POLICY REGRET FOR EMBEDDING MODEL ROUTING: CONTEXTUAL BANDITS WITH LOW-RANK EXPERTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Modern recommendation systems increasingly rely on dynamically routing diverse queries to multiple embedding models. Despite its practical significance, this problem remains poorly understood under realistic conditions like adversarial queries, bandit feedback, and limited observability of models. We formalize embedding model routing as an adversarial contextual linear bandit with low-rank experts, where contexts are queries, actions are items, and experts are the embedding models working on low-rank latent representation spaces. We first establish that standard regret notions suffer from structural misspecification or statistical intractability, and we identify a *log-quadratic* policy class that is expressive enough to capture query-dependent model routing, yet structured enough to allow efficient online learning. Focusing on this log-quadratic policy optimization problem under bandit feedback – which is of independent interest – we propose a policy gradient algorithm called Hypentropy Policy Gradient (HPG). It provably adapts to the unknown low-rank structure under incomplete information and attains $\widetilde{\mathcal{O}}(s\sqrt{MT})$ linearized policy regret – where s, M, and T are the intrinsic rank of the experts, the number of models, and the number of rounds – thus avoiding a curse of dimensionality. We provide computationally efficient and parameter-free implementation of HPG.

1 INTRODUCTION

Modern recommendation systems and search engines rely heavily on dense embedding models for information retrieval (Yi et al., 2019; Karpukhin et al., 2020). These models map queries and items into low-dimensional "embedding spaces," enabling efficient retrieval based on the similarity scores (i.e., inner products). However, representing the full diversity of distinct user queries within a single embedding space is fundamentally limited: Due to models' intrinsic low-dimensional structures, no single embedding model can be uniformly optimal across all types of queries (Weller et al., 2026).

To illustrate, Figure 1 ranks Amazon items (Collins et al., 2022) by a semantic embedding `miniLM` and by a Q&A model `msmarco` against an LLM-generated ground truth. Gray dots are high-quality items (LLM top-25%) and colored dots are those one model retrieves in its top-30 while the other misses (see Appendix A for details). The errors and strengths are mutually exclusive, meaning *no single model dominates the other*, and some adaptive routing is essential.

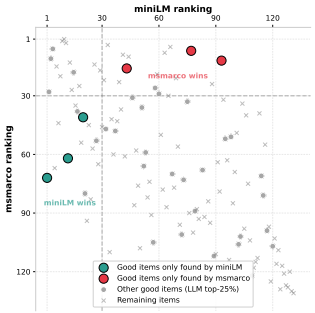

Figure 1: Rankings of two different embedding models

This limitation is already reflected in practice. Large-scale industrial e-commerce platforms by Alibaba and Meta use multiple embedding models to capture various semantic aspects of queries (Li et al., 2019a; Huang et al., 2020). Recent approaches in Retrieval-Augmented Generation (RAG) also route queries to specialized "expert models" trained on domain-specific data (Lee et al., 2025; Zhao et al., 2026). Nevertheless, most of such adaptive routing approaches are based on heuristics, which gives our main question: ***How can an online learner adaptively route queries to embedding models, using feedback only from the routed models, to achieve provable performance guarantees?*** Towards a theoretically grounded answer, this paper makes the following main contributions.

Modeling (Section 2). We frame the problem of routing queries to embedding models as a T-round adversarial contextual linear bandit with low-rank experts: In round t, the learner observes an adversarial query $q_t \in \mathbb{R}^{d_q}$ as *context* and faces an item set $\mathcal{A} \subseteq \mathbb{R}^{d_a}$ as *actions*. M embedding

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

models, as *experts*, each implicitly has a distribution over $\mathcal{A}$ via a latent embedding space of dimension $s \ll \min(d_q, d_a)$. The learner can only choose *one* model m_t to route, observe the selected model's recommendation, and receive an induced noisy reward. The goal is to compete with a fixed class of routing policies. This formulation captures the main challenges in realistic recommendation systems: adversarial query sequences, bandit reward feedback, and partial observability of model outputs.

Regret Notion (Section 3). A key challenge is identifying an appropriate performance benchmark for this problem. We find standard regret notions inadequate: Competing with the best fixed model (i.e., expert regret) ignores query heterogeneity, while competing with arbitrary query-dependent policies (i.e., contextual regret) is intractable. The commonly used log-linear policy class also fails to fit the underlying structures of embedding models. Instead, our main insight is, under the latent geometry of embedding models, the expected reward of each model is approximately a *low-rank quadratic* function of the query (Proposition 1). This leads to our log-quadratic policy class: expressive enough to capture near-optimal routing, while remaining amenable to efficient learning. Our claim is numerically justified on an Amazon ESCI dataset (Reddy et al., 2022) in Table 2.

Algorithm and Implementation (Sections 4 and 5). For log-quadratic policy optimization under bandit feedback – a problem that could be of independent interest – the quadratically large parameter space makes a vanilla policy gradient suffer from $\widetilde{\mathcal{O}}(d_q\sqrt{MT})$ linearized policy regret (a surrogate metric to resolve the non-convexity of softmax; see Section 3.2). To avoid the d_q dependency, we propose Hypentropy Policy Gradient (HPG), which automatically adapts to the unknown low-rank structure of embedding models and provably attains $\widetilde{\mathcal{O}}(s\sqrt{MT})$ linearized policy regret. HPG builds on Online Mirror Descent, whose Bregman projection is typically expensive; combining hypentropy regularization with a nuclear-norm parameter space brings it down to $\mathcal{O}(d_q^2 M)$ time per round (Theorem 4), and HPG can further be implemented parameter-freely, i.e., without knowing s.

Related works are discussed throughout the paper, with a comprehensive overview in Appendix B.

2 Setup: Contextual Bandits with Low-Rank Experts

Notations. For an integer $n \in \mathbb{Z}_+$, $[n]$ denotes the set $\{1, 2, \ldots, n\}$. We use $\mathcal{O}$ to hide absolute constants, and use $\widetilde{\mathcal{O}}$ to additionally hide poly-logarithmic factors. Let $\mathbb{S}^n \subseteq \mathbb{R}^{n\times n}$ be the space of symmetric $n \times n$ real matrices. For $X \in \mathbb{S}^n$, $\lambda_i(X)$ denotes the i-th largest eigenvalue of X. Let $\|X\|_* := \sum_i |\lambda_i(X)|$ be the nuclear norm and $\|X\|_2 := \max_i |\lambda_i(X)|$ be the spectral norm.

Consider a T-round game between a learner and the environment. In each round t, a *query* embedded into the d_q-dimensional Euclidean space, namely $q_t \in \mathcal{Q} := \mathbb{R}^{d_q}$, is presented to the learner as a context; we allow q_t to be arbitrary, i.e., chosen by an adaptive adversary. Although real queries can be in natural language, practical model routing pipelines first embed them via a "base encoder" (see, e.g., Lee et al., 2025). The learner holds a large set of candidate *items* $\mathcal{A} \subseteq \mathbb{R}^{d_a}$, the action set in bandits, and aims to recommend an $a_t \in \mathcal{A}$ that best answers q_t, measured by the expected *reward* $r(q_t, a_t)$ that Section 2.1 models as a similarity score on a latent space. Instead of directly working on $\mathcal{A}$, the learner is equipped with M embedding *models* as experts, each carrying a recommendation distribution $\xi_m(q_t)$ induced by its latent internal structure (Section 2.2). Given the cost of invoking large models, however, the learner can query only *one* model m_t for a single item a_{t,m_t} and receives bandit feedback on the induced reward alone (Tsiourvas et al., 2025); Section 2.3 details the protocol.

2.1 Misspecified Adversarial Contextual Linear Bandit

For query $q \in \mathcal{Q}$ and item $a \in \mathcal{A}$, we assume the expected reward $r(q, a)$ is roughly their similarity score $\langle U^* q, V^* a\rangle$ on a high-dimensional latent space $\mathbb{R}^{\max(d_a,d_q)}$, where $U^* \in \mathbb{R}^{\max(d_a,d_q)\times d_q}$ and $V^* \in \mathbb{R}^{\max(d_a,d_q)\times d_a}$ are fixed but unknown linear maps. Writing $\langle A, B\rangle_F := \mathrm{Tr}(A^\mathsf{T} B)$ for the Frobenius inner product of matrices, this is equivalently a $d_a \times d_q$-dimensional linear structure:

$$r(q, a) = \left\langle \Psi^*, aq^\mathsf{T} \right\rangle_F + \nu(q, a), \quad \text{where kernel } \Psi^* := (V^*)^\mathsf{T} U^* \in \mathbb{R}^{d_a\times d_q}, \tag{1}$$

and ν is a misspecification term capturing the non-linearity, with $\|\nu\|_\infty := \sup_{q,a} |\nu(q, a)| \le \epsilon$ for some fixed but unknown $\epsilon \in [0, 1)$ – a standard way of modeling misspecification in linear bandits (Gopalan et al., 2016; Ghosh et al., 2017; Foster et al., 2020) or linear Markov decision processes (Du et al., 2020; Vial et al., 2022; Jin et al., 2023). All of U^*, V^*, ν, and ϵ are unknown to the learner.

2.2 Embedding Models as Low-Rank Experts

As standard both in practical recommendation systems and theoretical research, we model embedding models via the two-tower architecture (Huang et al., 2013): Each model $m \in [M]$ projects queries and items independently onto its own embedding space $\mathbb{R}^{s_m}$, and recommends items to queries according to the softmax of similarity scores on $\mathbb{R}^{s_m}$. As observed by Weller et al. (2026), embedding models work in very low-dimensional representation spaces, hence we assume $s_m \ll d_a, d_q$. When presented with query $q \in \mathcal{Q}$, model m's *recommendation distribution* over the item set $\mathcal{A} \subseteq \mathbb{R}^{d_a}$ is:[1]

$$\xi_m(a \mid q) \propto \exp\Big(\big\langle U_m q, V_m a\big\rangle + \mu_m(q, a)\Big), \quad \forall a \in \mathcal{A}, \tag{2}$$

which parallels the reward in Eq. (1): $U_m \in \mathbb{R}^{s_m \times d_q}$ and $V_m \in \mathbb{R}^{s_m \times d_a}$ are two fixed but unknown linear maps, and $\mu_m(q, a)$ is a misspecification function capturing the non-linearity in the embedding model. We assume $\|\mu_m\|_\infty := \sup_{q,a} |\mu_m(q, a)| \le \epsilon$ for the same constant $\epsilon \in [0, 1)$. The key difference is that U_m and V_m are low-rank. For notational simplicity, we define $s := \max_m s_m$ and omit the subscript m in s_m. Consequently, when defining the *model kernel* $\Psi_m := V_m^\mathsf{T} U_m$ similar to Eq. (1), we have $\mathrm{rank}(\Psi_m) \le s$ and

$$\xi_m(a \mid q) \propto \exp\Big(\big\langle \Psi_m, a q^\mathsf{T} \big\rangle_F + \mu_m(q, a)\Big), \quad \text{where kernel } \Psi_m := V_m^\mathsf{T} U_m \text{ has a rank} \le s. \tag{3}$$

All components in Eq. (3), namely Ψ_m, μ_m, and s, are unknown to the learner. The name low-rank "experts" comes from the bandits with expert advice literature (Auer et al., 2002).

Regularity. We assume that all the embeddings are ℓ_2-normalized as $\|q\|_2, \|a\|_2 \le 1, \forall q \in \mathcal{Q}, a \in \mathcal{A}$; all model kernels ensure $\|\Psi_m\|_2 \le \rho$ for some $\rho \in [0, 1]$; and the reward kernel satisfies $\|\Psi^*\|_2 \le 1$.[2]

2.3 Interaction Protocol, Feedback Model, and Policy Regret

Before the game, the environment fixes the reward kernel Ψ^*, model kernels $\{\Psi_m\}_m$, and misspecification functions ν and μ_m in Eqs. (1) and (3). The learner only knows the query space $\mathcal{Q} = \mathbb{R}^{d_q}$, item set $\mathcal{A} \subseteq \mathbb{R}^{d_a}$, number of models M, and number of rounds T.[3] In each round $t = 1, 2, \ldots, T$:

1. the environment arbitrarily (perhaps adaptively adversarially) decides a query $q_t \in \mathcal{Q}$;
2. each model $m \in [M]$, observing the query q_t, crafts its distribution $\xi_m(q_t)$ via Eq. (3);
3. the learner, observing query q_t but not the distributions, chooses a model $m_t \in [M]$ to invoke;
4. the item recommended for this round is sampled from model m_t's distribution as $a_{t,m_t} \sim \xi_{m_t}(q_t)$;
5. the learner observes a noisy reward $r_t = r(q_t, a_{t,m_t}) + \eta_t$, where $r(q, a)$ is defined in Eq. (1). We assume η_t is a zero-mean and conditionally 1-sub-Gaussian noise (Abbasi-Yadkori et al., 2011).

The feedback is doubly limited. Beyond the *bandit reward feedback* common to recommendation systems (Li et al., 2010; Chapelle & Li, 2011) and to bandits with expert advice (Auer et al., 2002; Beygelzimer et al., 2011), model routing suffers an *expert observability* constraint: the learner sees neither what an unselected model $m \neq m_t$ would have recommended, nor the full distribution $\xi_{m_t}(q_t)$ behind the one item it does see. Assuming such counterfactuals is unrealistic given the cost of invoking multiple models (Tsiourvas et al., 2025; Zu et al., 2026). Section 3 show how the two together constitute significant challenges.

[1] Although embedding models are internally non-linear (Hendrycks & Gimpel, 2016; Devlin et al., 2019), the linear form in Eq. (2) is backed by the Platonic representation hypothesis and the representation alignment literature (Lenc & Vedaldi, 2015; Kornblith et al., 2019; Huh et al., 2024), which find the latent spaces of heterogeneous networks to be linearly inter-mappable. Specifically, Wang & Isola (2020) prove that contrastive learning enforces an *inner-product* geometry, in which case the residual non-linearity is nicely absorbed by μ_m.

[2] The first condition is a standard practice in contrastive learning (Wang & Isola, 2020, §3). The second condition arises because, prior to computing softmax over inner products, practitioners usually divide the logits by a large temperature, as common in Transformers (Vaswani et al., 2017), knowledge distillation (Hinton et al., 2015), and model calibration (Guo et al., 2017). Even if a sharper retrieval softmax pushed ρ towards 1, it only enlarges the residual term in Proposition 1; the main takeaway – quadratic leading term – is unaffected. The third condition ensures bounded expected rewards and is standard in the literature (Jin et al., 2023; Luo et al., 2021).

[3] While practical hard constraints may make the item set time-varying (Covington et al., 2016; Wang et al., 2018; Huang et al., 2020), incorporating such non-stationarity via sleeping bandits (Kleinberg et al., 2010) is orthogonal to our primary focus. We assume a stationary item set $\mathcal{A}$ to isolate our core challenge of online embedding model routing. The knowledge of T is for the ease of presentation, which can be relaxed via standard doubling tricks (Besson & Kaufmann, 2018). The low-rank parameter $s = \max_m s_m$ in Eq. (3) is unknown.

We consider the policy regret (Agarwal et al., 2014; Syrgkanis et al., 2016), which compares our learner to a fixed class of model routing policies. Formally, a *policy* is a mapping from query set $\mathcal{Q}$ to the probability simplex over all models, i.e., $\triangle := \{p \in [0,1]^M \mid \sum_{m=1}^M p_m = 1\}$. The space of all policies is $\triangle^{\mathcal{Q}}$. The policy regret is defined w.r.t. a fixed subset of policies, namely $\Pi \subseteq \triangle^{\mathcal{Q}}$, as:[4]

$$\mathfrak{R}_T(\Pi) := \sup_{\pi^* \in \Pi} \mathbb{E}\left[\sum_{t=1}^T \mathop{\mathbb{E}}_{m \sim \pi^*(q_t)} \left[\mathop{\mathbb{E}}_{a \sim \xi_m(q_t)}[r(q_t, a)]\right] - \sum_{t=1}^T r(q_t, a_t)\right]. \tag{4}$$

Properly deciding Π recovers several commonly used regret metrics: When $\Pi = \triangle^{\mathcal{Q}}$ takes the full policy space, it recovers the contextual regret, which is unfortunately intractable; when Π restricts all queries to a fixed model $m^* \in [M]$, it reduces to the expert regret, which essentially ignores the heterogeneity of queries; when $\Pi = \Pi_{\text{lin}}$ is the standard log-linear policy class, it turns out to be structurally misaligned with the underlying problem structure. In Section 3, we study the first question of this paper: what is the most appropriate choice of Π for embedding model routing?

3 Policy Class: Tradeoff between Alignment and Tractability

Instead of more common policy classes (discussed later in this section), as a main contribution of this paper, we argue that the most appropriate choice of Π is the following *log-quadratic* policy class:

$$\Pi_{\text{quad}} := \left\{\pi(m \mid q) \propto \exp(q^\mathsf{T} W_m q), \forall q \in \mathcal{Q} \;\middle|\; W_m \in \mathbb{S}^{d_q}, \forall m \in [M]\right\}, \tag{5}$$

where $\mathbb{S}^{d_q}$ is the space of symmetric $d_q \times d_q$ matrices. Π_{quad} defines a parametric class of model routing policies: The probability of routing a query q to a model m is determined by a softmax distribution over the *quadratic score* $q^\mathsf{T} W_m q$ (hence the name). We first justify that this log-quadratic class is well-aligned with the underlying structure of the embedding model routing problem.

Log-Quadratic Class. In Proposition 1, we establish that in embedding model routing, the expected reward of any model m under any query q is governed by a *low-rank quadratic* structure:

Proposition 1 (True Rewards are Roughly Low-Rank Quadratic)**.** *Given a query $q \in \mathcal{Q} = \mathbb{R}^{d_q}$, the expected reward of any model $m \in [M]$ is approximated by a low-rank quadratic form in q, namely*

$$R_m(q) := \mathop{\mathbb{E}}_{a \sim \xi_m(q)}[r(q,a)] = C(q) + q^\mathsf{T} W_m^* q + \delta_m(q), \quad \forall m \in [M], q \in \mathcal{Q} \subseteq \mathbb{R}^{d_q},$$

where $C(q)$ is a constant shared across all models (i.e., independent to m), $W_m^* \in \mathbb{S}^{d_q}$ is a symmetric matrix with a nuclear norm $\|W_m^*\|_* \le \rho \operatorname{rank}(W_m^*) \le 2s\rho$, and $\delta_m(q) = \mathcal{O}(\epsilon + \rho^2)$ is a residual term due to the non-linearity of ν and μ_m and to the softmax curvature $\|\Psi_m\|_2 \le \rho$.

The proof is in Appendix C.1. In light of Proposition 1, let us consider the following family of score-based embedding model router: Upon receiving a query q, the router calculates a score $s_m(q)$ for each model m, and performs routing via a softmax over scores with some temperature c^{-1}. Such an architecture is ubiquitous in classification or model routing systems (Devlin et al., 2019; Fedus et al., 2022). The optimal score-based routing policy is setting the score as the true reward, i.e.,

$$\pi^*(m \mid q) = \frac{\exp\big(cR_m(q)\big)}{\sum_{m'=1}^M \exp\big(cR_{m'}(q)\big)} \approx \frac{\exp\big(cq^\mathsf{T} W_m^* q\big)}{\sum_{m'=1}^M \exp\big(cq^\mathsf{T} W_{m'}^* q\big)} \propto \exp\Big(q^\mathsf{T}\big(cW_m^*\big)q\Big), \ \forall m. \tag{6}$$

Proposition 1 and Eq. (6) thus suggest that optimal model routing policies are naturally approximated by our log-quadratic class in Eq. (5), namely $\Pi_{\text{quad}} = \{\pi(m \mid q) \propto \exp(q^\mathsf{T} W_m q) \mid W_m \in \mathbb{S}^{d_q}, \forall m\}$, up to an residual $\delta_m(q)$. In addition to well-aligning with the underlying structure of embedding model routing, Π_{quad} is also tractable for learning: In Section 4, we design an efficient policy gradient algorithm, HPG, that attains $\widetilde{\mathcal{O}}(s\sqrt{MT})$ *linearized* policy regret (a surrogate metric we study; see Section 3.2) w.r.t. Π_{quad}. We thus conclude that the log-quadratic Π_{quad} *offers an ideal trade-off for embedding model routing*: well-approximating the optimal policies, while enabling efficient learning.

Having investigated our log-quadratic class, we examine why standard policy classes fall short. As depicted in Table 1 (whose log-quadratic baseline comes from flattening $\mathbb{W}$ and invoking Agarwal et al. 2021), we consider the following policy classes; detailed discussions are in Appendix C.

[4]Despite an adaptive adversary, Eq. (4) fixes the realized queries $q_1, q_2, \ldots, q_T$ and seeks the hindsight optimal $\pi^* \in \Pi$. The *strict* policy regret evaluating π^* on its induced queries must be $\Omega(T)$ (Arora et al., 2012).

Table 1: Comparison between Different Policy Classes or Regret Notions

Regret Notion	Aligned?	Tractable?	Regret	Computation	Comments
Expert Regret (constant policy)	✗	✓	$\sqrt{MT\log M}$ (Auer et al., 2002)	$\mathcal{O}(M)$ per round	Non-Adaptive Model Routing
Contextual Regret (unrestricted policy)	✓	✗	$\sqrt{d_a d_q T\log M\log T}$ (Li et al., 2019b)[a]	$\mathcal{O}(d_a^2 d_q^2 M)$ per round	Curse of Dimensionality
Log-Linear Regret ($\pi\propto\exp(\theta_m^\top q)$)	✗	✓	$\sqrt{d_q MT\log T}$ (Agarwal et al., 2021)[b]	$\mathcal{O}(d_q^2 M^2)$ per round	Structural Misspecification
Log-Quadratic Policy Regret ($\pi\propto\exp(q^\top W_m q)$)	✓	✓	$d_q\sqrt{MT\log T}$ (Agarwal et al., 2021)	$\mathcal{O}(d_q^4 M^2)$ per round	Curse of Dimensionality
			$s\sqrt{MT\log T}$ **(Ours)**	$\mathcal{O}(d_q^2 M)$ per round	**Efficient & Parameter-Free**

[a]This reduction only holds without misspecification ($\nu=\mu_m=0$) and when knowing model kernels (Ψ_m). "Curse of dimensionality" means, when treating d_q and d_a as the same order d, a regret bound that is linear in d.

[b]Their bound, as well as ours, only applies to *linearized* policy regret (Section 3.2); an exact guarantee under either class remains open.

