Efficient Near-Optimal Algorithm for Online Shortest Paths in Directed Acyclic Graphs with Bandit Feedback Against Adaptive Adversaries

Arnab Maiti ARNABM2@UW.EDU
University of Washington

Zhiyuan Fan FANZY@MIT.EDU
MIT

Kevin Jamieson JAMIESON@CS.WASHINGTON.EDU
University of Washington

Lillian J. Ratliff RATLIFFL@UW.EDU
University of Washington

Gabriele Farina GFARINA@MIT.EDU
MIT

Editors: Nika Haghtalab and Ankur Moitra

Abstract

In this paper, we study the online shortest path problem in directed acyclic graphs (DAGs) under bandit feedback against an adaptive adversary. Given a DAG $G = (V, E)$ with a source node v_{s} and a sink node v_{t}, let $\mathcal{X} \subseteq \{0,1\}^{|E|}$ denote the set of all paths from v_{s} to v_{t}. At each round t, we select a path $\mathbf{x}_t \in \mathcal{X}$ and receive bandit feedback on our loss $\langle \mathbf{x}_t, \mathbf{y}_t \rangle \in [-1, 1]$, where $\mathbf{y}_t$ is an adversarially chosen loss vector. Our goal is to minimize regret with respect to the best path in hindsight over T rounds. We propose the first computationally efficient algorithm to achieve a near-minimax optimal regret bound of $\widetilde{\mathcal{O}}(\sqrt{|E|T \log |\mathcal{X}|})$ with high probability against any adaptive adversary, where $\widetilde{\mathcal{O}}(\cdot)$ hides logarithmic factors in the number of edges $|E|$. Our algorithm leverages a novel loss estimator and a centroid-based decomposition in a nontrivial manner to attain this regret bound.

As an application, we show that our algorithm for DAGs provides state-of-the-art efficient algorithms for m-sets, extensive-form games, the Colonel Blotto game, shortest walks in directed graphs, hypercubes, and multi-task multi-armed bandits, achieving improved high-probability regret guarantees in all these settings.

Keywords: Directed acyclic graphs, online shortest path, regret minimization, bandit feedback, follow-the-regularized-leader, loss estimators, centroid-decomposition, combinatorial bandits

1. Introduction

Online decision-making is a well-studied area with applications in various domains, including recommendation systems, resource allocation, web ranking, shortest path planning, and portfolio selection (e.g., Lin et al. (2020); Chen et al. (2017); Frigó and Kocsis (2022); Gordon (2006); Das (2014)). In a typical online decision-making problem, a learner interacts with an adversary over multiple rounds. The learner is given a set of arms $\mathcal{X}$, and in each round t, selects an arm $\mathbf{x}_t \in \mathcal{X}$. Simultaneously, an adversary chooses a loss function $\mathbf{y}_t : \mathcal{X} \to \mathbb{R}$. The learner then incurs a loss of $\mathbf{y}_t[\mathbf{x}_t] \in [-1, 1]$ and observes only the incurred loss, not the full loss function $\mathbf{y}_t$. After T rounds of

∗. Arnab Maiti and Zhiyuan Fan contributed equally to this work.

interaction between the learner and the adversary, the learner's regret relative to a fixed arm $\mathbf{x} \in \mathcal{X}$ is defined as

$$R_T(\mathbf{x}) = \sum_{t=1}^{T} \mathbf{y}_t[\mathbf{x}_t] - \sum_{t=1}^{T} \mathbf{y}_t[\mathbf{x}].$$

The learner aims either to minimize the pseudo-regret, defined as $\max_{\mathbf{x}\in\mathcal{X}} \mathbb{E}[R_T(\mathbf{x})]$, or to provide high-probability guarantees on $\max_{\mathbf{x}\in\mathcal{X}} R_T(\mathbf{x})$. The latter is a stronger and more natural notion of regret, as $\mathbb{E}[\max_{\mathbf{x}\in\mathcal{X}} R_T(\mathbf{x})]$ can far exceed the pseudo-regret against an adaptive adversary.

A major breakthrough in this area was achieved by Auer et al. (2002), who proposed the EXP3.P algorithm, which attains a regret of $\widetilde{O}(\sqrt{KT})$ with high probability against an adaptive adversary, where K is the number of arms in $\mathcal{X}$. This bound is optimal, as there exists a minimax lower bound of $\Omega(\sqrt{KT})$ for this problem.

However, one can hope for better guarantees when the loss functions and the set of arms exhibit additional structure. Awerbuch and Kleinberg (2004) took a step in this direction by considering the online shortest path problem in directed acyclic graphs (DAGs) under bandit feedback. In this setting, an adversary assigns loss values to each edge of a given DAG, and the learner must select a path from the source to the sink. Here, the set of arms consists of all such paths, and the loss of a path is defined as the sum of the losses of its edges. