Minimizing Human Intervention in Online Classification

William Réveillard[1] **Vasileios Saketos**[1]

Alexandre Proutiere[1]
[1]KTH Royal Institute of Technology
{wilrev,saketos,alepro}@kth.se

Richard Combes[2]
[2]Univ. Paris-Saclay, CNRS, CentraleSupélec
richard.combes@centralesupelec.fr

Abstract

We introduce and study an online problem arising in question answering systems. In this problem, an agent must sequentially classify user-submitted queries represented by d-dimensional embeddings drawn i.i.d. from an unknown distribution. The agent may consult a costly human expert for the correct label, or guess on her own without receiving feedback. The goal is to minimize regret against an oracle with free expert access. When the time horizon T is at least exponential in the embedding dimension d, one can learn the geometry of the class regions: in this regime, we propose the Conservative Hull-based Classifier (`CHC`), which maintains convex hulls of expert-labeled queries and calls the expert as soon as a query lands outside all known hulls. `CHC` attains $\mathcal{O}(\log^d T)$ regret in T and is minimax optimal for $d = 1$. Otherwise, the geometry cannot be reliably learned without additional distributional assumptions. We show that when the queries are drawn from a subgaussian mixture, for $T \leq e^d$, a Center-based Classifier (`CC`) achieves regret proportional to $N \log N$ where N is the number of labels. To bridge these regimes, we introduce the Generalized Hull-based Classifier (`GHC`), a practical extension of `CHC` that allows for more aggressive guessing via a tunable threshold parameter. Our approach is validated with experiments, notably on real-world question-answering datasets using embeddings derived from state-of-the-art large language models.

1 INTRODUCTION

Question answering and customer support systems often start with no prior knowledge and must learn to respond to user queries over time. This work introduces an online classification problem addressing this challenge. Initially, human experts are needed to answer incoming queries, and each answered query is stored to aid future responses. However, expert interventions are costly, and the system should minimize unnecessary queries to the expert. This raises a fundamental question: *How can we design a system that progressively builds competence in answering queries while minimizing reliance on expert interventions?* We address this question in a setting where each query is represented by an embedding, typically produced by a language model. We formalize the problem as an online decision-making task, that we call *online classification with expert guidance*. At each round $t = 1, 2, \ldots$, an agent observes a query embedding $q_t \in \mathbb{R}^d$ drawn i.i.d. from an unknown distribution. There are N latent answer labels, each corresponding to some unknown region in the embedding space. The agent must assign a label to the query. She may either predict a label immediately or ask an expert to obtain the true label (incurring a cost). Crucially, if the agent guesses on its own, no label is revealed. The goal is to devise algorithms that minimize the cumulative cost after T queries, where the costs come from wrongly guessing the label or calling the expert.

Besides the introduction of the online classification with expert guidance problem, our contributions are as follows:

a. Algorithms We introduce two main algorithms: the Conservative Hull-based Classifier (`CHC`) and its generalization, `GHC`. Both exploit the geometry of class regions in the embedding space to guide decisions. `CHC` is conservative and never makes incorrect predictions; `GHC` introduces a tunable threshold that allows for more frequent guessing. This threshold controls the trade-off between accuracy and expert use, and is particularly beneficial in high-dimensional spaces such as those defined by LLM embeddings.

b. Theoretical Guarantees We provide the first regret bounds for the online classification with expert guidance problem. Using tools from convex geometry, we show that CHC achieves a regret scaling as $\mathcal{O}(\log^d T)$ after T queries, where d is the embedding dimension. These bounds are distribution-free and robust. Moreover, we show that in dimension $d = 1$, CHC is minimax optimal. To complement our distribution-free bounds, we additionally show that for well-separated subgaussian mixtures with equal weights, a Center-based Classifier (CC) attains regret proportional to $N \log N$ when $T \leq e^d$.

c. Empirical Validation We evaluate CHC, CC and GHC on both synthetic data and real-world question-answering tasks, using embeddings from various Large Language Models (LLMs). We demonstrate that leveraging higher-dimensional embeddings significantly reduces the overall regret. Additionally, we show that GHC outperforms CC and a sequential k-means baseline in all settings.

2 RELATED WORK

In this section, we review the literature on online classification and clustering. Additional related work on embedding-based retrieval and convex geometry is presented in Appendix D.

Online classification has a rich history in theoretical computer science and statistical machine learning (Littlestone, 1987; Ben-David et al., 2009; Rakhlin et al., 2015). A recent line of work studies streaming classification, where a learner must predict binary labels (Ben-Eliezer and Yogev, 2020; Alon et al., 2021; Montasser et al., 2025). These works differ from ours in several key respects: (i) their setting is adversarial; (ii) when the learner predicts incorrectly, the correct label is revealed; and (iii) their models do not involve the type of decision-making trade-offs present in ours.

In online classification with limited label information, Cesa-bianchi et al. (2004) studied binary online classification where the learner actively decides whether to ask for the correct label. Cesa-Bianchi et al. (2005) similarly considered a feedback model where information is only gained by an explicit action. Their model is adversarial, and less information is available to the learner than in our model: only the labels are observed. They obtain sublinear regret, but compared to a much easier baseline than ours, namely the best algorithm that guesses the same label at each round, whereas we compare to an oracle that always guesses the correct label. Bechavod et al. (2019) further studied one-sided feedback, revealing labels only upon positive predictions in an i.i.d. stream. In this work, we specifically exploit the geometric structure of the class regions to decide when to call the expert.

Finally, Kane et al. (2017) consider a pool-based active learning model, where the entire dataset is available from the start. Their goal is exact recovery of all labels for the entire pool, and they allow comparison queries ("which point is closer to the boundary between classes?"). In comparison, our model operates in an i.i.d. online streaming setting, and our notion of regret aims at balancing the costs of wrong guesses and expert calls.

Clustering refers to the task of grouping data points based on similarity, and is hence related to classification. Clustering is widely used to summarize and organize large datasets (Pandove et al., 2018), typically under the assumption that all data is available upfront in a batch setting. Clustering in high dimensions is notoriously difficult, both statistically and computationally (Steinbach et al., 2004). To overcome this, most batch clustering theoretical works assume strong generative or separability conditions. A rich line of work aims at characterizing the optimal clustering error rate and devising algorithms reaching these limits. For isotropic Gaussian Mixture Models (GMMs) (Pearson, 1894; Renshaw et al., 1987), it is shown in Lu and Zhou (2016) that the average number of misclustered points of Lloyd's algorithm (Lloyd, 1982) achieves the minimax error rate. More recent analyses extend to slightly more general settings, for example anisotropic GMMs (Chen and Zhang, 2024) and mixture with subexponential tails (Dreveton et al., 2024), for which optimal rates are obtained with a variant of Lloyd's algorithm. All such results rely on strong distributional assumptions. In contrast, our framework is distribution-free, and we notably derive regret bounds that hold even in the uniform setting, where no natural clusters exist. In fact, the class regions in our setting need only be defined by convex polytopes of the embedding space, a structure we use to specify the expert's labeling policy. Furthermore, our framework is online and incorporates a decision trade-off between prediction and expert consultation not found in these works.

Many modern applications—such as news feeds, social media, and interactive search—generate data as continuous streams. These scenarios have motivated the development of online clustering methods, which must operate in real time as data arrives. Choromanska and Monteleoni (2012) propose an online clustering algorithm that maintains an ensemble of batch k-means algorithms as "experts" and re-weights them by approximating their current clustering cost. Cohen-Addad et al. (2021) consider online k-means clustering: at each step the algorithm maintains k centers and incurs loss equal to the squared distance from the new point to the nearest center, aiming to minimize regret against

the best fixed clustering in hindsight. These methods focus on minimizing a distance-based cost rather than the classification/regret perspective we consider. Moreover, they do not capture the fundamental trade-off in our model between querying an expert and making an unsupported prediction.

3 MODEL AND OBJECTIVES

3.1 Online Classification Problem

We consider an online optimization problem in which users submit queries to an agent who can consult an expert for assistance. The interactions between the users, the agent, and the expert are as follows. In each round $t \geq 1$:

a. Query A user generates a query characterized by its embedding or representation $q_t \in \mathcal{E} \subset \mathbb{R}^d$ in an i.i.d. manner according to an unknown distribution μ. The agent must address the query. We assume that there are N possible labels $i \in [N] := \{1, \ldots, N\}$. Throughout the paper, the embedding space is either the hypercube $\mathcal{E} = \mathcal{I}^d := [0,1]^d$ or the unit sphere $\mathcal{E} = \mathcal{S}^{d-1} := \{x \in \mathbb{R}^d : \|x\|_2 = 1\}$ (this is usually the case when the embedding comes from an LLM).

b. Agent's Guess with or without Expert Guidance To label the query q_t, the agent has access to a dataset $\mathcal{D}_t$ consisting of query-label pairs that the expert labeled up to round t. She can then either (i) ask the expert, or (ii) make a guess. In the former case, the expert provides the correct label denoted by $i_t \in [N]$, and the pair (q_t, i_t) is appended to $\mathcal{D}_t$, i.e., $\mathcal{D}_{t+1} \leftarrow \mathcal{D}_t \cup \{(q_t, i_t)\}$. In the latter case, nothing is added to $\mathcal{D}_t$ (the correct label is not observed).

c. Partially Observed Reward The reward received at the end of round t is as follows. Let $\hat{\imath}_t$ denote the label provided by the agent. If the agent asked the expert, then $\hat{\imath}_t = i_t$ (she has the correct label), but the agent experiences a negative reward equal to $\alpha \leq 0$ corresponding to the cost of calling the expert. If she decides to guess without the expert guidance, then she does not observe the reward, and the latter is $\beta \geq 0$ if the label is correct ($\hat{\imath}_t = i_t$), and $\gamma \leq \alpha$ otherwise.

A learning algorithm is defined as a sequence of decisions made by the agent over time. Formally, such an algorithm π specifies the decision in round t as a function of the current query q_t and the past observations $\mathcal{D}_t$. Specifically, it is $\sigma(q_t, \mathcal{D}_t)$-measurable (where $\sigma(X)$ denotes the sigma-algebra generated by the random variable X). We denote the random reward obtained by the learning algorithm π in round t as $r_\pi(t)$.

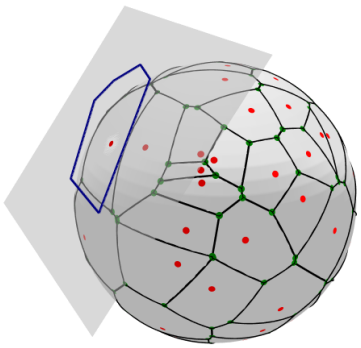


Figure 1: Voronoi tessellation of $\mathcal{S}^2$. In blue, gnomonic projection of a cell onto the tangent plane at its seed (used in Section 4.1).

3.2 Voronoi Regret

The performance of the learning algorithm used by the agent is measured by the expected cumulative reward over the first T rounds. Alternatively, this performance can be evaluated in terms of regret, defined as the difference between the expected reward of the agent and that of an Oracle algorithm with knowledge of the expert labeling policy. Throughout the paper, we assume that the expert labeling policy is dictated by a partition $\mathcal{C}_1, \ldots, \mathcal{C}_N$ of $\mathcal{E}$, unknown to the agent. The cell $\mathcal{C}_i$ corresponds to label i: whenever $q_t \in \mathcal{C}_i$, the expert provides label i.

A natural example of partition is constructed as follows. Assume that each label i also has a representation $s_i \in \mathcal{E}$ that is referred to as its *seed*. s_i may be interpreted as the typical query corresponding to the correct label i (i is the correct label to queries whose embeddings would be close to s_i). $\mathcal{E}$ can then be partitioned as the Voronoi tessellation generated by the seeds $s_1, \ldots, s_N$. This means that for any t, $i_t \in \arg\min_{i \in [N]} \|q_t - s_i\|_2$ and for any i, $\mathcal{C}_i = \{q \in \mathcal{E} : \|q - s_i\|_2 \leq \|q - s_j\|_2,\ \forall j \in [N] \setminus \{i\}\}$. This example of partition is justified as follows: When the query distribution μ is such that $\mathbb{P}(i_t = i) = 1/N$ and has a conditional density $p(q_t | i_t = i) = f(\|s_i - q_t\|_2)$ where f is decreasing, the expert's label corresponds to the Maximum Likelihood Estimator (MLE) for the correct label, assuming the s_i are known. Furthermore, when $\mathcal{E} = \mathcal{S}^{d-1}$ is the unit sphere, the expert assigns to a query q the index of the seed maximizing the cosine similarity $q^\top s_i$, as q, s_i have unit norm (cosine similarity is a standard measure of semantic similarity in embedding-based Natural Language Processing (NLP) models (Reimers and Gurevych, 2019)). In this case, the boundaries of the Voronoi cells are arcs of great circles, and the resulting cells form *spherical* convex polytopes, as illustrated in Fig. 1.

Since the Oracle algorithm knows the correct labels,

it earns a reward β per round. We hence define the regret of an algorithm π up to round T as: $R_\pi(T) = \beta T - \sum_{t=1}^{T} \mathbb{E}[r_\pi(t)]$. Our objective is to devise an algorithm that minimizes this regret.

The i.i.d. query assumption introduced in Section 3.1 is essential to make this objective meaningful: in an adversarial setting, any algorithm must incur linear regret. Indeed, in dimension one, an (even oblivious) adversary can binary-search the label boundary so that the agent either pays the expert nearly every round or suffers a constant error probability per round. Refer to Remark B.11 for a more detailed argument.

3.3 Notation and Assumptions

For a finite set of points $M = \{m_1, \dots, m_n\} \subset \mathbb{R}^d$, we denote their convex hull as $\mathrm{conv}(M) := \{\sum_{j=1}^{n} \alpha_j m_j : \alpha \geq 0, \sum_j \alpha_j = 1\}$, and their *spherical* convex hull as $\mathrm{conv}_s(M) := \mathcal{S}^{d-1} \cap \{\sum_{j=1}^{n} \alpha_j m_j : \alpha \geq 0\}$. A subset of $\mathbb{R}^d$ is said to be a convex polytope (resp. *proper* spherical convex polytope) if it can be written as $\mathrm{conv}(M)$ (resp. $\mathrm{conv}_s(M)$) for some finite set $M \subset \mathbb{R}^d$ (resp. $M \subset \mathcal{S}^{d-1}$). We will use the unifying notation $\mathrm{hull}_{\mathcal{E}}(M)$ to denote $\mathrm{conv}(M)$ if $\mathcal{E} = \mathcal{I}^d$ (hypercube) and $\mathrm{conv}_s(M)$ if $\mathcal{E} = \mathcal{S}^{d-1}$ (unit sphere). We write $f(T) = \mathcal{O}(g(T))$ if there exists a constant K that may depend on every model parameter besides T (including the dimension) such that $|f(T)| \leq K|g(T)|$ for T sufficiently large. Refer to Appendix A for a full notation table.

Our theoretical results are derived using some of the following assumptions.

Assumption 3.1. The embedding space $\mathcal{E}$ is either the hypercube $\mathcal{I}^d$ or the unit sphere $\mathcal{S}^{d-1}$.

Assumption 3.2. *(i)* When $\mathcal{E} = \mathcal{I}^d$, the cells $\{\mathcal{C}_i\}_{i\in[N]}$ are convex polytopes and form a partition of $\mathcal{E}$. When $\mathcal{E} = \mathcal{S}^{d-1}$, the cells $\{\mathcal{C}_i\}_{i\in[N]}$ form a partition of $\mathcal{E}$ and are proper spherical convex polytopes, and each cell $\mathcal{C}_i$ is contained in some open half-sphere $\mathcal{S}^+_{e_i} = \{v \in \mathcal{S}^{d-1}, v^\top e_i > 0\}$ where the pole e_i may vary with i.
(ii) In some contexts, we will further assume that the partition of $\mathcal{E}$ is a Voronoi tessellation with seeds $s_1, \dots, s_N \in \mathcal{E}$.

When $\mathcal{E} = \mathcal{S}^{d-1}$, the assumption that each cell lies within an open half-sphere is instrumental for applying the *gnomonic projection* (see Section 3.1 in Besau and Schuster (2016)) in our regret analysis. Let V denote the *natural volume measure* on $\mathcal{E}$, namely the Lebesgue measure λ on $\mathbb{R}^d$ if $\mathcal{E} = \mathcal{I}^d$, or the spherical Lebesgue measure ω on $\mathcal{S}^{d-1}$ if $\mathcal{E} = \mathcal{S}^{d-1}$.

Assumption 3.3. Depending on the context, we assume either:

(i) (Bounded density) μ is absolutely continuous with respect to V with bounded density $f_\mu = \frac{d\mu}{dV}$: there are constants $0 < c, C < \infty$ such that $c \leq f_\mu(x) \leq C$ for V-a.e. $x \in \mathcal{E}$.
(ii) (Subgaussian mixture) μ is a mixture of N σ-subgaussian distributions on $\mathbb{R}^d$ with component means s_i and mixture weights p_i. Furthermore, the mixture centers are sufficiently separated: $\delta^2_{\min} \geq 80\sigma^2 d$ where $\delta_{\min} := \min_{i\neq j} \|s_i - s_j\|_2$.

The plausibility of the center separation assumption depends on the geometry of the embedding space, a point we detail in Appendix E.5.

4 ALGORITHMS AND REGRET GUARANTEES

In this section, we present our algorithms together with their regret analyses. The first algorithm, `CHC` (Conservative Hull-based Classifier), is designed for large horizons T. It infers the geometry of the cells and makes cautious predictions, calling the expert whenever the correct label cannot be deduced from past observations. For smaller horizons, when $T \leq e^d$, we introduce `CC` (Center-based Classifier), a simple algorithm tailored to subgaussian mixtures: it queries the expert until the component centers are accurately estimated, and then predicts using the label of the nearest estimated center. Finally, we propose `GHC` (Generalized Hull-based Classifier), a flexible extension of `CHC` with a tunable threshold parameter that allows the learner to guess more frequently.

4.1 Conservative Hull-Based Classifier (`CHC`)

In round t, `CHC` maintains, for each possible label i, an estimated set $\hat{\mathcal{C}}_{i,t}$ of queries whose correct label is surely i based on past observations. This set is constructed based on the previous queries in $\mathcal{Q}_{i,t} := \{q \in \mathcal{E} : (q, i) \in \mathcal{D}_t\}$ for which the agent asked the expert and the expert label was i. The definition of $\hat{\mathcal{C}}_{i,t}$ depends on the choice of $\mathcal{E}$.

(i) If $\mathcal{E} = \mathcal{I}^d$ (hypercube), $\hat{\mathcal{C}}_{i,t} = \mathrm{conv}(\mathcal{Q}_{i,t})$ is the convex hull of the queries in $\mathcal{C}_i$ that triggered an expert call.
(ii) If $\mathcal{E} = \mathcal{S}^{d-1}$ (unit sphere), to ensure that $\hat{\mathcal{C}}_{i,t}$ is included in $\mathcal{E}$, we need to consider the spherical convex hull of $\mathcal{Q}_{i,t}$, $\hat{\mathcal{C}}_{i,t} = \mathrm{conv}_s(\mathcal{Q}_{i,t})$.
The `CHC` algorithm, whose pseudo-code is given in Algorithm 1, returns label i if $q_t \in \hat{\mathcal{C}}_{i,t}$. If $q_t \notin \cup_i \hat{\mathcal{C}}_{i,t}$, then the expert is called and provides the label. By construction, $\hat{\mathcal{C}}_{i,t} \subseteq \mathcal{C}_i$, so `CHC` always returns the correct label. In Fig. 2, we provide an example in dimension two of the convex hulls $\hat{\mathcal{C}}_{i,t}$ for $t = 200$ when μ is a mixture of truncated Gaussian distributions.

Algorithm 1 Conservative Hull-based Classifier (CHC)

Initialize $\mathcal{Q}_{i,1} \leftarrow \emptyset$ for $i \in [N]$
for $t = 1, \dots, T$ **do**
  if $\exists i \in [N] : q_t \in \text{hull}_{\mathcal{E}}(\mathcal{Q}_{i,t})$ **then**
    $\hat{\imath}_t \leftarrow i$
  else
    call expert, and set $\hat{\imath}_t \leftarrow i_t$
    $\mathcal{Q}_{i_t,t+1} \leftarrow \mathcal{Q}_{i_t,t} \cup \{q_t\}$
  end if
end for

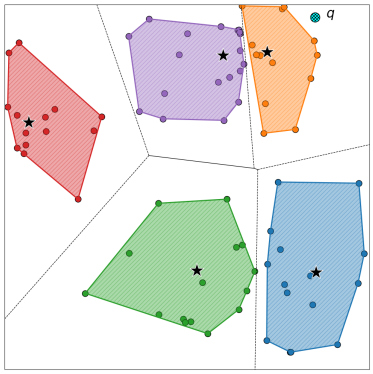


Figure 2: Hulls $\hat{\mathcal{C}}_{i,t}$ of CHC at $t = 200$. μ is a mixture of truncated Gaussian distributions with equal weights and covariance matrix $0.01I$. Stars are the seeds, circles are the queries that required an expert call.

Computational Complexity Importantly, to implement CHC, we do not need to compute convex hulls (this would be computationally prohibitive—algorithms such as QuickHull have a worst-case complexity that is exponential in $d/2$, see Theorem 3.2 in Barber et al. (1996)). Indeed, we only need to be able to check if a query $q \in \mathcal{E}$ belongs to the convex hull of some finite set of queries $\mathcal{Q}_i = \{p_1, \dots, p_n\}$. If $\mathcal{E} = \mathcal{I}^d$, this amounts to checking if there exists $\alpha \geq 0 \in \mathbb{R}^n$ such that $\sum_{i=1}^n \alpha_i = 1$ and $q = \sum_{i=1}^n \alpha_i p_i$. If $\mathcal{E} = \mathcal{S}^{d-1}$, the same condition must be checked but without requiring $\sum_{i=1}^n \alpha_i = 1$. Checking these conditions can be formulated as linear programs with n variables and $n + d + 1$ or $n + d$ constraints respectively. These programs can be solved in a time polynomial in the dimension and the number of queries by interior-point methods, see e.g. Khachiyan (1979); Karmarkar (1984).

Regret Analysis of CHC in Arbitrary Dimension To state the regret guarantees for CHC, we need to define the number $F(P)$ of *flags* of a polytope P. A flag of P is a sequence $(F_j)_{j=0}^{d-1}$ of faces[1] of P such that $\dim(F_j) = j$ and $F_0 \subset \cdots \subset F_{d-1}$. For example, for $d = 2$ and $P = [0, 1]^2$, a flag is a sequence formed by a vertex of the square and an edge incident to that vertex, and $F(P) = 4 \times 2 = 8$. Refer to Section 2.1 in Besau et al. (2018) for details. The following regret guarantees for CHC hold for any convex polytope partition $\{\mathcal{C}_i\}_{i\in[N]}$ of the embedding space by the expert.

Theorem 4.1. *Under Assumptions 3.2(i) and 3.3(i): (a) if $\mathcal{E} = \mathcal{I}^d$ and $d \geq 2$, then the regret of CHC satisfies*

$$R_{\text{CHC}}(T) \leq (\beta - \alpha)\frac{C}{c}\frac{\sum_{i=1}^N F(\mathcal{C}_i)}{(d+1)^{d-1}d!}\log^d(T) + \mathcal{O}(\log^{d-1}(T)\log\log(T)).$$

(b) if $\mathcal{E} = \mathcal{S}^{d-1}$, $d \geq 3$ and each cell $\mathcal{C}_i$ is contained in an open halfsphere $\mathcal{S}^+_{e_i}$, then the regret of CHC satisfies

$$R_{\text{CHC}}(T) \leq (\beta - \alpha)\frac{KC}{c}\frac{\sum_{i=1}^N F(\mathcal{C}_i)}{d^{d-2}(d-1)!}\log^{d-1}(T) + \mathcal{O}(\log^{d-2}(T)\log\log(T)),$$

where $K = \max_{i\in[N]}\left(\frac{\max_{y\in\mathcal{C}_i} y^\top e_i}{\min_{y\in\mathcal{C}_i} y^\top e_i}\right)^d$.

The proof of Theorem 4.1, provided in Appendix B.1.1, begins by noting that since CHC always returns the correct label, its regret is given by $R_{\text{CHC}}(T) = (\beta-\alpha)\mathbb{E}[C_T]$ where C_T denotes the total number of expert calls made up to round T. As the probability of calling the expert for a given query q_t is precisely the probability that q_t falls outside the current convex hull estimates at round t, we obtain the following expression for the regret:

$$R_{\text{CHC}}(T) = (\beta - \alpha)\sum_{t=1}^T\sum_{i=1}^N \mathbb{E}[\mu(\mathcal{C}_i \setminus \hat{\mathcal{C}}_{i,t})].$$

To upper bound $\mathbb{E}[\mu(\mathcal{C}_i \setminus \hat{\mathcal{C}}_{i,t})]$, we build upon a result from Bárány and Buchta (1993), which characterizes the expected volume of the convex hull of n points sampled uniformly at random from a convex polytope in $\mathbb{R}^d$. We extend this result in two directions: (i) to distributions with bounded density via a rejection sampling argument; (ii) to the spherical setting where points lie on $\mathcal{S}^{d-1}$, by mapping spherical polytopes to Euclidean polytopes in $\mathbb{R}^{d-1}$ using the gnomonic projection. The dependence on C/c (and CK/c in the spherical case) arises from the rejection sampling step and is likely an artifact of the proof technique—we conjecture that this dependence could be removed with a tighter analysis.

We demonstrate in Appendix E.2 that our analysis is sharp by deriving a distribution-free asymptotic lower

[1]Faces of a polytope are specific planar surfaces on its boundary, e.g., for a cube, the 0-dim faces are the vertices, the 1-dim faces are the 12 edges, and the 2-dim faces are the 6 squares forming the boundary.

bound on the regret of CHC that matches Theorem 4.1 when μ is uniform. This suggests that CHC does not exploit favorable scenarios where μ is highly concentrated around the seed queries of the cells.

Voronoi Regret. We finally derive explicit regret upper bounds when the cells form a Voronoi tessellation, as described in Section 3.2. The following corollary is obtained from Theorem 4.1 by controlling the total number of flags, $\sum_{i=1}^{N} F(\mathcal{C}_i)$. The proof, given in Appendix B.1.2, leverages an upper bound on the number of faces of a convex polytope from McMullen (1970), along with the observation that the number of facets of each cell $\mathcal{C}_i$ is at most $N-1+2d$ when $\mathcal{E}=\mathcal{I}^d$ (the hypercube), and at most $N-1$ when $\mathcal{E}=\mathcal{S}^{d-1}$ (the unit sphere).

Corollary 4.2. *Under Assumptions 3.2(ii) and 3.3(i): (a) if $\mathcal{E}=\mathcal{I}^d$ and $d\geq 2$, then the regret of* CHC *satisfies*

$$R_{\texttt{CHC}}(T) \leq \frac{8(\beta-\alpha)CN}{3c(d+1)^{d-1}}\left(\frac{2e(N+2d)}{d-1}\right)^{d/2}\log^d(T) + \mathcal{O}(\log^{d-1}(T)\log\log(T)).$$

(b) if $\mathcal{E}=\mathcal{S}^{d-1}$, $d\geq 3$ and each cell $\mathcal{C}_i$ is contained in an open half-sphere $\mathcal{S}^+_{e_i}$, then the regret of CHC *satisfies*

$$R_{\texttt{CHC}}(T) \leq \frac{4(\beta-\alpha)KCN}{cd^{d-2}}\left(\frac{2eN}{d-2}\right)^{(d-1)/2}\log^{d-1}(T) + \mathcal{O}(\log^{d-2}(T)\log\log(T)).$$

It may seem counterintuitive that the constant preceding the leading term (e.g., $\log^d T$ when $\mathcal{E}=\mathcal{I}^d$) decreases with increasing d. This behavior suggests that the error term conceals a more intricate dependency on the dimension. Precisely characterizing this dependency is difficult and touches upon unresolved questions in convex geometry (see Appendix E.3 for further discussion). Nonetheless, our upper bounds on the regret of CHC are most informative in regimes where T exceeds a threshold that grows exponentially with d, or equivalently, when the dimension is not excessively large. This is consistent with results from Chakraborti et al. (2021), which show that the expected volume of a convex hull formed by random points remains negligible until the number of samples is exponential in d. In our setting, this implies that a significant reduction in expert queries only begins once this sample threshold is crossed.

Minimax Optimality of CHC in Dimension One The following theorem, whose proof is given in Appendix B.1.3, provides a sharper regret bound in the case $d=1$, under a weaker condition of μ. An *exact* but unwieldy formula for $R_{\texttt{CHC}}(T)$ is also given by equation (4) in Appendix B.1.3. Whether this density-free bound can be generalized to dimension $d\geq 2$ is unclear, and it is discussed in Appendix E.4.