Constant Class is Non-Adaptive (Appendix C.2). When $\Pi=\Pi_{\text{const}}$ only contains constant policies $\pi(q)\equiv m$, we recover the *expert regret* of bandits with expert advice, whose benchmark answers every query with the *same* model. It is not merely coarse but structurally uninformative: Theorem 6 establishes $\mathfrak{R}_T(\Pi_{\text{const}})=\Omega(\sqrt{MT/\log d_a})$ even when the learner observes all models' recommendations, so a black-box EXP3 (Auer et al., 2002) – ignoring linear rewards and low-rank kernels entirely – is already near-optimal.

Unrestricted Class gives "Curse of Dimensionality" (Appendix C.3). The other extreme, the unrestricted $\Pi=\triangle^{\mathcal{Q}}$, recovers the contextual regret and is favorable in that the learner competes with *any* query-adaptive policy – yet it stays notorious even under **(A1)** no misspecification and **(A2)** known model kernels. Under both, $\mathfrak{R}_T(\triangle^{\mathcal{Q}})$ reduces to a $d_a d_q$-dimensional contextual linear bandit, on which VCL-SupLinUCB (Li et al., 2019b) attains $\mathfrak{R}_T^C=\mathcal{O}(\sqrt{d_a d_q T\log M\log T})$ regret at $\mathcal{O}(d_a^2 d_q^2 M)$ per-round cost. Neither of them is acceptable once query and item spaces are large: with $d_q=d_a=768$ as in RouterRetriever (Lee et al., 2025; Izacard et al., 2022), the bound stays vacuous for $T/(\log M\log T)\le 768^2\approx 5\times 10^5$ rounds and each round costs $3\times 10^{11}M$ operations – the *"curse of dimensionality"* of Footnote *a*. Other seemingly more promising reductions (e.g., to d_a-dimensional adversarial contextual linear bandits, or to Md_q^2-dimensional ones via the lifted feature $q_t q_t^\top$) also fail to refine the regret: even assuming i.i.d. queries on top of **(A1)** and **(A2)**, the correlation between recommendation distributions $\xi_m(\cdot\mid q_t)$ and rewards $r(q_t,\cdot)$ makes existing algorithms inapplicable, and the learner never chooses q_t (see Appendices C.3.3 and C.3.4).

Log-Linear Class is Misaligned (Appendix C.4). In reinforcement learning, the commonly used class is the *log-linear* $\Pi_{\text{lin}}:=\{\pi(m\mid q)\propto\exp(\theta_m^\top q)\mid\theta_m\in\mathbb{R}^{d_q},\forall m\}$ (Agarwal et al., 2021; Mei et al., 2020), whose Md_q-dimensional parameter space is far smaller than that of Π_{quad}. Based on Natural Policy Gradient (NPG; Kakade 2001), Agarwal et al. (2021) attain $\mathcal{O}(\sqrt{d_q MT\log T})$ *linearized* policy regret at $\mathcal{O}(d_q^2M^2)$ per round,[5] shrinking the vacuous regime above from 768^2 to $768M$ rounds. Π_{lin} is in fact already deployed: RouterRetriever (Lee et al., 2025) routes by the cosine similarity between q_t and each model's "representative training data," essentially a $\theta_m^\top q$ score. Yet Π_{lin} is structurally misaligned: Proposition 1 makes the true rewards *quadratic* in q, and a linear score cannot capture interactions between coordinates of q. Even an optimal policy regret w.r.t. Π_{lin} therefore leaves the learner sub-optimal at the actual routing task.

3.1 Numerical Verification on Amazon ESCI Dataset

In Table 2, we compare the optimal routing policies within four policy classes – the constant Π_{const}, the log-linear Π_{lin}, the log-quadratic Π_{quad}, and the unrestricted $\triangle^{\mathcal{Q}}$ – over the Amazon ESCI dataset (Reddy et al., 2022). This dataset benchmarks 97,345 difficult search queries in e-commerce: Each consumer query is associated with several candidate products (18.68 on average) with titles,

[5]Their linearization also resolves the non-convexity of softmax, but happens in a different space from ours; see Section 3.2 and Appendices C.4 and C.5 for more detailed discussions.

Table 2: Sub-Optimality Gap of Different Policy Classes on Amazon ESCI Dataset

Query Category	Num of Queries	Π_{const} Gap (↓)	Π_{lin} Gap (↓)	$\Pi_{\text{const}} \to \Pi_{\text{lin}}$ Improvement (↑)	Π_{quad} Gap (↓)	$\Pi_{\text{lin}} \to \Pi_{\text{quad}}$ Improvement (↑)
all	97,345	0.035	0.030	13.8%	**0.011**	**63.5%**
`nlqec`	169	0.110	0.004	96.0%	**<0.001**	**99.7%**
`behavioral`	1,253	0.049	0.006	87.8%	**<0.001**	**97.5%**
`parse-pattern`	4,250	0.046	0.014	69.8%	**0.002**	**89.0%**
`negations`	2,899	0.134	0.054	59.9%	**0.011**	**80.1%**
other	88,774	0.031	0.026	17.4%	**0.009**	**66.2%**

descriptions, and keywords, all in natural language. Each query-action pair is annotated with one of E(xact), S(ubstitute), C(omplement), or I(rrelevant), which we convert into numerical values as the reward function $r\colon \mathcal{Q} \times \mathcal{A} \to [0, 1]$. We consider $M = 8$ lightweight embedding models as candidate experts, spanning symmetric semantic encoders like `miniLM` (Wang et al., 2020; 2021), asymmetric search models like `msmarco-distilbert` (Reimers & Gurevych, 2019), and modern prompt-driven encoders (Wang et al., 2022; Xiao et al., 2024). Detailed experimental setup and discussions are in Appendix D.

To facilitate model routing, we embed queries and items into 768-dimensional vector spaces via CONTRIEVER (Izacard et al., 2022) before feeding them to the learner; this follows the practice of ROUTERRETRIEVER (Lee et al., 2025). We measure a routing policy π by its *sub-optimality gap* $\text{Gap}(\pi)$: averaged uniformly over ESCI queries, the expected reward that π forfeits relative to routing each query to its own best model (see Eq. (22) for the formal definition).

Table 2 reports $\text{Gap}(\pi^*_{\text{const}})$, $\text{Gap}(\pi^*_{\text{lin}})$, and $\text{Gap}(\pi^*_{\text{quad}})$ for the optimal policy of each class, together with the relative improvements $1 - \text{Gap}(\pi^*_{\text{lin}})/\text{Gap}(\pi^*_{\text{const}})$ and $1 - \text{Gap}(\pi^*_{\text{quad}})/\text{Gap}(\pi^*_{\text{lin}})$; how each optimum is computed is deferred to Appendix D. The first gap measures whether routing is needed at all: a large $\text{Gap}(\pi^*_{\text{const}})$ means no single model dominates the others on most queries. Besides the full dataset, we report a few "query categories" that Reddy et al. (2022) single out as exceptionally hard.

The pattern is consistent across rows: the log-linear class recovers only a small part of the constant-policy gap, whereas the log-quadratic class closes most of what remains. The contrast is sharpest where routing matters most: on the harder categories such as `nlqec` and `negations`, where the constant policy is far worse than its full-dataset average, Π_{quad} still closes over 90% of that gap. This evidences both the necessity of dynamic model routing and the low misspecification of Π_{quad}.

3.2 ADDITIONAL NOTATIONS AND LINEARIZED POLICY REGRET

We therefore focus on our log-quadratic Π_{quad} class, optimizing over its parameters rather than over $\triangle$ as bandit convex optimization would, since a point chosen in $\triangle$ cannot depend on the query. Since each policy is parameterized by M symmetric $d_q \times d_q$ matrices, we collect them into one diagonal block matrix $\boldsymbol{W} = \text{diag}(W_1, W_2, \ldots, W_M) \in \mathbb{S}^{Md_q}$, which we call a *parameter* and denote by a boldface letter; the parameter space $\mathbb{W} \subseteq \mathbb{S}^{Md_q}$ thus has dimension Md_q^2. According to Eq. (5), each parameter $\boldsymbol{W} \in \mathbb{W}$ induces a model routing policy:

$$\pi_{\boldsymbol{W}}(m \mid q) := \frac{\exp(q^\mathsf{T} W_m q)}{\sum_{m'=1}^{M} \exp(q^\mathsf{T} W_{m'} q)}, \quad \forall m = 1, 2, \ldots, M. \tag{7}$$

For each round $t \in [T]$, define the expected loss (negative reward) suffered by any parameter $\boldsymbol{W}$ as

$$\mathcal{L}_t(\boldsymbol{W}) := - \mathop{\mathbb{E}}_{m \sim \pi_{\boldsymbol{W}}(q_t)} \left[\mathop{\mathbb{E}}_{a \sim \xi_m(q_t)} [r(q_t, a)] \right], \quad \forall t \in [T], \boldsymbol{W} \in \mathbb{W}. \tag{8}$$

Then the policy regret w.r.t. Π_{quad}, namely $\mathfrak{R}_T(\Pi_{\text{quad}})$, is $\sup_{\boldsymbol{W}^* \in \mathbb{W}} \mathbb{E}[\sum_{t=1}^{T} \mathcal{L}_t(\boldsymbol{W}_t) - \mathcal{L}_t(\boldsymbol{W}^*)]$. From Proposition 1, we assume that an optimal $\boldsymbol{W}^* \in \mathbb{W}$ ensures $\|W_m^*\|_* \le 2s$, $\forall m$.

Unfortunately, since softmax is non-convex, neither policy (Eq. (7)) nor loss (Eq. (8)) is convex. Consistent with previous policy gradient analysis (detailed in Appendix C.4; Agarwal et al. 2021), we replace $\mathfrak{R}_T(\Pi_{\text{quad}})$ by a surrogate obtained from *local linearization* on the parameter space $\mathbb{W}$:

$$\widetilde{\mathfrak{R}}_T(\Pi_{\text{quad}}) := \sup_{\boldsymbol{W}^* \in \mathbb{W}} \mathbb{E} \left[\sum_{t=1}^{T} \left\langle \nabla \mathcal{L}_t(\boldsymbol{W}_t), \boldsymbol{W}_t - \boldsymbol{W}^* \right\rangle_F \right]. \tag{9}$$

Algorithm 1 **H**ypentropy **P**olicy **G**radient (HPG) for Model Routing

Input: Game length T, query set $\mathcal{Q} \subseteq \mathbb{R}^{d_q}$, candidate item set $\mathcal{A} \subseteq \mathbb{R}^{d_a}$, number of models M. Hypentropy parameter $\beta > 0$, learning rate $\eta > 0$, nuclear norm bound $\tau > 0$.

1: Initialize $\boldsymbol{W}_1$ as the $Md_q \times Md_q$ all-zero matrix, which is a diagonal block matrix in $\mathbb{S}^{Md_q}$.
2: **for** $t = 1, 2, \ldots, T$ **do**
3: Observe query q_t. Decide model m_t according to $\pi_{\boldsymbol{W}_t}(q_t)$ ▷ $\pi_{\boldsymbol{W}}$ *is in Eq. (7)*
4: Observe model m_t's recommendation $a_{t,m_t} \sim \xi_{m_t}(q_t)$ ▷ ξ_m *is in Eq. (3)*
5: Recommend item a_{t,m_t}. Observe $r_t = r(q_t, a_{t,m_t}) + \eta_t$ ▷ $r(q,a)$ *is in Eq. (1)*
6: Construct gradient estimator $\widehat{\boldsymbol{G}}_t$ using q_t, m_t, r_t ▷ $\widehat{\boldsymbol{G}}_t$ *is in Eq. (13)*
7: Perform an Online Mirror Descent (OMD) step over $\mathbb{B}_*^M(\tau)$. ▷ *Eqs. (10) to (12)*

Each summand recovers the Frank-Wolfe gap of online non-convex optimization (Lafond et al., 2015; Reddi et al., 2016), common in, e.g., online submodular optimization (Hassani et al., 2017; Chen et al., 2018; Zhang et al., 2019). See Appendix C.5 for a more technical comparison between $\mathfrak{R}_T(\Pi_{\text{quad}})$ and our linearized $\widetilde{\mathfrak{R}}_T(\Pi_{\text{quad}})$, whose one-step forms differ by an indefinite-sign factor $\log \frac{\pi^*}{\pi}$ that defeats gradient dominance arguments. While we consider $\widetilde{\mathfrak{R}}_T$ as a surrogate metric to handle non-convexity, tackling $\mathfrak{R}_T$ exactly remains an important open question.

4 Hypentropy Policy Gradient (HPG) for Model Routing

Our algorithm for the log-quadratic policy class Π_{quad} draws on the routing application only through the low-rankness that Proposition 1 grants $\boldsymbol{W}^*$. Moving from the log-linear class Π_{lin} to Π_{quad} improves approximation accuracy (see Proposition 1 and Eq. (6)), but comes at a cost: The parameter space dimension increases from $\dim(\Theta) = d_q M$ to $\dim(\mathbb{W}) = d_q^2 M$. Hence, naively flattening $\mathbb{W}$ into a $d_q^2 M$-dimensional vector space and applying policy gradient methods gives $\mathcal{O}(d_q\sqrt{MT\log T})$ regret and requires $\mathcal{O}(d_q^4 M)$ computation per round (Agarwal et al., 2021). Similar to the unrestricted class discussed in Section 3, both bounds can be prohibitively large.

Fortunately, since embedding models operate on low-dimensional representation spaces, Proposition 1 makes the optimal parameter $\boldsymbol{W}^*$ low-rank, which is the key to bypassing the curse of dimensionality: restricting learning to low-rank W_m's reduces the effective dimension to $\Theta(sd_qM)$. Exploiting this is not routine, though. The low-rank structure lies in the *parameter* space, not the *reward* space – the kernel Ψ^* in Eq. (1) is *not* low-rank, so low-rank or sparse regression (ridge, lasso) does not apply – and standard policy gradient is blind to it: NPG, equivalent to Online Mirror Descent (OMD) with a KL regularizer (Geist et al., 2019), is isotropic across directions and does not adapt to the rank.[6]

We therefore use the hypentropy regularizer (Ghai et al., 2020) to promote low-rankness (Section 4.1), paired with a REINFORCE-style gradient estimator for our bandit feedback and partial observability (Section 4.2); together they give **H**ypentropy **P**olicy **G**radient (HPG) in Algorithm 1. The estimator is unbiased and low-variance, giving provable guarantees under incomplete information (Theorem 3), and its structure also lets Section 5 implement HPG in $\mathcal{O}(d_q^2 M)$ time per round and parameter-freely.

4.1 Nuclear-Norm Ball, Hypentropy, and Online Mirror Descent

Since the optimal parameter $\boldsymbol{W}^*$ is low-rank, the first component of HPG restricts the parameter space to a *product nuclear-norm ball*:

$$\mathbb{B}_*^M(\tau) := \left\{\operatorname{diag}(W_1, W_2, \ldots, W_M) \;\middle|\; W_m \in \mathbb{S}^{d_q}, \|W_m\|_* \le \tau, \forall m\right\}. \tag{10}$$

Here $\|\cdot\|_*$, the nuclear norm (sum of absolute eigenvalues), is the usual convex surrogate for rank, and $\tau \ge 0$ is to be determined: since Proposition 1 gives $\|W_m^*\|_* \le 2s\rho \le 2s$, we have $\boldsymbol{W}^* \in \mathbb{B}_*^M(2s)$. For ease of presentation we assume s known for now, and drop this in Section 5.2. HPG then runs OMD on $\mathbb{B}_*^M(\tau)$ with the hyperbolic entropy (*hypentropy*; Ghai et al. 2020), which is strongly convex with respect to the nuclear norm, making it well-suited for promoting low-rank structures in the

[6] Vanilla NPG is only defined on the flattened vector space, where the low-rank structure of $\boldsymbol{W}^*$ is lost. Exploiting low-rankness requires working with matrix spaces, but the matrix analogue of NPG – OMD with von Neumann entropy $\Phi(X) = \operatorname{Tr}(X \log X - X)$ – requires *positive semi-definite* matrices (Tsuda et al., 2005).

parameter space.[7] Formally, for a parameter $\beta > 0$ to be specified later, the β-hypentropy of a symmetric matrix $\boldsymbol{W} \in \mathbb{S}^{Md_q}$ is defined as

$$\Phi_\beta(\boldsymbol{W}) := \sum_{i=1}^{Md_q} \left(\lambda_i \operatorname{arcsinh} \frac{\lambda_i}{\beta} - \sqrt{\lambda_i^2 + \beta^2} \right), \quad \forall \boldsymbol{W} \in \mathbb{S}^{Md_q}, \tag{11}$$

where $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{Md_q}$ are the eigenvalues of $\boldsymbol{W}$ in non-increasing order. Using Φ_β as the regularizer, the OMD update of HPG over the product ball $\mathbb{B}_*^M(\tau)$ takes the following form:

$$\boldsymbol{W}_{t+1} = \operatorname*{argmin}_{\boldsymbol{W} \in \mathbb{B}_*^M(\tau)} \left(\eta \langle \boldsymbol{W}, \widehat{\boldsymbol{G}}_t \rangle_F + D_\Phi^\beta(\boldsymbol{W} \| \boldsymbol{W}_t) \right), \quad \forall t = 1, 2, \ldots, T, \tag{12}$$

where $\eta > 0$ is a parameter to be determined; $\widehat{\boldsymbol{G}}_t$ is the (estimated) gradient defined in Section 4.2; and $D_\Phi^\beta(\boldsymbol{W} \| \boldsymbol{W}') := \Phi_\beta(\boldsymbol{W}) - \Phi_\beta(\boldsymbol{W}') - \langle \nabla \Phi_\beta(\boldsymbol{W}'), \boldsymbol{W} - \boldsymbol{W}' \rangle_F$ (detailed in Appendix E.1). This Bregman projection, usually computationally expensive, is implementable in $\mathcal{O}(d_q^2 M)$ time thanks to the hypentropy regularizer Φ_β and the nuclear-norm ball $\mathbb{B}_*^M(\tau)$ (Section 5.1).

4.2 Gradient Estimation and Policy Regret Bound

The other component of HPG is the gradient estimator $\widehat{\boldsymbol{G}}_t$. Full-information OMD would put the true $\nabla \mathcal{L}_t(\boldsymbol{W}_t)$ in its place in Eq. (12), but under bandit feedback and partial observability we see only one item $a_{t,m_t} \sim \xi_{m_t}(q_t)$ and one noisy reward r_t, and must estimate the gradient from that alone. We design a REINFORCE-style gradient estimator: let $\widehat{\boldsymbol{G}}_t = \operatorname{diag}(\widehat{G}_{t,1}, \widehat{G}_{t,2}, \ldots, \widehat{G}_{t,M})$, where

$$\widehat{G}_{t,m} = -r_t \big(\mathbb{1}[m_t = m] - \pi_{\boldsymbol{W}_t}(m \mid q_t) \big) q_t q_t^\mathsf{T}, \quad \forall m = 1, 2, \ldots, M. \tag{13}$$

Lemma 2 (Gradient Estimation)**.** *Let $\mathbb{E}_t$ be the expectation taken only w.r.t. the randomness in round t, i.e., conditional on the natural filtration up to round $(t-1)$. Then we have $\mathbb{E}_t[\widehat{\boldsymbol{G}}_t \mid q_t] = \nabla_{\boldsymbol{W}} \mathcal{L}_t(\boldsymbol{W}_t)$. Furthermore, $\sum_{m=1}^M \|\widehat{G}_{t,m}\|_2^2 \leq 2 r_t^2$ almost surely, and consequently $\mathbb{E}_t[\sum_{m=1}^M \|\widehat{G}_{t,m}\|_2^2 \mid q_t] \leq 10$.*

Lemma 2 indicates that $\widehat{\boldsymbol{G}}_t$ is conditionally unbiased and admits constant conditional second-order moments. While similar estimators and properties are widely used in policy gradient methods (see, e.g., Williams, 1992; Sutton et al., 1999), our log-quadratic class has a further property: the $\widehat{G}_{t,m}$ in Eq. (13) is *rank-one*, which is pivotal for the computational efficiency of HPG in Section 5.1.

Combining this gradient estimator with OMD updates yields the full HPG algorithm (see Algorithm 1).

Theorem 3 (Regret Bound)**.** *Assume that the optimal $\boldsymbol{W}^* \in \mathbb{W}$ ensures $\|W_m^*\|_* \leq 2s$, $\forall m$. When setting $\tau = 2s$, $\beta = 2s/d_q$ and $\eta = \sqrt{(M \log d_q)/T}$, HPG gives $\widetilde{\mathfrak{R}}_T(\Pi_{quad}) = \mathcal{O}(s\sqrt{MT \log d_q})$.*

Theorem 3 indicates that if the low-rank parameter s is known to the learner, the HPG algorithm attains $\widetilde{\mathcal{O}}(s\sqrt{MT})$ linearized policy regret against the best in-class parameter, thus provably adapting to the underlying low-rank structures of embedding models. In Section 5.2, we further show that when the low-rank parameter s is unknown, the HPG algorithm is still implementable in a *parameter-free* way that attains an almost identical regret bound. The full proofs are in Appendices E.3 and E.4.

5 Practical Implementation of the HPG Algorithm

While attaining $\widetilde{\mathcal{O}}(s\sqrt{MT})$ regret, two issues arise before implementing HPG for realistic model routing: the computational cost of the Bregman projection step in Eq. (12), and the knowledge of s in Theorem 3. We resolve both issues in this section, thus demonstrating the practicality of HPG.

5.1 Computationally Efficient Implementation

The Bregman projection of Eq. (12) is a convex program over the $M^2 d_q^2$-dimensional space $\mathbb{S}^{Md_q}$, on which a vanilla implementation of the interior point method would spend $\mathcal{O}(M^6 d_q^6)$ time per round

[7] The connection between hypentropy and low-rankness has been exploited by Woodworth et al. (2020), Pesme et al. (2021), Li et al. (2022b), Varre et al. (2023), and Jacobs et al. (2025) to understand the implicit bias/regularization in neural network training, by Wu & Rebeschini (2020; 2023) in sparse phase retrieval problems, and by Wu & Rebeschini (2021) for low-rank matrix sensing. However, we are unaware of existing works using this property for policy regret, bandit feedback, or embedding model routing.

