Riemannian Manifold Learning for Stackelberg Games with Neural Flow Representations

Larkin Liu[1,2] *, **Kashif Rasul**[3], **Yutong Chao**[1], **Jalal Etesami**[1, 4]
[1]Technische Universität München
[2]Riebaki AI
[3]Hugging Face, Inc.
[4]Munich Institute of Robotics and Machine Intelligence (MIRMI)
larkin.liu@tum.de, kashif.rasul@gmail.com, cyut@cit.tum.de, j.etesami@tum.de

Abstract

We present a novel framework for online learning in Stackelberg general-sum games, where two agents, the leader and follower, engage in sequential turn-based interactions. At the core of this approach is a learned diffeomorphism that maps the joint action space to a smooth spherical Riemannian manifold, referred to as the *Stackelberg manifold*. This mapping, facilitated by neural normalizing flows, ensures the formation of tractable isoplanar subspaces, enabling efficient techniques for online learning. Leveraging the linearity of the agents' reward functions on the Stackelberg manifold, our construct allows the application of linear bandit algorithms. We then provide a rigorous theoretical basis for regret minimization on the learned manifold and establish bounds on the simple regret for learning Stackelberg equilibrium. This integration of manifold learning into game theory uncovers a previously unrecognized potential for neural normalizing flows as an effective tool for multi-agent learning. We present empirical results demonstrating the effectiveness of our approach compared to standard baselines, with applications spanning domains such as cybersecurity and economic supply chain optimization.

Extended version — https://arxiv.org/abs/2502.05498

1 Introduction

A Stackelberg game consists of a sequential decision-making process involving two agents, a leader and a follower. This framework, introduced in (von Stackelberg 1934) models hierarchical strategic interactions where the leader moves first, anticipating the follower's best response, and then the follower reacts accordingly. These games have become central to understanding interactions in various fields, from economics to societal security, providing a formal method for analyzing situations where one party commits to a strategy before the other, affecting the subsequent decision-making process and reward outcomes. Over time, Stackelberg games have evolved to address more complex environments, incorporating factors like imperfect information and no-regret learning of system parameters. The solution revolves around finding a Stackelberg equilibrium, where the leader optimizes her strategy assuming the follower type, which affects how the follower optimizes his utility based on the leader's action. (Korzhyk, Conitzer, and Parr 2010; Kar et al. 2015).

Practical applications of Stackelberg games often face key challenges, primarily due to uncertainty about the follower's rationality or preferences, and imperfect information on reward outcomes. These issues complicate the leader's decision-making. In security domains, randomized strategies and robust optimization mitigate risks from incomplete information, as seen in systems like ARMOR and PROTECT (Jiang et al. 2013; Kar et al. 2015, 2017; Jain et al. 2011; Shieh et al. 2012). Stackelberg games are also used in supply chain optimization, addressing uncertainties like demand (Liu and Rong 2024; Cesa-Bianchi et al. 2023), and in conversational AI, where agents anticipate user behaviour to adjust responses (Nguyen et al. 2014). For non-cooperative multi-agent games with additive noise, sublinear regret is achievable via gradient-based methods like AdaGrad (Duchi, Hazan, and Singer 2011), though constrained by noise magnitude (Hsieh et al. 2023). These settings often extend to unlimited players, with regret worsening as the player count grows. We focus on two-player Stackelberg games with well-defined best response functions, common in economics and adversarial machine learning (Zhou and Kantarcioglu 2016; Wang et al. 2024).

Problem Setting: We consider a two-player Stackelberg game where player A leads and player B responds. Stackelberg games are sequential, meaning that the players take turns and, the follower can *best respond* to the leader's action, given information available to him. The best response of player B lies on a manifold within a subspace of the joint action space $\mathcal{A} \times \mathcal{B}$. We define this Stackelberg game setting in the framework of optimal transport, where the structure of the best response function $\mathfrak{B}(\cdot)$ gleans simplifications to the solution methodology to obtain Stackelberg regret. This research focuses on applying multi-armed bandit (MAB) methods, particularly in Stackelberg equilibrium settings, to achieve sublinear regret. It explores the utilization of geometric topologies to better understand agent behaviour and simplify computations in a game theoretic manner.

Key Contributions: We introduce a novel algorithm that advances Stackelberg learning under imperfect information, akin to (Balcan et al. 2015) and (Haghtalab et al. 2022), providing a systematic framework for efficiently solving equilibrium in such settings. Central to our work is a feature map

*Corresponding author.

using neural normalizing flows, transforming the joint action space into a tractable embedding, the *Stackelberg manifold*. Leveraging its geodesic properties and exploiting the linearity of the agents' reward functions on the manifold, our approach enables efficient computation of Stackelberg equilibria under no-regret learning, especially with parameter uncertainty. We also establish a rigorous theoretical foundation for optimizing Stackelberg games on spherical manifolds. Empirical simulations in supply chain management and cybersecurity validate our method, showing superior computational efficiency and regret minimization compared to standard baselines.

2 Formal Definitions

In a Stackelberg game, two players take turns executing their actions. Player A is the leader, she acts first with action $\mathbf{a}$ selected from her action space $\mathcal{A}$. Player B is the follower, he acts second with action $\mathbf{b} \in \mathcal{B}$. The follower acts in response to the leader's action, and both players earn a joint payoff as a function of their actions.

2.1 Repeated Stackelberg Games

In a repeated Stackelberg game, the leader chooses actions $\mathbf{a}^t \in \mathcal{A}$, and the follower reacts with actions $\mathbf{b}^t \in \mathcal{B}$ at each round $t = 1, \ldots, T$. The leader's strategy $\pi_A(\cdot|\mathcal{H}_t)$ is a probability distribution over the action space $\mathcal{A}$ which selects $\mathbf{a}^t$ based on past joint actions up to time t, i.e., $\mathcal{H}_t := \{(\mathbf{a}^\tau, \mathbf{b}^\tau) | \tau < t\}$. Similarly, the follower's strategy $\pi_B(\cdot|\mathcal{H}_t)$ is a conditional probability distribution over $\mathcal{B}$ which determines $\mathbf{b}^t$ given the full history, $\mathcal{H}_t := \mathcal{H}_t \cup \{\mathbf{a}^t\}$.

Best Response Strategy of the Follower: To be specific, the follower selects his best response strategy at round t by maximizing his expected reward function $\mu_B(\mathbf{a}, \mathbf{b}) : \mathcal{A} \times \mathcal{B} \to \mathbb{R}$ given that the leader has played action $\mathbf{a}^t$. Since we assume that the reward function solely depends on the most recent pairs of actions, the follower's best strategy is first order Markov, i.e., $\pi_B(\cdot|\mathcal{H}_t) = \pi_B(\cdot|\mathbf{a}^t)$. Formally, the follower's best response at round t is given by,

$$\pi_B^*(\mathbf{b}|\mathbf{a}^t) \equiv \underset{\pi_B \in \Pi_\mathcal{B}}{\operatorname{argmax}} \mathbb{E}_{\pi_B}[\mu_B(\mathbf{a}, \mathbf{b})|\mathbf{a} = \mathbf{a}^t], \tag{2.1}$$

$$\mathfrak{B}(\mathbf{a}^t) \equiv \{\mathbf{b} \in \mathcal{B} | \pi_B^*(\mathbf{b}|\mathbf{a}^t) > 0\}. \tag{2.2}$$

where $\Pi_\mathcal{B}$ is the space of probability distributions over the action space $\mathcal{B}$ and the expectation is taken with respect to the strategy of the follower. In this case, we can define the set of follower's best responses in Eq. (2.2). Analogously, the leader aims at maximizing the expected utility $\mathbb{E}[\mu_A(\mathbf{a}^t, \mathbf{b}^t)] : \mathcal{A} \times \mathcal{B} \to \mathbb{R}$ that is a deterministic function solely driven by her action $\mathbf{a}^t$ followed by the reaction of the follower $\mathbf{b}^t$.