Theorem 4.3. *Assume that*[2] $\mathcal{E}=[0,1]$ *and that μ has no atoms. Then for all $T\geq 1$, the regret of* CHC *satisfies $R_{\texttt{CHC}}(T)\leq 2(\beta-\alpha)N\log(T+1)$.*

Finally, we present, in the following theorem proved in Appendix B.1.4, minimax regret lower bounds (i.e., satisfied by *any* algorithm). These bounds match the regret guarantees for CHC up to a universal constant, hence CHC is minimax optimal in dimension one.

Theorem 4.4 (Lower bound on the minimax regret)**.** *Assume that $\mathcal{E}=[0,1]$ and that μ is uniform on $\mathcal{E}$. Denote by $\theta=(s_1,...,s_N)\in[0,1]^N$ a set of N query seeds in $[0,1]$. Then for all $T\geq 1$, the minimax regret satisfies*

$$\inf_{\pi}\max_{\theta\in[0,1]^N} R_\pi(T,\theta) \geq (\beta-\alpha)\frac{N-1}{64\sqrt{2}}\log\left(\frac{T+1}{2}\right) = \Omega((\beta-\alpha)(N-1)\log T).$$

4.2 Center-based Classifier (CC)

To complement our regret bounds in the large T regime, which hold under very mild distributional assumptions (Assumption 3.3(i)), we present regret bounds in the $T\leq e^d$ regime for a simple Center-based Classifier (CC), under Assumption 3.3(ii) on μ. In this section, the expert labeling policy is assumed to be given by the Voronoi tessellation with seeds s_i, as described in Section 3.2. CC proceeds in two phases:

Phase 1 (Explore) In the first phase, the expert is called at each round t, and CC stores estimates $\hat{s}_i(t) := \frac{1}{|\mathcal{Q}_{i,t}|}\sum_{q\in\mathcal{Q}_{i,t}} q$ once each label has been observed at least once. These yield an estimate for the minimum center gap $\delta_{\min}$ as $\hat{\delta}_{\min}(t)=\min_{i\neq j}\|\hat{s}_i(t)-\hat{s}_j(t)\|_2$. The first phase ends when each label has been observed sufficiently, i.e., at $T_1 := \min\{t\leq T : \forall i\in[N]\,|\mathcal{Q}_{i,t}|\geq \frac{108\sigma^2}{\hat{\delta}^2_{\min}(t)}(d+2\log T)\}$ with the convention $T_1=T$ if the stopping condition is not reached before round T.

Phase 2 (Commit) In the second phase ($t\in[T_1+1,T]$), the center estimates are no longer updated. CC guesses at each round the label of the closest estimated center: $\hat{\imath}_t=\arg\min_i\|q_t-\hat{s}_i(T_1)\|_2$.

Theorem 4.5. *Let $p_{\min}:=\min_{i\in[N]}p_i$. Under Assumptions 3.2(ii) and 3.3(ii), if $T\leq e^d$, the regret of* CC *satisfies*

$$R_{\texttt{CC}}(T)\leq \frac{41}{5}\frac{(\beta-\alpha)(\log N+1)}{p_{\min}}+2(\beta-\gamma)(N+1).$$

For mixtures with equal weights, the regret upper bound of CC is proportional to $N\log N$. In Appendix

[2]The theorem holds if $\mathcal{E}$ is an arbitrary interval of $\mathbb{R}$.

B.2, we present a more general upper bound of CC's regret which hold for all regimes of T along with its proof. It contains two main steps: (i) the definition of T_1 ensures that the center estimation error at the end of the first phase is bounded as $\|s_i - \hat{s}_i(T_1)\|_2 \leq \delta_{\min}/4$ with high probability; (ii) this entails that the probability of a wrong guess in the second phase is exponentially small in d.

When μ is a subgaussian mixture, Theorem 4.5 demonstrates that CC is effective in the $T \leq e^d$ regime, complementing the distribution-free guarantees of CHC which are most meaningful for large T. These regimes can be bridged: a hybrid approach—running CC in the early phases and switching to CHC once T exceeds e^d—would inherit the regret guarantees of Corollary 4.2 and Theorem 4.5 across both regimes. However, a more practical approach is to design a single algorithm that performs well in all settings. To this end, we introduce in the next section a tunable version of CHC.

4.3 Generalized Hull-based Classifier (GHC)

When the embedding space is high-dimensional—as is often the case when using representations from LLMs—the CHC algorithm introduced in Section 4.1 may struggle to perform effectively. In particular, the number of expert-labeled queries must be at least linear in the dimension d for the convex hulls to have non-zero volume, and exponential in d to cover a substantial portion of the space. Consequently, unless the query budget is large, CHC tends to be overly conservative, frequently deferring to the expert. Moreover, CHC fails to fully capitalize on favorable cases where the distribution μ is sharply concentrated around the seeds of each label. The CC algorithm on the other hand might not perform well when the horizon T is not much larger than $\frac{\log N}{p_{\min}}$ (before the seeds are well estimated). To address these limitations, we introduce GHC(τ), an extension of CHC specifically designed for high-dimensional settings. This variant incorporates a tunable threshold τ that enables the algorithm to take calculated risks, particularly when the density of μ is skewed toward known regions.

In GHC(τ), there is an initial phase where CHC is applied until all N labels have been observed. Then, all the hulls contain at least one query, and the agent guesses $\hat{\imath}_t = i$ when $d(q_t, \hat{\mathcal{C}}_{i,t}) \leq \tau d(q_t, \hat{\mathcal{C}}_{j,t})$ for all $j \neq i$ for some threshold $\tau \in [0, 1]$, where $d(q, S) := \inf_{q' \in S} \|q - q'\|_2$ denotes the Euclidean distance from q to a set S. Otherwise, the expert provides the label i_t and $\hat{\mathcal{C}}_{i_t,t}$ is updated. Note that the hulls $\hat{\mathcal{C}}_{i,t}$ depend on τ and will typically be formed with fewer samples as τ increases. If $\tau = 0$, GHC(0) corresponds to CHC; if $\tau = 1$, the expert is never called almost surely once all hulls are non-empty. Intuitively, if μ is concentrated around the seeds s_i, it is more likely that the distances $d(q_t, \hat{\mathcal{C}}_{i,t})$ are well separated. This separation allows for more aggressive guessing (by increasing the threshold τ) with little additional risk, ultimately leading to lower regret. The pseudo-code of GHC is provided in Algorithm 2.

Algorithm 2 Generalized Hull-based Classifier (GHC(τ))

Initialize $\mathcal{Q}_{i,1} \leftarrow \emptyset$ for $i \in [N]$
for $t = 1, \ldots, T$ **do**
  while $\exists\, i \in [N] : \mathcal{Q}_{i,t} = \emptyset$ **do**
    Apply Algorithm 1
  end while
  if $\exists i \in [N] : d(q_t, \text{hull}_{\mathcal{E}}(\mathcal{Q}_{i,t})) \leq \tau \min_{j \neq i} d(q_t, \text{hull}_{\mathcal{E}}(\mathcal{Q}_{j,t}))$ **then**
    $\hat{\imath}_t \leftarrow i$
  else
    Call expert, and set $\hat{\imath}_t \leftarrow i_t$
    $\mathcal{Q}_{i_t,t+1} \leftarrow \mathcal{Q}_{i_t,t} \cup \{q_t\}$
  end if
end for

Obtaining regret bounds for GHC is far more challenging than for CHC, the main difficulty being that the convex hulls $\hat{\mathcal{C}}_{i,t}$ are no longer formed by i.i.d. queries. This is discussed in more detail in Appendix E.6.

In Fig. 3, we present, for $\mathcal{E} = [0, 1]^2$, the decision regions of GHC(τ) in round $t = 250$ for a few values of τ. Here, μ is a mixture of Gaussian distributions with small variance. If q_t lands in the "correct guess" region (resp. "wrong guess" region), the agent guesses and labels q_t correctly (resp. incorrectly). If q_t lands in the "expert call" region, the expert provides its correct label. This figure illustrates the benefit of choosing a larger τ for distributions that are strongly concentrated around representative queries, which is typically the case in practice. Additional examples are provided in Fig. 6 in Appendix E.6.

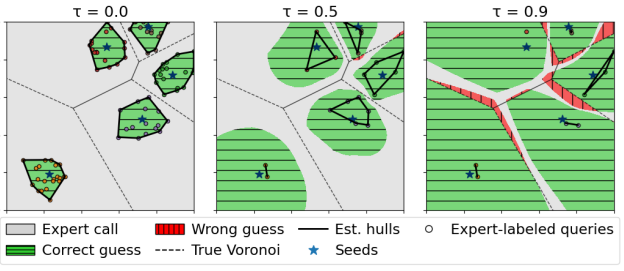


Figure 3: Decision regions of GHC(τ) for a mixture of truncated Gaussian distributions, covariance matrix $0.0025I$.

Computational Complexity To run GHC(τ) for $\tau > 0$, we need to compute the distance of q to a (spherical) convex hull of queries $p_1, \ldots, p_n$. If $\mathcal{E} = \mathcal{I}^d$, this can be performed in polynomial time without computing the convex hull, simply by computing the value of the quadratic program $\min_{\alpha \in \mathbb{R}^n} \|q - \sum_{i=1}^n \alpha_i p_i\|_2^2$ subject to $\alpha_i \geq 0$, $\sum_{i=1}^n \alpha_i = 1$. If $\mathcal{E} = \mathcal{S}^{d-1}$, the distance to the spherical convex hull is instead given as the value of the same optimization problem but where the last constraint is replaced by $\sum_{i=1}^n \alpha_i p_i \in \mathcal{S}^{d-1}$. We show in Appendix B.3 that this amounts to solving the non-negative least squares problem $\min_{\alpha \geq 0} \|q - \sum_{i=1}^n \alpha_i p_i\|_2^2$. If $\alpha^\star$ is its solution and $\sum_i \alpha_i^\star p_i \neq 0$, we establish that $d(q, \mathrm{conv}_s\{p_1, \ldots, p_n\}) = \sqrt{2 - 2\|\sum_{i=1}^n \alpha_i^\star p_i\|_2}$.

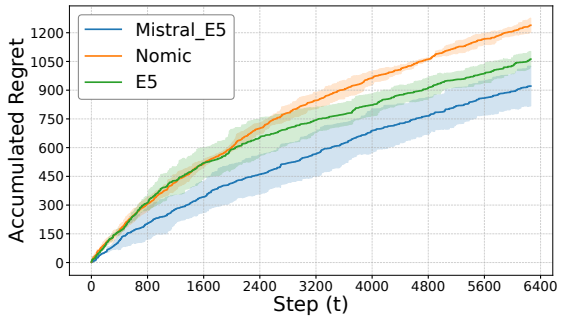
5 NUMERICAL EXPERIMENTS

In this section, we evaluate GHC on a real-world dataset using various embedding models. The reward values are instantiated as $\alpha = -1$, $\beta = +1$, $\gamma = -10$. Additional experiments on both synthetic and real-world datasets are deferred to Appendix F, where we notably show that GHC outperforms both CC and a sequential k-means baseline.

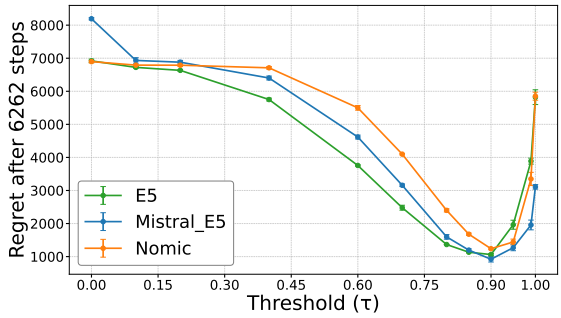
To evaluate GHC in a realistic setting, we introduce the Quora Question Group (QQG) dataset. Each question in the dataset is assigned one of $N = 1103$ labels, where each label corresponds to a group of questions that can be addressed by a single answer. We compare the performance of three different retrievers/embedding models: Nomic[3] (Nussbaum et al., 2024), E5[4](Wang et al., 2022) and Mistral_E5 [5] (Wang et al., 2024) (see Appendix F.2.2). We first encode each question with the chosen retrievers, producing embeddings of dimension 784 for Nomic, 1,024 for E5, and 4,096 for Mistral_E5. Given the relatively small size of our datasets (each label corresponds to a few questions only), we initialize the algorithm by picking an example question for each label, thereby obviating the first phase of GHC (where CHC is applied until every label is represented).

Fig. 4a shows the average accumulated regret for the three models when we use their respective optimal threshold. Results are averaged over three independent runs. As expected, Mistral_E5, the most recent and largest model in our experiments with 7 billion parameters, outperforms both E5 (330 million parameters) and Nomic (110 million parameters). Fig. 4b shows each model's performance across all threshold values τ. Observe that the optimal thresholds are quite high, which is expected since our embeddings are high-dimensional.

In Appendix F.2.3, we extend our evaluation to the ComQA (Abujabal et al., 2019) and CQADupStack (Hoogeveen et al., 2015) datasets. ComQA contains open-domain questions derived from Wikipedia, whereas CQADupStack comprises expert-domain questions from technical forums, including topics such as physics and mathematics. We provide additional details about our datasets in Appendix F.2.1.



(a) Accumulated regret



(b) Regret vs. threshold ($T = 6262$)

Figure 4: Comparison of GHC using different LLMs on Quora Question Groups dataset.

6 CONCLUSION

We introduced the problem of online classification with expert guidance and proposed algorithms tailored to different budget regimes. In particular, we developed the CC and CHC algorithms, which enjoy strong regret guarantees in low- and high-horizon settings, respectively, and further proposed the more versatile GHC algorithm, which demonstrates strong empirical performance across all regimes. A natural next step is to provide a theoretical regret analysis of this latter algorithm.

More broadly, the online learning problem studied here represents a first instance of a broader class of challenges aimed at minimizing costly human intervention during model training or fine-tuning. We anticipate that many more such problems will emerge, especially in the context of fine-tuning or adapting large foundation models using human feedback.

[3] huggingface.co/nomic-ai/nomic-embed-text-v1
[4] huggingface.co/intfloat/e5-large
[5] huggingface.co/intfloat/e5-mistral-7b-instruct

References

Abujabal, A., Roy, R. S., Yahya, M., and Weikum, G. (2019). Comqa: A community-sourced dataset for complex factoid question answering with paraphrase clusters. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 307–317, Minneapolis, MN. Association for Computational Linguistics.

Alon, N., Ben-Eliezer, O., Dagan, Y., Moran, S., Naor, M., and Yogev, E. (2021). Adversarial laws of large numbers and optimal regret in online classification. *SIAM Journal on Computing*, 0(0):STOC21–154–STOC21–210.

Anderssen, R. S., Brent, R. P., Daley, D. J., and Moran, P. A. P. (1976). Concerning $\int_0^1 \cdots \int_0^1 \left(x_1^2 + \cdots + x_k^2\right)^{1/2} dx_1 \cdots dx_k$ and a taylor series method. *SIAM Journal on Applied Mathematics*, 30(1):22–30.

Artstein-Avidan, S., Florentin, D., and Milman, V. (2012). *Order Isomorphisms on Convex Functions in Windows*, pages 61–122. Springer Berlin Heidelberg, Berlin, Heidelberg.

Bajaj, P., Campos, D., Craswell, N., Deng, L., Gao, J., Liu, X., Majumder, R., McNamara, A., Mitra, B., Nguyen, T., et al. (2016). Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

Bárány, I. (1989). Intrinsic volumes and f-vectors of random polytopes. *Math. Ann.*, 285(4):671–699.

Bárány, I. (2008). Random points and lattice points in convex bodies. *Bulletin of the American Mathematical Society*, 45(3):339–365.

Bárány, I. and Buchta, C. (1993). Random polytopes in a convex polytope, independence of shape, and concentration of vertices. *Mathematische Annalen*, 297(1):467–497.

Bárány, I., Fradelizi, M., Goaoc, X., Hubard, A., and Rote, G. (2020). Random polytopes and the wet part for arbitrary probability distributions. *Annales Henri Lebesgue*, 3:701–715.

Barber, C. B., Dobkin, D. P., and Huhdanpaa, H. (1996). The quickhull algorithm for convex hulls. *ACM Trans. Math. Softw.*, 22(4):469–483.

Bechavod, Y., Ligett, K., Roth, A., Waggoner, B., and Wu, S. Z. (2019). Equal opportunity in online classification with partial feedback. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

BehnamGhader, P., Adlakha, V., Mosbach, M., Bahdanau, D., Chapados, N., and Reddy, S. (2024). Llm2vec: Large language models are secretly powerful text encoders. *arXiv preprint arXiv:2404.05961*.

Ben-David, S., Pál, D., and Shalev-Shwartz, S. (2009). Agnostic online learning. In *COLT 2009 - The 22nd Conference on Learning Theory, Montreal, Quebec, Canada, June 18-21, 2009*.

Ben-Eliezer, O. and Yogev, E. (2020). The adversarial robustness of sampling. In *Proceedings of the 39th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, PODS'20, page 49–62, New York, NY, USA. Association for Computing Machinery.

Besau, F. and Schuster, F. (2016). Binary operations in spherical convex geometry. *Indiana University Mathematics Journal*, 65(4):1263–1288.

Besau, F., Schütt, C., and Werner, E. M. (2018). Flag numbers and floating bodies. *Advances in Mathematics*, 338:912–952.

Billera, L. J. and Ehrenborg, R. (2000). Monotonicity of the cd-index for polytopes. *Mathematische Zeitschrift*, 233(3):421–441.

Bárány, I. and Larman, D. G. (1988). Convex bodies, economic cap coverings, random polytopes. *Mathematika*, 35(2):274–291.

Cesa-bianchi, N., Gentile, C., and Zaniboni, L. (2004). Worst-case analysis of selective sampling for linear-threshold algorithms. In Saul, L., Weiss, Y., and Bottou, L., editors, *Advances in Neural Information Processing Systems*, volume 17. MIT Press.

Cesa-Bianchi, N., Lugosi, G., and Stoltz, G. (2005). Minimizing regret with label efficient prediction. *IEEE Transactions on Information Theory*, 51(6):2152–2162.

Chakraborti, D., Tkocz, T., and Vritsiou, B.-H. (2021). A note on volume thresholds for random polytopes. *Geometriae Dedicata*, 213(1):423–431.

Chen, X. and Zhang, A. Y. (2024). Achieving optimal clustering in gaussian mixture models with anisotropic covariance structures. In Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., and Zhang, C., editors, *Advances in Neural Information Processing Systems*, volume 37, pages 113698–113741. Curran Associates, Inc.

Choromanska, A. and Monteleoni, C. (2012). Online clustering with experts. In Glowacka, D., Dorard, L., and Shawe-Taylor, J., editors, *Proceedings of the Workshop on On-line Trading of Exploration and Exploitation 2*, volume 26 of *Proceedings of Machine Learning Research*, pages 1–18, Bellevue, Washington, USA. PMLR.

Cohen-Addad, V., Guedj, B., Kanade, V., and Rom, G. (2021). Online k-means clustering. In Banerjee, A. and Fukumizu, K., editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 1126–1134. PMLR.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T., editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dreveton, M., Gözeten, A., Grossglauser, M., and Thiran, P. (2024). Universal lower bounds and optimal rates: Achieving minimax clustering error in sub-exponential mixture models. In Agrawal, S. and Roth, A., editors, *Proceedings of Thirty Seventh Conference on Learning Theory*, volume 247 of *Proceedings of Machine Learning Research*, pages 1451–1485. PMLR.

Gao, T., Yao, X., and Chen, D. (2021). SimCSE: Simple contrastive learning of sentence embeddings. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t., editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Hoogeveen, D., Verspoor, K. M., and Baldwin, T. (2015). Cqadupstack: A benchmark data set for community question-answering research. In *Proceedings of the 20th Australasian Document Computing Symposium*, ADCS '15, New York, NY, USA. Association for Computing Machinery.

Hsu, D., Kakade, S., and Zhang, T. (2012). A tail inequality for quadratic forms of subgaussian random vectors. *Electron. Commun. Probab.*, 17(52).

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., Casas, D. d. l., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., et al. (2023). Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Kane, D. M., Lovett, S., Moran, S., and Zhang, J. (2017). Active classification with comparison queries. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE.

Karmarkar, N. (1984). A new polynomial-time algorithm for linear programming. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pages 302–311.

Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., and Yih, W.-t. (2020). Dense passage retrieval for open-domain question answering. In Webber, B., Cohn, T., He, Y., and Liu, Y., editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Khachiyan, L. G. (1979). A polynomial algorithm in linear programming. In *Doklady Akademii Nauk*, volume 244, pages 1093–1096. Russian Academy of Sciences.

Lewis, P. S. H., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., and Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Li, Z., Zhang, X., Zhang, Y., Long, D., Xie, P., and Zhang, M. (2023). Towards general text embeddings with multi-stage contrastive learning.

Littlestone, N. (1987). Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. In *28th Annual Symposium on Foundations of Computer Science (sfcs 1987)*, pages 68–77.

Lloyd, S. (1982). Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137.

Lu, Y. and Zhou, H. H. (2016). Statistical and computational guarantees of lloyd's algorithm and its variants.

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In Cam, L. M. L. and Neyman, J., editors, *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press.

McMullen, P. (1970). The maximum numbers of faces of a convex polytope. *Mathematika*, 17:179–184.

Montasser, O., Shetty, A., and Zhivotovskiy, N. (2025). Beyond worst-case online classification: Vc-based regret bounds for relaxed benchmarks.

Muennighoff, N., Tazi, N., Magne, L., and Reimers, N. (2022). Mteb: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*.

Nussbaum, Z., Morris, J. X., Duderstadt, B., and Mulyar, A. (2024). Nomic embed: Training a reproducible long context text embedder. *arXiv preprint arXiv:2402.01613*.

Pandove, D., Goel, S., and Rani, R. (2018). Systematic review of clustering high-dimensional and large datasets. *ACM Trans. Knowl. Discov. Data*, 12(2).

Pearson, K. (1894). III. contributions to the mathematical theory of evolution. *Philos. Trans. R. Soc. Lond. A*, 185(0):71–110.

Prochno, J., Schütt, C., and Werner, E. (2022). Best and random approximation of a convex body by a polytope. *Journal of Complexity*, 71:101652. Approximation and Geometry in High Dimensions.

Rakhlin, A., Sridharan, K., and Tewari, A. (2015). Online learning via sequential complexities. *Journal of Machine Learning Research*, 16(6):155–186.

Ratcliffe, J. G. (2019). *Foundations of hyperbolic manifolds*. Graduate texts in mathematics. Springer Nature, Cham, Switzerland, 3 edition.

Reimers, N. and Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In Inui, K., Jiang, J., Ng, V., and Wan, X., editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Renshaw, A. E., Titterington, D. M., Smith, A. F. M., and Makov, H. E. (1987). Statistical analysis of finite mixture distributions. *J. R. Stat. Soc. Ser. A*, 150(3):283.

Rosenblatt, M. (1952). Remarks on a Multivariate Transformation. *The Annals of Mathematical Statistics*, 23(3):470 – 472.

Ross, S. M. (2010). *Introduction to Probability Models*. Academic Press, San Diego, CA, 10th edition.

Schütt, C. (1991). The convex floating body and polyhedral approximation. *Isr. J. Math.*, 73(1):65–77.

Schütt, C. and Werner, E. M. (1990). The convex floating body. *Mathematica Scandinavica*, 66(2):275–290.

Springer, J. M., Kotha, S., Fried, D., Neubig, G., and Raghunathan, A. (2024). Repetition improves language model embeddings. *ArXiv*, abs/2402.15449.

Steinbach, M., Ertöz, L., and Kumar, V. (2004). *The Challenges of Clustering High Dimensional Data*, pages 273–309. Springer Berlin Heidelberg, Berlin, Heidelberg.

Thakur, N., Reimers, N., Rücklé, A., Srivastava, A., and Gurevych, I. (2021). Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663*.

van den Oord, A., Li, Y., and Vinyals, O. (2018). Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748.

Wang, L., Yang, N., Huang, X., Jiao, B., Yang, L., Jiang, D., Majumder, R., and Wei, F. (2022). Text embeddings by weakly-supervised contrastive pretraining. *arXiv preprint arXiv:2212.03533*.

Wang, L., Yang, N., Huang, X., Yang, L., Majumder, R., and Wei, F. (2024). Improving text embeddings with large language models. In Ku, L.-W., Martins, A., and Srikumar, V., editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11897–11916, Bangkok, Thailand. Association for Computational Linguistics.

Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., and Zhou, M. (2020). Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *CoRR*, abs/2002.10957.

Wang, Z., Hamza, W., and Florian, R. (2017). Bilateral multi-perspective matching for natural language sentences. *CoRR*, abs/1702.03814.

Ziegler, G. M. (1998). *Lecture on Polytopes*. Springer.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes] Our model, assumptions and algorithms are described in the main text.
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes] The algorithm's computational complexities are discussed in Sections 4 and 4.3. A regret analysis of `CHC` is also provided.
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes] The code used for the experiments is provided in the supplemental material, and will be made publicly accessible upon publication.
2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes] Assumptions are collected in Section 3.3 and every theoretical result is stated with the relevant ones.
   (b) Complete proofs of all theoretical results. [Yes] Detailed proofs are provided in the appendices.
   (c) Clear explanations of any assumptions. [Yes] Assumptions are motivated when they are introduced in Section 3 (e.g. i.i.d. assumption, convex polytope assumption).
3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes] All the code used to produce figures and tables is provided in the supplemental material, and will be made publicly accessible upon publication.
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes] The threshold parameter of `GHC` used in the experiments is specified, and all additional details about our setup are provided In Appendix F.
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes] The numerical results of Section 5 and Appendix F are reported with the corresponding standard deviation when applicable.
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes] Hardware specifications and compute resources are provided in Appendix F.3.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes] The prior works that introduced the datasets or LLMs we use for our numerical experiments are all cited in the paper and direct links are provided.
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Yes] The QQG dataset is provided in the supplemental material.
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]
5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable] We conduct no such experiment.
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Table of Contents

A NOTATION

Problem Setting	
N	Number of labels/classes.
$\mathcal{E}$	Embedding space: hypercube $\mathcal{I}^d$ or unit sphere S^{d-1}.
d	Dimension of the embedding space $\mathcal{E}$.
α, β, γ	reward for expert call, correct guess and incorrect guess respectively.
$s_i \in \mathcal{E}$	Unknown seed (representative) of queries with label i.
$\mathcal{C}_i$	Voronoi cell for queries with label i, defined by seeds s_j.
T	Time horizon (budget).
$q_t \in \mathcal{E}$	Query received at round t.
μ	Distribution from which q_t is sampled.
f_μ	Density of μ w.r.t. Lebesgue or spherical measure.
C, c	Upper and lower bounds on f_μ.
$\delta_{\min}$	Minimum gap $\min_{i \neq j} \|s_i - s_j\|_2$ between centers.
$i_t \in [N]$	Correct label of q_t.
$\hat{\imath}_t \in [N]$	Agent's guess for label of q_t.
$\mathcal{D}_t$	Expert-labeled pairs available at step t.
$\mathcal{F}_t$	Filtration of history up to round t.
$r_\pi(t)$	Reward at round t.
$R_\pi(T)$	Cumulative regret for algorithm π: $\beta T - \sum_{t=1}^{T} \mathbb{E}[r_\pi(t)]$.
C_T	Total number of expert calls.
M_T	Total number of incorrect guesses (mistakes).
Algorithm & Analysis	
conv	Euclidean convex hull.
conv_s	Spherical convex hull.
$\text{hull}_\mathcal{E}$	conv if $\mathcal{E} = \mathcal{I}^d$, conv_s if $\mathcal{E} = \mathcal{S}^{d-1}$.
$d(q, S) = \inf_{s \in S} \|q - s\|_2$	Euclidean distance from q to S.
$\mathcal{Q}_{i,t}$	Set of queries labeled as i by the expert up to round t.
$\hat{\mathcal{C}}_{i,t}$	Convex hull of $\mathcal{Q}_{i,t}$, estimate of $\mathcal{C}_i$ at round t.
$\tau \in [0, 1]$	Threshold parameter.
`CHC`	Conservative Hull-based Classifier.
`GHC`(τ)	Generalized Hull-based Classifier with threshold τ.
`CC`	Center-based Classifier.
$f(T) = \mathcal{O}(g(T))$	There exists K independent of T (potentially depending on other parameters) such that $\|f(T)\| \leq K\|g(T)\|$ for $T \geq T_0 \in \mathbb{N}$
$n_i(t) = \sum_{s=1}^{t} \mathbb{1}\{i_s = i\}$	Number of queries with true label i up to t.
$\mu_{\mathcal{C}_i}$	Conditional distribution $\mu(\cdot \cap \mathcal{C}_i)/\mu(\mathcal{C}_i)$.
w^ν	ν-measure of the wet part.
$F(P)$	Number of flags of polytope P, i.e., increasing sequences $(F_j)_{j=0}^{d-1}$ of j-dimensional faces of P.
λ	Lebesgue measure on $\mathbb{R}^d$.
ω	Spherical Lebesgue measure on $\mathcal{S}^{d-1}$.
log	Natural logarithm (base e).
$\text{Bin}(n, p)$	Binomial distribution with parameters n and p.