(Boyd & Vandenberghe, 2004) – some $2 \times 10^{17} M$ floating point operations at $d_q = 768$. Fortunately, since the hypentropy Φ_β and nuclear-norm ball $\mathbb{B}_*^M(\tau)$ are both defined w.r.t. eigenvalues, and diagonal block matrices preserve eigenvalues, we establish a *direct sum property* (Lemma 12 in Appendix E.2). This allows us to treat each model's parameter separately. Furthermore, we prove that each m only requires $\mathcal{O}(d_q^2)$ time per round by maintaining an eigen-decomposition of $W_{t,m}$:

$$W_{t,m} = U_{t,m} \operatorname{diag}(\lambda_{t,m}) U_{t,m}^\mathsf{T}, \quad \forall t \in [T], \tag{14}$$

where $U_{t,m} \in \mathbb{R}^{d_q \times d_q}$ and $\lambda_{t,m} \in \mathbb{R}^{d_q}$. Moving between the primal and mirror spaces is free: as detailed in Appendix E.1, $\nabla\Phi_\beta$ and its inverse simply apply $\operatorname{arcsinh}(\cdot/\beta)$ and $\beta \sinh(\cdot)$ to the eigenvalues in $\mathcal{O}(d_q)$ time. The remaining of Eq. (12) then decomposes into two easy steps:

1. **Gradient Descent in Mirror Space.** Perform $\widetilde{Y}_{t+1,m} = \nabla\Phi_\beta(W_{t,m}) - \eta \widehat{G}_{t,m}$ and map back to $\widetilde{W}_{t+1,m} = (\nabla\Phi_\beta)^{-1}(\widetilde{Y}_{t+1,m})$. Eigen-decomposing $\widetilde{Y}_{t+1,m}$ generally takes $\mathcal{O}(d_q^3)$ time, but as $\widehat{G}_{t,m}$ is rank-one (Eq. (13)), *eigen-updating* from $W_{t,m}$ takes only $\mathcal{O}(d_q^2)$ (Bunch et al., 1978).
2. **Projection onto $\mathbb{B}_*(\tau)$ based on D_Φ^β.** Usually expensive as well, but our $\mathbb{B}_*(\tau)$ and Φ_β both depend only on the *eigenvalues*, so von Neumann's trace inequality gives

$$W_{t+1,m} = U_{t+1,m} \operatorname{diag}(\lambda_{t+1,m}) U_{t+1,m}^\mathsf{T}, \quad \lambda_{t+1,m} = \operatorname*{argmin}_{\|\lambda\|_1 \le \tau} \sum_{i=1}^{d_q} D_\phi^\beta(\lambda_i \| \widetilde{\lambda}_{t+1,m,i}), \tag{15}$$

   where D_ϕ^β is the Bregman divergence of $\phi_\beta(x) = x \operatorname{arcsinh} \frac{x}{\beta} - \sqrt{x^2 + \beta^2}$: a d_q-dimensional convex program, solvable in $\mathcal{O}(d_q \log d_q)$ time.

In Appendix F.1, we prove Eq. (12) is equivalent to these steps. This gives the following theorem:
Theorem 4 (Computational Cost)**.** *HPG is implementable in $\mathcal{O}(d_q^2 M)$ time complexity per round.*

In practice one can do better still: Appendix F.1 shows the projection step is itself low-rank, so eigen-truncating to the top-τ eigenvalues brings the eigen-update down to $\mathcal{O}(\tau d_q)$, and OMD updates can be batched in the mirror space at the cost of the strict guarantee.

5.2 Parameter-Free Implementation

The remaining issue is the nuclear-norm parameter τ in Algorithm 1: In Theorem 3, we set τ to $2s$ where s, the maximum rank of model kernels, is *unknown* to the learner. Fortunately, utilizing the parameter-free online learning technique (Cutkosky & Orabona, 2018) and a clipping argument, the HPG algorithm can be implemented without *any* prior information on s while enjoying a similar regret guarantee. The detailed algorithm description and proof are deferred to Appendix F.2.
Theorem 5 (Parameter-Free HPG)**.** *Assume that the optimal $\boldsymbol{W}^* \in \mathbb{W}$ ensures $\|W_m^*\|_* \le 2s$, $\forall m$, but the low-rank parameter s is unknown. Then the parameter-free HPG algorithm (Algorithm 2) ensures a linearized policy regret bound of $\widetilde{\mathfrak{R}}_T(\Pi_{\mathit{quad}}) = \mathcal{O}(s\sqrt{MT \log(d_q T)} + sM\sqrt{\log T}\log(sT))$.*

Once $T \gtrsim M \log T$, the first term dominates and this matches the $\mathcal{O}(s\sqrt{MT \log d_q})$ in Theorem 3 up to poly-logarithmic factors. HPG's favorable properties all remain: adapting to the unknown low-rank structure, avoiding $\operatorname{poly}(d_q)$ dependencies, and admitting an efficient implementation.

6 Conclusion and Future Research

In this paper, we formalize the online embedding model routing problem as an adversarial contextual linear bandit with low-rank experts. One key challenge is that the learner only invokes a single model per round, and observes neither the full recommendation distribution nor what an unselected model would have returned. The standard benchmarks all fail here: The constant class is non-adaptive, the unrestricted class is intractable, the log-linear class is structurally misaligned. We instead identified a log-quadratic class by proving each model's expected reward is a low-rank quadratic form (Proposition 1); we numerically verified our claim on Amazon ESCI dataset. For the resulting log-quadratic policy optimization problem, which can be of independent interest, Hypentropy Policy Gradient (HPG) attains $\widetilde{\mathcal{O}}(s\sqrt{MT})$ linearized policy regret with no $\operatorname{poly}(d_q)$ dependence, can be computationally efficiently implemented, and does not require knowing the low-rank parameter s. Transferring our guarantees from the linearized surrogate $\widetilde{\mathfrak{R}}_T$ to the exact policy regret $\mathfrak{R}_T$ is left for future investigation, where an indefinite-sign factor $\log(\pi^*/\pi)$ defeats gradient dominance arguments. Hardness results – whether an $\Omega(s)$-dependency is necessary for Π_{quad} – also remain open.

AI USE STATEMENT

(This section is **required** and does not count toward the page limit.)

In this work, we used generative AI tools for generating synthetic data sets for Figure 1; writing and executing the codes for Figure 1 and Table 2; editing the paper to improve readability and meet the page limit; summarizing existing literature; and checking the correctness of proofs in the Appendix. We have not used generative AI tools for developing the theoretical model in Section 2; proposing, formulating, proving, or refining the claims in Proposition 1 and Theorems 3 to 5; or assisting in the writing of the proofs. The design of research methodology; translation; cleaning or reformatting datasets; qualitative or thematic data analysis; and interpreting results are not applicable to this work. We have reviewed all AI-assisted work. We proofread all LLM-generated codes. We checked the literature survey against their abstracts. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. *Advances in Neural Information Processing Systems*, 24:2312–2320, 2011.

Alekh Agarwal, Daniel Hsu, Satyen Kale, John Langford, Lihong Li, and Robert Schapire. Taming the monster: A fast and simple algorithm for contextual bandits. In *International Conference on Machine Learning*, pp. 1638–1646. PMLR, 2014.

Alekh Agarwal, Sham M Kakade, Jason D Lee, and Gaurav Mahajan. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *Journal of Machine Learning Research*, 22(98):1–76, 2021.

Carlo Alfano, Rui Yuan, and Patrick Rebeschini. A novel framework for policy mirror descent with general parameterization and linear convergence. *Advances in Neural Information Processing Systems*, 36:30681–30725, 2023.

Zeyuan Allen-Zhu, Sébastien Bubeck, and Yuanzhi Li. Make the minority great again: First-order regret bound for contextual bandits. In *International Conference on Machine Learning*, pp. 186–194. PMLR, 2018.

Chamy Allenberg, Peter Auer, László Györfi, and György Ottucsák. Hannan consistency in on-line learning in case of unbounded losses under partial monitoring. In *International Conference on Algorithmic Learning Theory*, pp. 229–243. Springer, 2006.

Raman Arora, Ofer Dekel, and Ambuj Tewari. Online bandit learning against an adaptive adversary: from regret to policy regret. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pp. 1747–1754, 2012.

Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. MS MARCO: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*, 2016.

Lilian Besson and Emilie Kaufmann. What doubling tricks can and can't do for multi-armed bandits. *arXiv preprint arXiv:1803.06971*, 2018.

Alina Beygelzimer, John Langford, Lihong Li, Lev Reyzin, and Robert Schapire. Contextual bandit algorithms with supervised learning guarantees. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 19–26. JMLR Workshop and Conference Proceedings, 2011.

Jalaj Bhandari and Daniel Russo. Global optimality guarantees for policy gradient methods. *Operations Research*, 72(5):1906–1927, 2024.

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

James R Bunch, Christopher P Nielsen, and Danny C Sorensen. Rank-one modification of the symmetric eigenproblem. *Numerische Mathematik*, 31(1):31–48, 1978.

Nicolò Cesa-Bianchi, Khaled Eldowa, Emmanuel Esposito, and Julia Olkhovskaya. Improved regret bounds for bandits with expert advice. *Journal of Artificial Intelligence Research*, 83, 2025.

Olivier Chapelle and Lihong Li. An empirical evaluation of Thompson sampling. *Advances in Neural Information Processing Systems*, 24, 2011.

Zachary Chase, Shinji Ito, and Idan Mehalel. A tight lower bound for non-stochastic multi-armed bandits with expert advice. *arXiv preprint arXiv:2511.00257*, 2025.

Kamalika Chaudhuri, Yoav Freund, and Daniel J Hsu. A parameter-free hedging algorithm. *Advances in neural information processing systems*, 22, 2009.

Lin Chen, Hamed Hassani, and Amin Karbasi. Online continuous submodular maximization. In *International Conference on Artificial Intelligence and Statistics*, pp. 1896–1905. PMLR, 2018.

Lingjiao Chen, Matei Zaharia, and James Zou. FrugalGPT: How to use large language models while reducing cost and improving performance. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856.

Yu Chen, Jiatai Huang, Yan Dai, and Longbo Huang. uniINF: Best-of-both-worlds algorithm for parameter-free heavy-tailed MABs. In *The Thirteenth International Conference on Learning Representations*, 2025.

Wei Chu, Lihong Li, Lev Reyzin, and Robert Schapire. Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 208–214. JMLR Workshop and Conference Proceedings, 2011.

Andrzej Cichocki, Toshihisa Tanaka, Frank Nielsen, and Sergio Cruces. Mirror descent and exponentiated gradient algorithms using trace-form entropies. *Entropy*, 27(12):1243, 2025.

Jasmine Collins, Shubham Goel, Kenan Deng, Achleshwar Luthra, Leon Xu, Erhan Gundogdu, Xi Zhang, Tomas F Yago Vicente, Thomas Dideriksen, Himanshu Arora, Matthieu Guillaumin, and Jitendra Malik. ABO: Dataset and benchmarks for real-world 3D object understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21126–21136, 2022.

Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for Youtube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pp. 191–198, 2016.

Ashok Cutkosky and Francesco Orabona. Black-box reductions for parameter-free online learning in Banach spaces. In *Conference On Learning Theory*, pp. 1493–1529. PMLR, 2018.

Yan Dai, Haipeng Luo, Chen-Yu Wei, and Julian Zimmert. Refined regret for adversarial MDPs with linear function approximation. In *International Conference on Machine Learning*, pp. 6726–6759. PMLR, 2023.

Varsha Dani, Thomas P Hayes, and Sham M Kakade. Stochastic linear optimization under bandit feedback. In *21st Annual Conference on Learning Theory*, number 101, pp. 355–366, 2008.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4171–4186, 2019.

Jialin Dong and Lin Yang. Does sparsity help in learning misspecified linear bandits? In *International Conference on Machine Learning*, pp. 8317–8333. PMLR, 2023.

Simon S Du, Sham M Kakade, Ruosong Wang, and Lin F Yang. Is a good representation sufficient for sample efficient reinforcement learning? In *International Conference on Learning Representations*, 2020.

Ky Fan. On a theorem of Weyl concerning eigenvalues of linear transformations I. *Proceedings of the National Academy of Sciences*, 35(11):652–655, 1949.

Maryam Fazel, Rong Ge, Sham Kakade, and Mehran Mesbahi. Global convergence of policy gradient methods for the linear quadratic regulator. In *International Conference on Machine Learning*, pp. 1467–1476. PMLR, 2018.

William Fedus, Barret Zoph, and Noam Shazeer. Switch Transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

Dylan Foster and Alexander Rakhlin. Beyond UCB: Optimal and efficient contextual bandits with regression oracles. In *International Conference on Machine Learning*, pp. 3199–3210. PMLR, 2020.

Dylan Foster, Alekh Agarwal, Miroslav Dudík, Haipeng Luo, and Robert Schapire. Practical contextual bandits with regression oracles. In *International Conference on Machine Learning*, pp. 1539–1548. PMLR, 2018.

Dylan J Foster, Claudio Gentile, Mehryar Mohri, and Julian Zimmert. Adapting to misspecification in contextual bandits. *Advances in Neural Information Processing Systems*, 33:11478–11489, 2020.

Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

Matthieu Geist, Bruno Scherrer, and Olivier Pietquin. A theory of regularized markov decision processes. In *International Conference on Machine Learning*, pp. 2160–2169. PMLR, 2019.

Gianmarco Genalti, Lupo Marsigli, Nicola Gatti, and Alberto Maria Metelli. (ε, u)-adaptive regret minimization in heavy-tailed bandits. In *The Thirty Seventh Annual Conference on Learning Theory*, pp. 1882–1915. PMLR, 2024.

Udaya Ghai, Elad Hazan, and Yoram Singer. Exponentiated gradient meets gradient descent. In *Algorithmic Learning Theory*, pp. 386–407. PMLR, 2020.

Avishek Ghosh, Sayak Ray Chowdhury, and Aditya Gopalan. Misspecified linear bandits. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

Aditya Gopalan, Odalric-Ambrym Maillard, and Mohammadi Zaki. Low-rank bandits with latent mixtures. *arXiv preprint arXiv:1609.01508*, 2016.

Ming Gu and Stanley C Eisenstat. A stable and efficient algorithm for the rank-one modification of the symmetric eigenproblem. *SIAM Journal on Matrix Analysis and Applications*, 15(4):1266–1276, 1994.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pp. 1321–1330. PMLR, 2017.

Andras Gyorgy and Pooria Joulani. Adapting to delays and data in adversarial multi-armed bandits. In *International Conference on Machine Learning*, pp. 3988–3997. PMLR, 2021.

Hamed Hassani, Mahdi Soltanolkotabi, and Amin Karbasi. Gradient methods for submodular maximization. *Advances in Neural Information Processing Systems*, 30, 2017.

Elad Hazan and Tomer Koren. The computational power of optimization in online learning. In *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*, pp. 128–141, 2016.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Xiaoyan Hu, Ho fung Leung, and Farzan Farnia. An online learning approach to prompt-based selection of generative models and LLMs. In *Forty-second International Conference on Machine Learning*, 2025.

Jiatai Huang, Yan Dai, and Longbo Huang. Adaptive best-of-both-worlds algorithm for heavy-tailed multi-armed bandits. In *International Conference on Machine Learning*, pp. 9173–9200. PMLR, 2022.

Jiatai Huang, Yan Dai, and Longbo Huang. Banker online mirror descent: A universal approach for delayed online bandit learning. In *International Conference on Machine Learning*, pp. 13814–13844. PMLR, 2023.

Jui-Ting Huang, Ashish Sharma, Shuying Sun, Li Xia, David Zhang, Philip Pronin, Janani Padmanabhan, Giuseppe Ottaviano, and Linjun Yang. Embedding-based retrieval in Facebook search. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2553–2561, 2020.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*, pp. 2333–2338, 2013.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. Position: The platonic representation hypothesis. In *Forty-First International Conference on Machine Learning*, 2024.

Shinji Ito. On the minimax regret for contextual linear bandits and multi-armed bandits with expert advice. *Advances in Neural Information Processing Systems*, 37:61793–61812, 2024.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856.

Tom Jacobs, Chao Zhou, and Rebekka Burkholz. Mirror, mirror of the flow: How does regularization shape implicit bias? In *International Conference on Machine Learning*, pp. 26673–26699. PMLR, 2025.

Kyoungseok Jang, Kwang-Sung Jun, Se-Young Yun, and Wanmo Kang. Improved regret bounds of bilinear bandits using action space analysis. In *International Conference on Machine Learning*, pp. 4744–4754. PMLR, 2021.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C Park. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 7036–7050, 2024.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. LLM-blender: Ensembling large language models with pairwise ranking and generative fusion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pp. 14165–14178, 2023.

Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I Jordan. Provably efficient reinforcement learning with linear function approximation. *Mathematics of Operations Research*, 48(3):1496–1521, 2023.

Wittawat Jitkrittum, Harikrishna Narasimhan, Ankit Singh Rawat, Jeevesh Juneja, Congchao Wang, Zifeng Wang, Alec Go, Chen-Yu Lee, Pradeep Shenoy, Rina Panigrahy, Aditya Krishna Menon, and Sanjiv Kumar. Universal model routing for efficient LLM inference. In *The Fourteenth International Conference on Learning Representations*, 2026.

Kwang-Sung Jun, Rebecca Willett, Stephen Wright, and Robert Nowak. Bilinear bandits with low-rank structure. In *International Conference on Machine Learning*, pp. 3163–3172. PMLR, 2019.

Sham Kakade. A natural policy gradient. In *Proceedings of the 15th International Conference on Neural Information Processing Systems: Natural and Synthetic*, pp. 1531–1538, 2001.

Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pp. 267–274, 2002.

Sham M Kakade, Shai Shalev-Shwartz, and Ambuj Tewari. Regularization techniques for learning with matrices. *The Journal of Machine Learning Research*, 13(1):1865–1890, 2012.

Satyen Kale. Multiarmed bandits with limited expert advice. In *Conference on Learning Theory*, pp. 107–122. PMLR, 2014.

Varun Kanade and Thomas Steinke. Learning hurdles for sleeping experts. *ACM Transactions on Computation Theory*, 6(3):1–16, 2014.

Yue Kang, Cho-Jui Hsieh, and Thomas Chun Man Lee. Efficient frameworks for generalized low-rank matrix bandit problems. *Advances in Neural Information Processing Systems*, 35:19971–19983, 2022.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 6769–6781, 2020.

Robert Kleinberg, Alexandru Niculescu-Mizil, and Yogeshwer Sharma. Regret bounds for sleeping experts and bandits. *Machine Learning*, 80(2):245–272, 2010.

Fang Kong, XiangCheng Zhang, Baoxiang Wang, and Shuai Li. Improved regret bounds for linear adversarial MDPs via linear optimization. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=KcmWZSk53y.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pp. 3519–3529. PMlR, 2019.

Jean Lafond, Hoi-To Wai, and Eric Moulines. On the online Frank-Wolfe algorithms for convex and non-convex optimizations. *arXiv preprint arXiv:1510.01171*, 2015.

Hyunji Lee, Luca Soldaini, Arman Cohan, Minjoon Seo, and Kyle Lo. RouterRetriever: Routing over a mixture of expert embedding models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 11995–12003, 2025.

Karel Lenc and Andrea Vedaldi. Understanding image representations by measuring their equivariance and equivalence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 991–999, 2015.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. {GS}hard: Scaling giant models with conditional computation and automatic sharding. In *International Conference on Learning Representations*, 2021.

Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. Multi-interest network with dynamic routing for recommendation at Tmall. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 2615–2623, 2019a.

Gene Li, Pritish Kamath, Dylan J Foster, and Nati Srebro. Understanding the eluder dimension. *Advances in Neural Information Processing Systems*, 35:23737–23750, 2022a.

Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, pp. 661–670, 2010.

Yingkai Li, Yining Wang, and Yuan Zhou. Nearly minimax-optimal regret for linearly parameterized bandits. In *Conference on Learning Theory*, pp. 2173–2174. PMLR, 2019b.

Zhiyuan Li, Tianhao Wang, Jason D Lee, and Sanjeev Arora. Implicit bias of gradient descent on reparametrized models: On equivalence to mirror descent. *Advances in Neural Information Processing Systems*, 35:34626–34640, 2022b.

Haolin Liu, Chen-Yu Wei, and Julian Zimmert. Bypassing the simulator: Near-optimal adversarial linear contextual bandits. *Advances in Neural Information Processing Systems*, 36:52086–52131, 2023.

Haolin Liu, Artin Tajdini, Andrew Wagenmaker, and Chen-Yu Wei. Corruption-robust linear bandits: Minimax optimality and gap-dependent misspecification. *Advances in Neural Information Processing Systems*, 37:24277–24325, 2024.

Yangyi Lu, Amirhossein Meisami, and Ambuj Tewari. Low-rank generalized linear bandit problems. In *International Conference on Artificial Intelligence and Statistics*, pp. 460–468. PMLR, 2021.

Haipeng Luo, Chen-Yu Wei, and Chung-Wei Lee. Policy optimization in adversarial MDPs: Improved exploration via dilated bonuses. *Advances in Neural Information Processing Systems*, 34:22931–22942, 2021.

Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H Chi. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1930–1939, 2018.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pp. 9802–9822, 2023.

H Brendan McMahan and Francesco Orabona. Unconstrained online linear learning in Hilbert spaces: Minimax algorithms and normal approximations. In *Conference on Learning Theory*, pp. 1020–1039. PMLR, 2014.

Jincheng Mei, Chenjun Xiao, Csaba Szepesvari, and Dale Schuurmans. On the global convergence rates of softmax policy gradient methods. In *International Conference on Machine Learning*, pp. 6820–6829. PMLR, 2020.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. MTEB: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 2014–2037, 2023.

Gergely Neu and Julia Olkhovskaya. Efficient and robust algorithms for adversarial linear contextual bandits. In *Conference on Learning Theory*, pp. 3049–3068. PMLR, 2020.

Gergely Neu and Julia Olkhovskaya. Online learning in MDPs with linear function approximation and bandit feedback. *Advances in Neural Information Processing Systems*, 34:10407–10417, 2021.

Quang H Nguyen, Thinh Dao, Duy C Hoang, Juliette Decugis, Saurav Manchanda, Nitesh V Chawla, and Khoa D Doan. Metallm: A high-performant and cost-efficient dynamic framework for wrapping llms. *arXiv preprint arXiv:2407.10834*, 2024.

Julia Olkhovskaya, Jack Mayo, Tim van Erven, Gergely Neu, and Chen-Yu Wei. First-and second-order bounds for adversarial linear contextual bandits. *Advances in Neural Information Processing Systems*, 36:61625–61644, 2023.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M Waleed Kadous, and Ion Stoica. RouteLLM: Learning to route LLMs from preference data. In *The Thirteenth International Conference on Learning Representations*, 2025.

Francesco Orabona. Dimension-free exponentiated gradient. *Advances in Neural Information Processing Systems*, 26, 2013.

Francesco Orabona. A modern introduction to online learning. *arXiv preprint arXiv:1912.13213*, 2019.

Francesco Orabona and Dávid Pál. Coin betting and parameter-free online learning. *Advances in Neural Information Processing Systems*, 29, 2016.

Andrea Papenmeier, Dagmar Kern, Daniel Hienert, Alfred Sliwa, Ahmet Aker, and Norbert Fuhr. Dataset of natural language queries for e-commerce. In *Proceedings of the 2021 conference on human information interaction and retrieval*, pp. 307–311, 2021.

Scott Pesme, Loucas Pillaud-Vivien, and Nicolas Flammarion. Implicit bias of sgd for diagonal linear networks: a provable benefit of stochasticity. *Advances in Neural Information Processing Systems*, 34:29218–29230, 2021.