Stackelberg Equilibrium: Consider a follower whose best response is optimal. We denote this scenario as Stackelberg Oracle (SOC) learning. From the leader's perspective, the uncertainty is not necessarily over the system, but rather the strategy of the follower $\pi_B(\cdot)$. *Stackelberg equilibrium* (π_A^*, π_B^*) is achieved when the follower is best responding, according to Eq. (2.2), and the leader acts with an optimal policy given the best response of the follower,

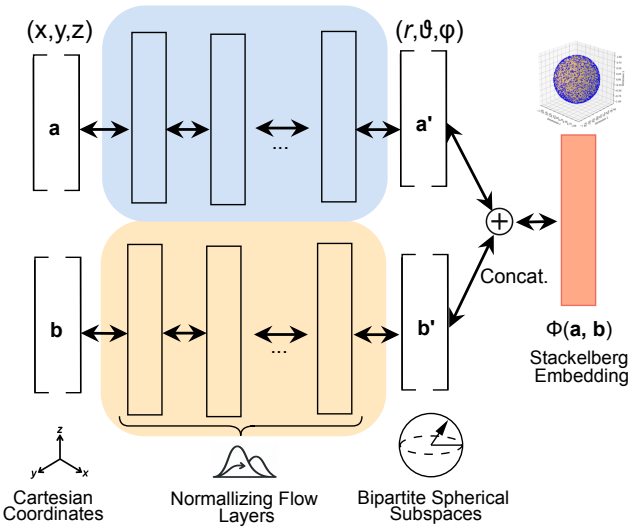

Figure 1: **Bipartite & Bijective Neural Flow Architecture:** We illustrate the bipartite structure of the normalizing flow architecture. Two players present joint actions $(\mathbf{a}, \mathbf{b})$, where each vector is mapped separately through a series of bijective transforms consisting of normalizing flow layers. Each player's action independently controls one subspace of the spherical manifold. The sequence of bijective transformations retain a fully bijective network from the ambient joint action space to the manifold space $\mathbf{\Phi}(\mathbf{a}, \mathbf{b})$. The network is invertible by design, and features a bipartite input.

$$\pi_A^* \equiv \arg \max_{\pi_A \in \Pi_\mathcal{A}} \mathbb{E}_{\pi_A, \pi_B^*}[\mu_A(\mathbf{a}, \mathbf{b})], \tag{2.3}$$

$$\mathbb{E}_{\pi_A, \pi_B^*}[\mu_A] = \int_\mathcal{A} \pi_A(\mathbf{a}) \int_\mathcal{B} \mu_A(\mathbf{a}, \mathbf{b}) \pi_B^*(\mathbf{b}|\mathbf{a}) \, d\mathbf{b} \, d\mathbf{a}.$$

2.2 The Stackelberg Manifold

To address the complexity of solving for Stackelberg equilibrium under uncertainty, we propose mapping actions from the ambient joint action space onto a well behaved spherical manifold $\mathbf{\Phi}$. This approach offers key advantages. First it simplifies the problem by optimizing on an intrinsic geometric structure (e.g., a unit sphere) enabling faster computation and convenient constraint enforcement. The manifold's smoothness also allows for efficient optimization via methods like Riemannian gradient descent (Bonnabel 2013). The core idea is to shift the complexity of learning in Stackelberg games by transforming the action space into a more tractable representation. Rather than relying on classical multi-agent learning, we instead learn a neural representation for $\mathbf{\Phi}$ to enable simplified equilibrium learning.

This concept of mapping the data from the ambient space, in our case defined by the joint action space $\mathcal{A} \times \mathcal{B}$, onto a latent space $\mathbf{\Phi}$ has been explored in several prior works. For a well defined manifold, the typical approach is to learn a diffeomorphism between the ambient data space, and the objective manifold, which is a subspace of the ambient data space (Rezende et al. 2020) (Gemici, Rezende, and Mohamed 2016). Suppose the manifold is not given, or there lies flexibility in defining the structure of such a manifold, certain

manifold learning techniques could be devised (Brehmer and Cranmer 2020). These approaches typically define invertible probability density maps between the ambient data space, the latent space, and the manifold space.

2.3 Normalizing Flows for Action Space Projection

We leverage *normalizing flows* to map a compact joint action space $\mathcal{A} \times \mathcal{B} \subseteq [0, K]^D$ with arbitrarily large $K \in \mathbb{R}$, onto a manifold, $\mathbf{\Phi}$ embedded in $\mathbb{R}^D$ (Rezende and Mohamed 2015; Dinh, Sohl-Dickstein, and Bengio 2016; Papamakarios et al. 2021). Normalizing flows are a class of generative models that transform a high dimensional simple distribution into a complex one through a series of invertible bijective mappings using neural networks that are computationally tractable. The joint action space consists of actions taken by two agents, denoted as $\mathbf{a} \in \mathcal{A}$ and $\mathbf{b} \in \mathcal{B}$, modelled via normalizing flows to ensure bijectivity and a tractable density estimate. Let $x \in \mathcal{A} \times \mathcal{B}$, the model density $p_X(x)$ for a data point $x \in \mathbb{R}^D$ is given by,

$$p_X(x) = p_Z(f_{\texttt{nf}}(x)) \left| \det \left(\frac{\partial f_{\texttt{nf}}(x)}{\partial x} \right) \right|. \tag{2.4}$$

Here, Z represents the latent space with a simple distribution, and $|\det (\partial f_{\texttt{nf}}(x)/\partial x)|$ is the Jacobian determinant of the transformation $f_{\texttt{nf}} : \mathbb{R}^D \to \mathbb{R}^D$. Several open-source methodologies and codebases have been developed to address this manifold mapping problem via normalizing flows (Brehmer and Cranmer 2020). We adopt the `nflows` package from (Durkan et al. 2020) into our approach. The key contribution of our application is the isolation of the input heads into two separate partitions of normalizing flows followed by concatenation of the outputs (see Fig. 1). This allows us to control the subspace induced by the leader's action, $\mathbf{a} \in \mathcal{A}$, independently. (We provide detailed model specifications in Appendix D.) The neural flow network is designed to be invertible, but could experience some reconstruction error due to numerical instability or rounding errors - which we aim to minimize. Our empirical results demonstrate that this error is negligible (see Table 1).

2.4 Specifications of the Feature Map $\phi(\mathbf{a}, \mathbf{b})$

Feature Map $\phi(\cdot)$: Common in the linear bandit literature, we propose a *feature map* $\phi : \mathcal{A} \times \mathcal{B} \mapsto \mathbb{R}^D$ which maps the joint action spaces of the agents, $\mathcal{A} \times \mathcal{B}$, to $\mathbb{R}^D$ (Zanette et al. 2021; Moradipari et al. 2022; Amani, Alizadeh, and Thrampoulidis 2019). Further, we introduce a concept known as the *Stackelberg manifold*, denoted by $\mathbf{\Phi}$, which is defined as the image of ϕ over the joint action space domain $\mathcal{A} \times \mathcal{B}$,

$$\mathbf{\Phi} \equiv Im(\phi) = \{\phi(\mathbf{a}, \mathbf{b}) | \mathbf{a} \in \mathcal{A}, \mathbf{b} \in \mathcal{B}\}. \tag{2.5}$$

In principle, the mapping ϕ can be constructed via any means. In our case, a normalizing neural flow network (but possibly any other architecture), but should abide by imposed characteristics described later in this Section. To be precise, $\hat{\phi}$ should denote our learned representation of the *ideal* map ϕ. Provided that we only have access to $\hat{\phi}$, purely for notational convenience, we will use ϕ to represent $\hat{\phi}$ moving forward.