B PROOFS OF SECTION 4

In this section, we provide complete proofs of all the statements in Section 4.

B.1 Proofs of Section 4.1

Remark B.1. Throughout this appendix, when the notations $\mathcal{O}(\cdot)$ or $o(\cdot)$ are used, the limit is w.r.t. the number of sampled points n or the number of queries t, T depending on the context. All other problem parameters—such as the cells and their dimension—are fixed, so the hidden constants may depend on them.

B.1.1 Proof of Theorem 4.1

(a) Case $\mathcal{E} = \mathcal{I}^d$. We rely on the following Theorem from Bárány and Buchta (1993).

Proposition B.2 (Theorem 2 from Bárány and Buchta (1993))**.** *Let P a convex polytope in $\mathbb{R}^d$ with $d \geq 2$ and $n \geq 1$ points $p_1, \dots, p_n$ sampled independently and uniformly at random in P, with convex hull P_n. Then*

$$\mathbb{E}[\lambda(P \setminus P_n)] = \frac{\lambda(P)F(P)}{(d+1)^{d-1}(d-1)!}\frac{\log^{d-1} n}{n} + \mathcal{O}\left(\frac{\log^{d-2}(n)\log\log n}{n}\right)$$

where $F(P)$ is the number of flags of P, i.e., the number of sequences $F_0 \subset F_1 \subset \cdots \subset F_{d-1}$ of i-dimensional faces of P.

Using a rejection sampling argument, it can be generalized to bounded densities as follows.

Corollary B.3. *Let P a convex polytope in $\mathbb{R}^d$ with $d \geq 2$ and $n \geq 1$ points $p_1, \dots, p_n$ sampled independently from a distribution ν supported on P with a density $f_P = \mathrm{d}\nu/\mathrm{d}\lambda$ that satisfies $c \leq f_P(x) \leq C$ for λ-a.e. $x \in P$ for $c > 0$, $C < \infty$. Denote by P_n the convex hull of $p_1, \dots, p_n$. Then*

$$\mathbb{E}\left[\nu(P \setminus P_n)\right] \leq \frac{C}{c}\frac{F(P)}{(d+1)^{d-1}(d-1)!}\frac{\log^{d-1} n}{n} + \mathcal{O}\left(\frac{\log^{d-2}(n)\log\log n}{n}\right)$$

We suspect that the dependency in C/c of Corollary B.3 is an artifact of our proof technique and could be improved. This corollary is proved further below.

Proof of Theorem 4.1(a). Let us denote by $(\mathcal{F}_s)_{s\leq t}$ the filtration generated by the query and expert feedback history history up to round t. The conditional probability of calling the expert at time t is given as

$$\mathbb{P}(q_t \in \bigcup_{i=1}^{N}(\mathcal{C}_i \setminus \hat{\mathcal{C}}_{i,t})|\mathcal{F}_{t-1}) = \sum_{i=1}^{N}\mu(\mathcal{C}_i \setminus \hat{\mathcal{C}}_{i,t}). \tag{1}$$

Since `CHC` always guesses correctly, the total regret suffered at time T is then

$$R_{\texttt{CHC}}(T) = (\beta-\alpha)\mathbb{E}[C_T] = (\beta-\alpha)\sum_{t=1}^{T}\mathbb{E}[\mathbb{P}(q_t \in \bigcup_{i=1}^{N}(\mathcal{C}_i \setminus \hat{\mathcal{C}}_{i,t})|\mathcal{F}_{t-1})] = (\beta-\alpha)\sum_{t=1}^{T}\sum_{i=1}^{N}\mathbb{E}[\mu(\mathcal{C}_i \setminus \hat{\mathcal{C}}_{i,t})].$$

By assumption 3.3(i), the density of q_t is bounded above by C and bounded below by c. The conditional density of q_t given $q_t \in \mathcal{C}_i$ is given by $f_{\mathcal{C}_i}(q) := f_\mu(q)/\mu(\mathcal{C}_i) \in [c/\mu(\mathcal{C}_i), C/\mu(\mathcal{C}_i)]$ so it is also bounded above and below, and the ratio of the upper and lower bounds on $f_{\mathcal{C}_i}$ is also equal to C/c. Let $n_i(t)$ be the number of points in $\mathcal{C}_i$ at time t. Since $\mathbb{P}(q_t \in \mathcal{C}_i) = \mu(\mathcal{C}_i)$, $n_i(t)$ is binomial with parameters $(t, \mu(\mathcal{C}_i))$ and we have $\mathbb{E}[n_i(t)] = t\mu(\mathcal{C}_i)$. Conditionally to $n_i(t) = n$, $\hat{\mathcal{C}}_{i,t}$ is distributed as the convex hull P_n of n points sampled at random within $\mathcal{C}_i$ with density $f_{\mathcal{C}_i}$. Denoting by $\mu_{\mathcal{C}_i}$ the conditional distribution of q_t given that $q_t \in \mathcal{C}_i$, $\mathbb{E}[\mu(\mathcal{C}_i \setminus \hat{\mathcal{C}}_{i,t})|n_i(t) = n] = \mathbb{E}[\mu(\mathcal{C}_i \setminus P_n)|n_i(t) = n] = \mu(\mathcal{C}_i)\mathbb{E}[\mu_{\mathcal{C}_i}(\mathcal{C}_i \setminus P_n)]$ by independence between P_n and $n_i(t)$. At time t, applying Corollary B.3 conditionally to $n_i(t) = n > 0$ we obtain

$$\mathbb{E}[\mu(\mathcal{C}_i \setminus \hat{\mathcal{C}}_{i,t})|n_i(t) = n] = \frac{C}{c}\frac{\mu(\mathcal{C}_i)F(\mathcal{C}_i)}{(d+1)^{d-1}(d-1)!}\frac{\log^{d-1} n}{n} + \mathcal{O}\left(\frac{\log^{d-2}(n)\log\log n}{n}\right).$$

If $n = 0$, we simply have $\mathbb{E}[\mu(\mathcal{C}_i \setminus \hat{\mathcal{C}}_{i,t})|n_i(t) = n] = \mu(\mathcal{C}_i)$. Thus, for φ_1 defined by $\varphi_1(x) = \log^{d-1}(x)/x$ for $x > 0$ and $\varphi_1(0) = 0$ and φ_2 defined by $\varphi_2(x) = \log^{d-2}(x)(\log\log x)/x$ for $x > 1$, $\varphi_2(0) = \mu(\mathcal{C}_i), \varphi_2(1) = 0$, we have

$$\mathbb{E}[\mu(\mathcal{C}_i \setminus \hat{\mathcal{C}}_{i,t})] = \frac{C}{c}\frac{\mu(\mathcal{C}_i)F(\mathcal{C}_i)}{(d+1)^{d-1}(d-1)!}\mathbb{E}[\varphi_1(n_i(t))] + \mathcal{O}\left(\mathbb{E}[\varphi_2(n_i(t))]\right).$$

Applying Lemma C.1 to $n_i(t)$ gives

$$\mathbb{E}[\varphi_1(n_i(t))] = \frac{\log^{d-1}(t\mu(\mathcal{C}_i))}{t\mu(\mathcal{C}_i)} + \mathcal{O}(1/t)$$

and

$$\mathbb{E}[\varphi_2(n_i(t))] = \frac{\log^{d-2}(t\mu(\mathcal{C}_i))\log\log(t\mu(\mathcal{C}_i))}{t\mu(\mathcal{C}_i)} + \mathcal{O}(1/t).$$

Using the fact that $t\mu(\mathcal{C}_i) \leq t$, we obtain

$$\mathbb{E}[\mu(\mathcal{C}_i \setminus \hat{\mathcal{C}}_{i,t})] \leq \frac{C}{c}\frac{F(\mathcal{C}_i)}{(d+1)^{d-1}(d-1)!}\frac{\log^{d-1}(t)}{t} + \mathcal{O}\left(\frac{\log^{d-2} t\log\log t}{t}\right).$$

This then entails a bound on the regret as

$$R_{\texttt{CHC}}(T) \leq (\beta-\alpha)\frac{C}{c}\sum_{i=1}^{N}\frac{F(\mathcal{C}_i)}{(d+1)^{d-1}(d-1)!}\sum_{t=1}^{T}\frac{\log^{d-1}(t)}{t} + \mathcal{O}\left(\sum_{t=1}^{T}\frac{\log^{d-2} t\log\log t}{t}\right).$$

To finish the proof, we simplify the sums over t. For $x \geq t_0 = \lceil e^{d-1}\rceil$, $x \mapsto \varphi_1(x)$ is decreasing, so that

$$\begin{aligned}\sum_{t=1}^{T}\frac{\log^{d-1}(t)}{t} &= \sum_{t=1}^{t_0-1}\frac{\log^{d-1} t}{t} + \sum_{t=t_0}^{T}\frac{\log^{d-1} t}{t}\\ &\leq \int_1^T \frac{\log^{d-1} x}{x}\mathrm{d}x + \mathcal{O}(1)\\ &= \frac{\log^d T}{d} + \mathcal{O}(1).\end{aligned}$$

Similarly, $x \mapsto \varphi_2(x)$ is decreasing for x large enough, so that

$$\begin{aligned}\sum_{t=1}^{T}\frac{\log^{d-2} t\log\log t}{t} &= \int_{e^2}^{T}\frac{\log^{d-2} x\log\log x}{x}\mathrm{d}x + \mathcal{O}(1)\\ &= \frac{\log^{d-1}(T)((d-1)\log\log(T)-1)}{(d-1)^2} + \mathcal{O}(1).\end{aligned}$$

Finally,

$$\begin{aligned}R_{\texttt{CHC}}(T) &\leq (\beta-\alpha)\frac{C}{c}\sum_{i=1}^{N}\frac{F(\mathcal{C}_i)}{(d+1)^{d-1}(d-1)!}\frac{\log^d T}{d} + \mathcal{O}(\log^{d-1}(T)\log\log(T))\\ &= (\beta-\alpha)\frac{C}{c}\frac{\sum_{i=1}^{N}F(\mathcal{C}_i)}{(d+1)^{d-1}d!}\log^d T + \mathcal{O}(\log^{d-1}(T)\log\log(T)).\end{aligned}$$

□

Proof of Corollary B.3. In the proof, we omit the subscript P and simply denote the density f_P by f. For a uniform sample $q_1, \ldots, q_n$ from P with density $u(x) = 1/\lambda(P)$ for $x \in P$, we denote its convex hull by $Q_n = \mathrm{conv}(q_1, \ldots, q_n)$. We want to upper bound $E_{f,n} := \mathbb{E}[\nu(P \setminus P_n)]$ by relating it to $E_{u,n} := \mathbb{E}[\lambda(P \setminus Q_n)]$, which can be bounded by Proposition B.2. We proceed as follows:

(i) Start from the sample $p_1, \dots, p_n$ generated from ν. We denote by f their density.

(ii) For each $i \in [n]$, sample U_i uniform on $[0,1]$ independently, and keep p_i if and only if $U_i \leq c/f(p_i)$. Note that $0 < c/C \leq c/f(p_i) \leq 1$.

Let J be the set of indices of the points p_i that are kept.

- By construction, $\mathbb{P}(i \in J | p_i = x) = \mathbb{P}(U_i \leq c/f(p_i) | p_i = x) = \mathbb{P}(U_i \leq c/f(x)) = c/f(x)$ so for any measurable set A, $\mathbb{P}(p_i \in A \cap i \in J) = \int_A f(x) \frac{c}{f(x)} \mathrm{d}x = c\lambda(A)$ and $\mathbb{P}(i \in J) = c\lambda(P)$. This implies that the conditional density of the kept points is u. Therefore, conditionally on $K := |J| = k$, the set of kept points $\{p_i | i \in J\}$ is distributed as $\{q_1, \dots, q_k\}$ with q_i independent and uniform on P.
- Since $\mathbb{P}(i \in J) = c\lambda(P)$, the random number of kept points K follows a $\mathrm{Bin}(n, c\lambda(P))$ distribution.
- Consider $P_n^J = \mathrm{conv}\{p_i | i \in J\}$. Clearly $P_n^J \subseteq P_n$ so $\lambda(P \setminus P_n) \leq \lambda(P \setminus P_n^J)$.

This means that $\nu(P \setminus P_n) = \int_{P \setminus P_n} f(x) \mathrm{d}x \leq C\lambda(P \setminus P_n) \leq C\lambda(P \setminus P_n^J)$, so that

$$\begin{aligned} E_{f,n} &\leq C\mathbb{E}[\lambda(P \setminus P_n^J)] \\ &= C \sum_{k=0}^{n} \mathbb{P}(K = k) \mathbb{E}[\lambda(P \setminus P_n^J) | K = k]. \end{aligned}$$

Conditionally on $K = k$, since $\{p_i | i \in J\}$ is distributed as $\{q_1, \dots, q_k\}$, P_n^J is distributed as Q_k. Thus

$$E_{f,n} \leq C \sum_{k=0}^{n} \mathbb{P}(K = k) E_{u,k}.$$

Then, by Proposition B.2, $E_{u,k} = \frac{\lambda(P)F(P)}{(d+1)^{d-1}(d-1)!} \frac{\log^{d-1}(k)}{k} + \mathcal{O}\left(\frac{\log^{d-2} k \log\log k}{k}\right)$ if $k > 1$, and $\lambda(P)$ if $k = 0$. Applying Lemma C.1 (as in the proof of Theorem 4.1) gives $\mathbb{E}[E_{u,K}] \leq \frac{\lambda(P)F(P)}{(d+1)^{d-1}(d-1)!} \frac{\log^{d-1}(nc\lambda(P))}{nc\lambda(P)} + \mathcal{O}\left(\frac{\log^{d-2} n \log\log n}{n}\right)$ which ensures

$$E_{f,n} \leq \frac{C}{c} \frac{F(P)}{(d+1)^{d-1}(d-1)!} \frac{\log^{d-1}(n)}{n} + \mathcal{O}\left(\frac{\log^{d-2} n \log\log n}{n}\right).$$

□

(b) Case $\mathcal{E} = \mathcal{S}^{d-1}$. To derive the regret bound in the spherical setting, we first obtain a variant of Corollary B.3 by projecting spherical convex polytopes to Euclidean polytopes via the *gnomonic projection* (see e.g. Besau and Schuster (2016)), similarly to Besau et al. (2018) in the proof of their Theorem 1.3. This result holds for distributions with bounded densities w.r.t. the spherical Lebesgue measure ω.

Corollary B.4. *Let P a spherically convex polytope in $\mathcal{S}^{d-1}$ contained in an open halfsphere $\mathcal{S}_e^+ = \{v \in \mathcal{S}^{d-1}, v^\top e > 0\}$ with $d \geq 3$ and $n \geq 1$ points $p_1, \dots, p_n$ sampled independently from a distribution ν supported on P with a density $f_P = \mathrm{d}\nu/\mathrm{d}\omega$ that satisfies $c \leq f_P(x) \leq C$ for ω-a.e. $x \in P$ for $c > 0$, $C < \infty$. Denote by P_n the spherical convex hull of $p_1, \dots, p_n$. Then*

$$\mathbb{E}\left[\nu(P \setminus P_n)\right] \leq \frac{C}{c} \left(\frac{\max_{y \in P} y^\top e}{\min_{y \in P} y^\top e}\right)^d \frac{F(P)}{d^{d-2}(d-2)!} \frac{\log^{d-2} n}{n} + \mathcal{O}\left(\frac{\log^{d-3}(n) \log\log n}{n}\right)$$

The derivation of the regret bound is then exactly the same as in the case $\mathcal{E} = \mathcal{I}^d$ and is consequently omitted.

Proof of Corollary B.4. Let

$$g_e : \mathcal{S}_e^+ \to \mathbb{R}^{d-1}$$
$$x \mapsto \frac{x}{x^\top e} - e$$

denote the gnomonic projection from the open halfsphere $\mathcal{S}_e^+$ onto the hyperplane tangent to $\mathcal{S}^{d-1}$ at the pole e of $\mathcal{S}_e^+$. This projection is bijective and its inverse is given by $g_e^{-1}(y) = (y+e)/\|y+e\|_2$ (Besau et al., 2018). Let $Q = g_e(P)$. Q is a Euclidean convex polytope in $\mathbb{R}^{d-1}$. Let $\rho = g_e\#\nu$ denote the pushforward measure of ν onto Q, i.e., $\rho(A) = \nu(g_e^{-1}(A))$ for all measurable set $A \subseteq Q$. ρ is a probability measure on Q. Let $q_i = g_e(p_i)$ for $i = 1, \dots, n$. The points q_i are independent samples from ρ. Let $Q_n = \mathrm{conv}(q_1, \dots, q_n)$ be their Euclidean convex hull.

A key property of the gnomonic projection is that it maps spherical geodesics (arcs of great circles) to Euclidean straight lines. Consequently, it maps the spherical convex hull of a set of points to the Euclidean convex hull of the projected points:

$$g_e(P_n) = g_e(\mathrm{conv}_s(p_1, \dots, p_n)) = \mathrm{conv}(g_e(p_1), \dots, g_e(p_n)) = Q_n.$$

Since g_e is injective, it preserves the set difference

$$g_e(P \setminus P_n) = g_e(P) \setminus g_e(P_n) = Q \setminus Q_n$$

so that $P \setminus P_n = g_e^{-1}(Q \setminus Q_n)$. By the definition of the pushforward measure ρ,

$$\rho(Q \setminus Q_n) = \nu(P \setminus P_n).$$

Taking the expectation yields

$$\mathbb{E}[\rho(Q \setminus Q_n)] = \mathbb{E}[\nu(P \setminus P_n)]. \tag{2}$$

We now verify the conditions of Corollary B.3 for our measure ρ on Q. By assumption, ν has a density $f_P = \mathrm{d}\nu/\mathrm{d}\omega$ w.r.t. the spherical Lebesgue measure ω satisfying $0 < c \le f_P(y) \le C$ for ω-a.e. every $y \in P$. The pushforward measure $\rho = g_e\#\nu$ has a density $f_Q = d\rho/d\lambda$ on Q given by $f_Q(x) \propto f_P(g_e^{-1}(x))(1+\|x\|^2)^{-d/2}$ for $x \in Q$ (see Besau et al. (2018), p.36). On the compact Q, the density f_Q is bounded below by $c' = \min_{x\in Q} f_Q(x) > 0$ and above by $C' = \max_{x\in Q} f_Q(x) < \infty$, where $\frac{C'}{c'} = \frac{C}{c}\left(\frac{1+r_{\max}^2}{1+r_{\min}^2}\right)^{d/2}$ for $r_{\max} = \max_{x\in g_e(P)} \|x\|$, $r_{\min} = \min_{x\in g_e(P)} \|x\|$. Note that for $x = g_e(y) \in g_e(P)$, $\|x\|^2 = \|\frac{y}{y^\top e} - e\|^2 = \frac{1}{(y^\top e)^2} - 1$ since $y, e \in \mathcal{S}^{d-1}$. Thus

$$\left(\frac{1+r_{\max}^2}{1+r_{\min}^2}\right)^{d/2} = \left(\frac{\max_{y\in P} 1/(y^\top e)^2}{\min_{y\in P} 1/(y^\top e)^2}\right)^{d/2} = \left(\frac{\max_{y\in P} y^\top e}{\min_{y\in P} y^\top e}\right)^{d}$$

since $y^\top e > 0$ for every $y \in P$.

Thus, the conditions of Corollary B.3 are satisfied for ρ on $Q \subset \mathbb{R}^{d-1}$. Applying it gives

$$\mathbb{E}[\rho(Q \setminus Q_n)] \le \frac{C'}{c'} \frac{F(P)}{d^{d-2}(d-2)!} \frac{\log^{d-2} n}{n} + \mathcal{O}\left(\frac{\log^{d-3}(n) \log\log n}{n}\right). \tag{3}$$

Combining inequality (3) with the equality of expectations (2), we obtain the desired result for the spherical case:

$$\mathbb{E}[\nu(P \setminus P_n)] \le \frac{C'}{c'} \frac{F(P)}{(d+1)^{d-1}(d-1)!} \frac{\log^{d-1} n}{n} + \mathcal{O}\left(\frac{\log^{d-3}(n) \log\log n}{n}\right).$$

□

Remark B.5. As can be seen in the proof of Corollary B.4, the constant $K = \min_{i\in[N]} \left(\frac{\max_{y\in\mathcal{C}_i} y^\top e_i}{\min_{y\in\mathcal{C}_i} y^\top e_i}\right)^d$ that appears in the statement of Theorem 4.1(b) comes from the bounds on the conditional densities after applying the gnomonic projections g_{e_i} to each cell $\mathcal{C}_i$. The e_i, $i \in [N]$ can be chosen to minimize K. In Fig. 1, we used $e_i = s_i$, but this choice is not optimal in general.

B.1.2 Proof of Corollary 4.2

The bound on the number of flags that we derive is the following.

Lemma B.6. *The total number of flags $\sum_{i=1}^N F(\mathcal{C}_i)$ in the Voronoi tessellation of $\mathcal{E}$ is bounded as*

- $\frac{8}{3}Nd!\left(\frac{2e(N+2d)}{d-1}\right)^{d/2}$ *if* $\mathcal{E}=\mathcal{I}^d$
- $4N(d-1)!\left(\frac{2eN}{d-2}\right)^{(d-1)/2}$ *if* $\mathcal{E}=\mathcal{S}^{d-1}$

We first show the following result, which is essentially a dual version of the upper bound from Section 2 of Besau et al. (2018).

Lemma B.7. *If a convex polytope P of $\mathbb{R}^d$ has k faces of dimension $d-1$,*

$$F(P)\leq\begin{cases}(2\ell)!\frac{k}{k-\ell}\binom{k-\ell}{\ell} & \text{if } d=2\ell\\ 2(2\ell+1)!\binom{k-\ell-1}{\ell} & \text{if } d=2\ell+1.\end{cases}$$

Proof of Lemma B.7. Consider the dual polytope P^o of P, which has k vertices and the same number of flags as P. By the upper bound theorem (McMullen (1970), or Theorem 6.6 in Billera and Ehrenborg (2000)), $F(P^o)\leq F(C_d(k))$ where $C_d(k)$ is the cyclic polytope in $\mathbb{R}^d$ with k vertices. Since $C_d(k)$ is simplicial, $F(C_d(k))=d!f_{d-1}(C_d(k))$, where $f_{d-1}(C_d(k))$ is the number of $(d-1)$-dimensional faces of $C_d(k)$. Hence, we have

$$F(P)\leq d!f_{d-1}(C_d(k)).$$

An exact formula for $f_{d-1}(C_d(k))$ is known and is given by

$$f_{d-1}(C_d(k))=\begin{cases}\frac{k}{k-\ell}\binom{k-\ell}{\ell} \text{ if } d=2\ell\\ 2\binom{k-\ell-1}{\ell} \text{ if } d=2\ell+1\end{cases}$$

(see Ziegler (1998), p.25). This concludes the proof. □

Proof of Lemma B.6. First consider the case $\mathcal{E}=\mathcal{I}^d$. As $\mathcal{C}_i$ is a convex polytope formed by at most $N-1+2d$ hyperplanes/facets of the hypercube $\mathcal{E}$ ($N-1$ for the hyperplanes, $2d$ for the facets of the hypercube), it has at most $N-1+2d$ faces of dimension $d-1$. The previous lemma applies and it gives

$$\sum_{i=1}^N F(\mathcal{C}_i)\leq\begin{cases}N(2\ell)!\frac{N+2d-1}{N+2d-1-\ell}\binom{N+2d-1-\ell}{\ell} \text{ if } d=2\ell\\ 2N(2\ell+1)!\binom{N+2d-1-\ell-1}{\ell} \text{ if } d=2\ell+1.\end{cases}$$

In particular, using $\binom{n}{k}\leq(\frac{en}{k})^k$, $\sum_{i=1}^N F(\mathcal{C}_i)\leq 2Nd!\frac{N+2d-1}{N+2d-1-\lfloor d/2\rfloor}(\frac{e(N+2d-1-\lfloor d/2\rfloor)}{\lfloor d/2\rfloor})^{\lfloor d/2\rfloor}$. Since $\lfloor d/2\rfloor\leq\frac{N+2d-1}{4}$, we have $N+2d-1-\lfloor d/2\rfloor\geq\frac{3}{4}(N+2d-1)$, so $\frac{N+2d-1}{N+2d-1-\lfloor d/2\rfloor}\leq 4/3$, and

$$\sum_{i=1}^N F(\mathcal{C}_i)\leq\frac{8}{3}Nd!\left(\frac{2e(N+2d)}{d-1}\right)^{d/2}.$$

When $\mathcal{E}=\mathcal{S}^{d-1}$, the argument is slightly simpler since only the $N-1$ boundaries of the Voronoi tessellation can form facets of the cells (the space $\mathcal{S}^{d-1}$ does not have boundaries to consider that might increase their number). More precisely, $\mathcal{C}_i$ is a spherically convex polytope formed by at most $N-1$ great spheres of $\mathcal{E}$, so it has at most $N-1$ faces of dimension $d-2$. We first remark that under the open halfsphere assumption, each cell has at least d facets (Theorem 6.3.7 in Ratcliffe (2019)). Consequently, we must have $N\geq d+1$. A spherical analogue of Lemma B.7 applies and it gives

$$\sum_{i=1}^N F(\mathcal{C}_i)\leq\begin{cases}N(2\ell)!\frac{N-1}{N-1-\ell}\binom{N-1-\ell}{\ell} \text{ if } d-1=2\ell\\ 2N(2\ell+1)!\binom{N-1-\ell-1}{\ell} \text{ if } d-1=2\ell+1\end{cases}$$

(note that the binomial coefficients are well defined since the open halfsphere assumption implies $N \geq d+1$). In particular, using $\binom{n}{k} \leq (\frac{en}{k})^k$, $\sum_{i=1}^{N} F(\mathcal{C}_i) \leq 2N(d-1)! \frac{N-1}{N-1-\lfloor (d-1)/2 \rfloor} (\frac{e(N-1-\lfloor (d-1)/2 \rfloor)}{\lfloor (d-1)/2 \rfloor})^{\lfloor (d-1)/2 \rfloor}$. Since $\lfloor (d-1)/2 \rfloor \leq \frac{N-1}{2}$, we have $N-1-\lfloor (d-1)/2 \rfloor \geq \frac{1}{2}(N-1)$, so $\frac{N-1}{N-1-\lfloor (d-1)/2 \rfloor} \leq 2$, and

$$\sum_{i=1}^{N} F(\mathcal{C}_i) \leq 4N(d-1)! \left(\frac{2eN}{d-2}\right)^{(d-1)/2}.$$

□

Injecting the result of Lemma B.6 in the regret bound of Theorem 4.1 directly gives Corollary 4.2.

B.1.3 Proof of Theorem 4.3

We first show the following result.

Proposition B.8. *Consider $n \geq 1$ points $p_1, \ldots, p_n$ sampled independently at random from a distribution ν supported on an interval $P \subseteq \mathbb{R}$ with continuous cumulative distribution function. Sort the p_i in increasing order as $p_{(1)} \leq \cdots \leq p_{(n)}$ and denote $P_n = [p_{(1)}, p_{(n)}]$ their convex hull. We have*

$$\mathbb{E}[\nu(P \setminus P_n)] = \frac{2}{n+1}.$$

Proof of Proposition B.8. Consider F_P the cumulative distribution function (c.d.f.) of p_i. Since F_P is continuous, the $q_i := F_P(p_i)$ are independent and uniformly distributed on $[0,1]$. Since F_P is nondecreasing, we also have $q_{(1)} \leq \cdots \leq q_{(n)}$. Letting $Q_n = [q_{(1)}, q_{(n)}]$ the convex hull of $\{q_i\}_{i=1}^n$, we thus have $F_P(P_n) = Q_n$ (the c.d.f. preserves convex hulls in dimension one). Since P is an interval, $P_n \subseteq P$, and

$$\begin{aligned}\nu(P \setminus P_n) &= \nu(P) - \nu(P_n) \\ &= 1 - (F_P(p_{(n)}) - F_P(p_{(1)})) \\ &= 1 - (q_{(n)} - q_{(1)}).\end{aligned}$$

The order statistics of uniform random variables on $[0,1]$ are well-known: $\mathbb{E}[q_{(i)}] = \frac{i}{n+1}$, consequently $1 - \mathbb{E}[q_{(n)} - q_{(1)}] = 1 - \frac{n-1}{n+1} = \frac{2}{n+1}$. □

We can now apply Proposition B.8 to $\nu = \mu_{\mathcal{C}_i}$ for each $i \in [N]$ to derive the regret bound of Theorem 4.3.

Proof of Theorem 4.3. Recall that $R_{\texttt{CHC}}(T) = 2\sum_{t=1}^{T}\sum_{i=1}^{N} \mathbb{E}[\mu(\mathcal{C}_i \setminus \hat{\mathcal{C}}_{i,t})]$. Let $n_i(t) \sim \mathrm{Bin}(t, \mu(\mathcal{C}_i))$ the number of points that landed in $\mathcal{C}_i$ up to time t. Conditionally to $n_i(t) = n$, $\hat{\mathcal{C}}_{i,t}$ is distributed as the convex hull Q_n of n points sampled at random within $\mathcal{C}_i$ with density $f_{\mathcal{C}_i}(x) = f_\mu(x)/\mu(\mathcal{C}_i)$. Thus $\mathbb{E}[\mu(\mathcal{C}_i \setminus \hat{\mathcal{C}}_{i,t}) | n_i(t)] = \mathbb{E}[\mu(\mathcal{C}_i \setminus Q_{n_i(t)}) | n_i(t)] = \mu(\mathcal{C}_i)\mathbb{E}[\mu_{\mathcal{C}_i}(\mathcal{C}_i \setminus Q_{n_i(t)}) | n_i(t)]$. Applying Proposition B.8 and noting that for $n = 0$, $2/(n+1)$ bounds from above the L.H.S. of Proposition B.8, we obtain

$$\mathbb{E}[\mu(\mathcal{C}_i \setminus \hat{\mathcal{C}}_{i,t}) | n_i(t)] \leq \frac{2}{n_i(t)+1}\mu(\mathcal{C}_i).$$

This gives $R_{\texttt{CHC}}(T) \leq 4\sum_{t=1}^{T}\sum_{i=1}^{N}\mathbb{E}[\frac{1}{n_i(t)+1}]\mu(\mathcal{C}_i)$. If $X \sim \mathrm{Bin}(n,p)$,

$$\begin{aligned}
\mathbb{E}\left[\frac{1}{X+1}\right] &= \sum_{k=0}^{n}\frac{1}{k+1}\binom{n}{k}p^k(1-p)^{n-k} \\
&= \frac{1}{(n+1)p}\sum_{k=0}^{n}\binom{n+1}{k+1}p^{k+1}(1-p)^{n-k} \\
&= \frac{1}{(n+1)p}\sum_{k=1}^{n+1}\binom{n+1}{k}p^{k}(1-p)^{n+1-k} \\
&= \frac{1}{(n+1)p}(1-(1-p)^{n+1}) \\
&\leq \frac{1}{(n+1)p}
\end{aligned}$$

thus $\mathbb{E}[\frac{1}{n_i(t)+1}] \leq \frac{1}{(t+1)\mu(\mathcal{C}_i)}$. This yields a regret of

$$\begin{aligned}
R_{\texttt{CHC}}(T) &\leq 2(\beta-\alpha)\sum_{t=1}^{T}\frac{1}{t+1}\sum_{i=1}^{N}\frac{\mu(\mathcal{C}_i)}{\mu(\mathcal{C}_i)} \\
&\leq 2(\beta-\alpha)N\log(T+1)\,.
\end{aligned}$$

□

Remark B.9. By carefully inspecting the proof, one can notice that

$$\begin{aligned}
\mathbb{E}\left[\mu(\mathcal{C}_i \setminus \hat{\mathcal{C}}_{i,t})\right] &= \mathbb{E}\left[\mathbb{E}\left[\mu(\mathcal{C}_i \setminus \hat{\mathcal{C}}_{i,t})|n_i(t)\right]\right] \\
&= \mathbb{E}\left[\mathbb{1}\{n_i(t)=0\} + \frac{2}{n_i(t)+1}\mathbb{1}\{n_i(t)\neq 0\}\right] \\
&= (1-\mu(\mathcal{C}_i))^t + 2\left[\frac{(1-(1-\mu(\mathcal{C}_i))^{t+1}}{(t+1)\mu(\mathcal{C}_i)} - (1-\mu(\mathcal{C}_i))^t\right] \\
&= \frac{2(1-(1-\mu(\mathcal{C}_i))^{t+1}}{(t+1)\mu(\mathcal{C}_i)} - (1-\mu(\mathcal{C}_i))^t.
\end{aligned}$$

An *exact* expression of the regret of CHC can then be obtained:

$$\begin{aligned}
R_{\texttt{CHC}}(T) &= (\beta-\alpha)\sum_{i=1}^{N}\sum_{t=1}^{T}\left[\frac{2(1-(1-\mu(\mathcal{C}_i))^{t+1})}{t+1} - \mu(\mathcal{C}_i)(1-\mu(\mathcal{C}_i))^t\right] \\
&= (\beta-\alpha)\left(2N\left(\sum_{t=1}^{T}\frac{1}{t+1}\right) - 2\sum_{i=1}^{N}\sum_{t=1}^{T}\frac{(1-\mu(\mathcal{C}_i))^{t+1}}{t+1} - \sum_{i=1}^{N}(1-\mu(\mathcal{C}_i)(1-(1-\mu(\mathcal{C}_i))^T)\right) \\
&= (\beta-\alpha)\left(2N\left(\sum_{t=1}^{T}\frac{1}{t+1}\right) - 2\sum_{i=1}^{N}\sum_{t=1}^{T}\frac{(1-\mu(\mathcal{C}_i))^{t+1}}{t+1} - \left[N-1-\sum_{i=1}^{N}(1-\mu(\mathcal{C}_i)^{T+1}\right]\right). \qquad (4)
\end{aligned}$$

B.1.4 Proof of Theorem 4.4

The proof strategy of Proposition 4.4 is to first derive a regret lower bound on the Bayesian regret for an adequately chosen prior on the set of N seeds $\theta = (s_1, \ldots, s_N) \in [0,1]^N$. The minimax regret bound then follows as an immediate corollary. First, note that given θ, the correct label to $q_t \in [0,1]$ is the index of the closest seed:

$$i_t(\theta) = \arg\min_{i\in[N]} |q_t - s_i|.$$

At any time t, the learner may ask the expert and obtain the correct label $i_t(\theta)$. Let $e_t = 1$ if the expert is consulted, and $e_t = 0$ otherwise. We define $v_t = e_t i_t(\theta)$ the expert output at time t. The learner attempts to

guess the label $i_t(\theta)$ with some estimate $\hat{\imath}_t$ using the past observations. We use the notation $v^t = (v_1, ..., v_t)$, $e^t = (e_1, ..., e_t)$ and $q^t = (q_1, ..., q_t)$.

An algorithm is therefore comprised of two components:

(i) Estimation: $\hat{\imath}_t$ which can be an arbitrary function of q_t, v^t

(ii) Consultation: e_t which can be an arbitrary function of q_t, v^{t-1}

Optimality of MAP estimation. Consider θ drawn according to a prior distribution with density p_θ, and define the Bayesian regret

$$\bar{R}(T) := \int_{[0,1]^N} R(T, \theta) p_\theta(\theta) d\theta = (\beta - \gamma) \sum_{t=1}^{T} \mathbb{P}(\hat{\imath}_t \neq i_t(\theta)) + (\beta - \alpha) \sum_{t=1}^{T} \mathbb{E}(e_t)$$

where we emphasize that θ is random and drawn according to p_θ, and we omit the subscript π in the regret notation for visual clarity.

Proposition B.10. *The Maximum A Posteriori (MAP) estimator*

$$\hat{\imath}_t \in \arg\max_{i \in [N]} \mathbb{P}(i_t(\theta) = i | q_t, v^{t-1})$$

minimizes $\bar{R}(T)$.

Proof of Proposition B.10. Let us prove that there exists an optimal Bayesian algorithm such that $\hat{\imath}_t$ is the MAP estimator:

$$\hat{\imath}_t \in \arg\max_{i \in [N]} \mathbb{P}(i_t(\theta) = i | q_t, v^{t-1})$$

Now consider an arbitrary algorithm, its Bayesian regret is

$$\begin{aligned}
\bar{R}(T) &= (\beta - \gamma) \sum_{t=1}^{T} \left[1 - \sum_{i=1}^{N} \mathbb{P}(\hat{\imath}_t = i, i_t(\theta) = i) \right] + (\beta - \alpha) \sum_{t=1}^{T} \mathbb{E}(e_t) \\
&= (\beta - \gamma) \sum_{t=1}^{T} \left[1 - \sum_{i=1}^{N} \mathbb{E}\left(\mathbb{P}(\hat{\imath}_t = i, i_t(\theta) = i | q_t, v^{t-1}) \right) \right] + (\beta - \alpha) \sum_{t=1}^{T} \mathbb{E}(e_t) \\
&= (\beta - \gamma) \sum_{t=1}^{T} \left[1 - \sum_{i=1}^{N} \mathbb{E}\left(\mathbb{1}\{\hat{\imath}_t = i\} \mathbb{P}(i_t(\theta) = i | q_t, v^{t-1}) \right) \right] + (\beta - \alpha) \sum_{t=1}^{T} \mathbb{E}(e_t) \\
&\geq (\beta - \gamma) \sum_{t=1}^{T} \left[1 - \mathbb{E}\left(\max_{i \in [N]} \mathbb{P}(i_t(\theta) = i | q_t, v^{t-1}) \right) \right] + (\beta - \alpha) \sum_{t=1}^{T} \mathbb{E}(e_t)
\end{aligned}$$

where we used the tower property of expectations and the fact that $\hat{\imath}_t \in [N]$, and where the lower bound is attained if $\hat{\imath}_t$ is the MAP $\hat{\imath}_t \in \arg\max_{i \in [N]} \mathbb{P}(i_t(\theta) = i | q_t, v^{t-1})$. □

We have proven that, given any algorithm, we may modify it by replacing its estimation part by the MAP estimator and doing so always results in lower Bayesian regret. We emphasize that changing the estimation part while keeping the consultation part constant does not change q^t nor does it change v^t, so that it has no impact on the information available at time t in order to perform the estimation.

Some properties of the MAP. Note that if the expert is consulted ($e_t = 1$), then the MAP is simply $\hat{\imath}_t = i_t(\theta)$ since the correct label is available. Furthermore, if $q_t \in \text{conv}(\{q_{t_j} : e_{t_j} = 1, v_{t_j} = i\})$ then the MAP is $\hat{\imath}_t = i$, since the only possible label is i.

Fig. 5 illustrates the construction behind the lower bound in dimension one.

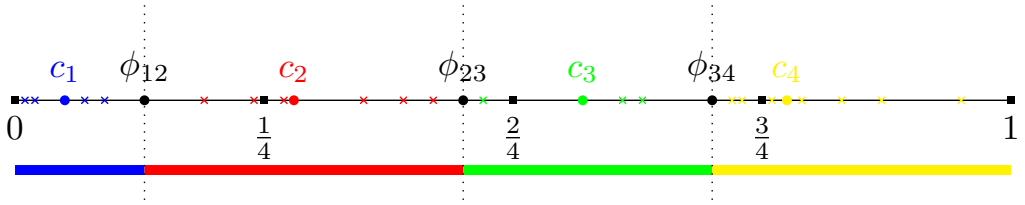


Figure 5: Lower bound construction in dimension one, $\phi_{ij} = (s_i + s_j)/2$

Proof of Theorem 4.4. Consider $2 \leq t \leq T$ (the theorem is trivial if $T = 1$) and the following prior on θ: $s_1, ..., s_N$ are drawn independently and s_i is uniformly distributed on the interval $I_i = [(i-1)/N, i/N]$. Either the expert is consulted, so that $e_t = 1$ and a regret of $\beta - \alpha$ is incurred, or a regret at least equal to $\beta - \gamma$ times the error rate of the MAP estimator is incurred. Since $\gamma \leq \alpha$, the instantaneous regret $1 - r(t)$ is lower bounded as:

$$\begin{aligned}
1 - r(t) &\geq (\beta - \gamma)\mathbb{P}(e_t = 0)\mathbb{P}(\hat{\imath}_t \neq i_t(\theta)|e_t = 0) + (\beta - \alpha)\mathbb{P}(e_t = 1) \\
&\geq (\beta - \alpha)\left(\mathbb{P}(e_t = 0)\mathbb{P}(\hat{\imath}_t \neq i_t(\theta)|e_t = 0) + 1 - \mathbb{P}(e_t = 0)\right) \\
&\geq (\beta - \alpha)\mathbb{P}(\hat{\imath}_t \neq i_t(\theta)|e_t = 0).
\end{aligned}$$

We now focus on the error rate of the MAP knowing that $e_t = 0$. Denote by $l_{i,t} = \min\{q_s : s \leq t, e_s = 1, v_s = i\}$ for $2 \leq i \leq N$ and $l_{1,t} = 0$. Similarly $r_{i,t} = \max\{q_s : e_s = 1, v_s = i\}$ for $1 \leq i \leq N-1$ and $r_{N,t} = 0$. Since $s_i \in I_i$ for all i we have $s_i \leq s_{i+1}$ for all i and it follows that $l_{i,t} \leq r_{i,t} \leq l_{i+1,t}$ for all i. If $q_t \in \cup_{i\in[N]}[l_{i,t}, r_{i,t}]$ then $\hat{\imath}_t = i_t(\theta)$ so that the correct label is guessed with probability one. If $q_t \notin \cup_{i\in[N]}[l_{i,t}, r_{i,t}]$ define i such that $r_i \leq q_t \leq l_{i+1}$ so that there are only two possible labels: namely i and $i+1$, and the learner must guess whether or not $q_t \leq \phi_{i,i+1} = (s_i + s_{i+1})/2$. Since s_i and s_{i+1} are uniformly distributed on I_i and I_{i+1} we have $\phi_{i,i+1} = (i + z - 1/2)/N$ in distribution where z follows the Irwin-Hall distribution with two parameters, i.e., $z \in [0, 2]$ is distributed as the sum of two i.i.d uniformly distributed random variables over $[0, 1]$ with density $\min(z, 2 - z)$. To ease notation we define $h(z) = \min(z, 2 - z)$ and $I_i(a, b) = \int_{Na-i+1/2}^{Nb-i+1/2} h(z)dz$.

Therefore

$$\begin{aligned}
\mathbb{P}(i_t(\theta) = i|q_t, v_{t-1}, e_t = 0) &= \mathbb{P}(q_t \leq \phi_{i,i+1}|q_t, v_{t-1}) \\
&= \mathbb{P}(q_t \leq \phi_{i,i+1}|\phi_{i,i+1} \in [r_{i,t}, l_{i+1,t}]) \\
&= \mathbb{P}(q_t \leq (i + z - 1/2)/N|(i + z - 1/2)/N \in [r_{i,t}, l_{i+1,t}]) \\
&= \mathbb{P}(Nq_t - i + 1/2 \leq z|z \in [Nr_{i,t} - i + 1/2, Nl_{i+1,t} - i + 1/2]) \\
&= \frac{I_i(q_t, l_{i+1,t})}{I_i(r_{i,t}, l_{i+1,t})}
\end{aligned}$$

and similarly

$$\mathbb{P}(i_t(\theta) = i - 1|q_t, v_{t-1}, e_t = 0) = \frac{I_i(r_{i,t}, q_t)}{I_i(r_{i,t}, l_{i+1,t})}$$

Furthermore, the error rate of the MAP knowing q_t is given by

$$\begin{aligned}
\mathbb{P}(\hat{\imath}_t \neq i_t(\theta)|q_t, v_{t-1}, e_t = 0) &= \min_{j\in\{i,i+1\}} \mathbb{P}(i_t(\theta) = j|q_t, v_{t-1}, e_t = 0) \\
&= \frac{\min(I_i(q_t, l_{i+1,t}), I_i(r_{i,t}, q_t))}{I_i(r_{i,t}, l_{i+1,t})}
\end{aligned}$$

and the error rate is found by integrating this function over $[0, 1]$ since q_t is drawn uniformly at random:

$$\mathbb{P}(\hat{\imath}_t \neq i_t(\theta)|e_t = 0) = \sum_{i=1}^{N-1} \int_0^1 \mathbb{E}\left(\frac{\min(I_i(q, l_{i+1,t}), I_i(r_{i,t}, q))}{I_i(r_{i,t}, l_{i+1,t})} \mathbb{1}(q \in [r_{i,t}, l_{i+1,t}])\mathrm{d}q\right).$$

While this expression can be computed in closed form, it is unwieldy, and we will now lower bound it. Define the median point $m_t \in [r_{i,t}, l_{i+1,t}]$ such that

$$I_i(m_t, l_{i+1,t}) = I_i(r_{i,t}, m_t) = \frac{1}{2} I_i(r_{i,t}, l_{i+1,t})$$

Consider $q \leq m_t$, we have

$$\begin{aligned}\min(I_i(q, l_{i+1,t}), I_i(r_{i,t}, q)) &= I_i(r_{i,t}, q) \\ &= I_i(r_{i,t}, m_t) - I_i(q, m_t) \\ &\geq \frac{1}{2} I_i(r_{i,t}, l_{i+1,t}) - 2|m_t - q|\end{aligned}$$

where we used the definition of m_t and the fact that $h(z) \leq 2$. Similarly let $q \geq m_t$, we have

$$\begin{aligned}\min(I_i(q, l_{i+1,t}), I_i(r_{i,t}, q)) &= I_i(q, l_{i+1,t}, q) \\ &= I_i(m_t, l_{i+1,t}) - I_i(m_t, q) \\ &\geq \frac{1}{2} I_i(r_{i,t}, l_{i+1,t}) - 2|m_t - q|\end{aligned}$$

Putting both cases together, and using the fact that the expression is always positive:

$$\frac{\min(I_i(q, l_{i+1,t}), I_i(r_{i,t}, q))}{I_i(r_{i,t}, l_{i+1,t})} \geq \max\left(0, \frac{1}{2} - \frac{2|m_t - q|}{I_i(r_{i,t}, l_{i+1,t})}\right).$$

Integrating this bound over q,

$$\begin{aligned}\int_0^1 \frac{\min(I_i(q, l_{i+1,t}), I_i(r_{i,t}, q))}{I_i(r_{i,t}, l_{i+1,t})} \mathbb{1}(q \in [r_{i,t}, l_{i+1,t}]) \mathrm{d}q &\geq \int_0^1 \max\left(0, \frac{1}{2} - \frac{2|m_t - q|}{I_i(r_{i,t}, l_{i+1,t})}\right) \mathrm{d}q \\ &= \frac{1}{4} I_i(r_{i,t}, l_{i+1,t})\end{aligned}$$

so the error rate is lower bounded by

$$\mathbb{P}(\hat{\imath}_t \neq i_t(\theta) | e_t = 0) \geq \frac{1}{4} \sum_{i=1}^{N-1} \mathbb{E}\left(I_i(r_{i,t}, l_{i+1,t})\right).$$

Consider $\epsilon_t = 1/t \leq 1/\sqrt{2}$ since $t \geq 2$ and consider the event $E_{i,t}$ that $\phi_{i,i+1} \in [(i - 1/2 + 1/\sqrt{2})/N, (i - 1/2 + 2 - 1/\sqrt{2})/N]$, and that $q_s \notin [\phi_{i,i+1} - \epsilon_t/2, \phi_{i,i+1} + \epsilon_t/2]$ for all $s \leq t$. This event has probability

$$\begin{aligned}\mathbb{P}(E_{i,t}) &= \int_{(i-1/2+1/\sqrt{2})/N}^{(i-1/2+2-1/\sqrt{2})/N} \mathbb{P}(q_s \notin [p - \epsilon_t/2, p + \epsilon_t/2] s \leq t | \phi_{i,i+1} = p) \mathbb{P}(\phi_{i,i+1} = p) dp \\ &= I(1/\sqrt{2}, 2 - 1/\sqrt{2})(1 - \epsilon_t)^t \\ &= \frac{1}{2}(1 - \epsilon_t)^t\end{aligned}$$

and when it occurs $r_{i,t} \leq \phi_{i,i+1} - \epsilon_t/2 \leq \phi_{i,i+1} - \epsilon_t/2 \leq l_{i+1,t}$ and in turn $1/\sqrt{8} \leq N r_{i,t} - (i - 1/2) \leq N l_{i+1,t} - (i - 1/2) \leq 2 - 1/\sqrt{8}$, which then implies that $I_i(r_{i,t}, l_{i+1,t} \geq (l_{i+1,t} - r_{i,t}) \min_{z \in [r_{i,t}, l_{i+1,t}]} h(z) \geq \epsilon_t/\sqrt{8}$. Hence taking expectations

$$\begin{aligned}\mathbb{E}\left(I_i(r_{i,t}, l_{i+1,t})\right) &\geq \frac{\epsilon_t}{2\sqrt{2}} \mathbb{P}(E_{i,t}) \\ &= \frac{1}{t 4\sqrt{2}} \left(1 - \frac{1}{t}\right)^t\end{aligned}$$

Putting everything together gives a lower bound on the Bayesian regret

$$\begin{aligned}\int_{[0,1]^N} R(T,\theta)p_\theta(\theta)d\theta &\geq (\beta-\alpha)\sum_{t=1}^{T}\mathbb{P}(\hat{\imath}_t \neq i_t(\theta)|e_t=0)\\ &\geq (\beta-\alpha)\frac{N-1}{16\sqrt{2}}\sum_{t=2}^{T}\frac{1}{t}\left(1-\frac{1}{t}\right)^t\\ &\geq (\beta-\alpha)\frac{N-1}{64\sqrt{2}}\log\left(\frac{T+1}{2}\right)\end{aligned}$$

where we used $(1-\frac{1}{t})^t \geq (1-\frac{1}{2})^2$ for $t \geq 2$ and $\sum_{t=2}^{T}\frac{1}{t} \geq \log\left(\frac{T+1}{2}\right)$ in the last inequality.

We have proven that, for any $T \geq 1$, the Bayesian regret of any algorithm is lower bounded by

$$(\beta-\alpha)\frac{N-1}{64\sqrt{2}}\log\left(\frac{T+1}{2}\right) = \Omega((N-1)\log T).$$

By corollary, the same bound holds for the minimax regret. □

Remark B.11 (Adversarial Setting). As mentioned in Section 3, any algorithm must suffer linear regret in an adversarial setting. consider dimension $d=1$, the interval $\mathcal{E}=[0,1]$ and $N=2$ labels. Denote by ϕ the boundary between the two labels so that $\mathcal{C}_1=[0,\phi]$ and $\mathcal{C}_2=[\phi,1]$, and the adversary selects the query points as follows:

$$q_t = \frac{l_t+r_t}{2},$$

obeying the following recursion:

(i) If $t=1$ then $[l_1,r_1]=[0,1]$

(ii) If $t\geq 2$ and $\phi \leq q_t$ then $[l_{t+1},r_{t+1}]=[l_t,q_t]$

(iii) If $t\geq 2$ and $\phi > q_t$ then $[l_{t+1},r_{t+1}]=[q_t,r_t]$

The query points can be seen as a binary search procedure over $[0,1]$ in an attempt to find ϕ. Then, for every time t:

(i) either the learner calls the expert, which incurs a cost of α

(ii) otherwise the learner must attempt to guess whether or not $\phi \leq q_t$ and the optimal guessing strategy is the MAP estimator, as explained in the proof of Theorem 4.4.

By construction, even in the most favorable case where the learner called the expert at all times from time 1 to time $t-1$, then the error rate of the MAP estimator is exactly $1/2$, because the a posteriori distribution of ϕ is uniform over $[l_t,r_t]$ and the a posteriori probability of $\phi \leq q_t$ and $\phi > q_t$ are equal, i.e., both labels are equiprobable. In other words, the adversary can always make sure that the MAP does not perform better than guessing uniformly at random. This incurs an average cost of guessing of $(\beta+\gamma)/2$.

It is noted that the adversary is oblivious, so that it does not adapt to the decisions selected by the learner, and the regret is indeed linear.

B.2 Proofs of Section 4.2

In this section, we present a more general version of Theorem B.13 along with its proof. We recall that the expert labeling policy is given by the Voronoi tessellation with seeds s_i, as described in Section 3.2. We show that in the regime $T \leq e^d$, if the queries are distributed as a subgaussian mixture with sufficiently separated centers, a simple Center-based Classifier (`CC`) can achieve $N\log N$ regret after T rounds when the mixture is homogeneous. `CC`'s general regret bound holds under the following condition, which implies 3.3(ii).

Assumption B.12. The distribution μ of the queries is given by the following mixture model: $i \in [N]$ is chosen with probability p_i, then q_t is sampled from $s_i + w_i$ where $s_i \in \mathbb{R}^d$ is the component center and w_i is subgaussian with parameter $\sigma > 0$: for all $\lambda \in \mathbb{R}^d$,

$$\mathbb{E}[\exp(\lambda^\top w_i)] \le \exp\left(\frac{\sigma^2\|\lambda\|_2^2}{2}\right).$$

Furthermore, the mixture centers are sufficiently separated:

$$\delta_{\min}^2 \ge c\sigma^2 d$$

where $\delta_{\min} = \min_{i\neq j}\|s_i - s_j\|_2$ and $c \ge 32$.

On $\mathcal{E} = \mathcal{I}^d$, Assumption B.12 is for instance verified if $w_i \in [-L, L]^d$ for some $L^2 \le \delta_{\min}^2/(cd)$. Note that we always have $\delta_{\min}^2 \le d$ on $\mathcal{I}^d$, so the latter requires $L^2 \le 1/c$.

Below, we state our general upper bound on CC's regret, from which Theorem B.13 directly follows.

Theorem B.13. *Under Assumption B.12, the regret of* CC *is bounded as*

$$R_{\texttt{CC}}(T) \le \frac{(\beta-\alpha)(\log N + 1)}{p_{\min}}\left(1 + \frac{192(d + 2\log T)}{cd}\right) + (2\beta - \gamma - \alpha)N + (\beta - \gamma)T(e^{-\frac{c-32}{48}d} + Te^{-\frac{c-8}{12}d}).$$

In particular, if $\delta_{\min}^2 \ge 80\sigma^2 d$ *and* $T \le e^d$,

$$R_{\texttt{CC}}(T) \le \frac{41}{5}\frac{(\beta-\alpha)(\log N + 1)}{p_{\min}} + 2(\beta - \gamma)(N + 1).$$

We denote by j_t the *generative* label of q_t, i.e., $q_t = s_{j_t} + w_{j_t}$. For the analysis, it is easier to work under the assumption that the expert provides the generative labels j_t instead of the Voronoi labels i_t. We will make use of the following lemma.

Lemma B.14. *Under assumption B.12, the event* $E := \{\forall t \in [T]\, i_t = j_t\}$ *occurs with probability* $\mathbb{P}(E) \ge 1 - Te^{-\frac{c-8}{12}d}$.

Proof. E^c occurs with probability

$$\begin{aligned}
\mathbb{P}(E^c) &\le \sum_{t=1}^T \mathbb{P}(\|q_t - s_{i_t}\|_2 \le \|q_t - s_{j_t}\|_2) \\
&\le \sum_{t=1}^T \mathbb{P}(\|w_{j_t} + s_{j_t} - s_{i_t}\|_2 \le \|w_{j_t}\|_2) \\
&\le \sum_{t=1}^T \mathbb{P}(\|w_{j_t}\|_2 \ge \delta_{\min}/2) \\
&\le \sum_{t=1}^T \sum_{j=1}^N \mathbb{P}(j_t = j)\mathbb{P}(\|w_j/\sigma\|_2^2 \ge \frac{\delta_{\min}^2}{4\sigma^2})
\end{aligned}$$

By Lemma C.2, for any $j \in [N]$ and $\eta > 0$,

$$\mathbb{P}(\|w_j/\sigma\|_2^2 \ge d + 2\sqrt{d\eta} + 2\eta) \le e^{-\eta}.$$

By the AM-GM inequality, $d + 2\sqrt{d\eta} + 2\eta \le d + (d + \eta) + 2\eta = 2d + 3\eta$. Hence

$$\mathbb{P}(\|w_j/\sigma\|_2^2 \ge 2d + 3\eta) \le e^{-\eta}.$$

We then pick η such that $2d + 3\eta = \frac{\delta_{\min}^2}{4\sigma^2}$. This requires $\delta_{\min}^2 \ge 8\sigma^2 d$ and gives $\eta = \frac{\delta_{\min}^2}{12\sigma^2} - \frac{2}{3}d$.

Finally, $\mathbb{P}(i_t \neq j_t) \le \exp\left(-(\frac{\delta_{\min}^2}{12\sigma^2} - \frac{2}{3}d)\right)$ so that $\mathbb{P}(E^c) \le Te^{-\frac{c-8}{12}d}$.

□

Generative Setting vs. Voronoi Setting This lemma ensures that for the sake of the analysis, we can assume that the ground-truth labels are the generative labels j_t. More precisely, denote by $\mathcal{R}_{\texttt{CC}}(T)$ the (random) regret of the CC algorithm, and by $\mathcal{R}'_{\texttt{CC}}(T)$ the random regret of the CC algorithm if the ground-truth labels (and those provided by the expert) were the generative labels j_t. Under the event E, the Voronoi and generative labels match, so that

$$\mathcal{R}_{\texttt{CC}}(T) = \mathcal{R}_{\texttt{CC}}(T)\mathbb{1}\{E\} + \mathcal{R}_{\texttt{CC}}(T)\mathbb{1}\{E^c\} \leq \mathcal{R}'_{\texttt{CC}}(T)\mathbb{1}\{E\} + (\beta-\gamma)T\mathbb{1}\{E^c\}.$$

This means that under Assumption B.12, the Voronoi regret of the CC algorithm is bounded as

$$R_{\texttt{CC}}(T) \leq \mathbb{E}[\mathcal{R}'_{\texttt{CC}}(T)] + (\beta-\gamma)T^2 e^{-\frac{c-8}{12}d}. \tag{5}$$

Note that the regret of CC in the generative setting can be expressed as

$$R'_{\texttt{CC}}(T) := \mathbb{E}[\mathcal{R}'_{\texttt{CC}}(T)] = (\beta-\alpha)\mathbb{E}[T_1] + (\beta-\gamma)\sum_{t=T_1+1}^{T}\mathbb{P}(\hat{\imath}_t \neq j_t)$$

The rest of the analysis focuses on bounding $\mathbb{E}[T_1]$ and $\mathbb{P}(\hat{\imath}_t \neq j_t)$ for $t > T_1$. We recall that $n_i(t) = \sum_{s=1}^{t}\mathbb{1}\{i_s = i\}$ denotes the number of queries with true label i up to round t. For the analysis, we use the definition $\hat{s}_i(t) = \frac{1}{n_i(t)}\sum_{s=1}^{t} q_s \mathbb{1}\{i_s = i\}$ for any $t \in [T]$. Since the expert is always called in the first phase of CC, this matches the definition of the estimator in Section 4.2, that was only defined for $t \leq T_1$.

Lemma B.15. *Let $\delta \in (0,1)$ and consider the CC algorithm whose first phase ends at $T_1 = \min\{t \leq T : \forall i \in [N]\, n_i(t) \geq \hat{m}(t)\}$ where $\hat{m}(t) := \frac{108\sigma^2}{\hat{\delta}^2_{\min}(t)}(d + \log(NT/\delta))$. Under the generative setting, the event*

$$A(\delta) := \{\forall i \in [N], \forall t \in [T]\ : n_i(t)\|s_i - \hat{s}_i(t)\|_2^2 \leq 3\sigma^2(d+\log(NT/\delta))\}$$

holds with probability larger than $1-\delta$. Additionally, we have under $A(\delta)$:

(i) If the first phase terminates, then for all $i \in [N]$, $\|s_i - \hat{s}_i(T_1)\|_2 \leq \frac{\delta_{\min}}{4}$

(ii) $T_1 \leq \tau_m := \min\{t \in \mathbb{N} : n_i(t) \geq m\}$

where $m := \frac{192\sigma^2(d+\log(NT/\delta))}{\delta^2_{\min}}$.

Proof of Lemma B.15. Let $t \in [T]$ and $i \in [N]$. Conditionally on $n_i(t) = n > 0$, $\hat{s}_i(t) - s_i$ is distributed as $\frac{1}{n}\sum_{j=1}^{n} x_{j,i}$ where the $x_{j,i}$ are independent and σ-subgaussian. Thus, $\frac{1}{n}\sum_{j=1}^{n} x_{j,i}$ is $(\sigma/\sqrt{n})$-subgaussian. Lemma C.2 gives, for any $\eta > 0$ and $n > 0$,

$$\begin{aligned}
&\mathbb{P}(n_i(t)\|s_i - \hat{s}_i(t)\|_2^2 \geq \sigma^2(d + 2\sqrt{d\eta} + 2\eta) \,|\, n_i(t) = n) \\
&= \mathbb{P}(\|\frac{1}{n}\sum_{j=1}^{n} x_{j,i}\|_2^2 \geq (\sigma^2/n)(d + 2\sqrt{d\eta} + 2\eta)) \\
&\leq e^{-\eta}.
\end{aligned}$$

and the same bound holds unconditionally by taking the expectation (if $n = 0$, the conditional bound trivially holds). By the AM-GM inequality, $d + 2\sqrt{d\eta} + 2\eta \leq d + (d+\eta) + 2\eta \leq 3(d+\eta)$. Thus, for $\eta = \log(NT/\delta) > 0$ we have, with probability larger than $1 - \frac{\delta}{NT}$,

$$n_i(t)\|s_i - \hat{s}_i(t)\|_2^2 \leq 3\sigma^2(d + \log(NT/\delta)).$$

A union bound over $i \in [N]$ and $t \in [T]$ yields that the event $A(\delta)$ holds with probability $1-\delta$.

(i) Under $A(\delta)$, if $T_1 < T$ then $n_i(T_1) \geq \hat{m}(T_1)$ for all $i \in [N]$ by definition of T_1. This implies that for all $i \in [N]$, $\sqrt{\hat{m}(T_1)}\|s_i - \hat{s}_i(T_1)\|_2 \leq \sigma\sqrt{3(d + \log(NT/\delta))}$, i.e.,

$$\|s_i - \hat{s}_i(T_1)\|_2 \leq \frac{\hat{\delta}_{\min}(T_1)}{6}.$$

Additionally, under this event,

$$|\|s_i - s_j\|_2 - \|\hat{s}_i(T_1) - \hat{s}_j(T_1)\|_2| \leq \|s_i - \hat{s}_i(T_1)\|_2 + \|s_j - \hat{s}_j(T_1)\|_2 \leq 2\frac{1}{6}\hat{\delta}_{\min}(T_1).$$

This implies that

$$\|\hat{s}_i(T_1) - \hat{s}_j(T_1)\|_2 \leq \|s_i - s_j\|_2 + \frac{1}{3}\hat{\delta}_{\min}(T_1),$$

and by taking the minimum on the right hand side,

$$\|\hat{s}_i(T_1) - \hat{s}_j(T_1)\|_2 \leq \delta_{\min} + \frac{1}{3}\hat{\delta}_{\min}(T_1)$$

for some i, j. Consequently, $\hat{\delta}_{\min}(T_1) \leq \delta_{\min} + \frac{1}{3}\hat{\delta}_{\min}(T_1)$ which finally gives

$$\|s_i - \hat{s}_i(T_1)\|_2 \leq \frac{1}{6}\hat{\delta}_{\min}(T_1) \leq \frac{1}{4}\delta_{\min}.$$

(ii) under $A(\delta)$, either $\tau_m \geq T$ so that $T_1 \leq T \leq \tau_m$, or $\tau_m < T$ and we have $n_i(\tau_m) \geq m$ for all $i \in [N]$, hence $\max_{i\in[N]} \|s_i - \hat{s}_i(\tau_m)\|_2 \leq \sigma\sqrt{\frac{3(d+\log(NT/\delta))}{m}} = \delta_{\min}/8$. Thus, $\hat{\delta}_{\min}(\tau_m) \geq \delta_{\min} - 2(\delta_{\min}/8) = 3\delta_{\min}/4$, i.e., $\hat{m}(\tau_m) \leq m$. Therefore, $n_i(\tau_m) \geq m \geq \hat{m}(\tau_m)$ so the stopping condition is verified at τ_m, which yields $T_1 \leq \tau_m$.

□

B.2.1 Bounding the First Phase Regret

Proposition B.16. *Under the generative setting, the first phase regret is bounded as*

$$(\beta - \alpha)\mathbb{E}[T_1] \leq \frac{(\beta - \alpha)(\log N + 1)}{p_{\min}}\left(1 + \frac{192\sigma^2(d + \log(NT/\delta))}{\delta_{\min}^2}\right) + (\beta - \alpha)T\delta.$$

Proof of Proposition B.16. By Lemma B.15, under $A(\delta)$, we have $T_1 \leq \tau_m$. In general, $T_1 \leq T$. Thus

$$\begin{aligned} T_1 &= T_1\mathbb{1}\{A(\delta)\} + T_1\mathbb{1}\{A(\delta)^c\} \\ &\leq \tau_m + T\mathbb{1}\{A(\delta)^c\}. \end{aligned}$$

As $A(\delta)$ holds with probability larger than $1 - \delta$, this yields

$$\mathbb{E}[T_1] \leq \mathbb{E}[\tau_m] + T\delta.$$

To bound $\mathbb{E}[\tau_m]$, first note that it is trivially upper bounded by $\lceil m \rceil \mathbb{E}[\tau_1]$ where $\tau_1 = \min\{t \in \mathbb{N} : n_i(t) \geq 1\}$ is the number of rounds required to observe each label once. Computing $\mathbb{E}[\tau_1]$ is known as a coupon-collector problem. Let $\tau_{1,i}$ denote the time to discover a i-th new label after $i - 1$ are already found. Then $\tau_1 = \sum_{i=1}^N \tau_{1,i}$. The probability of observing a given label is bounded below by $p_{\min} = \min_{i\in[N]} p_i$. Thus the probability of observing a i-th new label is $q_i \geq (N-(i-1))p_{\min}$. $\tau_{1,i}$ is geometric with success probability q_i, so $\mathbb{E}[\tau_{1,i}] = \frac{1}{q_i} \leq \frac{1}{(N-(i-1))p_{\min}}$. Consequently,

$$\begin{aligned} \mathbb{E}[\tau_1] &\leq \frac{1}{p_{\min}}\sum_{i=1}^N \frac{1}{N - (i-1)} \\ &= \frac{\sum_{i=1}^N 1/i}{p_{\min}} \\ &\leq \frac{\log N + 1}{p_{\min}}. \end{aligned}$$

This entails that $\mathbb{E}[\tau_m] \leq \lceil m \rceil \frac{\log N+1}{p_{\min}}$, so that

$$\mathbb{E}[T_1] \leq (\lceil m \rceil / p_{\min})(\log N + 1) + T\delta.$$

This yields

$$\begin{aligned}(\beta-\alpha)\mathbb{E}[T_1] &\leq (\beta-\alpha)(\lceil m \rceil/p_{\min})(1+\log N) + (\beta-\alpha)T\delta \\ &\leq \frac{(\beta-\alpha)(\log N+1)}{p_{\min}}\left(1 + \frac{192\sigma^2(d+\log(NT/\delta))}{\delta_{\min}^2}\right) + (\beta-\alpha)T\delta.\end{aligned}$$

□

B.2.2 Bounding the Second Phase Regret

Proposition B.17. *Let $\delta \in (0,1)$. Under the generative setting, for $t > T_1$, the probability of making a wrong guess is bounded as*

$$\mathbb{P}(\hat{\imath}_t \neq j_t) \leq e^{-\frac{c-32}{48}d} + \delta.$$

Consequently, the second phase regret is bounded as

$$(\beta-\gamma)\sum_{t=T_1+1}^{T} \mathbb{P}(\hat{\imath}_t \neq j_t) \leq (\beta-\gamma)T\left[e^{-\frac{c-32}{48}d} + \delta\right].$$

Proof of Proposition B.17. We first write $\mathbb{P}(\hat{\imath}_t \neq j_t) = \mathbb{P}(\hat{\imath}_t \neq j_t | A(\delta))\mathbb{P}(A(\delta)) + \mathbb{P}(\hat{\imath}_t \neq j_t | A(\delta)^c)\mathbb{P}(A(\delta)^c) \leq \mathbb{P}(\hat{\imath}_t \neq j_t | A(\delta)) + \delta$. Let $e_i := s_i - \hat{s}_i(T_1)$. By Lemma B.15, under $A(\delta)$, we have $\|e_i\|_2 \leq \delta_{\min}/4$ for all $i \in [N]$. We show that if $\|w_{j_t}\|_2 < \delta_{\min}/4$, we must have $\hat{\imath}_t = j_t$. First, note that for any $j \neq j_t$,

$$\|q_t - s_j\|_2 = \|w_{j_t} + s_{j_t} - s_j\|_2 \geq \|s_{j_t} - s_j\|_2 - \|w_{j_t}\|_2 \geq \delta_{\min} - \delta_{\min}/4 \geq 3\delta_{\min}/4 > \|w_{j_t}\|_2.$$

In particular, $j_t = i_t$. Note further that for any $j \neq j_t$,

$$\|q_t - \hat{s}_j\|_2 = \|q_t - s_j + e_j\|_2 \geq \|q_t - s_j\|_2 - \|e_j\|_2 \geq 3\delta_{\min}/4 - \delta_{\min}/4 = \delta_{\min}/2.$$

Additionally,

$$\|q_t - \hat{s}_{j_t}\|_2 = \|w_{j_t} + e_{j_t}\|_2 \leq \delta_{\min}/4 + \delta_{\min}/4 = \delta_{\min}/2,$$

so for any $j \in [N]$,

$$\|q_t - \hat{s}_j\|_2 \geq \|q_t - \hat{s}_{j_t}\|_2.$$

This yields $\hat{\imath}_t = j_t = i_t$. Consequently,

$$\mathbb{P}(\hat{\imath}_t \neq j_t | A(\delta)) \leq \mathbb{P}(\|w_{j_t}\|_2 \geq \delta_{\min}/4 | A(\delta)) = \mathbb{P}(\|w_{j_t}\|_2 \geq \delta_{\min}/4)$$

since q_t is independent from q_s for $s < t$. Hence

$$\mathbb{P}(\hat{\imath}_t \neq j_t | A(\delta)) \leq \sum_{j=1}^{N} \mathbb{P}(j_t = j)\mathbb{P}(\|w_j/\sigma\|_2^2 \geq \frac{\delta_{\min}^2}{16\sigma^2}).$$

By Lemma C.2, for any $j \in [N]$ and $\eta > 0$,

$$\mathbb{P}(\|w_j/\sigma\|_2^2 \geq 2d + 3\eta) \leq e^{-\eta}.$$

We then pick η such that $2d + 3\eta = \frac{\delta_{\min}^2}{16\sigma^2}$. This requires $\delta_{\min}^2 \geq 32\sigma^2 d$ and gives $\eta = \frac{\delta_{\min}^2}{48\sigma^2} - \frac{2}{3}d$.

Thus, $\mathbb{P}(\hat{\imath}_t \neq j_t | A(\delta)) \leq \exp\left(-(\frac{\delta_{\min}^2}{48\sigma^2} - \frac{2}{3}d)\right)$. Using $\delta_{\min}^2 \geq c\sigma^2 d$, this finally yields

$$\mathbb{P}(\hat{\imath}_t \neq j_t) \leq e^{-\frac{c-32}{48}d} + \delta.$$

□

We can now derive the full regret bound of the CC algorithm.

Proof of Theorem B.13. Combining Propositions B.16 and B.17 and choosing $\delta = N/T$, we obtain (under the generative setting)

$$R'_{\mathtt{CC}}(T) \leq \frac{(\beta-\alpha)(\log N+1)}{p_{\min}}\left(1+\frac{192\sigma^2(d+2\log T)}{\delta_{\min}^2}\right)+(2\beta-\gamma-\alpha)N+(\beta-\gamma)Te^{-\frac{c-32}{48}d}.$$

Using $\delta_{\min}^2 \geq c\sigma^2 d$ and combining the previous inequality with (5), we get under the Voronoi setting

$$R_{\mathtt{CC}}(T) \leq \frac{(\beta-\alpha)(\log N+1)}{p_{\min}}\left(1+\frac{192(d+2\log T)}{cd}\right)+(2\beta-\gamma-\alpha)N+(\beta-\gamma)Te^{-\frac{c-32}{48}d}+(\beta-\gamma)T^2e^{-\frac{c-8}{12}d}.$$

In particular, if $\delta_{\min}^2 \geq 80\sigma^2 d$, this implies

$$R_{\mathtt{CC}}(T) \leq \frac{(\beta-\alpha)(\log N+1)}{p_{\min}}\left(1+\frac{192(d+2\log T)}{80d}\right)+(2\beta-\gamma-\alpha)N+(\beta-\gamma)Te^{-d}+(\beta-\gamma)T^2e^{-6d}$$

When we additionally have $T \leq e^d$,

$$\begin{aligned} R_{\mathtt{CC}}(T) &\leq \frac{41}{5}\frac{(\beta-\alpha)(\log N+1)}{p_{\min}}+(2\beta-\gamma-\alpha)N+(\beta-\gamma)(1+e^{-4d}) \\ &\leq \frac{41}{5}\frac{(\beta-\alpha)(\log N+1)}{p_{\min}}+2(\beta-\gamma)(N+1). \end{aligned}$$

□

Remark B.18. In the proof above, under the condition $\delta_{\min}^2 \geq 80\sigma^2 d$, we assumed $T \leq e^d$ to obtain a simple bound that scales with $(\log N)/p_{\min}$. We remark that weaker conditions on T can still yield interesting bounds. For instance, $T \leq e^d\sqrt{(\log N)/p_{\min}}$ yields a bound that scales with $\frac{\log N}{p_{\min}}(1+\frac{\log((\log N)/p_{\min})}{d})$:

$$\begin{aligned} R_{\mathtt{CC}}(T) &\leq \frac{(\beta-\alpha)(\log N+1)}{p_{\min}}\left(\frac{41}{5}+\frac{12}{5}\frac{\log((\log N)/p_{\min})}{d}\right)+(2\beta-\gamma-\alpha)N+(\beta-\gamma)(1+e^{-4d})(\log N)/p_{\min} \\ &\leq \frac{(\beta-\alpha)(\log N+1)}{p_{\min}}\left(\frac{41}{5}+\frac{12}{5}\frac{\log((\log N)/p_{\min})}{d}\right)+2(\beta-\gamma)(N+(\log N)/p_{\min}). \end{aligned}$$

Remark B.19. In the proof of Proposition B.16, we used the simple upper bound $\mathbb{E}[\tau_1] \leq (\log N+1)/p_{\min}$. $\mathbb{E}[\tau_1]$ can in fact be computed exactly as $I := \int_0^\infty \left(1-\prod_{i=1}^N(1-e^{-p_i t})\right)\mathrm{d}t$ (see example 5.17 in Ross (2010)). Using this formula would yield a sharper but less readable regret bound that scales with I instead of $(\log N)/p_{\min}$ in the $T \leq e^d$ regime. We also note that an alternative bound is $I \leq \sum_{i=1}^N 1/p_i$, which can be sharper than $I \leq (\log N+1)/p_{\min}$ depending on the value of the weights p_i.

B.3 Proofs of Section 4.3

In this section, we show that the Euclidean projection on the spherical convex hull of a set of points $\{p_i\}_{i=1}^n$ is the normalized projection on the positive hull of $\{p_i\}_{i=1}^n$ if this projection is non-zero, and is one of the p_i otherwise.

Proposition B.20. *Let $\mathcal{H}_n$ be the positive hull generated by points $p_1,\dots,p_n \in \mathcal{S}^{d-1}$, i.e., $\mathcal{H}_n = \{\sum_{i=1}^n \alpha_i p_i, \alpha \geq 0\}$, and let $P_n = \mathcal{S}^{d-1} \cap \mathcal{H}_n$ their spherical convex hull. For any $q \in \mathcal{S}^{d-1}$:*

(i) If $\mathrm{proj}_n := \arg\min_{x\in\mathcal{H}_n}\|q-x\| \neq 0$, *then* $\arg\max_{x\in P_n} q^\top x = \dfrac{\mathrm{proj}_n}{\|\,\mathrm{proj}_n\,\|}$ *and*

$$d(q,P_n) = \sqrt{2-2\|\,\mathrm{proj}_n\,\|}.$$

(ii) If $\mathrm{proj}_n = 0$, *then* $\max_{x \in P_n} q^\top x = \max_{1 \le i \le n} q^\top p_i$ *and*

$$d(q, P_n) = \sqrt{2 - 2 \max_{1 \le i \le n} q^\top p_i}.$$

Proof of Proposition B.20. First note that for any $x \in \mathcal{S}^{d-1}$, $\|q - x\|_2^2 = 2 - 2q^\top x$.. Thus, computing

$$\min_{\alpha \in \mathbb{R}^n} \|q - \sum_{i=1}^n \alpha_i p_i\|_2^2 \quad \text{subject to} \quad \alpha_i \ge 0, \|\sum_{i=1}^n \alpha_i p_i\|_2 = 1$$

amounts to computing

$$\max_{\alpha \in \mathbb{R}^n} q^\top (\sum_{i=1}^n \alpha_i p_i) \quad \text{subject to} \quad \alpha_i \ge 0, \|\sum_{i=1}^n \alpha_i p_i\|_2 = 1.$$

Since $\mathcal{H}_n$ is a closed convex set, by the closest point property, $\forall x \in \mathcal{H}_n$, $(q - \mathrm{proj}_n)^\top (x - \mathrm{proj}_n) \le 0$. Applying this to $x = 0 \in \mathcal{H}_n$ and $x = 2\,\mathrm{proj}_n \in \mathcal{H}_n$, we obtain $(q - \mathrm{proj}_n)^\top \mathrm{proj}_n = 0$, i.e., $q^\top \mathrm{proj}_n = \| \mathrm{proj}_n \|^2$.
First assume that $\mathrm{proj}_n \neq 0$ and let $x \in P_n$. From $(q - \mathrm{proj}_n)^\top (x - \mathrm{proj}_n) \le 0$, $q^\top \mathrm{proj}_n = \| \mathrm{proj}_n \|^2$ and Cauchy-Schwartz, we get

$$q^\top x \le \mathrm{proj}_n^\top x \le \| \mathrm{proj}_n \|.$$

Conversely, this upper bound is attained for $x^* = \frac{\mathrm{proj}_n}{\| \mathrm{proj}_n \|} \in P_n$, as

$$q^\top x^* = \frac{q^\top \mathrm{proj}_n}{\| \mathrm{proj}_n \|} = \| \mathrm{proj}_n \|.$$

Thus, if $\mathrm{proj}_n \neq 0$, $d(q, P_n) = \|q - \frac{\mathrm{proj}_n}{\| \mathrm{proj}_n \|}\| = \sqrt{2 - 2\frac{q^\top \mathrm{proj}_n}{\| \mathrm{proj}_n \|}} = \sqrt{2 - 2\| \mathrm{proj}_n \|}$.

If $\mathrm{proj}_n = 0$, the closest point property gives $q^\top x \le 0$ for all $x \in \mathcal{H}_n$. Let $x = \sum_{i=1}^n \alpha_i p_i \in P_n$. Then

$$q^\top x \le \sum_{i=1}^n \alpha_i (q^\top p_i) \le M \sum_{i=1}^n \alpha_i \le M$$

because $1 = \| \sum_{i=1}^n \alpha_i p_i \| \le \sum_{i=1}^n \alpha_i \|p_i\| = \sum_{i=1}^n \alpha_i$, and $M \le 0$. This is obviously attained by one of the $p_i \in P_n$. This means that $\max_{x \in P_n} q^\top x = \max_{1 \le i \le n} q^\top p_i$, so that

$$d(q, P_n) = \sqrt{2 - 2 \max_{1 \le i \le n} q^\top p_i} \ge \sqrt{2}.$$

□

C CONCENTRATION TOOLS

In this section, we provide concentration tools that are useful for our analysis. First, we present a lemma that controls the expectation of specific functions of a binomial random variable, which is useful for handling the randomness in the number of points landing per cell prior to applying our intermediary results on random polytopes, that are stated for a deterministic number of points (see e.g. Corollary B.3). In the statement below, $\mathcal{O}(\cdot)$ is asymptotic in n and hides dependencies in d and p.

Lemma C.1. *Let* $X_n \sim \mathrm{Bin}(n, p)$ *for* $p \in [0, 1]$. *For*

(i) $\varphi_1 : x \mapsto \frac{\log^{d-1} x}{x}$ *with* $\varphi_1(0) = 0$

(ii) $\varphi_2 : x \mapsto \frac{\log^{d-2} x \log\log x}{x}$ *with* $\varphi_2(0) = \alpha \in \mathbb{R}, \varphi_2(1) = 0$

we have

$$\mathbb{E}[\varphi_i(X_n)] \le \varphi_i(\mathbb{E}(X_n)) + \mathcal{O}\left(\frac{1}{n}\right).$$

Proof of Lemma C.1. Let $p \in (0,1)$ (otherwise X_n is constant almost surely and the result follows). Since $\mathbb{E}[X_n] = np$, we write $\varphi_1(X_n) = \varphi_1(X_n)\mathbb{1}\{|X_n - np| < np/2\} + \varphi_1(X_n)\mathbb{1}\{|X_n - np| \geq np/2\}$. From the multiplicative Chernoff bound, we know that $\mathbb{P}(|X_n - np| \geq np/2) \leq 2e^{-np/12}$. For $x \geq 1$, φ_1 is maximized at $x = e^{d-1}$, for which it is equal to $(\frac{d-1}{e})^{d-1}$, and we have

$$\mathbb{E}[\varphi_1(X_n)\mathbb{1}\{|X_n - np| \geq np/2\}] \leq 2\left(\frac{d-1}{e}\right)^{d-1} e^{-np/12}.$$

On the other interval, namely $[np/2, 3np/2]$, we perform a Taylor approximation. Around $x = np$, we write $\varphi_1(x) = \varphi_1(np) + \varphi_1'(\xi)(x - np)$ where ξ is between x and np. φ_1' is given by $\varphi_1'(x) = \frac{(d - \log(x) - 1)\log^{d-2}(x)}{x^2}$. For n large enough $|\varphi_1'|$ is decreasing on $[np/2, 3np/2]$ and $|\varphi_1'(np/2)| \leq 4|d - \log(np/2) - 1|\frac{\log^{d-2}(np/2)}{(np)^2} \leq 4\frac{\log^{d-1}(np)}{(np)^2}$. Then

$$\begin{aligned}
\mathbb{E}[\varphi_1(X_n)\mathbb{1}\{|X_n - np| < np/2\}] &\leq \varphi_1(np) + |\varphi_1'(np/2)|\mathbb{E}[|X_n - np|] \\
&\leq \varphi_1(np) + 4\frac{\log^{d-1}(np)}{(np)^2}\sqrt{np(1-p)} \\
&\leq \varphi_1(np) + 4\sqrt{1-p}\frac{\log^{d-1}(np)}{(np)^{3/2}}.
\end{aligned}$$

Thus for n large enough,

$$\begin{aligned}
\mathbb{E}[\varphi_1(X_n)] &\leq \varphi_1(np) + 4\sqrt{1-p}\frac{\log^{d-1}(np)}{(np)^{3/2}} + 2(\frac{d-1}{e})^{d-1}e^{-np/12} \\
&= \varphi_1(np) + \mathcal{O}(1/n).
\end{aligned}$$

For the function φ_2, adapting the previous arguments gives $\mathbb{E}[\varphi_2(X_n)\mathbb{1}\{|X_n - np| \geq np/2\}] \leq Me^{-np/12} + \varphi_2(0)\mathbb{P}(X_n = 0) = Me^{-np/12} + \alpha(1-p)^n$ where M is the maximum of φ_2 for $x \geq 2$, and for n large enough, $\mathbb{E}[\varphi_2(X_n)\mathbb{1}\{|X_n - np| < np/2\}] \leq \varphi_2(np) + 4\frac{\log^{d-3}(np)}{(np)^2}[\log(\log(np))\log(np) + 1]\sqrt{np(1-p)}$. This gives

$$\mathbb{E}[\varphi_2(X_n)] \leq \varphi_2(np) + \mathcal{O}(1/n).$$

□

The following subgaussian concentration inequality will also be useful for the analysis of `CC`.

Lemma C.2 (Theorem 2.1 in Hsu et al. (2012))**.** *if X is σ-subgaussian, then for any $\lambda > 0$,*

$$\mathbb{P}(\|AX\|_2^2 \geq \sigma^2(\mathrm{tr}(\Sigma) + 2\sqrt{\mathrm{tr}(\Sigma^2)\lambda} + 2\|\Sigma\|_{op}\lambda)) \leq e^{-\lambda}$$

for $\Sigma = A^\top A$.

D ADDITIONAL RELATED WORK

In this section, we provide further context on areas central to our theoretical analysis and experimental methodology, complementing Section 2.

Convex Geometry Our theoretical analysis relies on results regarding the approximation of convex bodies by random polytopes. We refer to the recent survey of Prochno et al. (2022), and to the books of Ziegler (1998) and Ratcliffe (2019) for general background on Euclidean and spherical geometry.

Embedding-based Retrieval Our problem formulation (Section 3) assumes queries are represented by embeddings, and our real-world experiments (Section 5, Appendix F.2.3) utilize state-of-the-art embedding models derived from LLMs. Such embeddings are foundational to many modern Natural Language Processing (NLP) applications: for example, in contemporary applications of NLP to real-world question-answering environments, where knowledge may reside in documentation, Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) has emerged as a pivotal technique. A RAG system comprises two main components: a Retriever and a Large Language Model. Retrievers excel at representing similar words and sentences closely in the embedding space, thereby understanding language patterns effectively. Notable retriever models, such as Dense Passage Retrieval (DPR) (Karpukhin et al., 2020), EmbEddings from bidirEctional Encoder rEpresentations (E5) (Wang et al., 2022), and General Text Embedding (GTE) (Li et al., 2023), leverage pretrained architectures such as BERT (Devlin et al., 2019) to initialize encoders. These encoders are fine-tuned to ensure that the cosine similarity between the query and passage accurately captures their true relationship.

