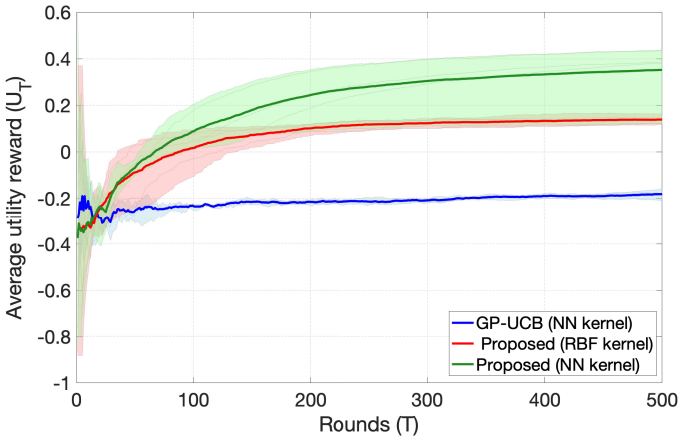


FIGURE 7. Assessing performance of `Heteroscedastic GP-UCB` (paired with Neural Network kernel) under two scenarios. In scenario 1, it outperforms GP-UCB [35] with a Neural Network kernel, validating the effect of considering heteroscedastic noise. In scenario 2, it also outperforms the radial basis function (RBF) kernel, validating our kernel choice.

TABLE 2. Details of uncertain and fixed parameters for numerical simulation.

(a) Uncertain parameters

Uncertain parameters	**Distribution**
N	Uniform$(20, 24)$
η^c and η^d	Uniform(0.9,0.98)
E_{initial}	Uniform(10,15)
E_{desired}	Uniform(20, $N\overline{P}_t\eta^c\Delta t$)

(b) Fixed parameters

Fixed parameters	**Values**
$\underline{E}$	5 kWh
$\overline{E}$	90 kWh
$\overline{P}_t$ and $\underline{P}_t$	11 kW
c	0.7
c'	1.2
J'	80 €
Δ	10 €
v^h	0.9
v^l	0
Δt	15 minutes

C.2. **Additional results**

We present an ablation study in Figure 7 to decouple the contributions of heteroscedastic variance modeling and kernel selection. First, to isolate the impact of the noise model, we compare our approach against standard homoscedastic GP-UCB [35] equipped with the Neural Network kernel. Second, to validate the kernel geometry, we substitute the NN kernel with the standard Radial Basis Function (RBF) kernel within our framework. The results demonstrate that our proposed method outperforms both variations, confirming that both the heteroscedastic structure and the non-stationary kernel are essential for our problem setup.

References

[1] Charalambos D Aliprantis and Kim C Border. *Infinite dimensional analysis: a hitchhiker's guide*. Springer, 2006.

[2] Manjari Asawa and Demosthenis Teneketzis. Multi-armed bandits with switching penalties. *IEEE Transactions on Automatic Control*, 41(3):328–348, 2002.

[3] Moshe Babaioff, Michal Feldman, and Noam Nisan. Combinatorial agency. In *Proceedings of the 7th ACM Conference on Electronic Commerce*, pages 18–28, 2006.

[4] Francesco Bacchiocchi, Matteo Castiglioni, Alberto Marchesi, and Nicola Gatti. Learning optimal contracts: How to exploit small action spaces. In *International Conference on Representation Learning*, volume 2024, pages 11944–11970, 2024.

[5] Jorge Barrera and Alfredo Garcia. Dynamic incentives for congestion control. *IEEE Transactions on Automatic Control*, 60(2):299–310, 2014.

[6] Dimitri P Bertsekas. *Control of uncertain systems with a set-membership description of the uncertainty.* PhD thesis, Massachusetts Institute of Technology, 1971.

[7] Alec N Brooks et al. Vehicle-to-grid demonstration project: Grid regulation ancillary service with a battery electric vehicle. 2002.

[8] Sayak Ray Chowdhury and Aditya Gopalan. On kernelized multi-armed bandits. In *International Conference on Machine Learning*, pages 844–853. PMLR, 2017.

[9] Eric W Cope. Regret and convergence bounds for a class of continuum-armed bandit problems. *IEEE Transactions on Automatic Control*, 54(6):1243–1253, 2009.

[10] Paul Dütting, Michal Feldman, and Inbal Talgam-Cohen. Algorithmic contract theory: A survey. *Foundations and Trends® in Theoretical Computer Science*, 16(3-4):211–411, 2024.

[11] Paul Dütting, Tim Roughgarden, and Inbal Talgam-Cohen. Simple versus optimal contracts. In *Proceedings of the 2019 ACM Conference on Economics and Computation*, pages 369–387, 2019.

[12] Lawrence C Evans. *Measure theory and fine properties of functions*. Chapman and Hall/CRC, 2025.

[13] Izrail Solomonovich Gradshteyn and Iosif Moiseevich Ryzhik. *Table of integrals, series, and products*. Academic press, 2014.

[14] Christopher Harris. You're hired! an examination of crowdsourcing incentive models in human resource tasks. In *Proceedings of the Workshop on Crowdsourcing for Search and Data Mining (CSDM) at the Fourth ACM International Conference on Web Search and Data Mining (WSDM)*, pages 15–18. Hong Kong, China, 2011.