Definition 2.1. ***Bipartite Spherical Map*** *$\phi(\mathbf{a}, \mathbf{b})$: Let $\mathbf{a} \in \mathcal{A}$ and $\mathbf{b} \in \mathcal{B}$, and define a mapping $\phi : \mathcal{A} \times \mathcal{B} \to \mathcal{S}^{(D-1)}$ from Cartesian coordinates to spherical coordinates on the D-dimensional unit sphere $\mathcal{S}^{(D-1)}$. The spherical coordinates are partitioned such that , $\mathbf{a}$ parametrizes a subset of the spherical coordinates $\nu_{\mathbf{a}}(\mathbf{a})$, and $\mathbf{b}$ parametrizes the remaining coordinates $\nu_{\mathbf{b}}(\mathbf{b})$. Also, $\nu_{\mathbf{a}} \cap \nu_{\mathbf{b}} = \emptyset$, meaning the partitions are disjoint. Thus, the full mapping is given by,*

$$\phi(\mathbf{a}, \mathbf{b}) := \big(\nu_{\mathbf{a}}(\mathbf{a}), \nu_{\mathbf{b}}(\mathbf{b})\big)^{\intercal} \in \mathcal{S}^{(D-1)}.$$

Mapping to a Spherical Manifold: The transformation from spherical coordinates to Cartesian coordinates is used to map input features onto an D-dimensional spherical manifold. Therefore, in addition to the properties of our feature map ϕ, we also enforce ϕ as a bipartite spherical map from Def. 2.1. This bipartite spherical map constructs a disjoint spherical mapping to parameterize two subspaces in $\mathbf{\Phi}$. Given two heads in the neural architecture, the head from A specifically controls the azimuthal spherical coordinate and the head from B controls other coordinates. The justification for this mapping involves trade-off between learning an optimal embedding and reducing the complexity inherent in the native multi-agent problem. When specific multi-agent optimization problems are too complex to solve in their native forms (see Section 5 for examples), we leverage normalizing flows as an enabling link (Brehmer and Cranmer 2020; Durkan et al. 2020) to transform the problem into a simpler representation. (A visualization of the empirical mapping results, showcasing the learned bipartite mapping to $\mathbf{\Phi}$ as a 3D spherical surface, is provided in Appendix E.1 and E.2. This visualization is generated by varying $\mathbf{a}$ or $\mathbf{b}$ to create longitudinal or latitudinal subspaces.)

Manifold Learning: To construct the Stackelberg manifold $\mathbf{\Phi}$, data is first sampled uniformly from the ambient Cartesian space. Given the architecture (see Fig. 1), the Cartesian data is fed through the neural flow architecture to the corresponding image. We train the model parameters such that the image satisfies properties of invertibility for accurate reconstruction. In addition, we train the model to produce an ideal $\mathbf{\Phi}$, which should be a smooth Riemannian manifold, be measurable, compact, and forms Lipschitz-continuous image corresponding to the domain $\mathcal{A} \times \mathcal{B}$, naturally admitting geometric analysis (see Appendix C.2 for details). Furthermore, we would like to ensure maximal spread across the surface, and ensure stability under small perturbations. We construct this manifold by minimizing a loss function $\mathcal{L}(\phi)$, denoted in Eq. (2.6).

$$\mathcal{L}(\phi) = \alpha_N \mathcal{L}_\phi^N + \alpha_R \mathcal{L}_\phi^R + \alpha_L \underbrace{\Big| \|\nabla_{\mathbf{a}} \phi\| + \|\nabla_{\mathbf{b}} \phi\| - C \Big|}_{\text{Lipschitz Loss: } \mathcal{L}_\phi^L}$$
$$+ \alpha_P \underbrace{\text{Var}\Big(\phi(\mathbf{a}, \mathbf{b}) - \phi(\mathfrak{J}_\sigma(\mathbf{a}, \mathbf{b})) \Big)}_{\text{Perturbation Loss: } \mathcal{L}_\phi^P}. \tag{2.6}$$

Loss Function Descriptions: $\mathcal{L}(\phi)$ is an aggregate loss function composed of a convex combination of separate loss functions designed to achieve the ideal manifold behaviour. The normalizing flow loss $\mathcal{L}_\phi^N$ penalizes misalignment of

NFL	N	$\mathsf{Dim}_{\mathcal{A}}$	$\mathsf{Dim}_{\mathcal{B}}$	D	$\mathcal{L}_\phi^N$	$\mathcal{L}(\phi)$
2	1,000	2	2	4	1.78×10^{-7}	2.9×10^{-2}
4	10,000	10	5	15	8.68×10^{-3}	4.5×10^{-1}
6	50,000	10	10	30	9.66×10^{-2}	1.1×10^{-3}
4	10,000	3	3	7	9.38×10^{-8}	5.6×10^{-2}
6	50,000	5	5	10	9.86×10^{-2}	6.8×10^{-1}
2	1,000	10	10	20	1.08×10^{-6}	2.9×10^{-3}

Table 1: Reconstruction error across neural flow configurations and joint action dimensions. All samples, with size N, were trained on a space uniformly sampled from $[0, 1]$, on dimensions $\mathsf{Dim}_{\mathcal{A}}$ and $\mathsf{Dim}_{\mathcal{B}}$, varying the number of NF layers (NFL), and Stackelberg manifold dimension D. All samples trained on 20,000 epochs, and reconstruction error conducted on separately sampled test set of the same sample size.

transformed data with the base distribution while accounting for volume changes from invertible transformations, per Eq. (2.4). Minimizing $\mathcal{L}_\phi^N$ enables efficient bijective mapping from complex data to simpler distributions (see Appendix A.7). $\mathcal{L}_\phi^R$ is the geodesic repulsion loss, penalizing close pairwise elements to maximize coverage over the target manifold (see Appendix A.6). The Lipschitz loss, $\mathcal{L}_\phi^L$, penalizes large gradient deviations with respect to $\mathbf{a}$ and $\mathbf{b}$, keeping the sum of absolute gradients near target $C \in \mathbb{R}$. $\mathfrak{J}_\sigma(\mathbf{a}, \mathbf{b}) : \mathcal{A} \times \mathcal{B} \mapsto \mathcal{A} \times \mathcal{B}$ is a Gaussian perturbation function on the joint action space (defined in Appendix A.5). The perturbation loss, $\mathcal{L}_\phi^P$, denoted as the variance between $\phi(\mathbf{a}, \mathbf{b})$ and $\phi(\mathfrak{J}\sigma(\mathbf{a}, \mathbf{b}))$, should be minimized further ensure Lipschitzness. (We present detailed justification of the loss function design in Appendix C.2.)

2.5 Reward Function

Reward Mechanisms: A Stackelberg game provides two reward functions $\mu_A(\mathbf{a}, \mathbf{b})$ and $\mu_B(\mathbf{a}, \mathbf{b})$. Both of which are linearizable with sub-Gaussian noises, ϵ_A and ϵ_B, i.e.,

$$\mu_A(\mathbf{a}, \mathbf{b}) = \langle \theta_A^*, \phi(\mathbf{a}, \mathbf{b}) \rangle + \epsilon_A, \tag{2.7}$$

$$\mu_B(\mathbf{a}, \mathbf{b}) = \langle \theta_B^*, \phi(\mathbf{a}, \mathbf{b}) \rangle + \epsilon_B. \tag{2.8}$$

We assume zero-mean sub-Gaussian distribution for both ϵ_A and ϵ_B. The objective is to learn the parameters $\theta_A^* \in \mathbb{R}^D$, and possibly as an extension problem θ_B^*. The parameters of the model can be estimated via parameterized regression,

$$\hat{\theta}_t = (\phi_{1:t}\phi_{1:t}^\top + \lambda_{\text{reg}} I)^{-1} \phi_{1:t}^\top \mu_{1:t}, \tag{2.9}$$

where, for A and B, respectively, $\phi_{1:t}$ represents the sequence of $\phi(\cdot)$ values via the feature map given the action sequences $\mathbf{a}_{1:t}$ and $\mathbf{b}_{1:t}$, λ_{reg} serves as a regularization parameter, I is the identity matrix, and $\mu_{1:t}$ are the historical rewards of players A or B (depending on the subscript). Here, we extend the reward structure of classical linear bandits in (Abbasi-Yadkori, Pál, and Szepesvári 2011) to a setting where two players jointly decide on the action sequence. (We provide the conditions for which the estimator is consistent in Appendix A.2.)

Lemma 2.1. ***Linear Relation for Smooth Invertible Maps:*** *Suppose that Y can be expressed as $Y = \langle \tilde{\theta}, \tilde{\phi}(X) \rangle$, where $\tilde{\phi} : \mathbb{R}^d \to \mathbb{R}^d$ is smooth and bijective, and $\tilde{\theta} \in \mathbb{R}^d$. Then, for any $k \geq d$, there exists an alternative set of parameters $\theta \in \mathbb{R}^k$ and corresponding map $\phi : \mathbb{R}^d \to \mathbb{R}^k$ such that, $Y = \langle \theta, \phi(X) \rangle$. (Please see Appendix C.4 for proof.)*

Proof Sketch: The result follows by embedding $\tilde{\phi}(X)$ into $\mathbb{R}^k$ ($k \geq d$) via an alternative map while preserving the inner product structure. Then we construct a diffeomorphism $T(y) = \phi(\tilde{\phi}^{-1}(y))$ between $\tilde{\phi}(X)$ and any alternative smooth bijection $\phi(X)$. By defining $\theta = (J_T^\top)^{-1}\tilde{\theta}$ via the Jacobian of T to ensure $Y = \langle \theta, \phi(X) \rangle$ holds.

Linearity by Design: Lemma 2.1 provides us a theoretical basis for exchanging one feature map $\tilde{\phi}$, to another ϕ, so long as the original feature map, $\tilde{\phi}$, is smooth and bijective. This preserves the functional representation of the primal expression $\langle \tilde{\theta}, \tilde{\phi} \rangle$ under $\langle \theta, \phi \rangle$. Therefore, in principle, an infinite amount of valid alternative mappings of ϕ exist, allowing us to construct a mapping with desired properties. Subsequently, we could adopt the linear bandit framework (Chu et al. 2011; Cesa-Bianchi and Lugosi 2006; Lattimore and Szepesvári 2020) for our Stackelberg manifold $\mathbf{\Phi}$, ensuring that the structure of the reward function remains equivalent for any alterative feature map ϕ.

3 Optimization of Stackelberg Games

Optimization under Parameter Uncertainty: In general we can solve Stackelberg games as a bilevel optimization problem. Suppose that after observing t samples under parameter uncertainty, for some no-regret learning algorithm, the uncertainty among parameters θ, is characterized by,

$$\texttt{Ball}(\theta^*, \mathcal{C}_\theta(t)) := \Big\{ \theta : \|\theta^* - \theta\| \leq \mathcal{C}_\theta(t) \Big\}, \tag{3.1}$$

with probability at least $1 - \delta(t)$. In this formulation, $\|\cdot\|$ denotes some norm in the space of parameters. Assuming a *pessimistic* leader, the optimization problem under parameter uncertainty at round t can be expressed as,

$$\pi_A^* \equiv \arg \max_{\pi_A \in \Pi_A} \min_{\theta_A} \mathbb{E}_{\pi_A, \pi_B^*(\pi_A)} \big[\langle \theta_A, \phi(\mathbf{a}, \mathbf{b}) \rangle \big], \tag{3.2}$$

$$\pi_B^*(\pi_A) \equiv \arg \max_{\pi_B \in \Pi_B} \max_{\theta_B} \mathbb{E}_{\pi_A, \pi_B} \big[\langle \theta_B, \phi(\mathbf{a}, \mathbf{b}) \rangle \big], \tag{3.3}$$

Given $\pi_B^*(\cdot)$ in Eq. (3.3), let us define,

$$\underline{\mathcal{H}}(\theta_A^*, t) \equiv \max_{\pi_A \in \Pi_A} \min_{\theta_A} \mathbb{E}_{\pi_A, \pi_B^*(\pi_A)} \big[\langle \theta_A, \phi(\mathbf{a}, \mathbf{b}) \rangle \big], \tag{3.4}$$

$$\overline{\mathcal{H}}(\theta_A^*, t) \equiv \max_{\pi_A \in \Pi_A} \max_{\theta_A} \mathbb{E}_{\pi_A, \pi_B^*(\pi_A)} \big[\langle \theta_A, \phi(\mathbf{a}, \mathbf{b}) \rangle \big]. \tag{3.5}$$

such that, $\theta_A \in \texttt{Ball}(\theta_A^*, \mathcal{C}_\theta(t))$, and $\theta_B \in \texttt{Ball}(\theta_B^*, \mathcal{C}_\theta(t))$.

The above expressions represent the pessimistic and optimistic leader's estimates of her average reward. We can see from the structure of Eq. (3.2) to Eq. (3.5), the resemblance to a bi-level optimization problem. The solutions to such problems are often computationally demanding and/or complex to formulate (Beck, Ljubić, and Schmidt 2023; Sinha, Malo, and Deb 2017). (We further provide a discussion of such optimization methods in Appendices B.1 and B.2.)

4 Geodesic Online Learning

To enable efficient multi-agent online learning on the Stackelberg manifold, $\mathbf{\Phi}$, we enforce $\mathbf{\Phi}$ to be a *locally convex manifold*. Our definition of a locally convex manifold is a one where the geodesic between any two points on the manifold is contained within or forms a geodesically convex set (see Appendix Def. C.1). More specifically, we employ the use of a *spherical manifold*, which is locally convex. Under geodesic convexity, the follower's optimal best response strategy is uniquely deterministic (see Appendix Lemma C.1).

4.1 Bilevel Optimization

Provided that we can transform data from the joint action space (or ambient space) onto a spherical manifold, we can leverage the properties of the D-sphere to determine the best response for the follower and minimize the corresponding Stackelberg regret. Consider the reward function structure outlined in Section 2.5. In general, the reward of each agent has the form $\mu = \langle \theta, \phi(\mathbf{a}, \mathbf{b}) \rangle$, where θ represents a D-dimensional vector, and $\phi(\mathbf{a}, \mathbf{b}) \in \mathbf{\Phi}$ encodes the joint action representation. In the Stackelberg game, θ_A and θ_B, referred to as *objective vectors*, characterize each agent's reward parameters. The leader's first-mover advantage defines a restricted subspace on $\mathbf{\Phi}$, within which the follower must then optimize his reward. Specifically, he must find the element in $\mathbf{\Phi}$ that maximizes the inner product subject to the leader's constraints given the follower's best response, $\mathfrak{B}(\mathbf{a})$, defined in Eq. (2.2), thereby attaining a Stackelberg equilibrium.

We denote the *geodesic distance* between two vectors, denoted as $\mathfrak{G}(\theta_A, \theta_B)$, for a unit-spherical manifold, as,

$$\mathfrak{G}(\theta_A, \theta_B) = \arccos\Big(\frac{\langle \theta_A, \theta_B \rangle}{\|\theta_A\| \|\theta_B\|}\Big). \tag{4.1}$$

In a D-dimensional sphere, a fully cooperative game exhibits co-directional objective vectors (i.e. without divergence), implying that the inner-product-maximizing solution in $\mathbf{\Phi}$ must be collinear with θ_A, mutatis mutandis for θ_B. Moving forward, we use the convention ξ_{θ_A} and ξ_{θ_B} to denote the projection of the θ_A and θ_B onto $\mathbf{\Phi}$.