Recently, the NLP community has increasingly favored decoder architectures for creating embeddings (Springer et al., 2024; BehnamGhader et al., 2024), as these approaches have demonstrated superior performance over traditional encoder-based methods. Among these, Mistral-E5 (Wang et al., 2024) stands out as the state-of-the-art LLM for text retrieval. Retriever models are commonly evaluated using benchmarks such as BEIR (Thakur et al., 2021) and MTEB (Muennighoff et al., 2022).

E ADDITIONAL RESULTS AND DISCUSSIONS

In this section, we provide additional results and discussions:

(i) We discuss a slight refinement of `CHC`'s guessing rule;

(ii) We show that our analysis of `CHC` is sharp;

(iii) We discuss Theorems 4.1 and 4.3;

(iv) We discuss the center separation condition used in the `CC` analysis;

(v) We explain why the regret analysis of $\texttt{GHC}(\tau)$ for $\tau > 0$ is challenging.

E.1 `CHC` Guessing Rule Refinement

If all hulls $\hat{\mathcal{C}}_{i,t}$ for $i \in [N]$ are non-empty, there are cases where the cell (hence the label) of a query that lands outside of the convex hulls can be deduced. Indeed, if $\text{hull}_{\mathcal{E}}(\{q_t\} \cup \hat{\mathcal{C}}_{i,t}) \cap \hat{\mathcal{C}}_{j,t} \neq \emptyset$ for some $i \neq j$, then q_t cannot have label i. The decision rule of `CHC` may thus be slightly refined by returning label i when for all $k \in [N] \setminus \{i\}$,

$$\text{hull}_{\mathcal{E}}(\{q_t\} \cup \hat{\mathcal{C}}_{k,t}) \cap \hat{\mathcal{C}}_{j,t} \neq \emptyset$$

for some $j \neq k$. For example, in Fig. 2, q is necessarily in the top-right cell, and we know its correct label. We do not leverage this minor refinement in the analysis nor the implementation of the algorithm.

E.2 Asymptotic Lower bound on the Regret of `CHC`

Here, we make a slightly stronger assumption on the density of μ than Assumption 3.3(i):

Assumption E.1. The distribution μ is absolutely continuous with respect to V with density $f_\mu = \frac{d\mu}{dV}$. Furthermore, f_μ is continuous on $\mathcal{E}$ and $\inf_{x \in \mathcal{E}} f_\mu(x) > 0$.

Theorem E.2. *Under Assumptions 3.2 and E.1, the following result holds either if $\mathcal{E} = \mathcal{I}^d$ and $d \geq 2$ with $\eta = d$, or if $\mathcal{E} = \mathcal{S}^{d-1}$, $d \geq 3$ and each cell $\mathcal{C}_i$ is contained in an open half-sphere with $\eta = d - 1$:*

$$\liminf_{T \to \infty} \frac{R_{\texttt{CHC}}(T)}{\log^\eta T} \geq (\beta - \alpha) \frac{\sum_{i=1}^N F(\mathcal{C}_i)}{4\eta! \eta^\eta}.$$

The lower bounds established in Theorem E.2, are independent of the distribution μ and match the upper bounds of Theorem 4.1 in the uniform case ($C/c = 1$), up to a multiplicative factor linear in the dimension d. This

suggests that CHC does not exploit favorable scenarios where μ is highly concentrated around the seed queries of the Voronoi tessellation. It is important to stress that this lower bound is specific to the CHC algorithm and does not extend to all possible algorithms. The proof, presented below, relies on the connection between random polytopes and the so-called *floating body* of the polytope they approximate, a notion developed in Schütt and Werner (1990).

Proof of Theorem E.2. Recall that the expected number of expert calls is driven by the probability of a query q_t landing outside the current estimated hulls:

$$\mathbb{E}[C_T] = \sum_{t=1}^{T} \mathbb{E}[\mathbb{P}(q_t \notin \cup_i \hat{\mathcal{C}}_{i,t} | \mathcal{F}_{t-1})].$$

By equality (1), this probability is the expected measure of the uncovered region, $\sum_{i=1}^{N} \mathbb{E}[\mu(\mathcal{C}_i \setminus \hat{\mathcal{C}}_{i,t})] = \sum_{i=1}^{N} \mu(\mathcal{C}_i)\mathbb{E}[\mu_{\mathcal{C}_i}(\mathcal{C}_i \setminus \hat{\mathcal{C}}_{i,t})]$, where $\mu_{\mathcal{C}_i}$ is the conditional distribution of q_t given that $q_t \in \mathcal{C}_i$. This means that

$$R_{\texttt{CHC}}(T) = (\beta - \alpha) \sum_{t=1}^{T} \sum_{i=1}^{N} \mu(\mathcal{C}_i)\mathbb{E}[\mu_{\mathcal{C}_i}(\mathcal{C}_i \setminus \hat{\mathcal{C}}_{i,t})]. \tag{6}$$

Case $\mathcal{E} = \mathcal{I}^d$. Instead of applying Corollary B.3, we may alternatively use the following lemma.

Lemma E.3. *Let P a convex polytope in $\mathbb{R}^d$ with $d \geq 2$ and $n \geq 1$ points $p_1, \ldots, p_n$ sampled independently from a distribution ν supported on P with a density w.r.t. λ that is continuous and never zero on P. Denote by P_n the convex hull of $p_1, \ldots, p_n$. Then*

$$\mathbb{E}[\nu(P \setminus P_n)] \geq \frac{F(P)}{4d!d^{d-1}} \frac{\log^{d-1} n}{n} + o\left(\frac{\log^{d-1} n}{n}\right).$$

Lemma E.3 leverages the relationship between the uncovered region $P \setminus P_n$ and the wet part of ν, which we now define. For a distribution ν, the wet part of ν is defined in Bárány et al. (2020) as

$$W_\nu^t = \{x \in \mathbb{R}^d, \text{there is a halfspace } h \text{ with } x \in h \text{ and } \nu(h) \leq t\},$$

and the ν-measure of the wet part is

$$w^\nu(t) = \nu(W_t^\nu).$$

To understand this notion intuitively, consider the case where ν is uniform on the unit ball, and assume that the ball is filled with a volume t of water. The wet part W_ν^t represents the "trace" left by the water as the ball rolls on a flat surface, and $w^\nu(t)$ is the volume of that wet part. The ν-measure of the wet part is tightly linked to the expected missing volume $\mathbb{E}[1 - \nu(P_n)]$, as shown in e.g. Schütt (1991) and Bárány and Buchta (1993) for the case where ν is uniform on some convex polytope of $\mathbb{R}^d$, and in Bárány et al. (2020) for arbitrary measures in $\mathbb{R}^d$. Such results are sometimes stated in terms of the *floating body* of ν, which is the relative complement of its wet part in its support, see Schütt and Werner (1990).

Proof of Lemma E.3. By the lower bound of Theorem 1.2 in Bárány et al. (2020):

$$\mathbb{E}[\nu(P \setminus P_n)] \geq \frac{1}{4} w^\nu\left(\frac{1}{n}\right). \tag{7}$$

This lower bound is straightfoward to derive: one can note that $\mathbb{P}(x \notin P_n) \geq (1-t)^n$ for any $x \in W_\nu^t$, thus $\mathbb{E}[\nu(P \setminus P_n)] = \int_0^\infty \mathbb{P}(x \notin P_n) d\mu(x) \geq (1-t)^n w^\nu(t)$. Setting $t = 1/n$ yields the result (see Section 3.1 in Bárány et al. (2020) for a more detailed proof).

Then, by Corollary 1.2 in Besau et al. (2018), under Assumption 3.3(ii), the asymptotic behavior of w^ν is independent of ν:

$$w^\nu(\delta) \underset{\delta \to 0}{\sim} \frac{F(P)}{d!d^{d-1}} \delta \log^{d-1}(1/\delta)$$

where we recall that $F(P)$ is the number of flags of P. Consequently, the right hand side in inequality (7) is asymptotically equivalent to $\frac{F(P)}{d!d^{d-1}} \frac{1}{4n} \log^{d-1} n$. □

As in the proof of Theorem 4.1, we may write $\mathbb{E}[\mu_{\mathcal{C}_i}(\mathcal{C}_i \setminus \hat{\mathcal{C}}_{i,t})] = \mathbb{E}[\mathbb{E}[\mu_{\mathcal{C}_i}(\mathcal{C}_i \setminus Q_{n_i(t)})|n_i(t)]]$ where $n_i(t)$ is the number of points that landed in $\mathcal{C}_i$ up to time t and Q_n is the convex hull of n points sampled independently from $\mu_{\mathcal{C}_i}$. Applying Lemma E.3 to each $\mathcal{C}_i$ gives

$$\mathbb{E}[\mu_{\mathcal{C}_i}(\mathcal{C}_i \setminus \hat{\mathcal{C}}_{i,t})|n_i(t) = n] \geq \frac{F(\mathcal{C}_i)}{4d!d^{d-1}} \frac{\log^{d-1} n}{n} + o\left(\frac{\log^{d-1} n}{n}\right).$$

As $n_i(t)$ follows a $\mathrm{Bin}(t, \mu(\mathcal{C}_i))$ distribution, Lemma C.1 gives that as $t \to \infty$, $\frac{F(\mathcal{C}_i)}{4d!d^{d-1}} \mathbb{E}\left[\frac{\log^{d-1}(n_i(t))}{n_i(t)}\right] \underset{t\to\infty}{\sim} \frac{F(\mathcal{C}_i)}{4d!d^{d-1}} \frac{\log^{d-1} t}{t\mu(\mathcal{C}_i)}$, so that asymptotically in t,

$$\mathbb{E}[\mu_{\mathcal{C}_i}(\mathcal{C}_i \setminus \hat{\mathcal{C}}_{i,t})] \geq \frac{F(\mathcal{C}_i)}{4d!d^{d-1}} \frac{\log^{d-1} t}{t\mu(\mathcal{C}_i)} + o\left(\frac{\log^{d-1} t}{t}\right).$$

As $\sum_{t=1}^{T} \frac{\log^{d-1} t}{t} = \frac{\log^d T}{d} + \mathcal{O}(1)$ for any $d \geq 1$ (with the convention $0^0 = 1$), combining the previous identity with equality (6) yields

$$R_{\texttt{CHC}}(T) \geq (\beta - \alpha) \frac{\sum_{i=1}^{N} F(\mathcal{C}_i)}{4d!d^{d-1}} \frac{\log^d T}{d} + o\left(\log^d T\right)$$

which concludes the proof in the case $\mathcal{E} = \mathcal{I}^d$.

Case $\mathcal{E} = \mathcal{S}^{d-1}$. The exact same gnomonic projection argument as in the proof of Corollary B.4 can be used to obtain a spherical analogue of Lemma E.3, simply noting that the density of the pushforward measure on $\mathbb{R}^d$ will remain continuous and never zero on the projected convex polytope $g_e(P)$ if the original measure on $\mathcal{S}^{d-1}$ is:

Lemma E.4. *Let P a spherically convex polytope in $\mathcal{S}^{d-1}$ contained in an open halfsphere with $d \geq 2$ and $n \geq 1$ points $p_1, \ldots, p_n$ sampled independently from a distribution ν supported on P with a density w.r.t. ω that is continuous and never zero on P. Denote by P_n the convex hull of $p_1, \ldots, p_n$. Then*

$$\mathbb{E}[\nu(P \setminus P_n)] \geq \frac{F(P)}{4(d-1)!(d-1)^{d-2}} \frac{\log^{d-2} n}{n} + o\left(\frac{\log^{d-2} n}{n}\right).$$

The regret analysis is then exactly the same as in the case $\mathcal{E} = \mathcal{I}^d$. □

E.3 On the Error Term in Theorem 4.1

The error term in Theorem 4.1, e.g. $\mathcal{O}(\log^{d-1}(T) \log\log T)$ when $\mathcal{E} = \mathcal{I}^d$, depends on the following model parameters.

(i) The geometry of each cell $\mathcal{C}_i$. This dependency comes from our application of Proposition B.2 (Theorem 2 from Bárány and Buchta (1993)).

(ii) The bounds on the density of μ (3.3), and the volume of each $\mathcal{C}_i$. This dependency mainly comes from both the rejection sampling trick (Corollary B.3) and the application of Lemma C.1—it was applied for $p = c\lambda(\mathcal{C}_i)$ in the proof of Corollary B.3, and for $p = \mu(\mathcal{C}_i) \in [c\lambda(\mathcal{C}_i), C\lambda(\mathcal{C}_i)]$ in the proof of Theorem 4.1.

Below, we support the claim that it is very difficult to identify the precise dependency in the cell geometry (and in particular in the dimension) induced by Theorem 2 from Bárány and Buchta (1993), which was not tracked in their original proof. One way to attempt to identify this dependency is to exploit the relationship between random polytopes and the floating body of the polytope they approximate, as already leveraged in the proof of Theorem E.2. For example, Bárány et al. (2020) show in their Theorem 1.2 that if $p_1, \ldots, p_n$ are sampled according to some arbitrary ν on $\mathbb{R}^d$ with convex hull P_n,

$$\mathbb{E}[1 - \nu(P_n)] \leq w^{\nu}\left((d+2)\frac{\log n}{n}\right) + 2^{d+3} e \frac{\lceil \log n \rceil^d}{n^2}.$$

Here, the dependency in $(\log n)/n$ in the wet part measure is necessary because ν is arbitrary, but for more specific distributions (e.g. log-concave), this dependency may potentially be improved to $1/n$, as discussed in Section 2.4 of Bárány et al. (2020). Since $\mathbb{E}[1-\nu(P_n)]$ is precisely the measure of the region uncovered by P_n, we may attempt to upper bound $w^\nu(t)$ for fixed t to obtain a fully explicit upper bound on our quantity of interest. This is not too difficult when ν is uniformly distributed on the hypercube $\mathcal{I}^d$, as we show below.

Proposition E.5. *Denote by $w(t)$ the volume of the wet part of parameter t for ν uniform on $\mathcal{I}^d$. For any $n \geq 2$, $w(1/n) \leq \frac{2}{n}\sum_{k=0}^{d-1}\frac{\log(n/2)^k}{k!}$.*

Proof. Let $v(x) = \min\{\lambda(\mathcal{I}^d \cap H), x \in H \text{ and } H \text{ is a halfspace}\}$, and $u(x) = \lambda(\mathcal{I}^d \setminus (2x - \mathcal{I}^d))$. This is the volume of a *Macbeath* region, see e.g. Bárány (2008), where they recall in their Lemma 4.3 that $u(x) \leq 2v(x)$. Note also that the wet part for the uniform distribution on $\mathcal{I}^d$ is $\{x \in K, v(x) \leq t\lambda(\mathcal{I}^d)\}$. Then

$$u(x) = \lambda(\mathcal{I}^d \cap (2x - \mathcal{I}^d)) = \lambda([0,1]^d \cap ([2x_1 - 1, 2x_1] \times \cdots \times [2x_d - 1, 2x_d]).$$

By symmetry we may decompose the hypercube into 2^d subcubes based on which side of $1/2$ each coordinate lies and only consider the case $x \in S_0 := [0, 1/2]^d$. For $x \in S_0$, we have $u(x) = \lambda([0, 2x_1] \times \cdots \times [0, 2x_d]) = 2^d \prod_i x_i$. Thus

$$w(t) = \int_{\mathcal{I}^d} \mathbb{1}_{v(x)\leq t}\mathrm{d}x \leq \int_{\mathcal{I}^d} \mathbb{1}_{u(x)\leq 2t}\mathrm{d}x = 2^d \int_{S_0} \mathbb{1}_{u(x)\leq 2t}\mathrm{d}x.$$

Setting $y = 2x$, this is equal to

$$\begin{aligned} 2^d \int_{S_0} \mathbb{1}_{\prod x_i \leq 2t/2^d}\mathrm{d}x &= 2^d \int_{\mathcal{I}^d} \mathbb{1}_{\prod (y_i/2)\leq 2t/2^d}(1/2^d)\mathrm{d}y \\ &= \int_{\mathcal{I}^d} \mathbb{1}_{\prod y_i \leq 2t}\mathrm{d}y \\ &= \lambda(\{y \in [0,1]^d, \prod y_i \leq 2t\}). \end{aligned}$$

Now for any $0 < a \leq 1$, $\lambda(\{y \in [0,1]^d, \prod y_i \leq a\}) = a\sum_{k=0}^{d-1}\log(1/a)^k/k!$ (see Bárány and Larman (1988), p.288). Consequently, $w(t) \leq 2t\sum_{k=0}^{d-1}\log(1/(2t))^k/k!$ whenever $0 < t \leq 1/2$. Applying this to $t = 1/n$ yields the result. □

When ν is uniform on some arbitrary polytope P, explicitly bounding the volume of the wet part of P becomes significantly more difficult. Schütt (1991) proved in their Lemma 1.8 that for some t_d and c_d that only depends on d, if $0 < t < t_d$,

$$w^\nu(t) \leq \frac{F(P)}{d!d^{d-1}}t\log(\lambda(P)/t)^{d-1} + c_d F(P)t\log(\lambda(P)/t)^{d-2},$$

but no explicit expressions for t_d and c_d are provided. Their proof strategy is to first obtain an upper bound when P is a simplex, then form a partition of P by $F(P)$ simplices, and apply this bound to each simplex in the partition. By carefully tracking the constant appearing in the proof of their Lemma 1.3, one can check that even in the simplex case, their arguments only yield a likely very loose value of c_d that scales roughly as 2^{d^2}. As such, more refined arguments are required to identify the precise dependency in d of c_d, and to the best of our knowledge, this has not been elucidated in subsequent works.

E.4 On Generalizing Theorem 4.3 to $d \geq 2$

In dimension one, we derived the sharp density-free regret bound of Theorem 4.3 via a change of variable involving cumulative distribution functions (c.d.f.). Specifically, we used that:

(i) If q has a continuous c.d.f. F, $F(q)$ is uniformly distributed in $[0,1]$

(ii) This F maps convex hulls to convex hulls: $F(\mathrm{conv}\{p_1, \ldots, p_n\}) = \mathrm{conv}\{F(p_1), \ldots, F(p_n)\}$.

A natural idea to generalize Theorem 4.3 to higher dimensions $d \geq 2$ is to identify a transformation F in $\mathbb{R}^d$ with analogous properties. However, preserving convex hulls is a much more restrictive condition in dimensions

$d \geq 2$ that for instance forces F to be affine if it is one-to-one: this is the so-called fundamental theorem of affine geometry, see e.g. Theorem 2.16 in Artstein-Avidan et al. (2012). One may hope that a weaker but still sufficient condition for the analysis would hold: namely, by inspecting the proof of Theorem 4.3, it can be noted that the condition

$$\lambda(F(\text{conv}\{p_1, \dots, p_n\})) \geq \lambda(\text{conv}\{F(p_1), \dots, F(p_n)\}) \tag{8}$$

would suffice. A natural candidate for F is the Rosenblatt transformation, which maps continuous random vectors to vectors uniformly distributed on the hypercube by inductively projecting their coordinates with their conditional c.d.f. (Rosenblatt, 1952). We show below that unfortunately, (8) does not hold for the Rosenblatt transformation.

Counter-example for (8). Consider a random variable with support $[0,1]^2$ and density $f(x_1, x_2) = 2x_1$, $p_1 = (0,0)$, $p_2 = (1,0)$, $p_3 = (0,1)$. Then $\text{conv}\{p_1, p_2, p_3\} = \{(x,y) \in [0,1]^2 | x + y \leq 1\}$ is a triangle. Furthermore, the marginal and conditional c.d.f. are given by $F_1(x_1) = x_1^2$ and $F_{2|1}(x_2|x_1) = x_2$. The Rosenblatt transformation is then given by $F(x_1, x_2) = (x_1^2, x_2)$. In particular, $F(p_i) = p_i$ so $\text{conv}\{p_1, p_2, p_3\} = \text{conv}\{F(p_1), F(p_2), F(p_3)\}$, yet $F(\text{conv}\{p_1, p_2, p_3\}) = \{(x^2, y) | (x,y) \in [0,1]^2, x + y \leq 1\} = \{(x,y) | (x,y) \in [0,1]^2, \sqrt{x} + y \leq 1\} \subset \text{conv}\{p_1, p_2, p_3\}$. In fact, $\lambda(\text{conv}\{F(p_1), F(p_2), F(p_3)\}) = 1/2$, and $\lambda(F(\text{conv}\{p_1, p_2, p_3\})) = \int_0^1 (1 - \sqrt{u}) \mathrm{d}u = 1/3$.

This suggests that this transformation argument is not fruitful in higher dimensions, and that a more refined analysis is required.

E.5 Center Separation Assumption when $\mathcal{E} = \mathcal{I}^d$ and $\mathcal{E} = \mathcal{S}^{d-1}$

The CC regret bound of Theorem 4.5 relies on the separability assumption $\delta_{\min}^2 \geq c\sigma^2 d$ for some $c \geq 32$.

When $\mathcal{E} = \mathbb{R}^d$, this assumption is reasonable: since the expected squared Euclidean distance between two uniformly sampled points scales linearly with d (Anderssen et al., 1976), it is plausible for $\delta_{\min}^2$ to also scale with d. If $\mathcal{E} = \mathcal{I}^d$, it is reasonable if σ is small. In this context, the d in the numerator of the threshold formula

$$\hat{m}(t) = \frac{108\sigma^2(d + 2\log T)}{\hat{\delta}_{\min}^2(t)}$$

is effectively balanced by the implicit scaling of d in the denominator, making the threshold rule behave in a "dimensionless" manner.

However, this assumption may become unrealistic when $\mathcal{E} = \mathcal{S}^{d-1}$, as the average distance between uniformly sampled seeds is bounded by 2, irrespective of the dimension. This average distance in fact approaches $\sqrt{2}$ as $d \to \infty$, because the probability that two seeds are orthogonal approaches one as $d \to \infty$. Therefore, we cannot expect $\delta_{\min}^2$ to scale linearly with d. In this setting, the numerator of $\hat{m}(t)$ scales with d while the denominator does not.

Although Theorems B.13 and 4.5 still hold as stated for subgaussian mixtures supported on the sphere, a tailored analysis would employ concentration inequalities native to that geometry. They would require a weaker separability assumption and result in a different threshold rule, likely both without a linear dependence on d. Such an analysis is beyond the scope of this work. Our empirical evaluation explores this by treating the constant in the definition of the threshold $\hat{m}(t)$ as a tunable hyperparameter, so that we can find a practical balance suitable for each geometric setting.

E.6 Challenges of the GHC Regret Analysis

Analyzing the regret of GHC(τ) (Algorithm 2) for $\tau > 0$ introduces significant complexities not present in the analysis of the CHC algorithm (which corresponds to GHC with $\tau = 0$).

Non i.i.d. Hull Construction It can no longer be said that $\hat{\mathcal{C}}_{i,t} = \text{hull}_{\mathcal{E}}(\mathcal{Q}_{i,t})$ is the convex hull of the i.i.d. samples drawn from μ that landed in $\mathcal{C}_i$, which is crucial for applying results from random polytope theory, as they are typically stated for independently sampled points. Indeed, when $\tau > 0$, if q_t results in a guess (whether correct or incorrect), since no reward feedback is observed, q_t will not be used to form the hulls. $\hat{\mathcal{C}}_{i,t}$ is *only* the convex hull of the queries that either triggered an expert call or landed inside the current hulls. The queries are selected through a complex history-dependent process that depends on the geometric configuration of all estimated hulls $\hat{\mathcal{C}}_{j,t}$.

Bounding the Number of Incorrect Guesses When $\tau > 0$, the GHC may make incorrect guesses, and the regret is no longer directly proportional to the number of expert calls: rather,

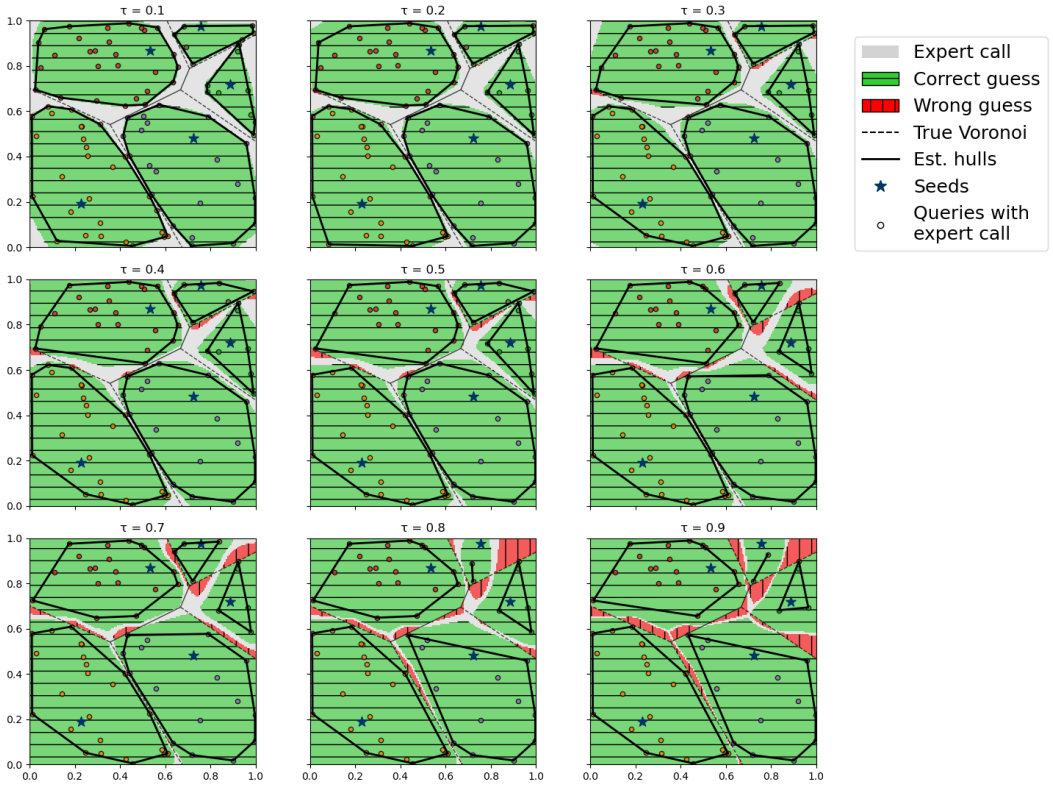
$$R_{\texttt{GHC}}(T) = (\beta - \gamma)\mathbb{E}[M_T] + (\beta - \alpha)\mathbb{E}[C_T]$$

where M_T is the total number of incorrect guesses (mistakes). A mistake occurs if $q_t \in C_k$ for some $k \in [N]$ while the algorithm guesses $i \neq k$. It can be checked that this implies $d(q_t, C_i) \leq \varepsilon_{i,t} + \tau\varepsilon_{k,t}$, where $\varepsilon_{j,t} = d_H(\mathcal{C}_j, \hat{\mathcal{C}}_{j,t})$ is the Hausdorff distance between the cell $\mathcal{C}_j$ and its estimate. Bounding these Hausdorff distances is already difficult due to the non-i.i.d. hull formation. Furthermore, in the simpler i.i.d. uniform case, bounds on the Hausdorff distance between a random polytope and the polytope it approximates scale as $n^{-1/d}$ where n is the number of points (Bárány, 1989) . This analysis approach would yield bounds on the number of mistakes scaling with $T^{(d-1)/d}$, which, asymptotically in T, is significantly worse than the $\log^d T$ bound of Theorem 4.1.