Manhin Poon, XiangXiang Dai, Xutong Liu, Fang Kong, John C.S. Lui, and Jinhang Zuo. Online multi-LLM selection via contextual bandits under unstructured context evolution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, 2026.

Sashank J Reddi, Suvrit Sra, Barnabás Póczos, and Alex Smola. Stochastic Frank-Wolfe methods for nonconvex optimization. In *2016 54th annual Allerton conference on communication, control, and computing (Allerton)*, pp. 1244–1251. IEEE, 2016.

Chandan K. Reddy, Lluís Màrquez, Fran Valero, Nikhil Rao, Hugo Zaragoza, Sambaran Bandyopadhyay, Arnab Biswas, Anlu Xing, and Karthik Subbian. Shopping queries dataset: A large-scale ESCI benchmark for improving product search. 2022.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992, 2019.

Daniel Russo and Benjamin Van Roy. Eluder dimension and the sample complexity of optimistic exploration. *Advances in Neural Information Processing Systems*, 26, 2013.

Yevgeny Seldin, Koby Crammer, and Peter Bartlett. Open problem: Adversarial multiarmed bandits with limited advice. In *Conference on Learning Theory*, pp. 1067–1072. PMLR, 2013.

Yevgeny Seldin, Peter Bartlett, Koby Crammer, and Yasin Abbasi-Yadkori. Prediction with limited advice and multiarmed bandits with paid observations. In *International Conference on Machine Learning*, pp. 280–287. PMLR, 2014.

Lior Shani, Yonathan Efroni, and Shie Mannor. Adaptive trust region policy optimization: Global convergence and faster rates for regularized mdps. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 5668–5675, 2020.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

Uri Sherman, Tomer Koren, and Yishay Mansour. Improved regret for efficient online reinforcement learning with linear function approximation. In *International Conference on Machine Learning*, pp. 31117–31150. PMLR, 2023.

Tal Shnitzer, Anthony Ou, Mírian Silva, Kate Soule, Yuekai Sun, Justin Solomon, Neil Thompson, and Mikhail Yurochkin. Large language model routing with benchmark datasets. *arXiv preprint arXiv:2309.15789*, 2023.

Matthew Streeter and H Brendan McMahan. No-regret algorithms for unconstrained online convex optimization. In *Advances in Neural Information Processing Systems*, volume 25, pp. 2402–2410, 2012.

Richard S. Sutton, David A. McAllester, Satinder P. Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 12, 1999.

Vasilis Syrgkanis, Akshay Krishnamurthy, and Robert Schapire. Efficient algorithms for adversarial contextual learning. In *International Conference on Machine Learning*, pp. 2159–2168. PMLR, 2016.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

Asterios Tsiourvas, Wei Sun, and Georgia Perakis. Causal LLM routing: End-to-end regret minimization from observational data. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Koji Tsuda, Gunnar Rätsch, and Manfred K Warmuth. Matrix exponentiated gradient updates for on-line learning and bregman projection. In *Journal of Machine Learning Research*, pp. 995–1018, 2005.

Tim van Erven, Jack Mayo, Julia Olkhovskaya, and Chen-Yu Wei. An improved algorithm for adversarial linear contextual bandits via reduction. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Aditya Vardhan Varre, Maria-Luiza Vladarean, Loucas Pillaud-Vivien, and Nicolas Flammarion. On the spectral bias of two-layer linear networks. *Advances in Neural Information Processing Systems*, 36:64380–64414, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Daniel Vial, Advait Parulekar, Sanjay Shakkottai, and R Srikant. Improved algorithms for misspecified linear markov decision processes. In *International Conference on Artificial Intelligence and Statistics*, pp. 4723–4746. PMLR, 2022.

John Von Neumann. Some matrix-inequalities and metrization of matric space. *Univ. Rev*, 1:286–300, 1937.

Jizhe Wang, Pipei Huang, Huan Zhao, Zhibo Zhang, Binqiang Zhao, and Dik Lun Lee. Billion-scale commodity embedding for e-commerce recommendation in Alibaba. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 839–848, 2018.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*, 2022.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pp. 9929–9939. PMLR, 2020.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33:5776–5788, 2020.

Wenhui Wang, Hangbo Bao, Shaohan Huang, Li Dong, and Furu Wei. MiniLMv2: Multi-head self-attention relation distillation for compressing pretrained transformers. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 2140–2151, 2021.

Orion Weller, Michael Boratko, Iftekhar Naim, and Jinhyuk Lee. On the theoretical limitations of embedding-based retrieval. In *The Fourteenth International Conference on Learning Representations*, 2026.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3–4):229–256, 1992.

Blake Woodworth, Suriya Gunasekar, Jason D Lee, Edward Moroshko, Pedro Savarese, Itay Golan, Daniel Soudry, and Nathan Srebro. Kernel and rich regimes in overparametrized models. In *Conference on Learning Theory*, pp. 3635–3673. PMLR, 2020.

Fan Wu and Patrick Rebeschini. A continuous-time mirror descent approach to sparse phase retrieval. *Advances in Neural Information Processing Systems*, 33:20192–20203, 2020.

Fan Wu and Patrick Rebeschini. Implicit regularization in matrix sensing via mirror descent. *Advances in Neural Information Processing Systems*, 34:20558–20570, 2021.

Fan Wu and Patrick Rebeschini. Nearly minimax-optimal rates for noisy sparse phase retrieval via early-stopped mirror descent. *Information and Inference: A Journal of the IMA*, 12(2):633–713, 2023.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. C-Pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 641–649, 2024.

Xinyang Yi, Ji Yang, Lichan Hong, Derek Zhiyuan Cheng, Lukasz Heldt, Aditee Kumthekar, Zhe Zhao, Li Wei, and Ed Chi. Sampling-bias-corrected neural modeling for large corpus item recommendations. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pp. 269–277, 2019.

Mingrui Zhang, Lin Chen, Hamed Hassani, and Amin Karbasi. Online continuous submodular maximization: From full-information to bandit feedback. *Advances in Neural Information Processing Systems*, 32, 2019.

Tong Zhao, Yutao Zhu, Yucheng Tian, and Zhicheng Dou. R3AG: Retriever routing for retrieval-augmented generation. *arXiv preprint arXiv:2604.22849*, 2026.

Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew M Dai, Quoc V Le, James Laudon, et al. Mixture-of-experts with expert choice routing. *Advances in Neural Information Processing Systems*, 35:7103–7114, 2022.

Julian Zimmert and Yevgeny Seldin. An optimal algorithm for adversarial bandits with arbitrary delays. In *International Conference on Artificial Intelligence and Statistics*, pp. 3285–3294. PMLR, 2020.

Lingkai Zu, Xiyue Peng, and Xin Liu. Barouter: A budget-adaptive online large language model router framework. In *Proceedings of the ACM Web Conference 2026*, pp. 7645–7656, 2026.

Technical Appendices

A Setup of Figure 1 and More Discussions **19**

B Additional Related Work **20**

C More Discussions on Policy Classes and Regret Notions **21**

C.1 Log-Quadratic Policy Class (Proposition 1) 21

C.2 Constant Policy Class and Expert Regret 22

C.3 Unrestricted Policy Class and Contextual Regret 24

C.4 Log-Linear Policy Class and Linearization in Policy Regret 26

C.5 Log-Quadratic Policy Class and Linearized Policy Regret 27

D Setup of Table 2 and More Discussions **28**

E Omitted Proofs for Hypentropy Policy Gradient **29**

E.1 Matrix Functions and Convex Analysis 29

E.2 Direct Sum Property 31

E.3 Gradient Estimation (Lemma 2) 31

E.4 Linearized Policy Regret Bound of HPG (Theorem 3) 32

F Practical Implementation of HPG **34**

F.1 Computational Cost Bound of HPG (Theorem 4) 34

F.2 Parameter-Free Implementation of HPG (Theorem 5) 35

A Setup of Figure 1 and More Discussions

In Figure 1, we consider a recommendation system scenario over the Amazon Berkeley Objects (ABO) dataset (Collins et al., 2022). Based on the "keywords" in the ABO dataset, we randomly picked 500 items with one of the following keywords: *'sofa', 'couch', 'chair', 'loveseat', 'ottoman.'* We consider the following specific user prompt, which is generated by a Large Language Model (LLM) API, Gemini-2.5-pro, to mimic a typical customer that has a diverse range of requirements: "*Looking for a modern, dark-colored lounge sofa (not a chair). Definitely NOT leather. Prefer fabric or linen with a cozy but structural design.*" It contains various components: vague style descriptions (modern, cozy but structural), color (dark-colored), object category (lounge sofa, not a chair), and material constraints (NOT leather, prefer fabric or linen).

In industrial recommendation systems like Youtube (Covington et al., 2016), Alibaba e-commerce (Wang et al., 2018), and Facebook (Huang et al., 2020), a filter step taking care of hard constraints is implemented independent of the embedding models. We capture this by invoking Gemini-2.5-pro with the following prompt: "*You are a strict data filter for a recommendation system. A user has provided the following search query: {user_prompt} Task: 1. Identify any STRICT negative constraints or hard requirements in the user's query (e.g., if they explicitly say "NOT X", then X is a hard constraint). 2. Evaluate the following items. If an item clearly violates a strict negative constraint from the query, it fails the filter. 3. Return ONLY a valid JSON dictionary where keys are Item IDs and values are boolean 'true' (passes constraints, or no strict constraints violated) or 'false' (fails strict constraints).*" The items satisfying all hard constraints in the user prompt stated above – 131 out of 500 – are passed onto the next stage of recommendation.

For the recommendation stage, we use two lightweight embedding models pre-trained using different methodologies: *(i)* the `miniLM` model by Wang et al. (2020; 2021, `all-MiniLM-L6-v2` on Huggingface), whose embeddings are optimized so that semantically similar sentences have high cosine similarity; and *(ii)* the `msmarco` model by Reimers & Gurevych (2019, `msmarco-distilbert-base-v4` on Huggingface), an asymmetric retrieval model trained on a large-scale dataset built from Bing search queries (Bajaj et al., 2016). Prior benchmarks show that text embedding models are strongly task-dependent: models optimized for semantic similarity and models optimized for query-document retrieval perform differently across downstream retrieval and similarity tasks (Thakur et al., 2021; Muennighoff et al., 2023). Thus, using one symmetric semantic model and one asymmetric retrieval model is expected to produce different rankings over the same candidate items.

To evaluate these two models, we again called the LLM API, Gemini-2.5-pro, with the following prompt: "*You are a recommendation system evaluator. A user queried: {user_prompt} Note: All items below have already passed the hard filter constraints of the query. Evaluate how perfectly each item satisfies the remaining soft constraints (modern, dark-colored, cozy but structural, fabric/linen). Task: Score each item from 0 to 100. Return ONLY a valid JSON dictionary where keys are Item IDs and values are integer scores.*" The resulting ranking is treated as the ground-truth.

In Figure 1, we visualize the rankings using a 2D scatter plot, with the x-axis and y-axis representing the ranks assigned by `miniLM` and `msmarco`, respectively. To show the discrepancy between different models without visual clutter, low-relevance items (ground-truth rank > 40, which is roughly 25%) are plotted as faint crosses, while high-quality items (ground-truth rank ≤ 40) are shown as gray dots. Furthermore, we also highlight representative "exclusively good" items identified as follows: An item ranked in the top tier by both the LLM ground-truth and one embedding model, while simultaneously being penalized by the other embedding model. Green dots are items uniquely captured by `miniLM`, and red dots are items uniquely captured by `msmarco`. This gives our main motivation: No single model dominates the other, and hence dynamic model routing is necessary.

B Additional Related Work

Model Routing. In large-scale industrial recommendation systems or search engines, multi-embedding frameworks were developed for better performance (Li et al., 2019a; Huang et al., 2020). Mixture-of-Experts (MoE) architectures were proposed in the natural language processing literature to aggregate representations and capture semantic intents (Shazeer et al., 2017; Ma et al., 2018; Lepikhin et al., 2021; Fedus et al., 2022; Zhou et al., 2022). In the era of large language models (LLMs), dynamic model routing (also known as model cascading or LLM routing) was proposed to balance the generation quality and the inference cost (Jiang et al., 2023; Shnitzer et al., 2023; Chen et al., 2024; Ong et al., 2025). More closely related to our context of embedding models, modern Retrieval-Augmented Generation (RAG) pipelines increasingly route queries to specialized expert retrievers (Mallen et al., 2023; Jeong et al., 2024; Lee et al., 2025; Zhao et al., 2026). Several prior works also drew heuristics by viewing model routing as (online or offline) contextual bandits (Nguyen et al., 2024; Ong et al., 2025; Hu et al., 2025; Tsiourvas et al., 2025; Jitkrittum et al., 2026; Zu et al., 2026; Poon et al., 2026), but they treated the underlying embedding models as complete black-boxes. On the other hand, in this paper, we model the embedding model routing rigorously as an adversarial contextual linear bandit with low-rank experts, and design a policy gradient algorithm with provable regret guarantee, computational cost bound, and parameter-free implementation.

Bandits with Expert Advice. Bandits with expert advice (Definition 1) were first studied by Auer et al. (2002), who proposed the EXP4 algorithm for $\mathcal{O}(\sqrt{KT\log M})$ expert regret where K, T, M are the numbers of arms, rounds, and experts. Beygelzimer et al. (2011) proposed EXP4.P, achieving the same bound with high probability. Seldin et al. (2013) raised the limited observation setup – where each round only N experts can be queried – as a COLT open problem. With full information, $\widetilde{\mathcal{O}}(\sqrt{MT/N})$ expert regret is possible (Seldin et al., 2014); under bandit feedback, Kale (2014) gave $\widetilde{\mathcal{O}}(\sqrt{\min(N,K)MT/N})$ expert regret and a near-matching lower bound. The unlimited observation setup has recently regained interest, with $\Omega(\sqrt{KT\log(M/K)})$ lower bounds established for learners with different levels of adaptivity (Ito, 2024; Cesa-Bianchi et al., 2025; Chase et al., 2025).

(Contextual) Linear Bandits. Linear bandits were studied by Dani et al. (2008), and the contextual version (Definition 2) was studied by Chu et al. (2011) and Abbasi-Yadkori et al. (2011) under the

adversarial-context stochastic-reward ("A-S") regime. The adversarial-context and adversarial-reward ("A-A") regime is intractable (Kanade & Steinke, 2014; Hazan & Koren, 2016; Neu & Olkhovskaya, 2020), thus Agarwal et al. (2014) and Syrgkanis et al. (2016) opted for policy regret.

The stochastic-context adversarial-reward ("S-A") regime was proposed by Neu & Olkhovskaya (2020) under the known context distribution assumption. Subsequent works (Neu & Olkhovskaya, 2021; Luo et al., 2021; Dai et al., 2023; Sherman et al., 2023; Kong et al., 2024) focused on the more general linear Markov decision process (MDP) setup; when specialized to bandits, they require extra assumptions or give sub-optimal regret. Liu et al. (2023) proposed an efficient $\widetilde{\mathcal{O}}(d^2\sqrt{T})$ algorithm and an inefficient $\widetilde{\mathcal{O}}(d\sqrt{T})$ algorithm, with d being the dimension and T being the number of rounds. Ito (2024) considered a finite-arm case and van Erven et al. (2025) refined the computational cost.

Misspecification in linear bandits was studied by, for example, Gopalan et al. (2016); Ghosh et al. (2017); Foster et al. (2020); Dong & Yang (2023); Liu et al. (2024), which we adopt in Eqs. (1) and (3). Richer non-linear regimes were also studied in the literature (Russo & Van Roy, 2013; Foster et al., 2018; Foster & Rakhlin, 2020; Li et al., 2022a), which are beyond the scope of this paper.

Policy Gradient Methods. Policy gradient (PG) methods are foundational for reinforcement learning and control (Williams, 1992; Sutton et al., 1999; Kakade, 2001). Global convergence of PG methods for the log-linear policy class was initiated by Fazel et al. (2018) in control theory and Bhandari & Russo (2024) in reinforcement learning. Agarwal et al. (2021) and Mei et al. (2020) proposed gradient dominance conditions to establish exact convergence rates in tabular settings, but the log-linear policy class remains open (see Appendix C.4). Policy gradient is also frequently designed from an OMD perspective (Geist et al., 2019; Shani et al., 2020), where various entropy regularizers exist (Alfano et al., 2023; Cichocki et al., 2025). However, we are unaware of existing works utilizing hypentropy in policy gradient to promote low-rankness of the learned parameter.

Parameter-Free Online Learning. Parameter-free online learning attains near-optimal regret without prior information on the comparator (Chaudhuri et al., 2009; Streeter & McMahan, 2012; Orabona, 2013; McMahan & Orabona, 2014). Orabona & Pál (2016) proposed a "coin-betting" framework, which enables black-box frameworks for parameter-free online learning in richer regimes (Cutkosky & Orabona, 2018). More generally, adaptivity to, e.g., optimal action's loss (Freund & Schapire, 1997; Allenberg et al., 2006; Allen-Zhu et al., 2018), feedback delays (Zimmert & Seldin, 2020; Gyorgy & Joulani, 2021; Huang et al., 2023), or properties of loss distributions (Huang et al., 2022; Genalti et al., 2024; Chen et al., 2025), has been widely studied in bandits or online learning.

C More Discussions on Policy Classes and Regret Notions

C.1 Log-Quadratic Policy Class (Proposition 1)

Proof of Proposition 1. We begin by defining the ideal (non-misspecified) recommendation distribution for expert m, which removes the misspecification function μ_m in Eq. (3):

$$\xi_m^*(a \mid q) = \frac{\exp\big(\langle \Psi_m, aq^\mathsf{T}\rangle\big)}{\sum_{a' \in \mathcal{A}} \exp\big(\langle \Psi_m, a'q^\mathsf{T}\rangle\big)}, \quad \forall m \in [M], q \in \mathcal{Q}, a \in \mathcal{A}.$$

Given the bounded misspecification $\|\mu_m\|_\infty \le \epsilon$ in Eq. (3), the ratio between the actual distribution and the ideal distribution is strictly bounded. Specifically, for any $a \in \mathcal{A}$:

$$\begin{aligned}\frac{\xi_m(a \mid q)}{\xi_m^*(a \mid q)} &= \exp\big(\mu_m(q,a)\big) \frac{\sum_{a' \in \mathcal{A}} \exp\big(\langle \Psi_m, a'q^\mathsf{T}\rangle\big)}{\sum_{a' \in \mathcal{A}} \exp\big(\langle \Psi_m, a'q^\mathsf{T}\rangle + \mu_m(q,a')\big)} \\ &\le \exp(\epsilon) \frac{\sum_{a' \in \mathcal{A}} \exp\big(\langle \Psi_m, a'q^\mathsf{T}\rangle\big)}{\exp(-\epsilon) \sum_{a' \in \mathcal{A}} \exp\big(\langle \Psi_m, a'q^\mathsf{T}\rangle\big)} = \exp(2\epsilon).\end{aligned}$$

By symmetry, the lower bound is $\exp(-2\epsilon)$. Consequently, the total variation distance is bounded by

$$D_{\mathrm{TV}}(\xi_m, \xi_m^*) \le \frac{1}{2} \sum_a \xi_m^*(a \mid q) |\exp(2\epsilon) - 1| = \mathcal{O}(\epsilon).$$

We therefore compare the expected action of model m when answering query q, namely $\overline{a}_m(q) := \mathbb{E}_{a \sim \xi_m}[a]$, to that under the ideal ξ_m^*, namely $\overline{a}_m^*(q) := \mathbb{E}_{a \sim \xi_m^*}[a]$. We have

$$\|\overline{a}_m(q) - \overline{a}_m^*(q)\|_2 = \Big\| \sum_{a \in \mathcal{A}} a\big(\xi_m(a \mid q) - \xi_m^*(a \mid q)\big) \Big\|_2 \le \max_{a \in \mathcal{A}} \|a\|_2 \cdot \sum_{a \in \mathcal{A}} |\xi_m - \xi_m^*| \le \mathcal{O}(\epsilon),$$

where we invoked the regularity assumption that $\|a\|_2 \le 1$ (Section 2.2).

Because ξ_m^* belongs to the exponential family, the expected action is the gradient of the log-partition function evaluated at $\Psi_m q$, i.e.,

$$\overline{a}_m^*(q) = \mathop{\mathbb{E}}_{a\sim\xi_m^*}[a] = \nabla A(\Psi_m q), \quad \text{where } A(z) := \log \sum_{a\in\mathcal{A}} \exp(\langle z, a\rangle).$$

Expand $\nabla A(z)$ around $z = 0$ to the second order:

$$\nabla A(z) = \nabla A(0) + \nabla^2 A(0) z + R_2(z),$$

where $\|R_2(z)\|_2 \le \mathcal{O}(\|z\|_2^2)$. Due to the log-partition function, $\nabla A(0) = \frac{1}{|\mathcal{A}|}\sum_{a\in\mathcal{A}} a := \mu_{\mathcal{A}}$ is the mean of all items, and $\nabla^2 A(0) = \frac{1}{|\mathcal{A}|}\sum_{a\in\mathcal{A}}(a-\mu_{\mathcal{A}})(a-\mu_{\mathcal{A}})^\mathsf{T} := \Sigma_{\mathcal{A}}$ is the covariance matrix.

Now, plug in $z = \Psi_m q$. By the regularity conditions that $\|q\|_2 \le 1$ and $\|\Psi_m\|_2 \le \rho$, we have $\|z\|_2 \le \rho$. Thus the remainder is bounded by $\|R_2(z)\|_2 \le \mathcal{O}(\rho^2)$. Combining it with the total variational bound between $\overline{a}_m(q)$ and $\overline{a}_m^*(q)$, we have

$$\overline{a}_m^*(q) = \mu_{\mathcal{A}} + \Sigma_{\mathcal{A}}\Psi_m q + \widetilde{e}_m(q), \quad \overline{a}_m(q) = \mu_{\mathcal{A}} + \Sigma_{\mathcal{A}}\Psi_m q + e_m(q),$$

where $\|\widetilde{e}_m(q)\|_2, \|e_m(q)\|_2 \le \mathcal{O}(\epsilon + \rho^2)$. Substituting $\overline{a}_m(q)$ into the reward function in Eq. (1),

$$\begin{aligned}\mathop{\mathbb{E}}_{a\sim\xi_m(q)}[r(q,a)] &= \mathop{\mathbb{E}}_{a\sim\xi_m(q)}\left[q^\mathsf{T}(\Psi^*)^\mathsf{T} a + \nu(q,a)\right] = q^\mathsf{T}(\Psi^*)^\mathsf{T}\overline{a}_m(q) + \mathop{\mathbb{E}}_{a\sim\xi_m}[\nu(q,a)] \\ &= \underbrace{q^\mathsf{T}(\Psi^*)^\mathsf{T}\mu_{\mathcal{A}}}_{\text{constant } C(q)} + \underbrace{q^\mathsf{T}(\Psi^*)^\mathsf{T}\Sigma_{\mathcal{A}}\Psi_m q}_{\text{quadratic } q^\mathsf{T} W_m q} + \underbrace{q^\mathsf{T}(\Psi^*)^\mathsf{T} e_m(q) + \mathop{\mathbb{E}}_{a\sim\xi_m}[\nu(q,a)]}_{\text{residual } \delta_m(q)}.\end{aligned}$$

We first focus on the quadratic term $q^\mathsf{T} W_m q$ where $W_m := (\Psi^*)^\mathsf{T}\Sigma_{\mathcal{A}}\Psi_m$. Since the quadratic form only depends on the symmetric part, we define $W_m^* = \frac{1}{2}(W_m + W_m^\mathsf{T}) \in \mathbb{S}^{d_q}$. It is also low-rank:

$$\operatorname{rank}(W_m^*) = \operatorname{rank}\left(\frac{1}{2}W_m + \frac{1}{2}W_m^\mathsf{T}\right) \le \operatorname{rank}(W_m) + \operatorname{rank}(W_m^\mathsf{T}) \le 2\operatorname{rank}(\Psi_m) \le 2s_m \le 2s.$$

Consequently, we have the following nuclear-norm bound:

$$\|W_m^*\|_* \le \operatorname{rank}(W_m^*) \times \|W_m^*\|_2 \le 2s \times \frac{1}{2}\left(\|W_m\|_2 + \|W_m^\mathsf{T}\|_2\right) \le 2s \times 1 \times 1 \times \rho = 2s\rho,$$

where the last step uses the regularity conditions that $\|\Psi^*\|_2 \le 1$, $\|a\|_2 \le 1$ for all $a \in \mathcal{A}$, and $\|\Psi_m\|_2 \le \rho$. For the residual term $\delta_m(q)$, because $\|q\|_2 \le 1$, $\|\Psi^*\|_2 \le 1$, and $|\nu| \le \epsilon$,

$$|\delta_m(q)| \le \|q\|_2 \cdot \|\Psi^*\|_2 \cdot \|e_m(q)\|_2 + \sup_{a\in\mathcal{A}} \nu(q,a) \le 1 \cdot 1 \cdot \mathcal{O}(\epsilon + \rho^2) + \epsilon = \mathcal{O}(\epsilon + \rho^2).$$

This finishes the proof: We have $\mathbb{E}_{a\sim\xi_m(q)}[r(q,a)] = C(q) + q^\mathsf{T} W_m^* q + \delta_m(q)$, such that $W_m^* \in \mathbb{S}^{d_q}$ satisfies $\|W_m^*\|_* \le \rho \operatorname{rank}(W_m^*) \le 2s\rho$, and that $|\delta_m(q)| \le \mathcal{O}(\epsilon + \rho^2)$. □

C.2 Constant Policy Class and Expert Regret

When picking the reference policy class as the constant class, the policy regret recovers the expert regret in the problem of bandits with expert advice (BwE). We begin by defining the BwE problem.

Definition 1 (Bandits with Expert Advice; Auer et al. 2002)**.** Consider a T-round K-armed bandit between a learner and an environment. In each round $t \in [T]$, the learner observes M experts' advice $\xi_{t,1}, \xi_{t,2}, \ldots, \xi_{t,M}$, which are probability distributions over $[K]$. The learner chooses one expert m_t and plays according to their advice, i.e., $a_t \sim \xi_{t,m_t}$. The learner suffers an adversarial loss ℓ_{t,a_t}. The learner minimizes *expert regret*, the sub-optimality gap compared to the optimal expert in hindsight:

$$\mathfrak{R}_T^E := \max_{m^*\in[M]} \mathbb{E}\left[\sum_{t=1}^T \left(\ell_{t,a_t} - \mathop{\mathbb{E}}_{a\sim\xi_{t,m^*}}[\ell_{t,a}]\right)\right]. \tag{16}$$

In the more restrictive *limited observability* setup (Seldin et al., 2013; Kale, 2014), there is a parameter $N \le M$. The learner can only pick a size-N subset of experts in each round to observe their advice. Everything else, including the performance metric of expert regret in Eq. (16), remains the same.

In our embedding model routing problem (Section 2), if picking Π as the class of constant policies:

$$\Pi_{\text{const}} := \big\{\pi(m \mid q) = \mathbb{1}[m = m_0], \forall q \in \mathcal{Q} \;\big|\; m_0 \in [M]\big\}, \tag{17}$$

which is parameterized by a single parameter $m_0 \in [M]$, our problem is almost Definition 1 with $N = 1$: Let $K = |\mathcal{A}|$, i.e., each item $a \in \mathcal{A}$ is an arm. View each model as an expert, with the round-t advice $\xi_{t,m}$ defined according to the $\xi_m(\cdot \mid q_t)$ in Eq. (3). The loss of arm a in round t is the $-r(q_t, a)$ in Eq. (1). The reason why we say "almost" is the correlation between expert advice and losses: In the standard bandits-with-expert-advice model (see, e.g., Kale, 2014, Footnote 2), the advice and losses must be independent conditional on the history (i.e., they can both be adaptive to the history, but cannot be correlated with each other). Fortunately, our claims in Section 3 still hold.

On the upper bound side, we ignore the "experts" structure and apply an EXP3 algorithm (Auer et al., 2002): In round t, the learner chooses one between M models, namely m_t, and the loss of model m in round t has a mean $\mathbb{E}_{a\sim\xi_m(q_t)}[r(q_t, a)]$, which is bounded by $[-1, 1]$. Since EXP3 allows adaptive losses, its policy regret w.r.t. Π_{const} is $\mathcal{O}(\sqrt{MT \log M})$ (Auer et al., 2002, Corollary 3.2).

On the lower bound side, we translate the construction of Kale (2014, Theorem 4) into our setting:
Theorem 6 (Regret Lower Bound w.r.t. Π_{const})**.** *There exists a group of instances with $d_a \gg M$, $d_q = Md_a$, and $s = d_a$, such that the policy regret w.r.t. Π_{const} must be $\Omega(\sqrt{MT/\log d_a})$. Moreover, even if we allow the learner to query multiple $N \le M$ models per round, the same bound remains.*

The proof of Theorem 6 is presented at the end of this section. We hence conclude that the constant policy class and the induced expert regret are inappropriate for embedding model routing: A black-box EXP3 ignoring the linear reward structures in Eq. (1) and the low-rank model structures in Eq. (3) attains near-optimal regret. This lower bound holds even with full observability on models (as long as $M \le d_a$, which is typically true since $d_a = 768$ in RouterRetriever; Lee et al. 2025).

We give two more remarks before concluding this section. First, one may ask with full observability, whether the Hedge algorithm (Freund & Schapire, 1997) – attaining $\mathcal{O}(\sqrt{T \log M})$ regret – becomes applicable. The answer is no: while full observability on models is obtained, the reward feedback is still bandit (see Section 2.3). Second, noticing that the s in Theorem 6 is chosen to be the same as d_a, one may ask whether the low-rank bilinear or matrix bandit algorithms (Jun et al., 2019; Jang et al., 2021; Lu et al., 2021; Kang et al., 2022) can be used to derive sharpened regret bounds and bypass the $\Omega(\sqrt{MT/\log d_a})$ lower bound. The answer is again no: the reward kernel Ψ^* in Eq. (1) can be general (especially, having a rank as large as d), and the learner also has no control over the query q_t.

C.2.1 Lower Bound on Policy Regret w.r.t. Constant Policy Class

Proof of Theorem 6. Write $d_q = MK$ and $d_a = K$. Let $\mathcal{A} = \{e_1, e_2, \ldots, e_K\} \in \mathbb{R}^K$ be unit vectors in the item space. For each round $t \in [T]$, the adversary samples $k_{t,1}, k_{t,2}, \ldots, k_{t,M}$ uniformly at random from $[K]$. The query is given as $q_t = \frac{1}{\sqrt{M}}[e_{k_{t,1}}^\mathsf{T}, e_{k_{t,2}}^\mathsf{T}, \ldots, e_{k_{t,M}}^\mathsf{T}]^\mathsf{T} \in \mathbb{R}^{MK}$. Each expert $m \in [M]$ has the following kernel (M matrices in total, and $\boldsymbol{I}_{K\times K}$ appears as the m-th):

$$\Psi_m = \alpha\sqrt{M}[\mathbf{0}_{K\times K} \quad \cdots \quad \mathbf{0}_{K\times K} \quad \boldsymbol{I}_{K\times K} \quad \mathbf{0}_{K\times K} \quad \cdots \quad \mathbf{0}_{K\times K}] \in \mathbb{R}^{K\times MK},$$

where α is a parameter controlling the softmax temperature in Eq. (3). The rank of Ψ_m is K. Setting misspecification $\mu_m \equiv 0$, the model m then with very high probability recommends the item $k_{t,m}$:

$$\xi_m(e_k \mid q_t) = \frac{\exp(\langle \Psi_m, e_k q_t^\mathsf{T}\rangle_F)}{\sum_{k'=1}^K \exp(\langle \Psi_m, e_{k'} q_t^\mathsf{T}\rangle_F)} = \frac{\exp(\alpha \mathbb{1}[k = k_{t,m}])}{(K-1) + \exp(\alpha)}, \quad \forall k \in [K].$$

It only remains to construct the true rewards. Following the construction by Kale (2014, Theorem 4), the environment picks a "good" expert $m^* \in [M]$ uniformly at random before the game, and set

$$\Psi^* = \delta\sqrt{M}[\mathbf{0}_{K\times K} \quad \cdots \quad \mathbf{0}_{K\times K} \quad \boldsymbol{I}_{K\times K} \quad \mathbf{0}_{K\times K} \quad \cdots \quad \mathbf{0}_{K\times K}] \in \mathbb{R}^{K\times MK},$$

where there are M matrices in total and $\boldsymbol{I}_{K\times K}$ appears as the m^*-th, and the $\delta > 0$ is a parameter enabling information theoretic arguments. This Ψ^* ensures $r(q_t, e_k) = \delta \mathbb{1}[k = k_{t,m^*}]$. That is, only the item recommended by the m^*-th expert has a reward of δ, whereas all others have zero reward.

We therefore resemble the hard instance constructed by Kale (2014, Theorem 4) by setting our α and δ according to their ϵ. Kale (2014, Theorem 4) proved that in bandits with expert advice with limited

observability, one suffers an expert regret lower bound of (with notations from Definition 1)

$$\mathfrak{R}_T^E = \Omega\left(\sqrt{\frac{\min(K, N/\log K)}{N} MT}\right).$$

Plugging in our configuration that $N \le M \ll d_a = K$, we come at the conclusion that, even if we allow full observability on models, the policy regret w.r.t. Π_{const} (Eq. (17)) is still at least

$$\mathfrak{R}_T(\Pi_{\text{const}}) = \Omega\left(\sqrt{\frac{N/\log d_a}{N} MT}\right) = \Omega\left(\sqrt{\frac{MT}{\log d_a}}\right). \qquad \square$$

C.3 Unrestricted Policy Class and Contextual Regret

Similar to Appendix C.2, we begin by defining the problem of contextual linear bandits and contextual regret. We use the adversarial formulation by Liu et al. (2023) to ease subsequent discussions.

Definition 2 (Adversarial Contextual Linear Bandit; Liu et al. 2023)**.** Consider a T-round game between a learner and an environment. In round t, a context, or equivalently, an action set $A_t \subseteq \mathbb{R}^d$, is presented to the learner. The learner picks action $a_t \in A_t$. At the same time, the environment (without observing a_t) chooses a loss vector $\ell_t \in \mathbb{R}^d$. The learner incurs and observes a noisy sample of the loss $\langle \ell_t, a_t \rangle$. The learner minimizes the contextual regret, defined as

$$\mathfrak{R}_T^C := \sup_{\pi^*} \mathbb{E}\left[\sum_{t=1}^{T}\left(\langle \ell_t, a_t\rangle - \mathop{\mathbb{E}}_{a\sim\pi^*(A_t)}[\langle \ell_t, a\rangle]\right)\right], \tag{18}$$

where π^* is any policy that maps a subset $A \subseteq \mathbb{R}^d$ to a (possibly randomized) action $a \in A$ therein.

To illustrate the hardness of matching the unrestricted policy class $\triangle^{\mathcal{Q}}$ via reduction to Definition 2, throughout this section, we make the following two assumptions as mentioned in Section 3:

A1. There is no misspecification in rewards or recommendation distributions so that everything is perfectly linear, i.e., $\nu(q, a) = \mu_m(q, a) = 0$ for all $q \in \mathcal{Q}$ and $a \in \mathcal{A}$ in Eqs. (1) and (3).

A2. The low-rank recommendation kernels of all models, namely the Ψ_m's in Eq. (3), are also known to the learner in advance. This allows the learner to calculate the following *expected recommendation*:

$$\overline{a}_m(q_t) := \mathop{\mathbb{E}}_{a\sim\xi_m(q_t)}[a] \in \mathbb{R}^{d_a}, \quad \forall m \in [M],$$

which means this is an even stronger assumption than full observability on models (Theorem 6).

Since these two assumptions strictly ease learning, but matching the unrestricted policy class remains notoriously hard as we detail in subsequent sections, this metric is only harder in the original setup.

C.3.1 Unrestricted Policy Class → A-S Contextual Linear Bandit

The first reduction, as sketched in Section 3 in the main text, is to the adversarial-context stochastic-loss ("A-S") contextual linear bandit problem. Specifically, take the d in Definition 2 as $d = d_a d_q$. The round-t action set (or context) is given as

$$A_t = \left\{\text{vec}(\overline{a}_m(q_t)\, q_t^\mathsf{T}) \;\middle|\; m \in [M]\right\} \subseteq \mathbb{R}^d = \mathbb{R}^{d_a d_q},$$

where the vec$(\cdot)$ means flattening a $\mathbb{R}^{m\times n}$ matrix into a $\mathbb{R}^{mn}$ vector: $A_{i,j} = \text{vec}(A)_{(i-1)\times n + j}$. Note that, assumption (A2) is used to ensure that the learner can calculate the action set A_t in each round t.

When choosing action $a = \text{vec}(\overline{a}_m(q_t)\, q_t^\mathsf{T}) \in A_t$ in round t, which is equivalent to choosing model m in round t, the expected loss (i.e., negative-reward) is

$$-\mathop{\mathbb{E}}_{a\sim\xi_m(q_t)}[r(q_t, a)] = -\left\langle \Psi^*, \mathop{\mathbb{E}}_{a\sim\xi_m(q_t)}[a]\, q_t^\mathsf{T}\right\rangle_F = -\left\langle \text{vec}(\Psi^*), \text{vec}(\overline{a}_m(q_t)\, q_t^\mathsf{T})\right\rangle, \tag{19}$$

where the first step uses the definition of $r(q, a)$ in Eq. (1) and also assumption (A1) to ensure the linearity, and the second step uses the property that for two matrices $A, B \in \mathbb{R}^{m\times n}$,

$$\langle A, B\rangle_F := \text{Tr}(A^\mathsf{T} B) = \sum_{i=1}^{m}\sum_{j=1}^{n} A_{i,j} B_{i,j} = \langle \text{vec}(A), \text{vec}(B)\rangle.$$

Therefore, the loss vector in Definition 2 is always chosen as $\ell_t = -\text{vec}(\Psi^*)$. This consequently reduces to an adversarial-context (because A_t's depend on q_t and are thus adversarial) and stochastic-loss (because ℓ_t is stationary) contextual linear bandit problem. We list a few available algorithms:

- OFUL (Abbasi-Yadkori et al., 2011): $\mathcal{O}(d\sqrt{T}\log T)$ contextual regret with $\mathcal{O}(|A_t|d^2)$ computation per round. A faster "rarely switching" implementation is available, but the regret is the same.
- SupLinUCB (Chu et al., 2011): $\mathcal{O}\left(\sqrt{dT\log^3(|A|T)}\right)$ contextual regret (where $|A| := \max_t |A_t|$; in our case we thus have $|A| = M$) with $\mathcal{O}(|A|d^2\log T)$ computation per round.
- VCL-SupLinUCB (Li et al., 2019b): $\mathcal{O}(\sqrt{dT\log T\log|A|})$ contextual regret with $\mathcal{O}(|A|d^2\log T)$ computation per round.

As we argued in Section 3, when $d = d_a d_q$ and $d_a = d_q = 768$ (Lee et al., 2025; Izacard et al., 2022), none of these regret bounds or computational cost guarantees are acceptable.

C.3.2 Unrestricted Policy Class → A-A Contextual Linear Bandit

A seemingly more promising reduction is rewriting Eq. (19) as a d_a-dimensional inner product:

$$-\mathop{\mathbb{E}}_{a\sim\xi_m(q_t)}[r(q_t,a)] = -\Big\langle \Psi^*, \mathop{\mathbb{E}}_{a\sim\xi_m(q_t)}[a]\, q_t^\mathsf{T}\Big\rangle_F = -\langle \Psi^* q_t, \overline{a}_m(q_t)\rangle,$$

which corresponds to Definition 2 with round-t action set $A_t = \big\{\overline{a}_m(q_t) \;\big|\; m\in[M]\big\} \subseteq \mathbb{R}^{d_a}$ and loss vector $\ell_t = -\Psi^* q_t \in \mathbb{R}^{d_a}$. While reducing the problem dimension greatly, the adversarial q_t – appearing in both action set $\mathcal{A}_t$ and loss vector ℓ_t – makes this problem intractable: Adversarial-context adversarial-loss (so-called "A-A") contextual linear bandits are computationally infeasible (Kanade & Steinke, 2014), admit $\Omega(|\triangle^{\mathcal{Q}}|)$ oracle-complexity lower bounds (Hazan & Koren, 2016), and are conjectured to bear $\Omega(T)$ regret lower bounds (Neu & Olkhovskaya, 2020).

C.3.3 Unrestricted Policy Class with i.i.d. Query → S-A Contextual Linear Bandit

Given the intractability comes from the fact that both the action set A_t and the loss vector ℓ_t are adversarial, what if we assume queries are i.i.d. in addition to the assumptions (A1) and (A2)?

A3. Assume that all queries q_t are i.i.d. samples from a fixed but unknown distribution $\mathbb{Q} \in \triangle(\mathcal{Q})$.

This assumption is reasonable in, e.g., large-scale e-commerce systems, where it is acceptable to expect that customers arrive uniformly at random from a large population. Now the action set $A_t = \big\{\overline{a}_m(q_t) \;\big|\; m\in[M]\big\}$ is i.i.d., and only the loss vector $\ell_t = -\Psi^* q_t$ is time-varying, does it fall into the tractable category of stochastic-context adversarial-loss ("S-A") contextual linear bandits? (While ℓ_t is also stochastic, a stochastic-loss setup requires a stationary ℓ_t.) This is because S-A contextual linear bandits admit an efficient algorithm with $\widetilde{\mathcal{O}}(d^2\sqrt{T})$ contextual regret (Liu et al., 2023). In our case where action sets have bounded sizes, $\mathcal{O}(\sqrt{dT\log|A|})$ is also possible (Ito, 2024).

However, the answer is negative. This is because all "S-A" contextual linear bandit algorithms require the "ghost sample" technique proposed by Neu & Olkhovskaya (2020). Roughly speaking, via the independence between losses and contexts, this technique allows one to analyze the contextual regret by "fixing" a single $q_0 \in \mathbb{Q}$ throughout the game. However, our q_t appears both in $A_t = \{\overline{a}_m(q_t)\}_m$ and in $\ell_t = -\Psi^* q_t$, thus inducing correlation. All existing algorithms are hence inapplicable (Neu & Olkhovskaya, 2020; Olkhovskaya et al., 2023; Liu et al., 2023; Ito, 2024; van Erven et al., 2025).

C.3.4 Unrestricted Policy Class → Lifted-Feature Linear Bandit

The reductions above all take $\overline{a}_m(q_t)$ as the feature, which is where assumption (A2) is needed. Lifting the *query* instead avoids it: Proposition 1 gives

$$\mathop{\mathbb{E}}_{a\sim\xi_m(q_t)}[r(q_t,a)] = C(q_t) + \big\langle W_m^*, q_t q_t^\mathsf{T}\big\rangle_F + \delta_m(q_t), \quad \forall m\in[M],$$

which corresponds to Definition 2 with $d = Md_q^2$, unknown $\mathrm{diag}(W_1^*, W_2^*, \ldots, W_M^*)$, and M actions placing $q_t q_t^\mathsf{T}$ in the m-th block (the model-independent $C(q_t)$ cancels across actions). SupLinUCB (Chu et al., 2011) or VCL-SupLinUCB (Li et al., 2019b) thus attain $\mathcal{O}(\sqrt{Md_q^2 T\log T\log M}) = \widetilde{\mathcal{O}}(d_q\sqrt{MT})$ contextual regret, up to the $\mathcal{O}((\epsilon+\rho^2)T)$ contributed by the residuals δ_m.

This is worse than the $\mathcal{O}(\sqrt{d_a d_q T\log M\log T})$ of Appendix C.3.1 whenever d_a and d_q are of the same order, so the curse of dimensionality persists. Nor is it removed by the low-rankness of W_m^*: as in Appendix C.2, the rank-dependent guarantees of low-rank bilinear and matrix bandits (Jun et al.,

2019; Jang et al., 2021; Lu et al., 2021; Kang et al., 2022) require *learner-chosen* matrix actions, whereas $q_t q_t^\mathsf{T}$ is decided by the adversary. Rank-dependent contextual regret hence remains out of reach, which motivates the log-quadratic policy class and HPG.

To conclude, matching the unrestricted policy class $\triangle^{\mathcal{Q}}$ or minimizing the contextual regret $\mathfrak{R}_T^C$ is desirable for embedding model routing. However, it is intractable – even after making the unrealistic assumptions of knowing model kernels and no misspecification – due to the extreme flexibility of $\pi^* \in \triangle^{\mathcal{Q}}$, which can impose an *arbitrary* decision boundary when routing $\mathcal{Q}$ to $\triangle$.