Lemma 4.1. ***Geodesic Distance and Closeness:*** *Let $\Phi \subset \mathbb{R}^D$ be a manifold serving as a boundary of a convex set in $\mathbb{R}^D$. Given θ, let $\xi_\theta \in \Phi$ be the point on the manifold that maximizes $\langle \theta, \xi_\theta \rangle$, and is orthogonal to $\mathbf{\Phi}$ at the point of intersection. For any two points on the manifold $\xi_{\theta_A}, \xi_{\theta_B} \in \Phi$, if the geodesic distance between ξ_θ and ξ_{θ_A} is greater than the geodesic distance between ξ_θ and ξ_{θ_B}, $\mathfrak{G}(\xi_\theta, \xi_{\theta_A}) > \mathfrak{G}(\xi_\theta, \xi_{\theta_B})$, then the dot product satisfies $\langle \theta, \xi_{\theta_A} \rangle < \langle \theta, \xi_{\theta_B} \rangle$. (Proof in Appendix C.6.)*

Sketch of Proof: We establish that on a spherical manifold, the dot product with ξ_θ decreases as geodesic distance from any point on the manifold to the ξ_θ increases, linking vector alignment to geodesic distance, and establish the inverse relation between dot product $\langle \xi_\theta, \xi \rangle$ and the cosine relationship. This allows us to frame our optimization problem as a geodesic distance minimization problem.

4.2 Regret Definitions

Definition 4.1. ***Stackelberg Regret:*** *We define Stackelberg regret, denoted as R_A^T for the leader, measuring the difference in cumulative rewards between a best responding follower and an optimal leader in a perfect information setting, against best responding follower and leader exhibiting bounded rationality, the leader acts rationally given the estimates of her expected reward function. Specifically,*

$$R_A^T = \sum_{t=1}^{T} \mathbb{E}\Big[\max_{\mathbf{a} \in \mathcal{A}} \mu_A(\mathbf{a}, \mathfrak{B}(\mathbf{a})) - \mu_A(\mathbf{a}^t, \mathfrak{B}(\mathbf{a}^t))\Big]$$

The leader selects $\mathbf{a}^t$ from policy π_A based on their estimates $\hat{\theta}_A$ and $\hat{\theta}_B$, following Eq. (3.2) and Eq. (3.3). Committing to π_A, the leader aims to maximize their reward while accounting for uncertainty in the follower's response, which is estimated rationally within a confidence interval. Our algorithm ensures a no-regret learning process by minimizing *Stackelberg regret* bounded by the proposed learning algorithm (Alg. 1). To evaluate this, we derive a closed-form expression for the gap between the leader's expected reward under the optimal policy and any algorithm.

Definition 4.2. ***Simple Regret:*** *We define the simple regret, where with probability $1-\delta$ at time t,*

$$\begin{aligned} reg(t) &\equiv \langle \theta_A^*, \phi(\mathbf{a}^*, \mathfrak{B}(\mathbf{a}^*)) \rangle - \langle \theta_A^*, \phi(\mathbf{a}^t, \mathfrak{B}(\mathbf{a}^t)) \rangle \quad (4.2) \\ &\leq \bar{\mathcal{H}}(\theta_A^*, t) - \underline{\mathcal{H}}(\theta_A^*, t) \quad (4.3) \end{aligned}$$

This assumes that the leader is acting under the bounded rationality assumption.

4.3 Quantifying Uncertainty on the Manifold

We now revisit the parameter uncertainty constraints introduced in Sec. 3, which dictate the uncertainty of a given learning algorithm, characterized by an uncertainty radius $\mathcal{C}_\theta(t)$. Given the feature map $\phi(\cdot)$, which adheres to the linear reward assumptions, particularly with respect to the covariance matrix of the regression (as outlined in Sec. 2.5), the learning leader can apply any bandit learning algorithm that imposes a high-probability bound on the parameter estimate. This constraint is formalized in Eq. (3.1) by the uncertainty region $\mathcal{C}_\theta(t)$. Let us define $\overleftrightarrow{\mathbf{\Phi}}_{\mathbf{a}}$ and $\overleftrightarrow{\mathbf{\Phi}}_{\mathbf{b}}$ as two subspaces, which we will use to analyze the leader's actions under these uncertainty constraints.

$$\overleftrightarrow{\mathbf{\Phi}}_{\mathbf{a}} := \{\phi(\mathbf{a}, \mathbf{b}') | \mathbf{b}' \in \mathcal{B}\}, \ \overleftrightarrow{\mathbf{\Phi}}_{\mathbf{b}} := \{\phi(\mathbf{a}', \mathbf{b}) | \mathbf{a}' \in \mathcal{A}\} \tag{4.4}$$

where $\overleftrightarrow{\mathbf{\Phi}}_{\mathbf{a}}$ and $\overleftrightarrow{\mathbf{\Phi}}_{\mathbf{b}}$ are the sub-spaces formed when fixing one of the agents' action, and varying the other action freely.

Lemma 4.2. ***Intersection of $\overleftrightarrow{\mathbf{\Phi}}_{\mathbf{a}}$ and $\overleftrightarrow{\mathbf{\Phi}}_{\mathbf{b}}$:*** *Given a bipartite spherical map from Def. 2.1, with $\mathbf{a}$ parameterizing the azimuthal (latitudinal) coordinates, the cardinality of the intersect between $\overleftrightarrow{\mathbf{\Phi}}_{\mathbf{a}}$ and $\overleftrightarrow{\mathbf{\Phi}}_{\mathbf{b}}$ will be non-empty. That is, $|\overleftrightarrow{\mathbf{\Phi}}_{\mathbf{a}} \cap \overleftrightarrow{\mathbf{\Phi}}_{\mathbf{b}}| > 0$. (Proof is in Appendix C.7.)*

The purpose of Lemma 4.2 is to highlight that, given the bipartite map, subspaces are guaranteed to intersect on the manifold. This is easy to visualize on a spherical manifold in

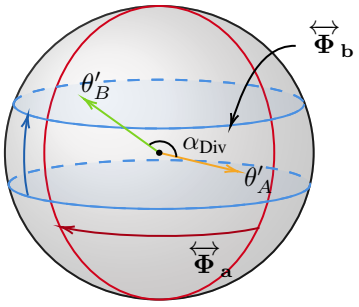

Figure 2: **Isoplanar subspaces for players A and B.** A visualization of the isoplanes $\overleftrightarrow{\mathbf{\Phi}}_{\mathbf{a}}$ and $\overleftrightarrow{\mathbf{\Phi}}_{\mathbf{b}}$ on a 2-sphere embedded in three dimensions is shown in on the left side. The isoplanes are depicted relative to the normalized objective vectors θ'_A and θ'_B, which lie on the manifold surface, separated by a divergence angle α_{Div}.

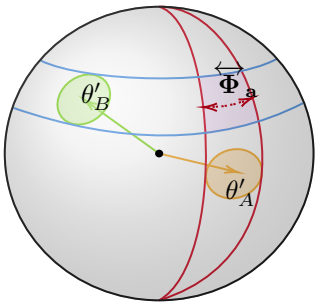

Figure 3: **Geodesic confidence balls for players A & B.** This figure illustrates the geodesic confidence balls, positioned on the surface of the spherical manifold. In three dimensions, it becomes evident that $\overleftrightarrow{\mathbf{\Phi}}_{\mathbf{a}}$ and $\overleftrightarrow{\mathbf{\Phi}}_{\mathbf{b}}$ are orthogonal at any point of intersection. This intersection, denoted by $\overleftrightarrow{\mathbf{\Phi}}_{\mathbf{b_a}}$, is where the joint action emerges, represented by a purple geodesic square indicating the uncertainty region.

the 2-sphere setting (e.g., longitudinal and latitudinal lines) but becomes challenging to perceive in higher dimensions. We rigorously argue that, just as in the 2-sphere case, the same principle holds in a D-sphere setting. The derivation of Lemma 4.2 first comes by isolating the subspaces in terms of angular coordinates. Next, due to the *Poincaré-Hopf theorem* (Poincaré 1885; Hopf 1927), the compactness of the smooth Riemmanian manifold imposes strong geometric constraints such that the two subspaces cannot avoid each other.