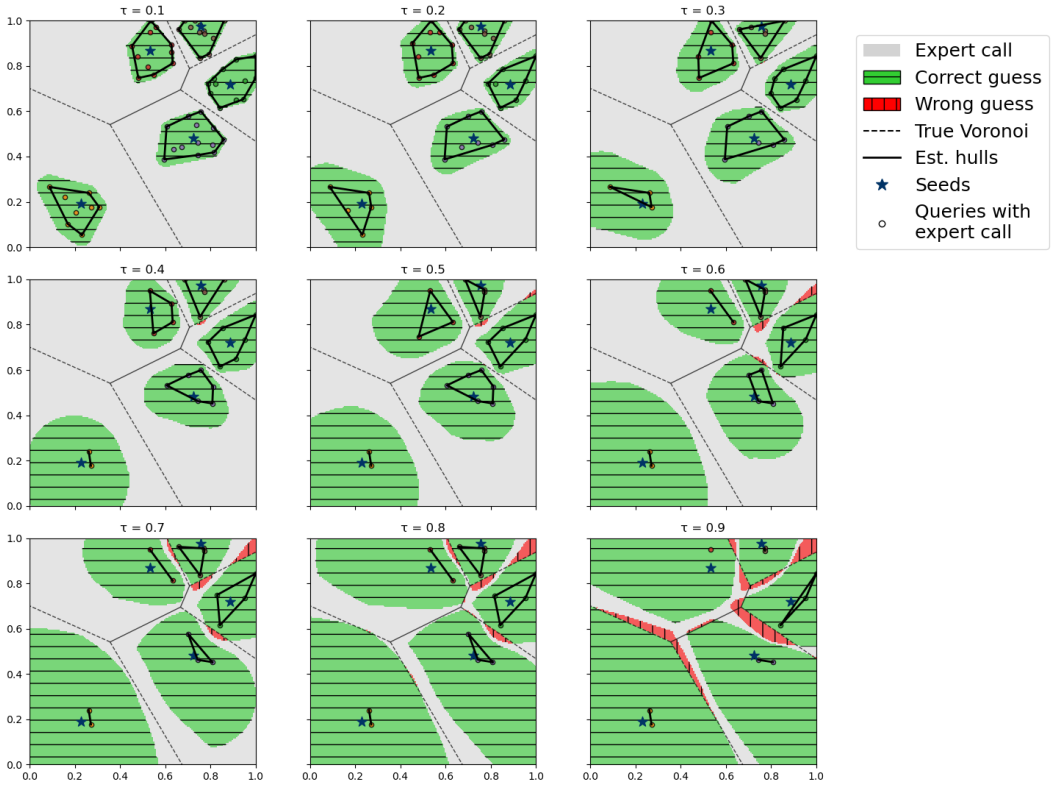
Bounding the Number of Expert Calls The region triggering an expert call is also more intricate: the convex hulls $\hat{\mathcal{C}}_{i,t}$ depend on τ. For a fixed hull, increasing τ reduces the probability of an expert call, but the hulls vary with τ: as τ increases fewer points are used to form them. For a fixed τ, reducing the hulls makes expert calls more likely. These two effects counteract each other and it is not directly obvious how to handle them.

These dynamics prevent a straightforward extension of the CHC analysis. Deriving tight regret bounds for GHC with $\tau > 0$ would likely require more involved techniques to handle these history-dependent processes.

Decision Regions To better visualize the impact of τ and the distribution of the queries in the decision rule of GHC we complement Fig. 3 from Section 4.3 with Fig. 6. The decision regions in round $t = 250$ for a run of $\texttt{GHC}(\tau)$ for $\tau \in \{0.1, 0.2, \ldots, 0.9\}$ are shown, for a uniform distribution and a mixture of truncated Gaussians with small covariance.



(a) Uniform



(b) Mixture of truncated Gaussians, covariance matrix $0.0025I$.

Figure 6: Decision regions of $\texttt{GHC}(\tau)$ for $t = 250$ and $\mathcal{E} = [0, 1]^2$.

F EXPERIMENTS DETAILS

In this section, we provide the results of numerous additional synthetic and real-world experiments, along with a more detailed description of the experiments of Section 5. Tthe reward values are always instantiated as $\alpha = -1$, $\beta = +1$, $\gamma = -10$.

F.1 Synthetic Experiments

To evaluate our algorithms in low dimensions, we perform synthetic experiments on two query domains for $N = 5$ labels:

(i) $\mathcal{I}^d$, $d \in \{1, 4, 10, 50\}$, where the seeds $s_1, \ldots, s_N$ are drawn uniformly on $\mathcal{I}^d$ and q_t is either drawn from the uniform distribution on $\mathcal{I}^d$ or from a homogeneous mixture of truncated Gaussians with covariance matrix $0.01I$ In this setting, `GHC` is applied using the distance to Euclidean convex hulls.

(ii) $\mathcal{S}^{d-1}$, $d \in \{2, 4, 10, 50\}$ where the seeds $s_1, \ldots, s_N$ are drawn uniformly on $\mathcal{S}^{d-1}$. The query q_t is sampled either uniformly on $\mathcal{S}^{d-1}$ or from a mixture, specifically $i \in [N]$ is drawn uniformly at random and $q_t = y_t/\|y_t\|$ where $y_t \sim \mathcal{N}(s_i, 0.01I)$. In this setting, `GHC` is applied using the distance to spherical convex hulls.

For each configuration, five independent datasets of size $T = 5000$ are generated. All evaluated algorithms are run on the same datasets. The expert labeling policy is assumed to be given by the Voronoi tessellation with seeds s_i. We report detailed metrics, namely:

(i) the average number of expert calls after every label has been observed at least once;

(ii) the average number of incorrect guesses;

(iii) the average Voronoi regret

over the $T = 5000$ rounds.

F.1.1 Evaluation of `CHC`, `GHC` and Comparison with Sequential k-Means Baseline

First, we report the performance of `GHC` and `CHC`. We recall that `CHC` corresponds to `GHC` for $\tau = 0$. We also present the results of a baseline algorithm based on sequential k-means (SKM) (MacQueen, 1967). At each round, it stores N estimated centroids $\{\hat{c}_i\}_{i\in[N]}$. If the algorithm guesses that the label of q is i, the corresponding centroid is updated as $\hat{c}_i \leftarrow \hat{c}_i + (q - \hat{c}_i)/\kappa_i$ where κ_i is the number of times label i was guessed up to the current round. This algorithm has two phases:

(i) In phase 1, it calls the expert up until all labels are observed.

(ii) In phase 2, it guesses the label of the nearest centroid.

Since this baseline never calls the expert after all labels are observed, it is a relevant comparison to `GHC` with threshold $\tau = 1$. As for `GHC` we do not report the number of calls in the first phase, although they are included in the regret calculation, as we describe below.

Results are reported in Tables 1 and 2 for $\mathcal{I}^d$ and $\mathcal{S}^{d-1}$ respectively. "C", "M" and "R" denote the mean $\pm$ standard deviation of the number of expert calls in the second phase, the number of incorrect guesses and the cumulative regret respectively.

Observe that the optimal threshold increases with the dimension: in dimension one, keeping a low threshold is optimal and the regret, mostly driven by the number of expert calls, is limited. Already in dimension 10, `CHC` almost calls the expert for the $5,000$ queries, as expected from the discussion following Corollary 4.2. As the dimension grows large, the optimal threshold appears to rapidly approach one. In the mixture case, the performance of `GHC` is surprisingly better in large dimension. A likely explanation in the $\mathcal{E} = \mathcal{I}^d$ is the $\sqrt{d}$ scaling of the average distance between the uniformly sampled seeds in $\mathcal{I}^d$ (Anderssen et al., 1976), making the Gaussian clusters more separated since their covariance is fixed. The same regret improvement is observed when $\mathcal{E} = \mathcal{S}^{d-1}$,

even though the average distance between the uniformly sampled seeds will remain bounded by 2. This average distance in fact approaches $\sqrt{2}$ as $d \to \infty$, because the probability that two seeds are orthogonal approaches one as $d \to \infty$. This entails that the centers are more evenly separated in higher dimensions and could explain why better regret is observed in these cases. The performance of SKM is worse than `GHC` for any threshold in the uniform settings, and comparable to `GHC` with high threshold in the mixture settings.

Table 1: Performance metrics of `GHC` and Sequential K-Means (SKM) on $\mathcal{I}^d$ ($T = 5000$).

Dim	Dist.	$\tau = 0.00$	$\tau = 0.10$	$\tau = 0.40$	$\tau = 0.60$	$\tau = 0.80$	$\tau = 0.90$	$\tau = 0.95$	$\tau = 1.00$	SKM
1	Unif.	C: 59 ± 7 M: 0 ± 0 R: 142 ± 11	C: 42 ± 6 M: 0 ± 0 **R: 110 ± 7**	C: 29 ± 3 M: 3 ± 3 R: 119 ± 32	C: 26 ± 2 M: 12 ± 3 R: 211 ± 39	C: 20 ± 3 M: 34 ± 10 R: 434 ± 113	C: 16 ± 3 M: 68 ± 20 R: 799 ± 212	C: 12 ± 2 M: 101 ± 33 R: 1161 ± 365	C: 0 ± 0 M: 375 ± 126 R: 4152 ± 1387	C: 0 ± 0 M: 1282 ± 389 R: 14128 ± 4275
	Mix.	C: 55 ± 10 M: 0 ± 0 R: 130 ± 21	C: 41 ± 9 M: 0 ± 0 **R: 106 ± 18**	C: 34 ± 4 M: 9 ± 5 R: 181 ± 53	C: 29 ± 1 M: 21 ± 7 R: 310 ± 78	C: 24 ± 1 M: 42 ± 14 R: 525 ± 148	C: 20 ± 1 M: 87 ± 22 R: 1013 ± 246	C: 15 ± 2 M: 155 ± 31 R: 1752 ± 336	C: 0 ± 0 M: 823 ± 389 R: 9070 ± 4275	C: 0 ± 0 M: 1011 ± 340 R: 11141 ± 3741
4	Unif.	C: 1476 ± 35 M: 0 ± 0 R: 2972 ± 72	C: 1657 ± 45 M: 0 ± 0 R: 3336 ± 92	C: 722 ± 29 M: 12 ± 4 **R: 1593 ± 35**	C: 478 ± 25 M: 60 ± 7 R: 1633 ± 55	C: 291 ± 13 M: 191 ± 15 R: 2706 ± 159	C: 194 ± 10 M: 376 ± 20 R: 4550 ± 201	C: 132 ± 5 M: 576 ± 46 R: 6618 ± 511	C: 0 ± 0 M: 2415 ± 324 R: 26584 ± 3559	C: 0 ± 0 M: 2730 ± 333 R: 30055 ± 3659
	Mix.	C: 1159 ± 7 M: 0 ± 0 R: 2337 ± 19	C: 933 ± 48 M: 0 ± 0 R: 1884 ± 100	C: 334 ± 15 M: 2 ± 2 R: 712 ± 28	C: 214 ± 13 M: 12 ± 3 **R: 573 ± 52**	C: 130 ± 10 M: 43 ± 11 R: 754 ± 113	C: 83 ± 8 M: 86 ± 15 R: 1136 ± 152	C: 55 ± 7 M: 120 ± 16 R: 1453 ± 173	C: 0 ± 0 M: 444 ± 304 R: 4902 ± 3346	C: 0 ± 0 M: 70 ± 7 R: 793 ± 94
10	Unif.	C: 4725 ± 25 M: 0 ± 0 R: 9489 ± 35	C: 4783 ± 24 M: 0 ± 0 R: 9605 ± 33	C: 3762 ± 47 M: 1 ± 0 R: 7569 ± 73	C: 2665 ± 50 M: 22 ± 5 R: 5607 ± 74	C: 1517 ± 30 M: 196 ± 10 **R: 5233 ± 94**	C: 903 ± 23 M: 489 ± 17 R: 7226 ± 186	C: 528 ± 22 M: 816 ± 48 R: 10068 ± 487	C: 0 ± 0 M: 2599 ± 512 R: 28628 ± 5600	C: 0 ± 0 M: 3533 ± 389 R: 38900 ± 4268
	Mix.	C: 4400 ± 23 M: 0 ± 0 R: 8821 ± 48	C: 2837 ± 43 M: 0 ± 0 R: 5695 ± 88	C: 132 ± 6 M: 0 ± 0 R: 284 ± 9	C: 23 ± 3 M: 0 ± 0 R: 66 ± 9	C: 3 ± 2 M: 0 ± 0 R: 27 ± 4	C: 1 ± 1 M: 0 ± 0 R: 23 ± 4	C: 1 ± 1 M: 0 ± 0 **R: 23 ± 4**	C: 0 ± 0 M: 1 ± 1 R: 31 ± 7	C: 0 ± 0 M: 0 ± 0 R: 20 ± 5
50	Unif.	C: 4981 ± 6 M: 0 ± 0 R: 10000 ± 0	C: 4981 ± 6 M: 0 ± 0 R: 10000 ± 0	C: 4981 ± 6 M: 0 ± 0 R: 10000 ± 0	C: 4981 ± 6 M: 0 ± 0 R: 10000 ± 0	C: 4976 ± 6 M: 0 ± 0 R: 9991 ± 3	C: 4569 ± 11 M: 35 ± 5 **R: 9559 ± 36**	C: 3284 ± 31 M: 356 ± 21 R: 10525 ± 177	C: 0 ± 0 M: 3155 ± 101 R: 34741 ± 1097	C: 0 ± 0 M: 3875 ± 307 R: 42659 ± 3378
	Mix.	C: 4989 ± 3 M: 0 ± 0 R: 10000 ± 0	C: 4989 ± 3 M: 0 ± 0 R: 10000 ± 0	C: 7 ± 2 M: 0 ± 0 R: 37 ± 2	C: 0 ± 0 M: 0 ± 0 R: 23 ± 5	C: 0 ± 0 M: 0 ± 0 R: 23 ± 5	C: 0 ± 0 M: 0 ± 0 R: 23 ± 5	C: 0 ± 0 M: 0 ± 0 R: 23 ± 5	C: 0 ± 0 M: 0 ± 0 **R: 23 ± 5**	C: 0 ± 0 M: 0 ± 0 R: 23 ± 5

Table 2: Performance metrics of `GHC` and Sequential K-Means (SKM) on $\mathcal{S}^{d-1}$ ($T = 5000$).

Dim	Dist.	$\tau = 0.00$	$\tau = 0.10$	$\tau = 0.20$	$\tau = 0.40$	$\tau = 0.60$	$\tau = 0.80$	$\tau = 1.00$	SKM
2	Unif.	C: 50 ± 7 M: 0 ± 0 R: 132 ± 12	C: 43 ± 5 M: 1 ± 1 **R: 125 ± 6**	C: 41 ± 3 M: 2 ± 1 R: 142 ± 7	C: 35 ± 2 M: 5 ± 3 R: 157 ± 30	C: 28 ± 1 M: 18 ± 6 R: 288 ± 69	C: 22 ± 2 M: 44 ± 11 R: 563 ± 128	C: 0 ± 0 M: 315 ± 148 R: 3495 ± 1633	C: 0 ± 0 M: 2431 ± 174 R: 26804 ± 1923
	Mix.	C: 61 ± 3 M: 0 ± 0 R: 138 ± 8	C: 47 ± 4 M: 0 ± 0 R: 114 ± 8	C: 38 ± 2 M: 1 ± 1 **R: 104 ± 13**	C: 29 ± 1 M: 3 ± 1 R: 110 ± 17	C: 24 ± 1 M: 9 ± 3 R: 166 ± 31	C: 18 ± 2 M: 22 ± 9 R: 299 ± 98	C: 0 ± 0 M: 345 ± 77 R: 3810 ± 847	C: 0 ± 0 M: 70 ± 58 R: 791 ± 645
4	Unif.	C: 602 ± 15 M: 0 ± 0 R: 1225 ± 31	C: 514 ± 11 M: 0 ± 0 R: 1053 ± 26	C: 438 ± 11 M: 2 ± 1 R: 922 ± 25	C: 329 ± 10 M: 14 ± 3 **R: 836 ± 40**	C: 246 ± 6 M: 55 ± 11 R: 1122 ± 108	C: 177 ± 7 M: 137 ± 4 R: 1878 ± 52	C: 0 ± 0 M: 1909 ± 339 R: 21016 ± 3728	C: 0 ± 0 M: 2329 ± 742 R: 25645 ± 8157
	Mix.	C: 531 ± 29 M: 0 ± 0 R: 1083 ± 56	C: 127 ± 4 M: 0 ± 0 R: 275 ± 5	C: 51 ± 4 M: 0 ± 0 R: 124 ± 11	C: 12 ± 2 M: 0 ± 0 R: 46 ± 7	C: 3 ± 2 M: 0 ± 0 R: 28 ± 3	C: 0 ± 0 M: 0 ± 0 R: 22 ± 6	C: 0 ± 0 M: 0 ± 0 **R: 22 ± 7**	C: 0 ± 0 M: 0 ± 0 R: 22 ± 7
10	Unif.	C: 2428 ± 72 M: 0 ± 0 R: 4878 ± 150	C: 2167 ± 74 M: 0 ± 0 R: 4357 ± 154	C: 1919 ± 74 M: 0 ± 0 R: 3862 ± 155	C: 1430 ± 54 M: 6 ± 3 R: 2952 ± 138	C: 1011 ± 42 M: 46 ± 6 **R: 2550 ± 123**	C: 632 ± 21 M: 194 ± 21 R: 3423 ± 225	C: 0 ± 0 M: 2990 ± 94 R: 32914 ± 1023	C: 0 ± 0 M: 3630 ± 358 R: 39956 ± 3931
	Mix.	C: 3735 ± 38 M: 0 ± 0 R: 7490 ± 74	C: 1439 ± 36 M: 0 ± 0 R: 2898 ± 63	C: 430 ± 13 M: 0 ± 0 R: 879 ± 21	C: 55 ± 3 M: 0 ± 0 R: 130 ± 7	C: 6 ± 3 M: 0 ± 0 R: 31 ± 7	C: 0 ± 0 M: 0 ± 0 R: 20 ± 11	C: 0 ± 0 M: 0 ± 0 **R: 20 ± 11**	C: 0 ± 0 M: 0 ± 0 R: 20 ± 11
50	Unif.	C: 4979 ± 5 M: 0 ± 0 R: 9983 ± 4	C: 4964 ± 6 M: 0 ± 0 R: 9953 ± 6	C: 4936 ± 7 M: 0 ± 0 R: 9898 ± 8	C: 4720 ± 9 M: 0 ± 0 R: 9464 ± 18	C: 4119 ± 20 M: 0 ± 0 R: 8262 ± 41	C: 2825 ± 13 M: 42 ± 8 **R: 6135 ± 73**	C: 0 ± 0 M: 3502 ± 53 R: 38549 ± 582	C: 0 ± 0 M: 3747 ± 99 R: 41246 ± 1091
	Mix.	C: 4990 ± 2 M: 0 ± 0 R: 10000 ± 0	C: 4990 ± 2 M: 0 ± 0 R: 10000 ± 0	C: 4971 ± 4 M: 0 ± 0 R: 9961 ± 12	C: 1650 ± 28 M: 0 ± 0 R: 3319 ± 56	C: 109 ± 7 M: 0 ± 0 R: 237 ± 15	C: 5 ± 2 M: 0 ± 0 R: 28 ± 5	C: 0 ± 0 M: 0 ± 0 **R: 19 ± 5**	C: 0 ± 0 M: 0 ± 0 R: 19 ± 5

F.1.2 Evaluation of CC and Comparison with GHC

Then, we evaluate the performance of the CC algorithm described in Section 4.2. Its exploration phase length is determined by the threshold

$$\hat{m}(t) = \frac{C\sigma^2}{\hat{\delta}^2_{\min}(t)}(d + 2\log T),$$

where $\sigma = 0.1$ and $C = 108$. As discussed in Appendix E.5, this rule is best suited for the Euclidean setting $\mathcal{E} = \mathcal{I}^d$. We therefore treat C as a tunable parameter to find a practical balance in all settings. The results for various values of C are reported in Tables 3 and 4. To allow a direct comparison with the performance of previous algorithms, we only report in "C" the number of expert calls after each label has been observed at least once (this corresponds to the number of expert calls in the second phase for the GHC and SKM algorithms).

Interestingly, our experiments suggest that a much more aggressive choice for C is preferable in practice, even in the hypercube setting where the theoretical analysis suggested $C = 108$. We also observe that as for the other algorithms, in the mixture settings, performance generally improves with dimension. Additionally, even in low dimensions, performance is poor in the uniform setting, which is expected since CC is tailored to settings where the sample mean of expert-labeled queries is a reliable estimate of the corresponding seed.

Table 3: Performance Metrics of CC on $\mathcal{I}^d$ ($T = 5000$).

Dim	Dist.	$C = 0.25$	$C = 1$	$C = 5$	$C = 10$	$C = 25$	$C = 50$	$C = 100$	$C = 108$
1	Unif.	C: 29 ± 20 M: 549 ± 38 **R: 6123 ± 390**	C: 98 ± 12 M: 541 ± 29 R: 6174 ± 333	C: 445 ± 50 M: 483 ± 40 R: 6231 ± 376	C: 851 ± 138 M: 447 ± 43 R: 6650 ± 243	C: 2120 ± 83 M: 310 ± 21 R: 7676 ± 113	C: 4173 ± 114 M: 90 ± 16 R: 9364 ± 94	C: 4985 ± 5 M: 0 ± 0 R: 10000 ± 0	C: 4985 ± 5 M: 0 ± 0 R: 10000 ± 0
	Mix.	C: 19 ± 9 M: 386 ± 71 **R: 4305 ± 770**	C: 71 ± 16 M: 407 ± 65 R: 4643 ± 718	C: 329 ± 22 M: 349 ± 30 R: 4517 ± 320	C: 657 ± 37 M: 309 ± 14 R: 4734 ± 156	C: 1566 ± 73 M: 252 ± 22 R: 5923 ± 163	C: 3119 ± 146 M: 141 ± 11 R: 7812 ± 246	C: 4990 ± 2 M: 0 ± 0 R: 10000 ± 0	C: 4990 ± 2 M: 0 ± 0 R: 10000 ± 0
4	Unif.	C: 0 ± 0 M: 2080 ± 344 R: 22913 ± 3771	C: 3 ± 3 M: 1890 ± 408 R: 20827 ± 4476	C: 60 ± 29 M: 1184 ± 170 R: 13175 ± 1814	C: 123 ± 26 M: 1017 ± 170 R: 11464 ± 1843	C: 333 ± 44 M: 825 ± 81 R: 9771 ± 857	C: 627 ± 51 M: 773 ± 44 R: 9785 ± 448	C: 1307 ± 101 M: 632 ± 34 R: 9598 ± 390	C: 1411 ± 113 M: 610 ± 44 **R: 9563 ± 486**
	Mix.	C: 0 ± 1 M: 252 ± 89 R: 2802 ± 976	C: 9 ± 4 M: 178 ± 53 R: 2006 ± 587	C: 49 ± 12 M: 98 ± 20 R: 1207 ± 239	C: 102 ± 23 M: 76 ± 11 **R: 1064 ± 142**	C: 238 ± 30 M: 70 ± 8 R: 1279 ± 124	C: 491 ± 52 M: 51 ± 8 R: 1568 ± 174	C: 944 ± 58 M: 43 ± 9 R: 2395 ± 193	C: 1037 ± 47 M: 43 ± 8 R: 2571 ± 150
10	Unif.	C: 0 ± 0 M: 2481 ± 356 R: 27328 ± 3902	C: 0 ± 0 M: 2481 ± 356 R: 27328 ± 3902	C: 45 ± 17 M: 1657 ± 205 R: 18348 ± 2213	C: 68 ± 20 M: 1452 ± 205 R: 16143 ± 2212	C: 199 ± 43 M: 1034 ± 93 R: 11808 ± 976	C: 395 ± 51 M: 893 ± 100 R: 10644 ± 1026	C: 834 ± 92 M: 735 ± 45 **R: 9782 ± 462**	C: 880 ± 95 M: 731 ± 46 R: 9831 ± 510
	Mix.	C: 0 ± 0 M: 1 ± 1 **R: 33 ± 7**	C: 0 ± 0 M: 1 ± 1 R: 33 ± 7	C: 8 ± 4 M: 0 ± 0 R: 44 ± 7	C: 25 ± 5 M: 0 ± 0 R: 75 ± 9	C: 62 ± 13 M: 0 ± 0 R: 149 ± 24	C: 116 ± 13 M: 0 ± 0 R: 257 ± 27	C: 229 ± 10 M: 0 ± 0 R: 483 ± 24	C: 248 ± 7 M: 0 ± 0 R: 520 ± 14
50	Unif.	C: 0 ± 0 M: 3020 ± 115 R: 33260 ± 1267	C: 0 ± 0 M: 3020 ± 115 R: 33260 ± 1267	C: 35 ± 18 M: 2579 ± 205 R: 28483 ± 2223	C: 160 ± 78 M: 1985 ± 364 R: 22197 ± 3857	C: 466 ± 82 M: 1277 ± 91 R: 15027 ± 887	C: 955 ± 96 M: 909 ± 52 R: 11957 ± 541	C: 2176 ± 164 M: 497 ± 44 R: 9866 ± 252	C: 2380 ± 203 M: 436 ± 38 **R: 9597 ± 143**
	Mix.	C: 0 ± 0 M: 0 ± 0 **R: 24 ± 9**	C: 0 ± 0 M: 0 ± 0 R: 24 ± 9	C: 0 ± 0 M: 0 ± 0 R: 24 ± 9	C: 7 ± 4 M: 0 ± 0 R: 39 ± 15	C: 24 ± 8 M: 0 ± 0 R: 72 ± 15	C: 43 ± 12 M: 0 ± 0 R: 110 ± 20	C: 81 ± 8 M: 0 ± 0 R: 186 ± 17	C: 90 ± 12 M: 0 ± 0 R: 205 ± 21

Finally, we provide in Tables 5 and 6 detailed results for the nearest-query distance modification of GHC introduced in Appendix F.2.3, where the distance to the closest point in each class is used instead of the distance to convex hulls. As expected, this version performs slightly worse than GHC, but it still outperforms CC in almost all settings, while being just as computationally efficient. Overall, these numerical results suggest using GHC when additional computation is affordable, and otherwise using nearest-query GHC. We summarize the results of all synthetic experiments in Table 7.

Table 4: Performance Metrics of `CC` on $\mathcal{S}^{d-1}$ ($T = 5000$).

Dim	Dist.	$C = 0.25$	$C = 1$	$C = 5$	$C = 10$	$C = 25$	$C = 50$	$C = 100$	$C = 108$
2	Unif.	C: 0 ± 0 M: 983 ± 117 R: 10877 ± 1270	C: 25 ± 17 M: 947 ± 68 R: 10525 ± 778	C: 131 ± 61 M: 905 ± 88 R: 10274 ± 920	C: 299 ± 63 M: 829 ± 69 R: 9775 ± 714	C: 807 ± 85 M: 697 ± 54 **R: 9347 ± 472**	C: 1574 ± 104 M: 579 ± 39 R: 9573 ± 354	C: 2895 ± 213 M: 355 ± 27 R: 9754 ± 319	C: 3135 ± 211 M: 316 ± 26 R: 9810 ± 331
	Mix.	C: 10 ± 5 M: 256 ± 122 R: 2853 ± 1341	C: 46 ± 24 M: 171 ± 169 R: 1993 ± 1825	C: 223 ± 21 M: 45 ± 15 **R: 963 ± 173**	C: 422 ± 67 M: 33 ± 8 R: 1224 ± 176	C: 1020 ± 94 M: 26 ± 3 R: 2347 ± 209	C: 1976 ± 75 M: 19 ± 4 R: 4177 ± 115	C: 3962 ± 125 M: 6 ± 2 R: 8011 ± 230	C: 4304 ± 103 M: 4 ± 1 R: 8666 ± 207
4	Unif.	C: 0 ± 0 M: 1714 ± 437 R: 18873 ± 4810	C: 0 ± 0 M: 1714 ± 437 R: 18873 ± 4810	C: 8 ± 4 M: 1359 ± 308 R: 14987 ± 3391	C: 25 ± 13 M: 959 ± 202 R: 10622 ± 2195	C: 42 ± 13 M: 823 ± 123 R: 9160 ± 1328	C: 79 ± 15 M: 635 ± 83 R: 7165 ± 904	C: 152 ± 21 M: 529 ± 98 R: 6148 ± 1053	C: 159 ± 18 M: 524 ± 83 **R: 6099 ± 889**
	Mix.	C: 0 ± 0 M: 0 ± 0 **R: 22 ± 7**	C: 0 ± 0 M: 0 ± 0 R: 22 ± 7	C: 4 ± 4 M: 0 ± 0 R: 30 ± 9	C: 12 ± 6 M: 0 ± 0 R: 46 ± 12	C: 34 ± 10 M: 0 ± 0 R: 89 ± 15	C: 75 ± 17 M: 0 ± 0 R: 172 ± 34	C: 140 ± 16 M: 0 ± 0 R: 301 ± 30	C: 154 ± 14 M: 0 ± 0 R: 329 ± 25
10	Unif.	C: 0 ± 0 M: 2920 ± 269 R: 32141 ± 2957	C: 0 ± 0 M: 2920 ± 269 R: 32141 ± 2957	C: 11 ± 7 M: 2271 ± 434 R: 25020 ± 4758	C: 52 ± 15 M: 1563 ± 238 R: 17316 ± 2602	C: 115 ± 26 M: 1096 ± 37 R: 12300 ± 379	C: 229 ± 34 M: 796 ± 121 R: 9235 ± 1272	C: 421 ± 25 M: 654 ± 52 R: 8060 ± 536	C: 460 ± 29 M: 611 ± 54 **R: 7657 ± 544**
	Mix.	C: 0 ± 0 M: 0 ± 0 **R: 22 ± 9**	C: 0 ± 0 M: 0 ± 0 R: 22 ± 9	C: 1 ± 2 M: 0 ± 0 R: 24 ± 12	C: 12 ± 3 M: 0 ± 0 R: 46 ± 9	C: 29 ± 9 M: 0 ± 0 R: 79 ± 12	C: 55 ± 4 M: 0 ± 0 R: 132 ± 7	C: 114 ± 6 M: 0 ± 0 R: 249 ± 14	C: 123 ± 9 M: 0 ± 0 R: 268 ± 14
50	Unif.	C: 0 ± 0 M: 3672 ± 197 R: 40407 ± 2165	C: 3 ± 5 M: 3604 ± 287 R: 39669 ± 3151	C: 190 ± 20 M: 1882 ± 125 R: 21104 ± 1337	C: 422 ± 13 M: 1338 ± 52 R: 15576 ± 552	C: 1099 ± 42 M: 760 ± 31 R: 10582 ± 284	C: 2200 ± 82 M: 412 ± 38 **R: 8947 ± 296**	C: 4632 ± 53 M: 39 ± 7 R: 9715 ± 71	C: 4947 ± 62 M: 2 ± 3 R: 9938 ± 89
	Mix.	C: 0 ± 0 M: 0 ± 0 **R: 26 ± 12**	C: 0 ± 0 M: 0 ± 0 R: 26 ± 12	C: 14 ± 5 M: 0 ± 0 R: 54 ± 2	C: 32 ± 8 M: 0 ± 0 R: 89 ± 7	C: 82 ± 18 M: 0 ± 0 R: 189 ± 32	C: 169 ± 27 M: 0 ± 0 R: 364 ± 47	C: 334 ± 19 M: 0 ± 0 R: 694 ± 42	C: 359 ± 24 M: 0 ± 0 R: 743 ± 49

Table 5: Performance Metrics on $[0, 1]^d$ for `GHC` with nearest-query distance ($T = 5000$).