C.4 Log-Linear Policy Class and Linearization in Policy Regret

The log-linear policy class is formally defined as follows:

$$\Pi_{\text{lin}} := \left\{\pi(m \mid q) \propto \exp(\theta_m^\mathsf{T} q), \forall q \in \mathcal{Q} \;\middle|\; \theta_m \in \mathbb{R}^{d_q}, \forall m \in [M]\right\}. \tag{20}$$

It assumes the routing policy is parameterized by M d_q-dimensional vectors, $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_M) \in (\mathbb{R}^{d_q})^M$. We call $\Theta := (\mathbb{R}^{d_q})^M$ the parameter space. The policy induced by parameter $\boldsymbol{\theta} \in \Theta$ is thus

$$\pi_{\boldsymbol{\theta}}(m \mid q) = \frac{\exp(\theta_m^\mathsf{T} q)}{\sum_{m'=1}^M \exp(\theta_{m'}^\mathsf{T} q)}, \quad \forall m \in [M], q \in \mathcal{Q}.$$

Equivalently, we can view $\boldsymbol{\theta}$ as a concatenated Md_q-dimensional vector and let

$$\phi(q, m) = [0^\mathsf{T} \quad \cdots \quad 0^\mathsf{T} \quad q^\mathsf{T} \quad 0^\mathsf{T} \quad \cdots 0^\mathsf{T}]$$

be the concatenation of M d_q-dimensional vectors with the m-th one being q and all remaining ones being 0. Thus $\theta_m^\mathsf{T} q = \boldsymbol{\theta}^\mathsf{T} \phi(q, m)$, recovering the log-linear class by Agarwal et al. (2021, §6.1.1).

As discussed by Agarwal et al. (2021), Π_{lin} is much harder than the *tabular* softmax policy class Π_{tab}, where each policy $\pi(m \mid q) \propto \exp(\theta'_{q,m})$ for some $\boldsymbol{\theta}' \in \Theta' := \mathbb{R}^{M|\mathcal{Q}|}$. Indeed, for Π_{tab}, it is possible to exactly handle the policy regret (Agarwal et al., 2021, Theorems 5.1 and 5.4). However, for the log-linear policy class Π_{lin}, where the policy $\pi(m \mid q) \propto \exp(\boldsymbol{\theta}^\mathsf{T} \phi(q, m))$ for some $\boldsymbol{\theta} \in \Theta := \mathbb{R}^{Md_q}$, Agarwal et al. (2021) instead considered a *linearized surrogate regret notion* instead of the policy regret. We now introduce their notations and results in more details. Since they focus on stochastic MDPs, we also assume i.i.d. queries (that is, $q_t \sim \mathbb{Q}$ with some fixed but unknown $\mathbb{Q}$) in this section.

For any parameter $\boldsymbol{\theta} \in \Theta$, let $\overline{\phi}_{\boldsymbol{\theta}}(q, m) = \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(m \mid q)$. Now consider a round $t \in [T]$ with parameter $\boldsymbol{\theta}_t \in \Theta$. For each model $m \in [M]$, define the advantage under $\boldsymbol{\theta}_t$ as

$$A_{\boldsymbol{\theta}_t}(q, m) = \mathop{\mathbb{E}}_{a \sim \xi_m(q)}[r(q, a)] - \sum_{m'=1}^M \pi_{\boldsymbol{\theta}_t}(m' \mid q) \mathop{\mathbb{E}}_{a \sim \xi_{m'}(q)}[r(q, a)].$$

The natural policy gradient (NPG) step in round t is then given as (Agarwal et al., 2021, §6.1.1)

$$\boldsymbol{w}_t := \mathop{\mathrm{argmin}}_{\boldsymbol{w} \in \Theta} \mathop{\mathbb{E}}_{q \sim \mathbb{Q}, m \sim \pi_{\boldsymbol{\theta}_t}(\cdot|q)} \left[\left(A_{\boldsymbol{\theta}_t}(q, m) - \boldsymbol{w}^\mathsf{T} \overline{\phi}_{\boldsymbol{\theta}_t}(q, m)\right)^2\right]. \tag{21}$$

Agarwal et al. (2021, Assumption 6.1 and Theorem 6.1) defined the following *transfer error* ϵ_{bias}, and had a final bound scaling with $\sqrt{\epsilon_{\text{bias}}} \times T$:[8]

$$\epsilon_{\text{bias}} := \max_{t \in [T]} \mathop{\mathbb{E}}_{q \sim \mathbb{Q}, m \sim \pi_{\boldsymbol{\theta}^*}(\cdot|q)} \left[\left(A_{\boldsymbol{\theta}_t}(q, m) - \boldsymbol{w}_t^\mathsf{T} \overline{\phi}_{\boldsymbol{\theta}_t}(q, m)\right)^2\right].$$

Let the expected reward – when taking $q \sim \mathbb{Q}$ into consideration – of policy $\pi_{\boldsymbol{\theta}_t}$ be $J(\boldsymbol{\theta}_t)$, defined as

$$J(\boldsymbol{\theta}_t) := \mathop{\mathbb{E}}_{q \sim \mathbb{Q}} \left[\mathop{\mathbb{E}}_{m \sim \pi_{\boldsymbol{\theta}_t}(q)} \left[\mathop{\mathbb{E}}_{a \sim \xi_m(q)}[r(q, a)]\right]\right].$$

[8] The bound of Agarwal et al. (2021) is stated w.r.t. single-round sub-optimality gaps, hence a linear dependency on T arises when converting to our cumulative regret setup. As an additional remark, their γ is the discount factor that only applies to multi-stage MDPs; in our bandit setup, we have $\gamma = 0$.

By direct arithmetic calculation (which is the famous Performance Difference Lemma specialized to bandits; Kakade & Langford 2002), the one-step true policy regret in $\mathfrak{R}_T(\Pi_{\text{lin}})$ decomposes as

$$J(\boldsymbol{\theta}^*) - J(\boldsymbol{\theta}_t) = \mathop{\mathbb{E}}_{q\sim\mathbb{Q}}\left[\sum_{m=1}^{M} \pi_{\boldsymbol{\theta}^*}(m \mid q) A_{\boldsymbol{\theta}_t}(q,m)\right]$$
$$= \mathop{\mathbb{E}}_{q\sim\mathbb{Q}}\left[\sum_{m=1}^{M} \pi_{\boldsymbol{\theta}^*}(m \mid q) \boldsymbol{w}_t^\mathsf{T} \overline{\phi}_{\boldsymbol{\theta}_t}(q,m)\right] + \mathop{\mathbb{E}}_{q\sim\mathbb{Q}}\left[\sum_{m=1}^{M} \pi_{\boldsymbol{\theta}^*}(m \mid q) \left(A_{\boldsymbol{\theta}_t}(q,m) - \boldsymbol{w}_t^\mathsf{T} \overline{\phi}_{\boldsymbol{\theta}_t}(q,m)\right)\right].$$

Instead of tackling $J(\boldsymbol{\theta}^*) - J(\boldsymbol{\theta}_t)$ directly, Agarwal et al. (2021) controlled the second term via Cauchy-Schwartz as $\sqrt{\epsilon_{\text{bias}}}$ and instead studied the first term. For log-linear policy class, we have $\overline{\phi}_{\boldsymbol{\theta}_t}(q,m) = \phi(q,m) - \mathbb{E}_{m'\sim\pi_{\boldsymbol{\theta}_t}(\cdot|q)}[\phi(q,m')]$ (Agarwal et al., 2021, §6.1.1). Thus, we know

$$\boldsymbol{w}_t^\mathsf{T} \overline{\phi}_{\boldsymbol{\theta}_t}(q,m) = w_{t,m}^\mathsf{T} q - \sum_{m'=1}^{M} \pi_{\boldsymbol{\theta}_t}(m' \mid q) w_{t,m'}^\mathsf{T} q.$$

Plugging this back into the first term and using the fact that $\sum_{m=1}^{M} \boldsymbol{\theta}^*(m \mid q) = 1$, we obtain:

$$\mathop{\mathbb{E}}_{q\sim\mathbb{Q}}\left[\sum_{m=1}^{M} \pi_{\boldsymbol{\theta}^*}(m \mid q) \boldsymbol{w}_t^\mathsf{T} \overline{\phi}_{\boldsymbol{\theta}_t}(q,m)\right] = \mathop{\mathbb{E}}_{q\sim\mathbb{Q}}\left[\sum_{m=1}^{M} \pi_{\boldsymbol{\theta}^*}(m \mid q) w_{t,m}^\mathsf{T} q - \sum_{m'=1}^{M} \pi_{\boldsymbol{\theta}_t}(m' \mid q) w_{t,m'}^\mathsf{T} q\right]$$
$$= \mathop{\mathbb{E}}_{q\sim\mathbb{Q}}\left[\sum_{m=1}^{M} \left(\pi_{\boldsymbol{\theta}^*}(m \mid q) - \pi_{\boldsymbol{\theta}_t}(m \mid q)\right) w_{t,m}^\mathsf{T} q\right],$$

Agarwal et al. (2021, Theorem 6.1 and Corollary 6.1) then proved that NPG attains $\sqrt{T}$-style regret under this *linearized surrogate regret*, but suffering a linear $\sqrt{\epsilon_{\text{bias}}} \times T$ overhead in the true policy regret. Their linearization, however, happens in a different space from our $\widetilde{\mathfrak{R}}_T(\Pi)$ in Eq. (9): theirs in the action space (the probability simplex), ours in the parameter space (the $\boldsymbol{W} \in \mathbb{W}$ of Section 3.2). It remains open whether these two linearized surrogate regret notions for the policy regret can be transformed to each other.

C.5 Log-Quadratic Policy Class and Linearized Policy Regret

Proposition 7 (Linearized Policy Regret vs Policy Regret)**.** *Consider the log-quadratic policy class Π_{quad} parameterized by $\boldsymbol{W} \in \mathbb{W}$; see Section 3.2. For each round $t \in [T]$ and model $m \in [M]$, let $\pi_{t,m} := \pi_{\boldsymbol{W}_t}(m \mid q_t)$, $\pi^*_{t,m} := \pi_{\boldsymbol{W}^*}(m \mid q_t)$, and $r_{t,m} := \mathbb{E}_{a\sim\xi_m(q_t)}[r(q_t,a)]$. Then the one-step regret in the policy regret w.r.t. Π_{quad} is (where $\mathcal{L}_t$ is defined in Eq. (8))*

$$\mathcal{L}_t(\boldsymbol{W}_t) - \mathcal{L}_t(\boldsymbol{W}^*) = \sum_{m=1}^{M} \pi^*_{t,m} \left(r_{t,m} - \sum_{m'=1}^{M} \pi_{t,m'} r_{t,m'}\right),$$

while that in the linearized policy regret $\widetilde{\mathfrak{R}}_T(\Pi_{\text{quad}})$ (defined in Eq. (9)) is

$$\left\langle \nabla_{\boldsymbol{W}} \mathcal{L}_t(\boldsymbol{W}_t), \boldsymbol{W}_t - \boldsymbol{W}^* \right\rangle_F = \sum_{m=1}^{M} \pi_{t,m} \left(r_{t,m} - \sum_{m'=1}^{M} \pi_{t,m'} r_{t,m'}\right) \log \frac{\pi^*_{t,m}}{\pi_{t,m}}.$$

Given the very similar forms in Proposition 7, it is tempting to claim that these two regret notions are multiplicatively close to each other via a *gradient dominance condition*; this unfortunately fails. The gradient dominance condition is given as follows.[9] Due to the projection onto the product nuclear-norm ball $\mathbb{B}_*^M(\tau)$, the maximum and minimum possible coefficient of a model m under any $W_m \in \mathbb{B}_*(\tau)$ are similar, hence giving $\max_m \frac{\pi^*_m}{\pi_m} \le \exp(2\tau)$. But the other term, $\log \frac{\pi^*_m}{\pi_m}$, is problematic: both it and the $(r_m - \sum_{m'=1}^{M} \pi_{m'} r_{m'})$ term (also known as the *advantage function* in reinforcement learning) can be negative, hence the true policy regret is *not* automatically bounded by $\exp(2\tau)$ times the linearized policy regret. We leave the exact policy regret for future research.

[9] This type of condition, also known as Łojasiewicz condition, has been developed by Agarwal et al. (2021) and Mei et al. (2020) to exactly control the policy regret of NPG w.r.t. the tabular softmax class $\Pi_{\text{tab}} := \{\pi(m \mid q) \propto \exp(\theta'_{q,m}) \mid \boldsymbol{\theta}' \in \mathbb{R}^{M|\mathcal{Q}|}\}$; see more in Appendix C.4. We also remark their analysis only works with i.i.d. queries.

Proof of Proposition 7. For notational simplicity, we focus on a single round $t \in [T]$ and omit all the subscript t's. By definition of $\mathcal{L}_t$ in Eq. (8), the one-step regret in $\mathfrak{R}_T(\Pi_{\text{quad}})$ (Eq. (4)) is

$$\begin{aligned}
\mathcal{L}_t(\boldsymbol{W}_t) - \mathcal{L}_t(\boldsymbol{W}^*) &= \sum_{m=1}^{M} (\pi^*_m - \pi_m) r_m \overset{(a)}{=} \sum_{m=1}^{M} (\pi^*_m - \pi_m) r_m - \sum_{m=1}^{M} (\pi^*_m - \pi_m) \left(\sum_{m'=1}^{M} \pi_{m'} r_{m'} \right) \\
&\overset{(b)}{=} \sum_{m=1}^{M} \pi^*_m \left(r_m - \sum_{m'=1}^{M} \pi_{m'} r_{m'} \right) - \sum_{m=1}^{M} \pi_m r_m + \sum_{m'=1}^{M} \pi_{m'} r_{m'} \\
&= \sum_{m=1}^{M} \pi^*_m \left(r_m - \sum_{m'=1}^{M} \pi_{m'} r_{m'} \right).
\end{aligned}$$

where (a) uses the fact that $\sum_m (\pi^*_m - \pi_m) C = (1-1)C = 0$ for any C and (b) uses $\sum_m \pi_m = 1$.

For the linearized policy regret $\widetilde{\mathfrak{R}}_T(\Pi_{\text{quad}})$ in Eq. (9), we have

$$\begin{aligned}
\left\langle \nabla_{\boldsymbol{W}} \mathcal{L}_t(\boldsymbol{W}_t), \boldsymbol{W}_t - \boldsymbol{W}^* \right\rangle_F &\overset{(a)}{=} \sum_{m=1}^{M} \left\langle - \sum_{m'=1}^{M} \pi_{m'} r_{m'} \big(\mathbb{1}[m' = m] - \pi_m \big) q_t q_t^\mathsf{T}, W_{t,m} - W^*_m \right\rangle_F \\
&= \sum_{m=1}^{M} \left(\pi_m r_m - \pi_m \sum_{m'=1}^{M} \pi_{m'} r_{m'} \right) q_t^\mathsf{T} \left(W^*_m - W_{t,m} \right) q_t \\
&\overset{(b)}{=} \sum_{m=1}^{M} \pi_m \left(r_m - \sum_{m'=1}^{M} \pi_{m'} r_{m'} \right) \log \frac{\pi^*_m}{\pi_m},
\end{aligned}$$

where (a) uses the direct sum property in Lemma 12 (proved in Appendix E.2) and $\nabla \mathcal{L}_t(\boldsymbol{W})$ derived in Eq. (26) (rewritten using the π_m and r_m notations), and (b) uses the definition of $\pi_{\boldsymbol{W}}$:

$$\begin{aligned}
\log \frac{\pi_{\boldsymbol{W}^*}(m \mid q_t)}{\pi_{\boldsymbol{W}_t}(m \mid q_t)} &= \log \left(\frac{\exp(q_t^\mathsf{T} W^*_m q_t)}{\sum_{m'=1}^{M} \exp(q_t^\mathsf{T} W^*_{m'} q_t)} \right) \Bigg/ \left(\frac{\exp(q_t^\mathsf{T} W_{t,m} q_t)}{\sum_{m'=1}^{M} \exp(q_t^\mathsf{T} W_{t,m'} q_t)} \right) \\
= \log \frac{\exp(q_t^\mathsf{T} W^*_m q_t)}{\exp(q_t^\mathsf{T} W_{t,m} q_t)} + \log \frac{\sum_{m'=1}^{M} \exp(q_t^\mathsf{T} W^*_{m'} q_t)}{\sum_{m'=1}^{M} \exp(q_t^\mathsf{T} W_{t,m'} q_t)} &= q_t^\mathsf{T} (W^*_m - W_{t,m}) q_t + C,
\end{aligned}$$

where C is (another) constant independent of m that automatically vanishes because

$$C \sum_{m=1}^{M} \pi_m \left(r_m - \sum_{m'=1}^{M} \pi_{m'} r_{m'} \right) = C \sum_{m=1}^{M} \pi_m r_m - C \left(\sum_{m=1}^{M} \pi_m \right) \sum_{m'=1}^{M} \pi_{m'} r_{m'} = 0. \qquad \square$$

D Setup of Table 2 and More Discussions

As mentioned in the main text, we first encode every natural language query and item into 768-dimensional vector spaces via CONTRIEVER (Izacard et al., 2022). For each embedded query $q \in \mathbb{R}^{768}$ and associated item $a \in \mathbb{R}^{768}$, the ESCI dataset labels it with one of four relevance scores: E (Exact), S (Substitute), C (Complement), or I (Irrelevant). We convert the four ordinal ESCI labels into numerical rewards as $E = 0.95$, $S = 0.70$, $C = 0.30$, and $I = 0.05$ (i.e., the reward function $r(q, a)$ in Eq. (1)). The specific values are a modeling choice that preserves the ordering of the labels and treats non-exact but related products (i.e., categories S and C) as partially useful.

We consider $M = 8$ embedding models as candidate experts, spanning three distinct methodologies: *(i)* symmetric semantic encoders capturing similarities (`all-MiniLM-L6`, `-L12`, and `paraphrase-MiniLM`; Wang et al. 2020; 2021); *(ii)* asymmetric search models trained on Q&A datasets (`multi-qa-MiniLM`, `msmarco-MiniLM`, and `msmarco-distilbert`; Reimers & Gurevych 2019; Bajaj et al. 2016); and *(iii)* modern prompt-driven encoders dynamically adapting to different tasks (`e5-small-v2` and `bge-small-en-v1.5`; Wang et al. 2022; Xiao et al. 2024).

The recommendation distribution of each model $m \in [M]$ on a query $q \in \mathcal{Q}$, namely the $\xi_m(q)$ in Eq. (3) but instead supported on the candidate items corresponding to this query, is the softmax over the

query-item similarity scores induced by this model. We then calculate the model's expected reward $R_m(q) = \mathbb{E}_{a\sim\xi_m(q)}[r(q,a)]$ and find the optimal model routing policies within each class:[10]

- $\pi^*_{\text{const}} \in \Pi_{\text{const}}$ maps each query to the best fixed model $m^* = \operatorname{argmax}_m \mathbb{E}_q[R_m(q)]$, where the expectation is taken over the uniform distribution over all queries in the ESCI dataset;
- $\pi^*_{\text{lin}} \in \Pi_{\text{lin}}$ is the best log-linear routing policy, i.e., in the form of $\pi(m \mid q) \propto \exp(\theta_m^\mathsf{T} q)$ for some $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_M) \in (\mathbb{R}^{768})^M$;
- $\pi^*_{\text{quad}} \in \Pi_{\text{quad}}$ is the best log-quadratic routing policy, i.e., in the form of $\pi(m \mid q) \propto \exp(q^\mathsf{T} W_m q)$ for some $\boldsymbol{W} = (W_1, W_2, \ldots, W_M) \in (\mathbb{R}^{768\times 768})^M$; and
- $\pi^*\colon \mathcal{Q} \to \triangle$ is the best unrestricted policy, mapping any query q to the embedding model (out of $M = 8$) that performs the best on this specific query q. This, by definition, is the best model routing policy that one can possibly hope for.

In Table 2, we report the expected reward gap of each policy compared to π^*, that is,

$$\text{Gap}(\pi) := \mathbb{E}_q\Big[\mathbb{E}_{m\sim\pi^*(q)}\big[R_m(q)\big] - \mathbb{E}_{m\sim\pi(q)}\big[R_m(q)\big]\Big], \quad \forall \pi \in \{\pi^*_{\text{const}}, \pi^*_{\text{lin}}, \pi^*_{\text{quad}}\}, \tag{22}$$

where the outer expectation is again taken over the uniform distribution over all queries in ESCI. We also report the relative improvement of π^*_{lin} over π^*_{const} and that of π^*_{quad} over π^*_{lin}. We see that $\text{Gap}(\pi^*_{\text{const}}) = 0.035$, $\text{Gap}(\pi^*_{\text{lin}}) = 0.030$, and $\text{Gap}(\pi^*_{\text{quad}}) = 0.011$ (which, are all pretty small because the expert embedding models we chose are quite powerful). We therefore observe that the improvement of π^*_{lin} over π^*_{const} is incremental (13.8%), whereas our proposed log-quadratic policy π^*_{quad} yields a 63.5% improvement over π^*_{lin} (and, 68.5% over π^*_{const}). This indicates that our analysis Proposition 1 is robust to the two discrepancies arising in the realistic ESCI dataset: *(i)* the reward function $r(q,a)$ is not generated according to the bilinear structure in Eq. (1), and *(ii)* the misspecification in recommendation model distributions – in Eq. (3) – can be large.

The ESCI dataset additionally includes several categories of challenging queries, where different embedding models are expected to exhibit different failure modes, making model routing more important. For example, `nlqec` consists of long natural-language search queries (Papenmeier et al., 2021). On this category, the best constant policy incurs an expected sub-optimality gap of 0.110 (where the expectation is taken uniformly over all `nlqec` queries), larger than the 0.035 gap over the full ESCI dataset; hence this category is indeed harder for any fixed expert model. The log-linear class reduces the gap to 0.004, closing 96.0% of the constant-policy gap, while our log-quadratic class nearly closes the gap entirely by another 99.7% improvement.

Another category is `negations`, which contains negation words such as "without." This category is not only hard for constant policies, but also hard for log-linear policies: the best $\pi^{\texttt{negations}}_{\text{lin}} \in \Pi_{\text{lin}}$ still incurs a sub-optimality gap of 0.054, whereas our log-quadratic policy reduces it to 0.011, an 80.1% reduction relative to the log-linear gap. Similar results hold for `behavioral`, where such queries' resulting clicks or purchases have non-representative distributions, and `parse-pattern`, where queries exhibit linguistic complexity such as quantities or multiple modifiers.

E Omitted Proofs for Hypentropy Policy Gradient

E.1 Matrix Functions and Convex Analysis

We first state a few properties of matrix functions and convex analysis without proof, mainly from Kakade et al. (2012). Since our matrices are always symmetric, we restrict our definitions to the symmetric matrix space $\mathbb{S}^n$, though most of them can be seamlessly extended to rectangular matrices.

Convex Analysis. We view $\mathbb{S}^n$ as a n^2-dimensional vector space equipped with the inner product:

$$\langle X, Y\rangle_F = \operatorname{Tr}(X^\mathsf{T} Y) = \sum_{i=1}^n \sum_{j=1}^n X_{i,j} Y_{i,j}, \quad \forall X, Y \in \mathbb{S}^n.$$

Over a convex domain $\mathcal{K} \subseteq \mathbb{S}^n$, a function $f\colon \mathcal{K} \to \mathbb{R}$ is α-strongly convex w.r.t. some norm $\|\cdot\|$ if

$$f(X) - f(Y) - \nabla f(Y)\,(X - Y) \geq \frac{\alpha}{2}\|X - Y\|^2, \quad \forall X, Y \in \mathbb{S}^n. \tag{23}$$

[10] Note that instead of online learning the optimal policies within different classes, in Table 2 we perform offline optimization in order to verify the structural alignment of each class.