Lemma 4.3. ***Pure Strategy of the Leader:*** *Given a spherical manifold, $\mathbf{\Phi}$, and isoplanar subspace, $\overleftrightarrow{\mathbf{\Phi}}_{\mathbf{a}}$ and $\overleftrightarrow{\mathbf{\Phi}}_{\mathbf{b}}$ for the longitudinal and lattitudinal subspaces respectively, the optimal strategy of the leader is that of a pure strategy, that is, $\pi^*_A(\mathbf{a}) \in \{0, 1\}$. (Proof is provided in Appendix C.9.)*

Lemma 4.3 argues that the intersection between $\overleftrightarrow{\mathbf{\Phi}}_{\mathbf{a}}$ and $\overleftrightarrow{\mathbf{\Phi}}_{\mathbf{b}}$ contains at most one element due to their orthogonality (see Appendix Lemma C.2). Consequently, no other actions on the manifold can further maximize the leader's reward. Intuitively, the positive curvature of the manifold ensures that once two non-degenerate isoplanes intersect, the intersection is a unique point that maximizes the dot product between the action and the objective vector.

Geodesic Isoplanar Subspace Alignment (GISA): We present an end-to-end procedure (in Algorithm 1) for constructing the Stackelberg manifold, $\mathbf{\Phi}$, and subsequently enabling online learning within it. The process begins with an offline, data-independent, training phase (Step 5 of Algorithm 1) to construct $\mathbf{\Phi}$. This is followed by iterative online learning performed directly on the manifold. As a result, optimal strategy learning reduces to geodesic distance minimization, providing a significantly more computationally efficient alternative to traditional game-theoretic approaches, such as bi-level optimization.

The general methodology in which we can compute the optimal leader strategy is that the leader can anticipate the follower strategy based on knowledge of follower's reward parameters ξ_{θ_B} and the isoplane $\overleftrightarrow{\mathbf{\Phi}}_{\mathbf{a}}$. We denote this homeomorphism as $f_1(\overleftrightarrow{\mathbf{\Phi}}_{\mathbf{a}}, \xi_{\theta_B}) : \overleftrightarrow{\mathbf{\Phi}}_{\mathbf{a}} \mapsto \overleftrightarrow{\mathbf{\Phi}}_{\mathbf{b}^*_{\mathbf{a}}}$. Thereafter, we

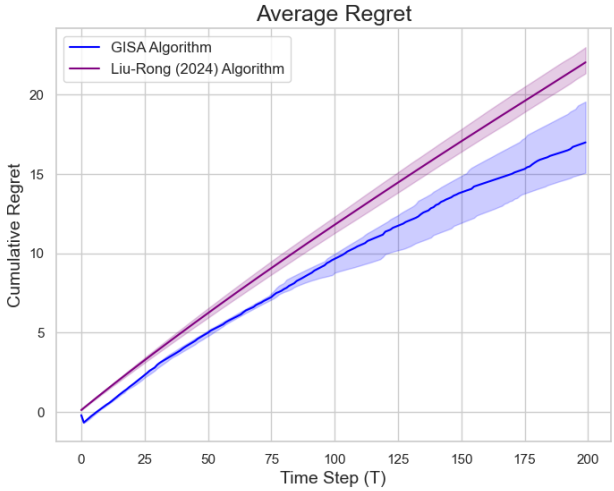

Figure 4: NPG Regret. Uncertainty regions denote upper and lower quartiles.

compute the geodesic distance minimizing distance from $\overleftrightarrow{\mathbf{\Phi}}_{\mathbf{b}^*_{\mathbf{a}}}$ to ξ_{θ_A} via injective map $f_2(\overleftrightarrow{\mathbf{\Phi}}_{\mathbf{b}^*_{\mathbf{a}}}, \xi_{\theta_A}) : \overleftrightarrow{\mathbf{\Phi}}_{\mathbf{b}^*_{\mathbf{a}}} \mapsto \mathbb{R}$. For illustration, referencing Fig. 3, the $\overleftrightarrow{\mathbf{\Phi}}_{\mathbf{a}}$ represents the subspace induced by the leader, visualized as a red trace, from where the follower selects his best response within. The follower will attempt to act within the subspace, such that it minimizes his geodesic distance to θ'_B. Thus, the leader's objective is to find $\mathbf{a} \in \mathcal{A}$ such that it minimizes the composition of $f_1 \circ f_2$, giving us the geodesic distance. This composition is defined as,

$$\overleftrightarrow{\mathbf{\Phi}}_{\mathbf{a}} \underset{f_1(\cdot, \xi_{\theta_A})}{\longmapsto} \overleftrightarrow{\mathbf{\Phi}}_{\mathbf{b}^*_{\mathbf{a}}} \underset{f_2(\cdot, \xi_{\theta_B})}{\longmapsto} \mathfrak{G}(\mathbf{a}, \mathbf{b}^*_{\mathbf{a}}) \in \mathbb{R}, \tag{4.5}$$

where $\theta' = \frac{\theta}{\|\theta\|}$ for A and B.

Theorem 1. ***Isoplane Stackelberg Regret:*** *For D-dimensional spherical manifolds embedded in $\mathbb{R}^D$ space,*

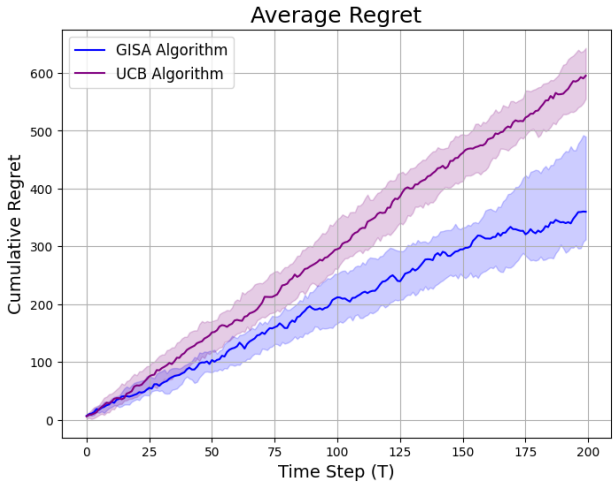

Figure 5: $\mathbb{R}^1$ Stackelberg Regret. Uncertainty regions denote upper and lower quartiles.

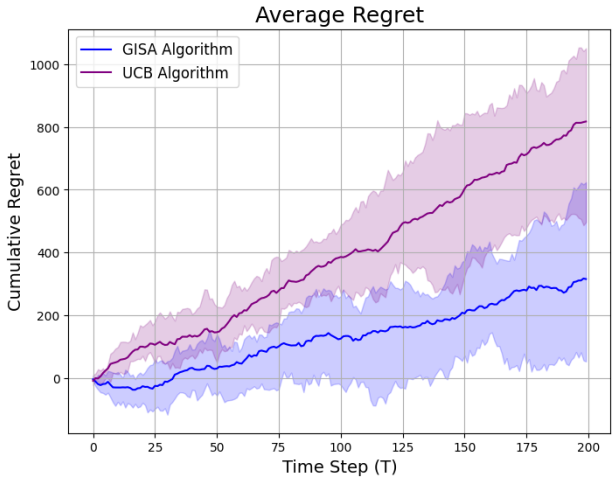

Figure 6: SSG Regret. Uncertainty regions denote upper and lower quartiles.

where $\phi(\mathbf{a}, \cdot)$ generates an isoplanes $\overleftrightarrow{\mathbf{\Phi}}_{\mathbf{a}}$, and the linear relationship to the reward function in Eq. (2.7) *and Eq.* (2.8)*, the simple regret, defined in Eq.* (4.3)*, of any learning algorithm with uncertainty parameter uncertainty $\mathcal{C}_\theta(t)$, refer to in Eq.* (3.1)*, is bounded by $\mathcal{O}(\arccos(1 - \mathcal{C}_\theta(t)^2/2))$. (Proof in Appendix C.13.)*