[15] Chien-Ju Ho, Aleksandrs Slivkins, and Jennifer Wortman Vaughan. Adaptive contract design for crowdsourcing markets: Bandit algorithms for repeated principal-agent problems. In *Proceedings of the fifteenth ACM conference on Economics and computation*, pages 359–376, 2014.

[16] Y Ho and D Teneketzis. On the interactions of incentive and information structures. *IEEE Transactions on Automatic Control*, 29(7):647–650, 1984.

[17] Yu-Chi Ho, P Luh, and Ramal Muralidharan. Information structure, stackelberg games, and incentive controllability. *IEEE Transactions on Automatic Control*, 26(2):454–460, 1981.

[18] Matthew Hoffman, Eric Brochu, Nando De Freitas, et al. Portfolio Allocation for Bayesian Optimization. In *UAI*, volume 11, pages 327–336, 2011.

[19] Bengt Holmstrom et al. Moral hazard and observability. *Bell journal of Economics*, 10(1):74–91, 1979.

[20] Johannes Kirschner and Andreas Krause. Information directed sampling and bandits with heteroscedastic noise. In *Conference On Learning Theory*, pages 358–384. PMLR, 2018.

[21] Jean-Jacques Laffont and David Martimort. *The theory of incentives: the principal-agent model*. Princeton university press, 2002.

[22] Shuying Lai, Jing Qiu, Yuechuan Tao, and Junhua Zhao. Pricing for electric vehicle charging stations based on the responsiveness of demand. *IEEE Transactions on Smart Grid*, 14(1):530–544, 2022.

[23] Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

[24] Niklas Lauffer, Mahsa Ghasemi, Abolfazl Hashemi, Yagiz Savas, and Ufuk Topcu. No-regret learning in dynamic stackelberg games. *IEEE Transactions on Automatic Control*, 69(3):1418–1431, 2023.

[25] G Little and JB Reade. Eigenvalues of analytic kernels. *SIAM journal on mathematical analysis*, 15(1):133–136, 1984.

[26] Chunhua Liu, KT Chau, Diyun Wu, and Shuang Gao. Opportunities and challenges of vehicle-to-home, vehicle-to-vehicle, and vehicle-to-grid technologies. *Proceedings of the IEEE*, 101(11):2409–2427, 2013.

[27] Wanchun Liu, Alex S Leong, and Daniel E Quevedo. Thompson sampling for networked control over unknown channels. *Automatica*, 165:111684, 2024.

[28] Thomas M. Cover and Joy A. Thomas. *Elements of information theory*. John Wiley and Sons, 1999.

[29] Chinmay Maheshwari, Kshitij Kulkarni, Manxi Wu, and Shankar Sastry. Adaptive incentive design with learning agents. *IEEE Transactions on Automatic Control*, 71(6):3619–3633, 2026.

[30] Anastasia Makarova, Ilnura Usmanova, Ilija Bogunovic, and Andreas Krause. Risk-averse heteroscedastic bayesian optimization. *Advances in Neural Information Processing Systems*, 34:17235–17245, 2021.

[31] Adam J Mersereau, Paat Rusmevichientong, and John N Tsitsiklis. A structured multiarmed bandit problem and the greedy policy. *IEEE Transactions on Automatic Control*, 54(12):2787–2802, 2009.

[32] Albrecht Pietsch. *Eigenvalues and s-numbers*. Cambridge university press Cambridge, 1987.

[33] Lillian J Ratliff and Tanner Fiez. Adaptive incentive design. *IEEE Transactions on Automatic Control*, 66(8):3871–3878, 2020.

[34] Paat Rusmevichientong and John N Tsitsiklis. Linearly parameterized bandits. *Mathematics of Operations Research*, 35(2):395–411, 2010.

[35] Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias W Seeger. Information-theoretic regret bounds for gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory*, 58(5):3250–3265, 2012.

[36] Koen van Heuveln, Rishabh Ghotge, Jan Anne Annema, Esther van Bergen, Bert van Wee, and Udo Pesch. Factors influencing consumer acceptance of vehicle-to-grid by electric vehicle drivers in the Netherlands. *Travel Behaviour and Society*, 24:34–45, 2021.

[37] Tonghan Wang, Paul Duetting, Dmitry Ivanov, Inbal Talgam-Cohen, and David C Parkes. Deep contract design via discontinuous networks. *Advances in Neural Information Processing Systems*, 36:65818–65836, 2023.

[38] Christopher KI Williams. Computation with infinite neural networks. *Neural Computation*, 10(5):1203–1216, 1998.

[39] Le Yang, Siyang Gao, Cheng Li, and Yi Wang. Stochastically constrained best arm identification with thompson sampling. *Automatica*, 176:112223, 2025.

[40] Xinyang Zhou, Emiliano Dall'Anese, Lijun Chen, and Andrea Simonetto. An incentive-based online optimization framework for distribution grids. *IEEE Transactions on Automatic Control*, 63(7):2019–2031, 2017.

[41] Banghua Zhu, Stephen Bates, Zhuoran Yang, Yixin Wang, Jiantao Jiao, and Michael I Jordan. The sample complexity of online contract design. In *Proceedings of the 24th ACM Conference on Economics and Computation*, pages 1188–1188, 2023.

[42] Jingxuan Zhu and Ji Liu. Distributed multiarmed bandits. *IEEE Transactions on Automatic Control*, 68(5):3025–3040, 2023.