When $\alpha = 0$, Eq. (23) recovers the standard definition of convexity. For a function $f\colon \mathcal{K} \to \mathbb{R}$ that is only (strongly) convex over a sub-domain $\mathcal{K} \subseteq \mathbb{S}^n$, we equivalently define $\overline{f}\colon \mathbb{S}^n \to \mathbb{R} \cup \{\infty\}$ as $\overline{f}(X) = \begin{cases} f(X), & X \in \mathcal{K} \\ +\infty, & X \notin \mathcal{K} \end{cases}$, which is (strongly) convex over $\mathbb{S}^n$. Its Fenchel dual is defined as

$$f^*(Z) := \sup_{X \in \mathbb{S}^n} \Big(\langle X, Z \rangle_F - \overline{f}(X) \Big), \quad \forall Z \in \mathbb{S}^n.$$

Matrix Functions. For any symmetric real matrix $X \in \mathbb{S}^n$, it can always be eigen-decomposed as

$$X = U \operatorname{diag}(\lambda(X)) U^\mathsf{T}, \quad \lambda(X) := [\lambda_1, \lambda_2, \ldots, \lambda_n],$$

with $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ being the eigenvalues of X and U being an orthogonal matrix. The following trace inequality will be used in analyzing the computational complexity (Appendix F.1):

Lemma 8 (Trace Inequality (Von Neumann, 1937; Fan, 1949)). *For $X, Y \in \mathbb{S}^n$, we have $\langle X, Y \rangle_F \leq \langle \lambda(X), \lambda(Y) \rangle$. The equality holds if and only if there exists an orthogonal matrix U such that*

$$X = U \operatorname{diag}(\lambda(X)) U^\mathsf{T}, \quad Y = U \operatorname{diag}(\lambda(Y)) U^\mathsf{T}.$$

A vector function $f\colon \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ is symmetric if $f(x)$ is invariant under any permutations of the components of x. Any such function can be lifted to $\mathbb{S}^n$ as a $F\colon \mathbb{S}^n \to \mathbb{R} \cup \{\infty\}$, defined as

$$F(X) := f(\lambda(X)), \quad \forall X = U \operatorname{diag}(\lambda(X)) U^\mathsf{T} \in \mathbb{S}^n.$$

Lemma 9 (Kakade et al. 2012, Theorem 28). *Let $F\colon \mathbb{S}^n \to \mathbb{R} \cup \{\infty\}$ be lifted from a symmetric $f\colon \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$. The Fenchel dual of F, namely F^*, is equal to the Fenchel dual of f lifted to $\mathbb{S}^n$.*

Lemma 10 (Kakade et al. 2012, Theorem 30). *Let $F\colon \mathbb{S}^n \to \mathbb{R} \cup \{\infty\}$ be lifted from a symmetric $f\colon \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$. The gradient of F is given as*

$$\nabla F(X) = U \operatorname{diag}\big(\nabla f(\lambda(X))\big) U^\mathsf{T}, \quad \forall X = U \operatorname{diag}(\lambda(X)) U^\mathsf{T} \in \mathbb{S}^n.$$

After defining gradients, we can define the Bregman divergence for any $F\colon \mathbb{S}^n \to \mathbb{R} \cup \{\infty\}$:

$$D_F(X \| Y) := F(X) - F(Y) - \langle \nabla F(Y), X - F \rangle_F, \quad \forall X, Y \in \mathbb{S}^n. \tag{24}$$

Hypentropy Properties. We now move on to the β-Hypentropy function $\Phi_\beta\colon \mathbb{S}^n \to \mathbb{R}$ used in our HPG algorithm, which is defined as (Ghai et al., 2020):

$$\Phi_\beta(X) := \sum_{i=1}^n \left(\lambda_i \operatorname{arcsinh} \frac{\lambda_i}{\beta} - \sqrt{\lambda_i^2 + \beta^2} \right), \quad \forall X \in \mathbb{S}^n,$$

where $\operatorname{arcsinh}(x) = \ln(x + \sqrt{x^2 + 1})$. It is lifted from the following vector function $\phi_\beta\colon \mathbb{R}^n \to \mathbb{R}$:

$$\phi_\beta(x) := \sum_{i=1}^n \left(x_i \operatorname{arcsinh} \frac{x_i}{\beta} - \sqrt{x_i^2 + \beta^2} \right), \quad \forall x \in \mathbb{R}^n.$$

We have $\frac{\partial}{\partial x_i} \phi_\beta(x) = \operatorname{arcsinh} \frac{x_i}{\beta}$. According to Lemma 10, for any $X \in \mathbb{S}^n$, we have

$$\nabla \Phi_\beta(X) = U \operatorname{diag} \left(\operatorname{arcsinh} \frac{\lambda(X)}{\beta} \right) U^\mathsf{T}, \quad \forall X = U \operatorname{diag}(\lambda(X)) U^\mathsf{T} \in \mathbb{S}^n. \tag{25}$$

Plugging Eq. (25) into Eq. (24) defines the Bregman divergence induced by Φ_β, namely D_Φ^β.

Finally, we include the following lemma, which ensures the hypentropy in $\mathbb{S}^n$ is strongly convex w.r.t. the nuclear norm $\|\cdot\|_*$ over any nuclear-norm ball.

Lemma 11 (Ghai et al. 2020, Theorem 14). *On the nuclear-norm ball $\{W \in \mathbb{S}^n \mid \|W\|_* \leq \tau\}$, Φ_β is $\big(2(\tau + \beta n)\big)^{-1}$-strongly convex w.r.t. the nuclear norm $\|\cdot\|_*$. That is,*

$$\Phi_\beta(X) - \Phi_\beta(Y) - \nabla \Phi_\beta(Y)\,(X - Y) \geq \frac{\big(2(\tau + \beta n)\big)^{-1}}{2} \|X - Y\|_*^2, \ \ \forall X, Y \in \mathbb{B}_*(\tau).$$

E.2 Direct Sum Property

Lemma 12 (Direct Sum Property)**.** *For any diagonal block matrix* $\boldsymbol{W} = \operatorname{diag}(W_1, W_2, \ldots, W_M)$*,*

$$\Phi_\beta(\boldsymbol{W}) = \sum_{m=1}^{M} \Phi_\beta(W_m),\ \|\boldsymbol{W}\|_* = \sum_{m=1}^{M} \|W_m\|_*,\ \|\boldsymbol{W}\|_2 = \max_{m\in[M]} \|W_m\|_2,$$

which implies $D_\Phi^\beta(\boldsymbol{W}\|\boldsymbol{W}') = \sum_{m=1}^{M} D_\Phi^\beta(W_m\|W_m')$*. Moreover, over the product nuclear-norm ball* $\mathbb{B}_*^M(\tau) := \left\{\boldsymbol{W} \in \mathbb{W} \;\middle|\; \|W_m\|_* \le \tau, \forall m\right\}$*,* Φ_β *is block-wise* $\left(2(\tau+\beta d_q)\right)^{-1}$*-strongly convex.*

Proof. Let $\lambda(X)$ denote the multiset of eigenvalues of a symmetric matrix X. Since $\boldsymbol{W} = \operatorname{diag}(W_1, W_2, \ldots, W_M) \in \mathbb{W}$ is a block-diagonal symmetric matrix, its eigenvalues are exactly the union of the eigenvalues of its diagonal blocks, i.e., $\lambda(\boldsymbol{W}) = \bigcup_{m=1}^{M} \lambda(W_m)$.

Because the nuclear norm $\|\cdot\|_*$, the spectral norm $\|\cdot\|_2$, and the hypentropy potential Φ_β are all spectral functions (i.e., defined w.r.t. eigenvalues), they can be decomposed into blocks:

$$\begin{aligned}
\|\boldsymbol{W}\|_* &= \sum_{\lambda\in\lambda(\boldsymbol{W})} |\lambda| = \sum_{m=1}^{M} \sum_{\lambda\in\lambda(W_m)} |\lambda| = \sum_{m=1}^{M} \|W_m\|_*, \\
\|\boldsymbol{W}\|_2 &= \max_{\lambda\in\lambda(\boldsymbol{W})} |\lambda| = \max_{m\in[M]} \max_{\lambda\in\lambda(W_m)} |\lambda| = \max_{m\in[M]} \|W_m\|_2, \\
\Phi_\beta(\boldsymbol{W}) &= \sum_{\lambda\in\lambda(\boldsymbol{W})} (\lambda+\beta)\log(\lambda+\beta) = \sum_{m=1}^{M} \sum_{\lambda\in\lambda(W_m)} (\lambda+\beta)\log(\lambda+\beta) = \sum_{m=1}^{M} \Phi_\beta(W_m).
\end{aligned}$$

From Lemma 10, the gradient of a spectral function evaluated at a block-diagonal matrix remains block-diagonal, i.e., $\nabla\Phi_\beta(\boldsymbol{W}') = \operatorname{diag}\big(\nabla\Phi_\beta(W_1'), \ldots, \nabla\Phi_\beta(W_M')\big)$. Further using the additivity of Φ_β and the fact that the Frobenius inner product also decomposes into the sum of block-wise inner products, the Bregman divergence decomposes as:

$$\begin{aligned}
D_\Phi^\beta(\boldsymbol{W}\|\boldsymbol{W}') &= \Phi_\beta(\boldsymbol{W}) - \Phi_\beta(\boldsymbol{W}') - \langle\nabla\Phi_\beta(\boldsymbol{W}'), \boldsymbol{W}-\boldsymbol{W}'\rangle_F \\
&= \sum_{m=1}^{M} \left(\Phi_\beta(W_m) - \Phi_\beta(W_m') - \langle\nabla\Phi_\beta(W_m'), W_m - W_m'\rangle_F\right) \\
&= \sum_{m=1}^{M} D_\Phi^\beta(W_m\|W_m').
\end{aligned}$$

From Lemma 11, over the d_q-dimensional nuclear-norm ball $\mathbb{B}_*(\tau)$, the hypentropy potential Φ_β is $\left(2(\tau+\beta d_q)\right)^{-1}$-strongly convex. For any $\boldsymbol{W} \in \mathbb{B}_*^M(\tau)$, we have $W_m \in \mathbb{B}_*(\tau)$ for all $m \in [M]$. Since $\Phi_\beta(\boldsymbol{W}) = \sum_{m=1}^{M} \Phi_\beta(W_m)$, the Hessian $\nabla^2\Phi_\beta(\boldsymbol{W})$ is block-diagonal where each block $\nabla^2\Phi_\beta(W_m)$ satisfies α-strong convexity. Thus, Φ_β is block-wise α-strongly convex over $\mathbb{B}_*^M(\tau)$. □

E.3 Gradient Estimation (Lemma 2)

Proof of Lemma 2. Recall that the round-t loss function for parameter $\boldsymbol{W}$, namely $\mathcal{L}_t(\boldsymbol{W})$, is

$$\mathcal{L}_t(\boldsymbol{W}) = -\mathop{\mathbb{E}}_{m'\sim\pi_{\boldsymbol{W}}(q_t)} \left[\mathop{\mathbb{E}}_{a\sim\xi_{m'}(q_t)} [r(q_t, a)]\right], \quad \forall t \in [T], \boldsymbol{W} \in \mathbb{W},$$

where $\pi_{\boldsymbol{W}}$ is the log-quadratic policy induced by $\boldsymbol{W}$ (see Eq. (7)). We intentionally changed all m's in the original definition to m', so that we can take the partial derivative w.r.t. any $m \in [M]$ as:

$$\frac{\partial}{\partial W_m}\mathcal{L}_t(\boldsymbol{W}) = -\sum_{m'=1}^{M} \left(\mathop{\mathbb{E}}_{a\sim\xi_{m'}(q_t)} [r(q_t, a)]\, \frac{\partial}{\partial W_m}\pi_{\boldsymbol{W}}(m' \mid q_t)\right).$$

By definition of $\pi_{\boldsymbol{W}}$ in Eq. (7), we have by the definition of softmax that

$$\frac{\partial}{\partial W_m}\pi_{\boldsymbol{W}}(m' \mid q_t) = \frac{\partial}{\partial W_m}\frac{\exp(q_t^\mathsf{T} W_{m'} q_t)}{\sum_{m''}\exp(q_t^\mathsf{T} W_{m''} q_t)} = \pi_{\boldsymbol{W}}(m' \mid q_t)\left(\mathbb{1}[m'=m] - \pi_{\boldsymbol{W}}(m \mid q_t)\right) q_t q_t^\mathsf{T}.$$

Therefore, we have for any $m \in [M]$ that

$$\frac{\partial}{\partial W_m}\mathcal{L}_t(\boldsymbol{W}) = -\mathop{\mathbb{E}}_{m'\sim\pi_{\boldsymbol{W}}(q_t)}\left[\mathop{\mathbb{E}}_{a\sim\xi_{m'}(q_t)}[r(q_t,a)]\ \left(\mathbb{1}[m'=m]-\pi_{\boldsymbol{W}}(m\mid q_t)\right)q_t q_t^\mathsf{T}\right]. \tag{26}$$

Hence conditional on the history before round t and the round-t query q_t, the expectation of $\widehat{G}_{t,m}$ in Eq. (13) – taken w.r.t. $m_t \sim \pi_{\boldsymbol{W}_t}(q_t)$, $a_t \sim \xi_{m_t}(q_t)$, and $r_t = r(q_t,a_t)+\eta_t$ – is given as

$$\begin{aligned}
&\mathbb{E}_t[\widehat{G}_{t,m}\mid q_t] = \mathbb{E}_t\Big[-r_t\big(\mathbb{1}[m_t=m]-\pi_{\boldsymbol{W}_t}(m\mid q_t)\big)q_tq_t^\mathsf{T}\ \Big|\ q_t\Big]\\
&= \mathop{\mathbb{E}}_{m_t\sim\pi_{\boldsymbol{W}_t}(q_t)}\left[\mathop{\mathbb{E}}_{a_t\sim\xi_{m_t}(q_t)}\left[\mathop{\mathbb{E}}_{\eta_t}\Big[-r_t\big(\mathbb{1}[m_t=m]-\pi_{\boldsymbol{W}_t}(m\mid q_t)\big)q_tq_t^\mathsf{T}\ \Big|\ q_t,m_t,a_t\Big]\ \Big|\ q_t,m_t\right]\right]\\
&= \mathop{\mathbb{E}}_{m_t\sim\pi_{\boldsymbol{W}_t}(q_t)}\left[\mathop{\mathbb{E}}_{a_t\sim\xi_{m_t}(q_t)}\Big[-r(q_t,a_t)\big(\mathbb{1}[m_t=m]-\pi_{\boldsymbol{W}_t}(m\mid q_t)\big)q_tq_t^\mathsf{T}\ \Big|\ q_t,m_t\Big]\right]\\
&= \mathop{\mathbb{E}}_{m_t\sim\pi_{\boldsymbol{W}_t}(q_t)}\left[-\mathop{\mathbb{E}}_{a_t\sim\xi_{m_t}(q_t)}[r(q_t,a_t)]\big(\mathbb{1}[m_t=m]-\pi_{\boldsymbol{W}_t}(m\mid q_t)\big)q_tq_t^\mathsf{T}\right]\\
&= \frac{\partial}{\partial W_m}\mathcal{L}_t(\boldsymbol{W}_t),\quad \forall m\in[M].
\end{aligned}$$

Thus we have $\mathbb{E}_t[\widehat{\boldsymbol{G}}_t\mid q_t] = \nabla_{\boldsymbol{W}}\mathcal{L}_t(\boldsymbol{W}_t)$ unbiased, as claimed. For the second-order moments,

$$\begin{aligned}
\sum_{m=1}^M\|\widehat{G}_{t,m}\|_2^2 &= r_t^2\|q_t\|_2^4\sum_{m=1}^M\big(\mathbb{1}[m_t=m]-\pi_{\boldsymbol{W}_t}(m\mid q_t)\big)^2\\
&\le r_t^2\left((1-\pi_{\boldsymbol{W}_t}(m_t\mid q_t))^2+\sum_{m=1}^M\pi_{\boldsymbol{W}_t}^2(m\mid q_t)\right)\le 2r_t^2,
\end{aligned}$$

where the first inequality uses $\|q_tq_t^\mathsf{T}\|_2 = \|q_t\|_2^2 \le 1$, and the second one uses the fact that $\pi_{\boldsymbol{W}_t}(q_t)\in\triangle$ is a probability distribution (hence $\sum_m\pi_{\boldsymbol{W}_t}^2(m\mid q_t)\le\sum_m\pi_{\boldsymbol{W}_t}(m\mid q_t)=1$).

Since $|r(q,a)|\le\|\Psi^*\|_2+\|\nu\|_\infty\le 1+\epsilon$ while η_t is zero-mean and conditionally 1-sub-Gaussian, we have

$$\mathbb{E}_t[r_t^2\mid q_t]\le(1+\epsilon)^2+1\le 5,\quad\text{and thus}\quad \mathbb{E}_t\left[\sum_{m=1}^M\|\widehat{G}_{t,m}\|_2^2\ \Big|\ q_t\right]\le 10.$$

□

E.4 Linearized Policy Regret Bound of HPG (Theorem 3)

Proof of Theorem 3. According to the assumption that the optimal $\boldsymbol{W}^*\in\mathbb{W}$ ensures $\|W_m^*\|_*\le\tau$, $\forall m$, we have $\boldsymbol{W}^*\in\mathbb{B}_*^M(\tau)$ (recall Eq. (10)). Recall the OMD one-step update rule in Eq. (12):

$$\boldsymbol{W}_{t+1} = \mathop{\mathrm{argmin}}_{\boldsymbol{W}\in\mathbb{B}_*^M(\tau)}\Big(\eta\big\langle\boldsymbol{W},\widehat{\boldsymbol{G}}_t\big\rangle_F+D_\Phi^\beta(\boldsymbol{W}\|\boldsymbol{W}_t)\Big).$$

Using the direct sum property Lemma 12 later proved in Appendix E.2, this is equivalent to

$$W_{t+1,m} = \mathop{\mathrm{argmin}}_{W\in\mathbb{B}_*(\tau)}\Big(\eta\big\langle W,\widehat{G}_{t,m}\big\rangle_F+D_\Phi^\beta(W\|W_{t,m})\Big),\quad\forall m\in[M], \tag{27}$$

where we recall that $\mathbb{B}_*(\tau):=\{W\in\mathbb{S}^{d_q}\mid\|W\|_*\le\tau\}$ is the nuclear-norm ball in $\mathbb{S}^{d_q}$. The rest of the proof follows from standard OMD analysis. Since $\nabla D_\Phi^\beta(W\|W_{t,m}) = \nabla\Phi_\beta(W)-\nabla\Phi_\beta(W_{t,m})$ (by definition of D_Φ^β in Eq. (24)), Eq. (27) gives the first-order optimality condition of

$$\Big\langle\eta\widehat{G}_{t,m}+\nabla\Phi_\beta(W_{t+1,m})-\nabla\Phi_\beta(W_{t,m}),W-W_{t+1,m}\Big\rangle_F\ge 0,\quad\forall W\in\mathbb{B}_*(\tau).$$

More specifically, for the m-th parameter in the optimum, namely $W_m^*\in\mathbb{B}_*(\tau)$, we have

$$\langle\eta\widehat{G}_{t,m},W_{t+1,m}-W_m^*\rangle_F\le\Big\langle\nabla\Phi_\beta(W_{t+1,m})-\nabla\Phi_\beta(W_{t,m}),W_m^*-W_{t+1,m}\Big\rangle_F$$

$$= D_\Phi^\beta(W_m^* \| W_{t,m}) - D_\Phi^\beta(W_m^* \| W_{t+1,m}) - D_\Phi^\beta(W_{t+1,m} \| W_{t,m}),$$

where the second step is the standard three-point identity (Orabona, 2019, Lemma 6.7). Adding $\langle \eta \widehat{G}_{t,m}, W_{t,m} - W_{t+1,m} \rangle_F$ on both sides and summing up from $t = 1, 2, \ldots, T$, we have

$$\sum_{t=1}^T \langle \widehat{G}_{t,m}, W_{t,m} - W_m^* \rangle_F \le \frac{D_\Phi^\beta(W_m^* \| W_{1,m}) - D_\Phi^\beta(W_m^* \| W_{T+1,m})}{\eta} + \sum_{t=1}^T \frac{\langle \eta \widehat{G}_{t,m}, W_{t,m} - W_{t+1,m} \rangle_F - D_\Phi^\beta(W_{t+1,m} \| W_{t,m})}{\eta}.$$

For the first term on the RHS, since $W_{1,m}$ is initialized as the all-zero matrix (Algorithm 1) and Bregman divergences are non-negative for convex functions (cf. Eqs. (23) and (24)), we have

$$\begin{aligned} & D_\Phi^\beta(W_m^* \| W_{1,m}) - D_\Phi^\beta(W_m^* \| W_{T+1,m}) \le \Phi_\beta(W_m^*) - \Phi_\beta(0) + 0 \\ &= \sum_{i=1}^{d_q} \left(\lambda_i(W_m^*) \operatorname{arcsinh} \frac{\lambda_i(W_m^*)}{\beta} - \sqrt{\lambda_i(W_m^*)^2 + \beta^2} + \beta \right) \\ &\le \sum_{i=1}^{d_q} |\lambda_i(W_m^*)| \operatorname{arcsinh} \frac{|\lambda_i(W_m^*)|}{\beta} \le \|W_m^*\|_* \log \frac{3\|W_m^*\|_*}{\beta}, \end{aligned}$$

where the second to last step uses the fact that $\operatorname{arcsinh}$ is an odd function, and the last step uses $\sqrt{x^2 + \beta^2} + x \le \sqrt{2} + 1$ for $|x| \le 1$ (Ghai et al., 2020, Eq. (12)) and $\|W_m^*\|_* := \sum_{i=1}^{d_q} |\lambda_i(W_m^*)|$.