Proof Sketch: The proof of Theorem 1 focuses on analyzing the geodesic distances on $\mathbf{\Phi}$ due to uncertainty. First, we argue that any norm-like confidence ball in Cartesian coordinates, $\texttt{Ball}(\cdot)$, can be transformed into a confidence bound into a geodesic distance-based confidence ball, $\texttt{Ball}_{\mathfrak{G}}(\cdot)$, in spherical coordinates (discussed in Lemma C.3 of the Appendix.) Due to orthogonality between $\overleftrightarrow{\mathbf{\Phi}}_{\mathbf{a}}$ and $\overleftrightarrow{\mathbf{\Phi}}_{\mathbf{b}}$, we argue that that the geodesic distance either remains the same or decreases when we projected from any $\texttt{Ball}(\cdot)'$ from $\overleftrightarrow{\mathbf{\Phi}}_{\mathbf{a}}$ to $\overleftrightarrow{\mathbf{\Phi}}_{\mathbf{b}}$ (discussed in Lemma C.4 of the Appendix.) This naturally extends to a bound on the maximum diameter of the projected confidence ball on $\overleftrightarrow{\mathbf{\Phi}}_{\mathbf{b}}$. This constitutes the best and worst possible outcomes due to misspecification in accordance with the formulas in Eq. (3.4) and Eq. (3.5). Consequently, one can see that our methodology enables the leader to simplify the follower's best response region on $\mathbf{\Phi}$, allowing derivation of high-probability worst-case outcomes and theoretical guarantees for simple regret under quantifiable uncertainty.

Algorithm 1: Geodesic Isoplanar Subspace Alignment (GISA) Algorithm

Input: Time horizon T, and confidence ball $\mathcal{C}_\theta(\cdot)$.
Output: Estimated optimal leader action $\hat{\mathbf{a}}$.
Initialize $\hat{\theta}_A$ and $\hat{\theta}_B$ uniformly at random.
Initialize reward and action histories, $\mathcal{U}$ and $\mathcal{H}$ as $\emptyset$.
Construct a Stackelberg embedding $\mathbf{\Phi}$ and feature map ϕ per specifications in Sec. 2.2.
for $t \in 1...T$ **do**
  if $\mathfrak{G}(\hat{\theta}_A, \hat{\theta}_B) < 2\mathcal{C}_\theta(t)$ **then**
    $\texttt{Phase 1}$: Select uniformly an action on the boundary of A's geodesic confidence ball.
    $\mathbf{a} \sim \texttt{Uniform}[\partial\,\texttt{Ball}_{\mathfrak{G}}(\hat{\theta}_A, \mathcal{C}_\theta(t))]$
  else
    $\texttt{Phase 2}$: Select $\mathbf{a}$ that minimizes the geodesic distance to $\hat{\theta}_B$ from $\texttt{Ball}_{\mathfrak{G}}(\hat{\theta}_A, \mathcal{C}_\theta(t))$.
    $\mathbf{a} \leftarrow \underset{\mathbf{a} \in \texttt{Ball}_{\mathfrak{G}}(\hat{\theta}_A, \mathcal{C}_\theta(t))}{\arg\min} \mathfrak{G}(\mathbf{a}, \hat{\theta}_B)$
  end if
  $\mathbf{b} \leftarrow \underset{\mathbf{b} \in \overleftrightarrow{\mathbf{\Phi}}_{\mathbf{a}}}{\arg\min} \mathfrak{G}(\mathbf{b}, \hat{\theta}_B)$
  $\hat{\mathbf{a}}^t, \hat{\mathbf{b}}^t \leftarrow \phi^{-1}(\mathbf{a}, \mathbf{b})$ Perform an inverse map.
  yield $\hat{\mathbf{a}}^t, \hat{\mathbf{b}}^t$, and obtain empirical reward μ_A^t, μ_B^t.
  $\mathcal{H} \leftarrow \mathcal{H} \cup (\hat{\mathbf{a}}^t, \hat{\mathbf{b}}^t), \quad \mathcal{U} \leftarrow \mathcal{U} \cup (\mu_A^t, \mu_B^t)$.
  Re-estimate $\hat{\theta}_A$ and $\hat{\theta}_B$ from $\mathcal{H}$ and $\mathcal{U}$, via Eq. (2.9).
end for
return $\hat{\mathbf{a}}^t$

5 Empirical Experiments

We present three practical instances of Stackelberg games and benchmark the GISA algorithm (Alg. 1) against a dual-UCB algorithm, where both agents use UCB-based no-regret learning (Blum and Mansour 2007). The $\mathbf{\Phi}$ transform abstracts away the need for exact reward structure knowledge, enabling well-behaved representations for online learning in new problem settings. GISA starts by sampling the action space until uncertainty intervals become disjoint, then enters a continuous learning phase, subject to the guarantees in Thm. 1. This method generalizes beyond problem-specific solutions and addresses Stackelberg game learning methods lacking closed-form solutions and/or computational feasibility. (All experimental details are presented in Appendix G.3.)

The Newsvendor Pricing Game (NPG): Modelled from (Cesa-Bianchi et al. 2023; Liu and Rong 2024), the NPG models a supply chain consisting of a supplier (leader) and retailer

(follower) in a repeated Stackelberg game. The leader's action space is $\mathbf{a} \in \mathbb{R}^1$, and the follower's is $\mathbf{b} \in \mathbb{R}^2$. The supplier dynamically prices the product to maximize reward, while the retailer optimizes pricing and order quantity based on stochastic demand, following classical Newsvendor theory (Arrow, Harris, and Marschak 1951; Petruzzi and Dada 1999). The asymmetric reward function and stochastic demand complicates online learning significantly. (See Fig. 4.)

$\mathbb{R}^1$ **Stackelberg Game:** In this game, the leader chooses an action, anticipating the follower's best response. Both players have one-dimensional action spaces. Nonlinear rewards and penalties complicate the equilibrium, requiring numerical methods to find optimal strategies. A real-world example is an energy grid management, where a utility company (leader) sets prices or output levels, considering consumers' (followers) usage and nonlinear factors like demand fluctuations or storage constraints. (See Fig. 5.)

Stackelberg Security Game (SSG) in $\mathbb{R}^5$: Motivated by (Balcan et al. 2015; Zhang and Malacaria 2021), this SSG models a defender (leader) allocating limited resources across multiple targets, anticipating an attacker's (follower) strategy. Both players select actions from $\mathbb{R}^5$, with rewards driven by the difference between the actions (i.e. $\mathbf{a} - \mathbf{b}$) and quadratic penalties for overextension. Resource constraints are imposed via weighted L_1-norms, limiting feasible actions. The Stackelberg equilibrium is defined by the leader's optimal resource allocation, accounting for the adversary's best response. Nonlinear constraints increase the problem complexity, conventionally requiring numerical solutions. (See Fig. 6.)

6 Conclusion

This work establishes a foundational connection between Stackelberg games and normalizing neural flows, marking a significant advancement in the study of equilibrium learning and manifold learning. By utilizing normalizing flows to map joint action spaces onto Riemannian manifolds, particularly spherical ones, we offer a novel, theoretically grounded framework with formal guarantees on simple regret. This approach represents the first application of normalizing flows in game-theoretic settings, specifically Stackelberg games, thereby opening new avenues for learning on spherical manifolds. Current limitations include the restriction to spherical manifolds, however, the key principles of geodesic distance minimization apply, and warrants future investigation. Our empirical results, grounded in realistic simulation scenarios, highlight promising improvements in both computational efficiency and regret minimization, underscoring the broad potential of this methodology across multiple domains in economics and engineering. Despite potential challenges related to numerical accuracy for the neural flow network, this integration of manifold learning into game theory exhibits strong implications for online learning, positioning neural flows as a promising tool for both machine learning and strategic decision-making.