Dim	Dist.	$\tau = 0.00$	$\tau = 0.20$	$\tau = 0.40$	$\tau = 0.60$	$\tau = 0.80$	$\tau = 0.95$	$\tau = 1.00$
1	Unif.	C: 0 ± 0 M: 375 ± 126 R: 4160 ± 1384	C: 124 ± 3 M: 2 ± 2 R: 302 ± 19	C: 59 ± 3 M: 3 ± 3 **R: 186 ± 33**	C: 35 ± 3 M: 12 ± 3 R: 237 ± 33	C: 21 ± 4 M: 34 ± 10 R: 445 ± 110	C: 12 ± 2 M: 101 ± 33 R: 1168 ± 363	C: 0 ± 0 M: 375 ± 126 R: 4160 ± 1384
	Mix.	C: 0 ± 0 M: 823 ± 389 R: 9071 ± 4275	C: 136 ± 9 M: 1 ± 1 R: 307 ± 21	C: 65 ± 5 M: 9 ± 5 **R: 245 ± 50**	C: 40 ± 1 M: 21 ± 7 R: 333 ± 78	C: 27 ± 1 M: 42 ± 14 R: 532 ± 151	C: 15 ± 2 M: 155 ± 31 R: 1753 ± 337	C: 0 ± 0 M: 823 ± 389 R: 9071 ± 4275
4	Unif.	C: 0 ± 0 M: 2155 ± 181 R: 23729 ± 1988	C: 4723 ± 8 M: 0 ± 0 R: 9477 ± 17	C: 3530 ± 16 M: 12 ± 3 R: 7224 ± 65	C: 2190 ± 24 M: 90 ± 6 **R: 5396 ± 95**	C: 1068 ± 29 M: 348 ± 16 R: 5988 ± 188	C: 319 ± 16 M: 894 ± 30 R: 10496 ± 305	C: 0 ± 0 M: 2155 ± 181 R: 23729 ± 1988
	Mix.	C: 0 ± 0 M: 422 ± 199 R: 4666 ± 2188	C: 3450 ± 45 M: 0 ± 0 R: 6928 ± 81	C: 1575 ± 31 M: 2 ± 1 R: 3196 ± 49	C: 760 ± 26 M: 13 ± 4 R: 1690 ± 79	C: 343 ± 22 M: 59 ± 7 **R: 1366 ± 99**	C: 98 ± 9 M: 187 ± 24 R: 2283 ± 267	C: 0 ± 0 M: 422 ± 199 R: 4666 ± 2188
10	Unif.	C: 0 ± 0 M: 2731 ± 236 R: 30075 ± 2577	C: 4983 ± 10 M: 0 ± 0 R: 10000 ± 0	C: 4972 ± 11 M: 0 ± 0 R: 9978 ± 2	C: 4704 ± 24 M: 18 ± 4 R: 9641 ± 38	C: 3235 ± 22 M: 276 ± 7 **R: 9544 ± 91**	C: 1027 ± 29 M: 1242 ± 24 R: 15751 ± 316	C: 0 ± 0 M: 2731 ± 236 R: 30075 ± 2577
	Mix.	C: 0 ± 0 M: 1 ± 1 R: 40 ± 12	C: 4635 ± 21 M: 0 ± 0 R: 9294 ± 45	C: 1081 ± 24 M: 0 ± 0 R: 2187 ± 54	C: 174 ± 5 M: 0 ± 0 R: 372 ± 13	C: 26 ± 3 M: 0 ± 0 R: 78 ± 15	C: 1 ± 1 M: 0 ± 0 **R: 29 ± 10**	C: 0 ± 0 M: 1 ± 1 R: 40 ± 12
50	Unif.	C: 0 ± 0 M: 3344 ± 52 R: 36828 ± 570	C: 4978 ± 3 M: 0 ± 0 R: 10000 ± 0	C: 4978 ± 3 M: 0 ± 0 R: 10000 ± 0	C: 4978 ± 3 M: 0 ± 0 **R: 10000 ± 0**	C: 4974 ± 4 M: 2 ± 1 R: 10019 ± 8	C: 3551 ± 32 M: 738 ± 25 R: 15268 ± 221	C: 0 ± 0 M: 3344 ± 52 R: 36828 ± 570
	Mix.	C: 0 ± 0 M: 0 ± 0 R: 24 ± 9	C: 4987 ± 5 M: 0 ± 0 R: 9998 ± 3	C: 254 ± 17 M: 0 ± 0 R: 532 ± 31	C: 0 ± 0 M: 0 ± 0 R: 24 ± 9	C: 0 ± 0 M: 0 ± 0 R: 24 ± 9	C: 0 ± 0 M: 0 ± 0 R: 24 ± 9	C: 0 ± 0 M: 0 ± 0 **R: 24 ± 9**

Table 6: Performance Metrics on $\mathcal{S}^{d-1}$ for `GHC` with nearest-query distance ($T = 5000$).

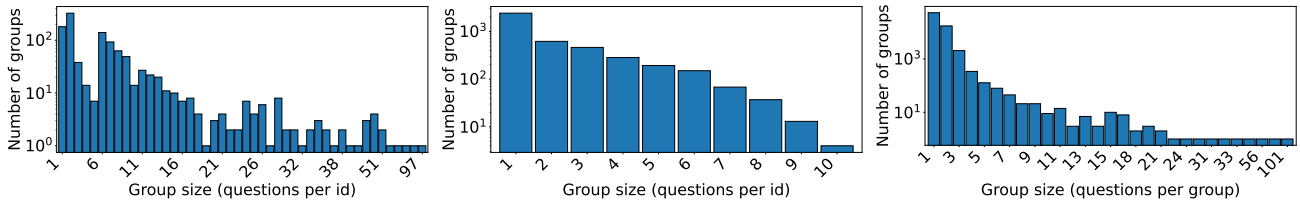
Dim	**Dist.**	$\tau = 0.00$	$\tau = 0.20$	$\tau = 0.40$	$\tau = 0.60$	$\tau = 0.80$	$\tau = 1.00$
2	Unif.	C: 0 ± 0 M: 315 ± 148 R: 3524 ± 1630	C: 142 ± 7 M: 2 ± 1 R: 371 ± 15	C: 69 ± 5 M: 5 ± 3 **R: 254 ± 42**	C: 39 ± 3 M: 18 ± 6 R: 340 ± 73	C: 24 ± 2 M: 44 ± 11 R: 596 ± 144	C: 0 ± 0 M: 315 ± 148 R: 3524 ± 1630
	Mix.	C: 0 ± 0 M: 345 ± 77 R: 3812 ± 849	C: 100 ± 5 M: 1 ± 1 R: 231 ± 17	C: 50 ± 5 M: 3 ± 1 **R: 155 ± 19**	C: 30 ± 2 M: 9 ± 3 R: 180 ± 30	C: 18 ± 3 M: 22 ± 9 R: 302 ± 100	C: 0 ± 0 M: 345 ± 77 R: 3812 ± 849
4	Unif.	C: 0 ± 0 M: 1939 ± 308 R: 21348 ± 3381	C: 4117 ± 19 M: 1 ± 1 R: 8272 ± 46	C: 2526 ± 21 M: 22 ± 6 R: 5319 ± 79	C: 1425 ± 9 M: 103 ± 11 **R: 4005 ± 126**	C: 696 ± 9 M: 307 ± 9 R: 4793 ± 92	C: 0 ± 0 M: 1939 ± 308 R: 21348 ± 3381
	Mix.	C: 0 ± 0 M: 0 ± 0 R: 22 ± 7	C: 121 ± 10 M: 0 ± 0 R: 264 ± 19	C: 20 ± 3 M: 0 ± 0 R: 62 ± 11	C: 4 ± 3 M: 0 ± 0 R: 31 ± 3	C: 0 ± 0 M: 0 ± 0 R: 22 ± 6	C: 0 ± 0 M: 0 ± 0 **R: 22 ± 7**
10	Unif.	C: 0 ± 0 M: 3039 ± 213 R: 33446 ± 2337	C: 4990 ± 3 M: 0 ± 0 R: 10000 ± 0	C: 4983 ± 5 M: 1 ± 0 R: 9993 ± 7	C: 4766 ± 7 M: 17 ± 5 **R: 9743 ± 51**	C: 3412 ± 25 M: 284 ± 12 R: 9971 ± 142	C: 0 ± 0 M: 3039 ± 213 R: 33446 ± 2337
	Mix.	C: 0 ± 0 M: 0 ± 0 R: 22 ± 9	C: 1769 ± 45 M: 0 ± 0 R: 3560 ± 86	C: 105 ± 5 M: 0 ± 0 R: 231 ± 7	C: 3 ± 2 M: 0 ± 0 R: 28 ± 8	C: 0 ± 0 M: 0 ± 0 R: 22 ± 9	C: 0 ± 0 M: 0 ± 0 **R: 22 ± 9**
50	Unif.	C: 0 ± 0 M: 3549 ± 83 R: 39061 ± 911	C: 4990 ± 2 M: 0 ± 0 R: 10000 ± 0	C: 4990 ± 2 M: 0 ± 0 R: 10000 ± 0	C: 4990 ± 2 M: 0 ± 0 **R: 10000 ± 0**	C: 4987 ± 2 M: 1 ± 1 R: 10005 ± 10	C: 0 ± 0 M: 3549 ± 83 R: 39061 ± 911
	Mix.	C: 0 ± 0 M: 0 ± 0 R: 26 ± 12	C: 4987 ± 6 M: 0 ± 0 R: 10000 ± 0	C: 4983 ± 7 M: 0 ± 0 R: 9992 ± 1	C: 1287 ± 44 M: 0 ± 0 R: 2600 ± 97	C: 12 ± 3 M: 0 ± 0 R: 49 ± 10	C: 0 ± 0 M: 0 ± 0 **R: 26 ± 12**

Table 7: Voronoi regret of all algorithms for each experimental setup ($T = 5000$). Results for CC and GHC are reported for the best-performing threshold for that setup.

$\mathcal{E}$	**Dim.**	**Dist.**	CC	CHC	GHC	**Nearest-query GHC**	**SKM**
$\mathcal{I}^d$	1	Unif.	6123 ± 390	142 ± 11	**110 ± 7**	186 ± 33	14128 ± 4275
		Mix.	4305 ± 770	130 ± 21	**106 ± 18**	245 ± 50	11141 ± 3741
	4	Unif.	9563 ± 486	2972 ± 72	**1593 ± 35**	5396 ± 95	30055 ± 3659
		Mix.	1064 ± 142	2337 ± 19	**573 ± 52**	1366 ± 99	793 ± 94
	10	Unif.	9782 ± 462	9489 ± 35	**5233 ± 94**	9544 ± 91	38900 ± 4268
		Mix.	33 ± 7	8821 ± 48	23 ± 4	29 ± 10	**20 ± 5**
	50	Unif.	9597 ± 143	10000 ± 0	**9559 ± 36**	10000 ± 0	42659 ± 3378
		Mix.	24 ± 9	10000 ± 0	**23 ± 5**	24 ± 9	**23 ± 5**
$\mathcal{S}^{d-1}$	2	Unif.	9347 ± 472	132 ± 12	**125 ± 6**	254 ± 42	26804 ± 1923
		Mix.	963 ± 173	138 ± 8	**104 ± 13**	155 ± 19	791 ± 645
	4	Unif.	6099 ± 889	1225 ± 31	**836 ± 40**	4005 ± 126	25645 ± 8157
		Mix.	**22 ± 7**	1083 ± 56	**22 ± 7**	**22 ± 7**	**22 ± 7**
	10	Unif.	7657 ± 544	4878 ± 150	**2550 ± 123**	9743 ± 51	39956 ± 3931
		Mix.	22 ± 9	7490 ± 74	**20 ± 11**	22 ± 9	**20 ± 11**
	50	Unif.	8947 ± 296	9983 ± 4	**6135 ± 73**	10000 ± 0	41246 ± 1091
		Mix.	26 ± 12	10000 ± 0	**19 ± 5**	26 ± 12	**19 ± 5**

F.2 Real-World Experiments

F.2.1 Real-World Datasets



(a) Quora Question Groups (b) ComQA (c) CQADupStack

Figure 7: Distributions of group sizes: number of questions per group (with the same answer) within the datasets.

In this section, we describe the question-answer datasets used in our real-world experiments. We say that two questions belong to the same *question group* if a single answer can adequately address both, i.e., if their correct answer label is identical.

Quora Question Groups (QQG): Quora Question Groups is derived from the Quora Question Pairs dataset (Wang et al., 2017). The original dataset comprises over 400,000 question pairs, each annotated with a binary label indicating whether the questions are paraphrases of each other. To transform this pairwise paraphrase data into coherent question groups, we represented the dataset as a graph where nodes correspond to questions and edges indicate paraphrase relationships.

To construct the Quora Question Groups dataset, we first extracted all connected components from the graph, treating each component as a candidate group. From these, we sampled a subset with the goal of obtaining approximately 1,200 groups, favoring components with larger group sizes. We then manually reviewed and cleaned the selected groups to ensure that all questions within a group could be addressed by the same answer. This involved discarding noisy or weakly related links and merging or splitting groups where appropriate. Throughout this process, we preserved the structural assumptions of the original dataset while focusing on high-quality, semantically coherent groupings. At the end of this process, we obtained $1,103$ distinct groups comprising a total of $7,365$ curated questions.

ComQA[6] **(Abujabal et al., 2019)**: ComQA consists of 11,214 English questions collected from the WikiAnswers forum and grouped into 4,834 paraphrase clusters by crowd workers. By design, ComQA emphasizes reasoning-intensive phenomena—namely compositional queries, explicit temporal constraints, and comparative or superlative constructions—making it a rigorous benchmark for research on complex factoid question answering. The middle panel of Fig. 7 presents the group size distribution of this dataset.

CQADupStack [7] **(Hoogeveen et al., 2015)**: CQADupStack is a public benchmark available to researchers working on question-answering learning tasks. It consists of complete threads from twelve StackExchange subforums (android, english, gis, mathematica, physics, programmers, statistics, superuser, tex, unix, webmasters, wordpress) annotated with community-marked duplicate links. To evaluate our classification approach across distinct expert domains, we apply it to the Mathematica, Physics, and Statistics parts of the dataset. The right panel of Fig. 7 presents the group size distribution of this dataset. The subset of the dataset we use comprises 99,785 questions organized into 74,519 groups. CQADupStack is the most challenging dataset used in our experiments, because of its relatively small average group size.

F.2.2 Large Language Models

In this section, we describe the Large Language Models (LLMs) selected for our experiments, along with the way they were trained and fine-tuned. We chose three models that are widely recognized in the literature and frequently cited due to their strong performance across a range of Natural Language Processing tasks.

E5 (Wang et al., 2022): EmbEddings from bidirEctional Encoder rEpresentations (E5) is available in three versions. The first version is initialized with MiniLM (Wang et al., 2020), the second is initialized with BERT-base

[6] https://paperswithcode.com/dataset/comqa
[7] https://github.com/D1Doris/CQADupStack

(Reimers and Gurevych, 2019), and the third is initialized with BERT-large. This model follows a bi-encoder architecture, where both the query and passage encoders are initialized with BERT. The training process consists of two stages. The first stage, called "unsupervised", uses a large number of unlabeled data, including title and passage pairs from Wikipedia, questions and answers from Reddit, and more. The InfoNCE contrastive loss (van den Oord et al., 2018) is employed to minimize the distance between related queries and passages, while maximizing the distance between unrelated queries and passages. The second stage involves training the model on high-quality human-annotated data, such as NLI (Gao et al., 2021), MS-MARCO (Bajaj et al., 2016), and Natural Questions (Karpukhin et al., 2020). During this stage, the model is trained with a loss that combines the KL divergence between the probability distribution of the label, as given by a cross-encoder teacher model, and the probability distribution generated by our E5 student model, along with the InfoNCE contrastive loss. This second stage further improves model performance on benchmarks such as BEIR (Thakur et al., 2021) and MTEB (Muennighoff et al., 2022).

NOMIC (Nussbaum et al., 2024): Nomic is initialized from BERT and modified to address long-context retrieval. Nomic consists of 100 million parameters and supports a sequence length of up to 2048. Nomic's pretraining is divided into three stages. The first stage focuses on Masked Language Modeling to learn longer sequence representations. The subsequent stages are supervised and unsupervised, both employing the InfoNCE contrastive loss. This model was trained on a significantly larger dataset compared to the previous versions, encompassing both supervised and unsupervised phases. Nomic uses task-specific prefixes—such as search, search query document, classification, and clustering—to distinguish between the behaviors of each task. For the purpose of our work, we used the clustering prefix.

Mistral_E5 (Wang et al., 2024): This is the first unidirectional decoder architecture we use for our work. The model is initialized from Mistral 7B Jiang et al. (2023) and consists of 7 billion parameters. The model takes as input the query q^+ and the task_definition and generates the instruction template:

$$q^+_{inst} = \text{'Instruct: \{task\_definition\} \textbackslash n Query: } \{q^+\}\text{'}$$

Where the task definition is:

"Given a web search query, retrieve relevant search queries that paraphrase the query."

The query and document instruction templates are then passed to the LLM. We obtain the query and document embeddings, h^+_q and h^+_d, from the last layer of the LLM at the [EOS] position. The model was trained on a large corpus of both original and synthetic data. The synthetic data was generated using advanced LLMs such as GPT-4.

F.2.3 Additional Real-World Experiments

In this section, we evaluate the performance of `GHC` (Algorithm 2), `CC` and a sequential k-means baseline using the datasets and LLMs described in Sections F.2.1 and F.2.2 respectively. The models we utilize are typically trained using a loss function based on cosine similarity. Since they output unit-normalized embeddings, maximizing cosine similarity is equivalent to minimizing Euclidean distance. Hence, the reliance of `GHC` on Euclidean distance naturally leverages the models' cosine-trained embeddings. As explained in Section 5, we always initialize the algorithms by manually picking an example question from each group. For `GHC`, this skips the first phase where `CHC` is applied until every group is represented. We then run our algorithms on the remaining questions in the dataset considered.

To align with the Retrieval-Augmented Generation (RAG) literature (Lewis et al., 2020) and improve computational efficiency, it is sensible to consider the following modification of the `GHC` algorithm: instead of computing distances to the convex hulls $\hat{\mathcal{C}}_{i,t}$ of each set of expert queries $\mathcal{Q}_{i,t}$, we compute the distance to the nearest query in $\mathcal{Q}_{i,t}$. We call "nearest-query" this distance. We compare the performance of `GHC` with three distances: the nearest-query distance, the distance to Euclidean convex hulls, and the distance to spherical convex hulls.

In Table 8, we present the average cumulative regret for a range of thresholds across all three retriever models—E5, Nomic, and Mistral_E5 on the QQG dataset. We observe that changing the distance does not meaningfully affect the regret. This is expected in high dimension, as the query hulls $\hat{\mathcal{C}}_{i,t}$ then typically occupy a negligible portion of the space, see discussion below Corollary 4.2. We have made the same observation for the two other

datasets. We also observe that the performance of each model is highly sensitive to the threshold, with optimal performance often concentrated around specific values (namely 0.85–0.90). This suggests the importance of careful threshold tuning for maximizing algorithm performance. Lastly, Mistral_E5, owing to its recent development and high-dimensional latent space, demonstrates lower cumulative regret across all distance metrics for the QQG dataset. This effect is particularly pronounced at higher thresholds, indicating its robustness to the choice of metric and its superior capacity for generalization across different geometric interpretations of similarity.

Table 8: Average cumulative regret of GHC using different distance metrics after 6262 steps on QQG dataset.

Threshold (τ)	**0.10**	**0.20**	**0.40**	**0.60**	**0.70**	**0.80**	**0.85**	**0.90**	**0.95**	**1.00**
Nearest-query										
E5	6722	6632	5750	3756	2479	1365	1132	**1062**	1964	5822
Nomic	6790	6624	6709	5499	4098	2398	1677	**1239**	1442	5852
Mistral_E5	8192	6932	6401	4616	3157	1596	1196	**921**	1269	3109
Spherical hull										
E5	7942	7582	6060	3398	2163	1422	**1101**	1192	1927	5734
Nomic	7920	7936	7706	5410	3608	1996	1442	**1140**	1422	4928
Mistral_E5	7888	7735	6723	4137	2637	1414	1062	**1002**	1293	4620
Euclidean hull										
E5	7949	7628	6073	3424	2238	1358	**1165**	1197	1773	5016
Nomic	7874	7916	7818	5652	3849	2178	1555	**1215**	1445	6457
Mistral_E5	7898	7742	6749	4217	2705	1476	1074	**836**	1204	4048

Next, we evaluate the sequential k-means (SKM) baseline described in Appendix F.1.1 and the tunable version of CC described in Appendix F.1.2, for each embedding model on the QQG dataset. The length of the exploration phase of CC is determined by the threshold

$$\hat{m}(t) = \frac{C\sigma^2}{\hat{\delta}^2_{\min}(t)}(d + 2\log T)$$

where we have chosen $\sigma = 0.1$ to allow easier comparison with our findings on synthetic datasets (note that the value of σ can be chosen arbitrarily as it amounts to scaling C by a fixed constant).

We report the regrets of both algorithms in Table 9. We observe that CC has two distinct behaviors depending on the value of C: if C is sufficiently small, $\hat{m}(t)$ typically remains bounded by 1 at each round t. As the algorithm is initialized with one example per group, this means that the expert is never called and the regret solely comes from the wrong guesses. If instead C is sufficiently large, $\hat{m}(t)$ typically remains larger than 1 and CC calls the expert up until another query for each group is observed. Given that on the QQG dataset, the number of groups $N = 1103$ is non-negligible compared to size of the dataset $T = 6262$, this results in CC calling the expert at each round t and high regret. This highlights that CC is poorly suited to settings where T is comparable to $N \log N$, as mentioned in Section 4.3. The SKM baseline performs better than CC for two out of three embedding models. Both algorithms are still significantly outperformed by properly tuned GHC using any of the distance metrics reported in Table 8.

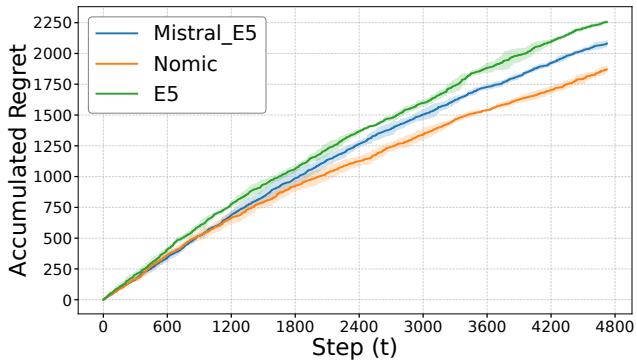
Table 9: Average cumulative regret of CC and SKM baseline after 6262 steps on QQG dataset.

Algorithm	CC					**SKM**
Threshold (C)	**0.0**	**0.01**	**0.025**	**0.05**	**0.1**	**N/A**
E5	**5533**	5533	12524	12524	12524	3531
Nomic	**5654**	5654	5654	5654	12524	6204
Mistral_E5	**3289**	12524	12524	12524	12524	2442

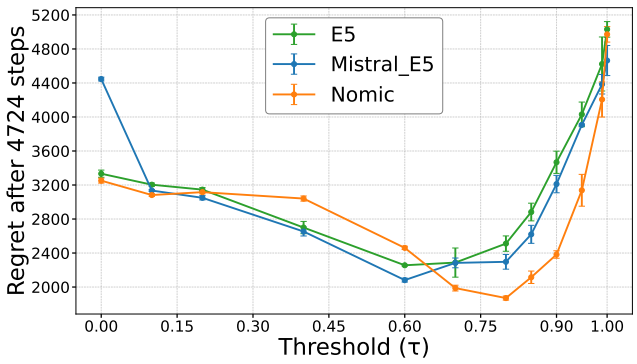
In Fig. 8, we present the performance of our models on the ComQA and CQADupStack datasets using the nearest-query distance. Nomic surprisingly outperforms Mistral_E5 on both datasets, achieving a lower accumulated regret of 1,870 compared to 2,081 on ComQA, and 7,549 compared to 7,586 on CQADupStack. We also observe that Mistral_E5 accumulates less regret than E5-large, which is an expected outcome given the design and capabilities of the two models.

Furthermore, by comparing the performance of the models between datasets, we can observe that the task becomes more challenging as the average number of questions per group decreases. Indeed, the lowest accumulated regret for QQG was 921 after handling around 6,200 questions, whereas it reached 1,870 (resp. 2,070) for ComQA (resp. CQADupStack) after handling 4,800 questions only. We also observe that the optimal threshold for QQG is 0.9, while it is 0.8 for ComQA. Among all datasets, CQADupStack is in principle the most challenging, with an average of 1.34 questions per group. This is reflected in the optimal threshold, which is significantly lower at 0.3. This suggests that the fewer the questions per group, the lower the confidence our model should have in its predictions.

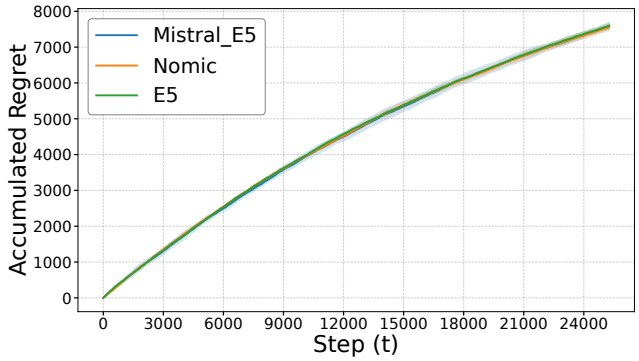
We note that even in the challenging datasets, ComQA and CQADupStack, the growth of the accumulated regret decreases over time. This indicates that the algorithm is learning effectively and consistently improving its performance over time. More specifically, in the CQADupStack dataset, the average regret per step decreases from about one-half after 3,000 steps to about one-third after 24,000 steps, indicating a clear improvement in performance over time even in the most challenging scenario. This trend is observed across all our experiments.



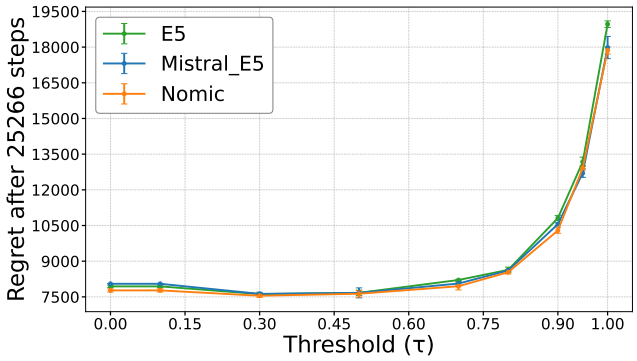
(a) Regret vs. step, ComQA.



(b) Regret vs. threshold ($T = 4724$), ComQA.



(c) Regret vs. step, CQADupStack.



(d) Regret vs. threshold ($T = 25266$), CQADupStack.

Figure 8: Comparison of `GHC` with different LLMs on ComQA and CQADupStack datasets.

F.3 Compute Resources for Experiments

Throughout our experiments, the only component that required substantial computational resources was the encoding of our datasets using Mistral_E5. As Mistral_E5 is a Large Language Model comprising 7 billion parameters, it necessitates the use of a high-memory GPU for inference, such as the NVIDIA H100 with 80 GB of memory. The encoding of QQG and ComQA takes approximately 20 minutes while the encoding of CQADupStack takes approximately 4 hours. Every other component of the experiments can be executed on a single workstation, provided it is capable of running continuously for up to two days to accommodate the more computationally intensive tasks.