For the second term, by Fenchel-Young inequality (Orabona, 2019, Lemma 6.32) and the block-wise $\big(2(\tau + \beta d_q)\big)^{-1}$-strong convexity of Φ_β (proved in Lemma 12), we have

$$\langle \eta \widehat{G}_{t,m}, W_{t,m} - W_{t+1,m} \rangle_F - D_\Phi^\beta(W_{t+1,m} \| W_{t,m}) \le \frac{\|\eta \widehat{G}_{t,m}\|_2^2}{2\big(2(\tau + \beta d_q)\big)^{-1}},$$

where $\|\cdot\|_2$, the spectral norm of a matrix, is the dual norm of the nuclear norm $\|\cdot\|_*$. Hence we have

$$\sum_{t=1}^T \langle \widehat{G}_{t,m}, W_{t,m} - W_m^* \rangle_F \le \frac{\|W_m^*\|_*}{\eta} \log \frac{3\|W_m^*\|_*}{\beta} + \sum_{t=1}^T (\tau + \beta d_q) \eta \|\widehat{G}_{t,m}\|_2^2, \quad \forall m \in [M].$$

Summing up $m \in [M]$ and using the assumption that $\|W_m^*\|_* \le \tau$, we arrive at

$$\sum_{t=1}^T \Big\langle \widehat{\boldsymbol{G}}_t, \boldsymbol{W}_t - \boldsymbol{W}^* \Big\rangle_F \le \frac{M\tau}{\eta} \log \frac{3\tau}{\beta} + (\tau + \beta d_q) \eta \sum_{t=1}^T \sum_{m=1}^M \|\widehat{G}_{t,m}\|_2^2. \tag{28}$$

Taking expectations on both sides and invoking the second part of Lemma 2 gives

$$\mathbb{E}\left[\sum_{t=1}^T \Big\langle \widehat{\boldsymbol{G}}_t, \boldsymbol{W}_t - \boldsymbol{W}^* \Big\rangle_F \right] \le \frac{M\tau}{\eta} \log \frac{3\tau}{\beta} + 10(\tau + \beta d_q) \eta T.$$

By the definition of linearized policy regret in Eq. (9), setting $\tau = 2s$, $\beta = 2s/d_q$, and $\eta = \sqrt{(M \log d_q)/T}$ therefore gives

$$\widetilde{\mathfrak{R}}_T(\Pi_{\text{quad}}) = \sup_{\boldsymbol{W}^* \in \mathbb{W}} \mathbb{E}\left[\sum_{t=1}^T \Big\langle \widehat{\boldsymbol{G}}_t, \boldsymbol{W}_t - \boldsymbol{W}^* \Big\rangle_F \right] \le 44 s \sqrt{MT \log d_q},$$

where we used the first part of Lemma 2 and also the assumption that $\|W_m^*\|_* \le 2s, \forall m$. $\square$

F Practical Implementation of HPG

F.1 Computational Cost Bound of HPG (Theorem 4)

Proof of Theorem 4. By the standard property of Bregman divergence, the single-step update of OMD in Eq. (12) is equivalent to the following two step procedure (Orabona, 2019, §6.4.3):

$$\nabla\Phi_\beta(\widetilde{\boldsymbol{W}}_{t+1}) = \nabla\Phi_\beta(\boldsymbol{W}_t) - \eta\widehat{\boldsymbol{G}}_t, \quad \boldsymbol{W}_{t+1} = \operatorname*{argmin}_{\boldsymbol{W}\in\mathbb{B}_*^M(\tau)} D_\Phi^\beta(\boldsymbol{W}\|\widetilde{\boldsymbol{W}}_{t+1}). \tag{29}$$

According to Lemma 12, both steps can be decomposed for each model. Therefore, we fix a single $m \in [M]$, and suppose we have the eigen-decomposition of $W_{t,m} \in \mathbb{S}^{d_q}$ available:

$$W_{t,m} = U_{t,m}\operatorname{diag}(\lambda_{t,m})U_{t,m}^\mathsf{T}, \quad \forall t \in [T], \tag{30}$$

where $U_{t,m} \in \mathbb{R}^{d_q\times d_q}$ and $\lambda_{t,m} \in \mathbb{R}^{d_q}$. We now prove that we can efficiently compute the eigen-decomposition of $W_{t+1,m}$, defined in Eq. (29), via the following steps (including the two mirror-space conversions folded into the first step of Section 5.1):

1. **Convert $\boldsymbol{W}_t$ to Mirror Space.** We find $Y_{t,m} = \nabla\Phi_\beta(W_{t,m})$. As defined in Eq. (25), we have
$$Y_{t,m} = U_{t,m}\operatorname{diag}(\gamma_{t,m})U_{t,m}^\mathsf{T}, \quad \gamma_{t,m,i} = \operatorname{arcsinh}(\lambda_{t,m,i}/\beta),\ \forall i \in [d_q]. \tag{31}$$
This element-wise update on the d_q eigenvalues takes $\mathcal{O}(d_q)$ time.
2. **Gradient Descent Update.** In the mirror space, we perform gradient descent $\widetilde{Y}_{t+1,m} = Y_{t,m} - \eta\widehat{G}_{t,m}$ and obtain the eigen-decomposition of $\widetilde{Y}_{t+1,m}$. Since $\widehat{G}_{t,m}$ is parallel to $q_tq_t^\mathsf{T}$, it is a rank-one matrix (see Eq. (13)). Therefore, eigen-decomposing $\widetilde{Y}_{t+1,m}$ based on that of $Y_{t,m}$ in Eq. (31) only takes $\mathcal{O}(d_q^2)$ time (see, e.g., Bunch et al., 1978; Gu & Eisenstat, 1994):
$$\widetilde{Y}_{t+1,m} = U_{t+1,m}\operatorname{diag}(\widetilde{\gamma}_{t+1,m})U_{t+1,m}^\mathsf{T}, \tag{32}$$
where $U_{t+1,m} \in \mathbb{R}^{d_q\times d_q}$ and $\widetilde{\gamma}_{t+1,m} \in \mathbb{R}^{d_q}$.
3. **Convert $\widetilde{\boldsymbol{Y}}_{t+1}$ to Primal Space.** Let $\widetilde{W}_{t+1,m} = (\nabla\Phi_\beta)^{-1}(\widetilde{Y}_{t+1,m})$. Using Lemma 10 again, it is equivalent to applying the map $(\phi_\beta)^{-1}(y) = \beta\sinh(y)$ to all the eigenvalues of $\widetilde{Y}_{t+1,m}$. Therefore, from Eq. (32), we obtain the eigen-decomposition of $\widetilde{W}_{t+1,m}$ in $\mathcal{O}(d_q)$ time as
$$\widetilde{W}_{t+1,m} = U_{t+1,m}\operatorname{diag}(\widetilde{\lambda}_{t+1,m})U_{t+1,m}^\mathsf{T}, \quad \widetilde{\lambda}_{t+1,m,i} = \beta\sinh(\widetilde{\gamma}_{t+1,m,i}),\ \forall i \in [d_q]. \tag{33}$$
4. **Projection onto $\mathbb{B}_*(\tau)$ based on D_Φ^β.** Expanding the Bregman divergence defined in Eq. (24), the projection step in Eq. (29) minimizes
$$D_\Phi^\beta(W\|\widetilde{W}_{t+1,m}) = \Phi_\beta(W) - \Phi_\beta(\widetilde{W}_{t+1,m}) + \left\langle\nabla\Phi_\beta(\widetilde{W}_{t+1,m}), W - \widetilde{W}_{t+1,m}\right\rangle_F.$$
Removing all terms independent to W and recalling that $\widetilde{Y}_{t+1,m} = \nabla\Phi_\beta(\widetilde{W}_{t+1,m})$, we write
$$\begin{aligned} W_{t+1,m} &= \operatorname*{argmin}_{W\in\mathbb{B}_*(\tau)} \left(\Phi_\beta(W) - \left\langle\widetilde{Y}_{t+1,m}, W\right\rangle_F\right) \\ &= \operatorname*{argmin}_{\|\lambda\|_1\le\tau} \operatorname*{argmin}_{U^\mathsf{T}U=I} \left(\sum_{i=1}^{d_q}\phi_\beta(\lambda_i) - \left\langle U_{t+1,m}\operatorname{diag}(\widetilde{\gamma}_{t+1,m})U_{t+1,m}^\mathsf{T}, U\operatorname{diag}(\lambda)U^\mathsf{T}\right\rangle_F\right), \end{aligned}$$
where we used the fact that any symmetric matrix $W \in \mathbb{B}_*(\tau)$ can be written as $U\operatorname{diag}(\lambda)U^\mathsf{T}$ with $\|\lambda\|_1 \le \tau$ (definition of nuclear norm) and $UU^\mathsf{T} = I$ (definition of orthogonal matrix). Fixing $\lambda \in \mathbb{R}^{d_q}$, apply the trace inequality in Lemma 8 to the second term: the inner product is at most $\langle\widetilde{\gamma}_{t+1,m}, \lambda\rangle$, with equality attained if and only if $U = U_{t+1,m}$. Therefore, we find
$$\operatorname*{argmin}_{\|\lambda\|_1\le\tau} \left(\sum_{t=1}^{d_q}(\phi_\beta(\lambda_i) - \widetilde{\gamma}_{t+1,m,i}\lambda_i)\right).$$
For each $i \in [d_q]$, minimizing this is equivalent to minimizing the Bregman divergence induced by ϕ_β (again, writing out the definition and removing all terms independent to λ_i). This gives
$$W_{t+1,m} = U_{t+1,m}\operatorname{diag}(\lambda_{t+1,m})U_{t+1,m}^\mathsf{T}, \quad \lambda_{t+1,m} = \operatorname*{argmin}_{\|\lambda\|_1\le\tau}\sum_{i=1}^{d_q} D_\phi^\beta(\lambda_i\|\widetilde{\lambda}_{t+1,m,i}), \tag{34}$$

Algorithm 2 Parameter-Free Hypentropy Policy Gradient

Initialize M 1D "coin-betting" algorithms (Cutkosky & Orabona, 2018, Algorithm 1), where the action sets are $[0, Z]$ with $Z := 2d_q$
Initialize a HPG algorithm with $\tau = 1, \beta = d_q^{-1}$, and $\eta = \sqrt{(M \log d_q)/T}$ (Algorithm 1)
for $t = 1, 2, \dots, T$ **do**
  Call each coin-betting algorithm for magnitudes $z_{t,1}, z_{t,2}, \dots, z_{t,M} \in [0, Z]$
  Call HPG algorithm for a direction $\boldsymbol{w}_t = \mathrm{diag}(w_{t,1}, w_{t,2}, \dots, w_{t,M}) \in \mathbb{B}_*^M(1)$
  Play according to the parameter $\boldsymbol{W}_t = \mathrm{diag}(z_{t,1}w_{t,1}, z_{t,2}w_{t,2}, \dots, z_{t,M}w_{t,M}) \in \mathbb{W}$
  Clip the observed reward as $\overline{r}_t = \min\big(\max(r_t, -R), R\big)$, where $R := 2 + \sqrt{2\log(2T^2)}$
  Let $\widehat{\boldsymbol{G}}_t = \mathrm{diag}(\widehat{G}_{t,1}, \widehat{G}_{t,2}, \dots, \widehat{G}_{t,M})$ be Eq. (13) evaluated at $\overline{r}_t$
  For each $m \in [M]$, pass $\langle \widehat{G}_{t,m}, w_{t,m} \rangle_F / R$ to the m-th coin-betting algorithm
  Pass $\widehat{\boldsymbol{G}}_t$ to the HPG algorithm as the gradient for the OMD step Eq. (12)

as claimed in the main text (Eq. (15)). This is implementable in $\mathcal{O}(d_q \log d_q)$ time as follows: By symmetry, the optimal λ_i must share the same sign as $\widetilde{\gamma}_{t+1,m,i}$. Let $x_i = |\lambda_i|$ and $z_i = |\widetilde{\gamma}_{t+1,m,i}|$,

$$\min_{x_i \geq 0} \sum_{i=1}^{d_q} \left(\phi_\beta(x_i) - z_i x_i\right) \text{ subject to } \sum_{i=1}^{d_q} x_i \leq \tau.$$

Introducing a Lagrange multiplier $\nu \geq 0$ for the sum constraint. Given any multiplier ν,

$$\phi'_\beta(x_i) - z_i + \nu \geq 0 \Longrightarrow x_i(\nu) = \beta \max\Big(\sinh(z_i - \nu), 0\Big), \tag{35}$$

and the optimal ν^* is the unique non-negative number satisfying $\sum_{i=1}^{d_q} x_i(\nu) \leq \tau$. Since $\sinh$ is monotone, ν^* is found after sorting $\{z_i\}_i$ in $\mathcal{O}(d_q \log d_q)$ time. This gives $x_i(\nu^*)$'s and Eq. (34).

Therefore, we have proved that moving from the eigen-decomposition of $W_{t,m}$ in Eq. (30) to that of $W_{t+1,m}$ in Eq. (34) takes $\mathcal{O}(d_q + d_q^2 + d_q + d_q \log d_q) = \mathcal{O}(d_q^2)$ time. Since there are M models to update in each round, this gives the claimed per-round complexity bound of $\mathcal{O}(d_q^2 M)$. □

In practice, one can sharpen the $\mathcal{O}(d_q^2 M)$ bound via approximation: Eq. (35) induces a low-rank structure, because the sum of $x_i(\nu^*)$'s cannot exceed τ. Therefore, a good approximation of HPG algorithm is only keeping the top-τ eigenvalues of $W_{t+1,m}$. This makes $\lambda_{t+1,m}$ τ-dimensional and $U_{t+1,m}$ in the shape of $d_q \times \tau$. The rank-one eigen-update in Eq. (32) is then solvable in $\mathcal{O}(\tau d_q)$ time. Under the $\tau \approx s$ configuration in Theorem 3, this only takes $\mathcal{O}(s d_q M)$ time per round.

F.2 Parameter-Free Implementation of HPG (Theorem 5)

In order to implement the HPG algorithm without knowing the low-rank parameter s or the nuclear-norm parameter $\tau = \max_m \|W_m^*\|_*$, we utilize the parameter-free online learning technique by Cutkosky & Orabona (2018). They provide a general framework for online linear optimization in any Banach space (a vector space with norm, e.g., our $\mathbb{S}^n$ equipped with the nuclear norm).

In our context, given the direct sum property in Lemma 12, we consider a single expert $m \in [M]$. We now work on the d_q^2-dimensional vector space equipped with nuclear norm $\|\cdot\|_*$. The action in round t is $W_{t,m} \in \mathbb{S}^{d_q}$, and the loss vector handed to the algorithm is the gradient estimator $\widehat{G}_{t,m}$ of Eq. (13), not the true gradient $\frac{\partial}{\partial W_m}\mathcal{L}_t(\boldsymbol{W}_t)$. Every bound below is therefore pathwise in $\widehat{\boldsymbol{G}}_t$, and we pass to the expected $\widetilde{\mathfrak{R}}_T(\Pi_{\text{quad}})$ of Eq. (9) only at the end.

Cutkosky & Orabona (2018) decompose $W_{t,m}$ into two parts: a magnitude $z_{t,m} = \|W_{t,m}\|_*$ and a direction $w_{t,m} = \frac{W_{t,m}}{\|W_{t,m}\|_*}$. The decision of $z_{t,m}$ is a one-dimensional decision-making problem over $\mathbb{R}$, and can be done via the *coin-betting* algorithm of Orabona & Pál (2016). For the direction $w_{t,m}$, which always has norm 1, our HPG algorithm over $\mathbb{B}_*(1)$ attains good regret. This gives Algorithm 2.

The coin-betting subroutine requires losses bounded by 1 a.s. but our reward r_t is unbounded due to the sub-Gaussian noise η_t. We adopt a standard clipping argument: We clip each r_t to $\overline{r}_t \in [-R, R]$

with $R := 2 + \sqrt{2\log(2T^2)}$, and thus the coin-betting loss $\langle \widehat{G}_{t,m}, w_{t,m}\rangle_F$ lies in $[-R, R]$ a.s. While this makes $\widehat{\boldsymbol{G}}_t$ biased, the bias can be controlled once $z_{t,m} = \|W_{t,m}\|_*$'s are bounded (see Eq. (36) in the proof). We therefore restrict each magnitude $z_{t,m}$ to $[0, Z]$ with $Z := 2d_q$ (this ensures that $\boldsymbol{W}^*$, satisfying $\|W_m^*\|_* \le 2s \le Z$, is not excluded), using the constraint set reduction of Cutkosky & Orabona (2018, Algorithm 3). Their Theorem 3 shows that this at most doubles the regret.

Proof of Theorem 5. We first write out the clipping error. Let $\boldsymbol{\Delta}_t$ be Eq. (13) evaluated at r_t minus the same expression at $\overline{r}_t$ (which is the $\widehat{\boldsymbol{G}}_t$ in Algorithm 2). Since $\boldsymbol{W}_t$ is decided before round t's randomness, the first half of Lemma 2 gives $\mathbb{E}_t[\widehat{\boldsymbol{G}}_t + \boldsymbol{\Delta}_t \mid q_t] = \nabla \mathcal{L}_t(\boldsymbol{W}_t)$. Consequently,

$$\widetilde{\mathfrak{R}}_T(\Pi_{\text{quad}}) = \mathbb{E}\left[\sum_{t=1}^{T} \left\langle \widehat{\boldsymbol{G}}_t, \boldsymbol{W}_t - \boldsymbol{W}^* \right\rangle_F\right] + \mathbb{E}\left[\sum_{t=1}^{T} \left\langle \boldsymbol{\Delta}_t, \boldsymbol{W}_t - \boldsymbol{W}^* \right\rangle_F\right], \tag{36}$$

where $\boldsymbol{W}^*$ is the optimal parameter $\boldsymbol{W}^* \in \mathbb{W}$ that we assume to satisfy $\|W_m^*\|_* \le 2s, \forall m$. Focusing on the parameter-free online learning error $\mathbb{E}[\sum_{t=1}^T \langle \widehat{\boldsymbol{G}}_t, \boldsymbol{W}_t - \boldsymbol{W}^*\rangle_F]$, we have for all m:

$$\begin{aligned}
&\sum_{t=1}^{T} \langle \widehat{G}_{t,m}, W_{t,m} - W_m^*\rangle_F = \sum_{t=1}^{T} \Big(\langle \widehat{G}_{t,m}, z_{t,m} w_{t,m}\rangle_F - \langle \widehat{G}_{t,m}, W_m^*\rangle_F \Big) \\
&= \sum_{t=1}^{T} \Big(z_{t,m} \langle \widehat{G}_{t,m}, w_{t,m}\rangle_F - \|W_m^*\|_* \langle \widehat{G}_{t,m}, w_{t,m}\rangle_F \Big) + \sum_{t=1}^{T} \Big(\|W_m^*\|_* \langle \widehat{G}_{t,m}, w_{t,m}\rangle_F - \langle \widehat{G}_{t,m}, W_m^*\rangle_F \Big).
\end{aligned}$$

The first term is the 1-dimensional online learning regret of $z_{t,m}$ w.r.t. the comparator $\|W_m^*\|_* \in [0, Z]$, with the round-t loss $g_{t,m} := \langle \widehat{G}_{t,m}, w_{t,m}\rangle_F$. Clipping gives $|g_{t,m}| \le \|\widehat{G}_{t,m}\|_2 \|w_{t,m}\|_* \le |\overline{r}_t| \le R$ (see Eq. (13)), so the rescaled losses $g_{t,m}/R$ fed to the subroutine lie in $[-1, 1]$ a.s. The coin-betting guarantee (Cutkosky & Orabona, 2018, Theorem 1) therefore applies pathwise as

$$\begin{aligned}
\sum_{t=1}^{T} \big(z_{t,m} - \|W_m^*\|_*\big) g_{t,m} &\le \mathcal{O}\left(R\|W_m^*\|_* L_m + \|W_m^*\|_* \sqrt{L_m \sum_{t=1}^{T} g_{t,m}^2} \right), \; L_m := \log\big(\|W_m^*\|_* T\big) \\
&\le \mathcal{O}\left(RsM \log(sT) + s\sqrt{M \log(sT) \sum_{t=1}^{T}\sum_{m=1}^{M} g_{t,m}^2} \right).
\end{aligned}$$

The second term, on the other hand, is $\|W_m^*\|_*$ times the HPG regret w.r.t. $\frac{W_m^*}{\|W_m^*\|_*} \in \mathbb{B}_*(1)$. Applying Eq. (28) block-wise with $\tau = 1$, $\beta = 1/d_q$, and $\eta = \sqrt{(M \log d_q)/T}$, and using $\|W_m^*\|_* \le 2s$,

$$\sum_{m=1}^{M}\sum_{t=1}^{T} \Big(\|W_m^*\|_* \langle \widehat{G}_{t,m}, w_{t,m}\rangle_F - \langle \widehat{G}_{t,m}, W_m^*\rangle_F \Big) \le 2s \left(\frac{M}{\eta} \log(3d_q) + 2\eta \sum_{t=1}^{T}\sum_{m=1}^{M} \|\widehat{G}_{t,m}\|_2^2 \right).$$

We now focus on the clipping error $\mathbb{E}[\sum_{t=1}^T \langle \boldsymbol{\Delta}_t, \boldsymbol{W}_t - \boldsymbol{W}^*\rangle_F]$. We have $\|\Delta_{t,m}\|_2 \le |r_t - \overline{r}_t| \le (|\eta_t| - u)_+$ for $u := R - 2 = \sqrt{2\log(2T^2)}$, because $|r(q,a)| \le 1 + \epsilon \le 2$. As η_t is conditionally 1-sub-Gaussian, integrating over the cases where $r_t \ne \overline{r}_t$ gives

$$\mathbb{E}_t\big[(|\eta_t| - u)_+\big] = \int_u^\infty \Pr\big[|\eta_t| > v\big] \, \mathrm{d}v \le \int_u^\infty 2e^{-v^2/2} \, \mathrm{d}v \le \frac{2}{u} e^{-u^2/2} = \frac{1}{uT^2} \le \frac{1}{T^2}.$$

Because we have $\|W_{t,m} - W_m^*\|_* \le Z + 2s \le 4d_q$ for all m, so the clipping error term is at most $T \cdot M \cdot T^{-2} \cdot 4d_q = \mathcal{O}(Md_q/T) = o(1)$. We now collect the three terms and take expectations. Since $g_{t,m}^2 \le \|\widehat{G}_{t,m}\|_2^2$ and clipping only shrinks $|r_t|$, Lemma 2 gives $\mathbb{E}[\sum_t \sum_m g_{t,m}^2] \le 10T$. As $\sqrt{\cdot}$ is concave, Jensen's inequality turns the huge square root into $\sqrt{10MT}$. Setting $\eta = \sqrt{(M \log d_q)/T}$,

$$\begin{aligned}
\widetilde{\mathfrak{R}}_T(\Pi_{\text{quad}}) &\le \mathcal{O}\Big(RsM \log(sT) + s\sqrt{MT \log(sT)} \Big) + \mathcal{O}\Big(s\sqrt{MT \log d_q} \Big) + o(1) \\
&= \mathcal{O}\Big(s\sqrt{MT \log(d_q T)} + sM\sqrt{\log T} \log(sT) \Big),
\end{aligned}$$

where the last step uses $R = \mathcal{O}(\sqrt{\log T})$ and $s \le d_q$. This proves Theorem 5. □