Ethical Statement

We affirm that this research complies with the accepted ethical standards in scientific research. All simulations and methodologies were conducted with integrity and transparency, without harm to individuals, groups, or the environment. From a perspective of social impact, we ensured that the theoretical and practical contributions of this work are aimed at advancing knowledge in a responsible and ethical manner, with no misuse or malicious application of the techniques proposed. Additionally, no conflicts of interest or external influences have compromised the objectivity or scientific rigour of this work.

Acknowledgements

The affiliated authors thank the departmental research funding from the Technical University of Munich School of Computation and Information Technology for their generous support. We also acknowledge Stefanos Leonardos, Vinzenz Thoma, and the paper reviewers for their the insightful technical feedback helping to refine our work.

References

Abbasi-Yadkori, Y.; Pál, D.; and Szepesvári, C. 2011. Improved algorithms for linear stochastic bandits. *Advances in neural information processing systems*, 24.

Amani, S.; Alizadeh, M.; and Thrampoulidis, C. 2019. Linear stochastic bandits under safety constraints. *Advances in Neural Information Processing Systems*, 32.

Arrow, K. J.; Harris, T.; and Marschak, J. 1951. Optimal inventory policy. *Econometrica: Journal of the Econometric Society*, 250–272.

Balcan, M.-F.; Blum, A.; Haghtalab, N.; and Procaccia, A. D. 2015. Commitment without regrets: Online learning in stackelberg security games. In *Proceedings of the sixteenth ACM conference on economics and computation*, 61–78.

Beck, Y.; Ljubić, I.; and Schmidt, M. 2023. A survey on bilevel optimization under uncertainty. *European Journal of Operational Research*, 311(2): 401–426.

Blum, A.; and Mansour, Y. 2007. From external to internal regret. *Journal of Machine Learning Research*, 8(6).

Bonnabel, S. 2013. Stochastic gradient descent on Riemannian manifolds. *IEEE Transactions on Automatic Control*, 58(9): 2217–2229.

Brehmer, J.; and Cranmer, K. 2020. Flows for simultaneous manifold learning and density estimation. *Advances in neural information processing systems*, 33: 442–453.

Cesa-Bianchi, N.; Cesari, T.; Osogami, T.; Scarsini, M.; and Wasserkrug, S. 2023. Learning the stackelberg equilibrium in a newsvendor game. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, 242–250.

Cesa-Bianchi, N.; and Lugosi, G. 2006. *Prediction, learning, and games*. Cambridge university press.

Chu, W.; Li, L.; Reyzin, L.; and Schapire, R. 2011. Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 208–214. JMLR Workshop and Conference Proceedings.

Dinh, L.; Sohl-Dickstein, J.; and Bengio, S. 2016. Density estimation using real nvp. In *International Conference on Machine Learning*, 1530–1538. PMLR.

Duchi, J.; Hazan, E.; and Singer, Y. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul): 2121–2159.

Durkan, C.; Bekasov, A.; Murray, I.; and Papamakarios, G. 2020. nflows: normalizing flows in PyTorch. https://doi.org/10.5281/zenodo.4296287. Accessed: 2024-02-04.

Gemici, M. C.; Rezende, D.; and Mohamed, S. 2016. Normalizing flows on riemannian manifolds. *arXiv preprint arXiv:1611.02304*.

Haghtalab, N.; Lykouris, T.; Nietert, S.; and Wei, A. 2022. Learning in Stackelberg Games with Non-myopic Agents. In *Proceedings of the 23rd ACM Conference on Economics and Computation*, 917–918.

Hopf, H. 1927. Vektorfelder in n-dimensionalen Mannigfaltigkeiten. *Mathematische Annalen*, 96(1): 225–250.

Hsieh, Y.-G.; Mertikopoulos, P.; Staudigl, M.; and Cevher, V. 2023. No-Regret Learning in Games with Noisy Feedback: Faster Rates and Adaptivity via Learning Rate Separation. *arXiv preprint arXiv:2206.06015*.

Jain, M.; Korzhyk, D.; Vanek, O.; Pechoucek, M.; Conitzer, V.; and Tambe, M. 2011. A double oracle algorithm for zero-sum security games on graphs.

Jiang, A.; Nguyen, T.; Tambe, M.; and Procaccia, A. 2013. Monotonic maximin: A robust Stackelberg solution against boundedly rational followers. *Conference on Decision and Game Theory for Security (GameSec)*.

Kar, D.; Fang, F.; Fave, D.; Sintov, N.; and Tambe, M. 2015. A game of thrones: when human behavior models compete in repeated Stackelberg security games. *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Kar, D.; Nguyen, T. H.; Fang, F.; Brown, M.; Sinha, A.; Tambe, M.; and Jiang, A. X. 2017. Trends and applications in Stackelberg security games. *Handbook of dynamic game theory*, 1–47.

Korzhyk, D.; Conitzer, V.; and Parr, R. 2010. Complexity of computing optimal Stackelberg strategies in security resource allocation games. *Proceedings of the 24th AAAI conference on artificial intelligence*, 805–810.

Lattimore, T.; and Szepesvári, C. 2020. *Bandit algorithms*. Cambridge University Press.

Liu, L.; and Rong, Y. 2024. No-Regret Learning for Stackelberg Equilibrium Computation in Newsvendor Pricing Games. *The 8th International Conference on Algorithmic Decision Theory*.

Moradipari, A.; Turan, B.; Abbasi-Yadkori, Y.; Alizadeh, M.; and Ghavamzadeh, M. 2022. Feature and parameter selection in stochastic linear bandits. In *International Conference on Machine Learning*, 15927–15958. PMLR.

Nguyen, T. H.; Yadav, A.; Tambe, M.; and Boutilier, C. 2014. Regret-based optimization and preference elicitation for Stackelberg security games with uncertainty. *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 756–762.

Papamakarios, G.; Nalisnick, E.; Rezende, D. J.; Mohamed, S.; and Lakshminarayanan, B. 2021. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57): 1–64.

Petruzzi, N. C.; and Dada, M. 1999. Pricing and the newsvendor problem: A review with extensions. *Operations research*, 47(2): 183–194.

Poincaré, H. 1885. Sur les courbes définies par les équations différentielles. *Journal de Mathématiques Pures et Appliquées*, 1: 167–244.

Rezende, D. J.; and Mohamed, S. 2015. Variational Inference with Normalizing Flows. In *International Conference on Machine Learning*, 1530–1538. PMLR.

Rezende, D. J.; Papamakarios, G.; Racaniere, S.; Albergo, M.; Kanwar, G.; Shanahan, P.; and Cranmer, K. 2020. Normalizing flows on tori and spheres. In *International Conference on Machine Learning*, 8083–8092. PMLR.

Shieh, E.; An, B.; Yang, R.; Tambe, M.; Baldwin, C.; et al. 2012. PROTECT: An application of computational game theory for the security of the ports of the United States. *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Sinha, A.; Malo, P.; and Deb, K. 2017. A review on bilevel optimization: From classical to evolutionary approaches and applications. *IEEE transactions on evolutionary computation*, 22(2): 276–295.

von Stackelberg, H. 1934. *Marktform und Gleichgewicht*. Vienna: Springer-Verlag.

Wang, J.; Hu, H.; Nguyen, D. P.; and Fisac, J. F. 2024. MAGICS: Adversarial RL with Minimax Actors Guided by Implicit Critic Stackelberg for Convergent Neural Synthesis of Robot Safety. *arXiv preprint arXiv:2409.13867*.

Zanette, A.; Dong, K.; Lee, J. N.; and Brunskill, E. 2021. Design of experiments for stochastic contextual linear bandits. volume 34, 22720–22731.

Zhang, Y.; and Malacaria, P. 2021. Bayesian Stackelberg games for cyber-security decision support. *Decision Support Systems*, 148: 113599.

Zhou, Y.; and Kantarcioglu, M. 2016. Modeling adversarial learning as nested stackelberg games. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 350–362. Springer.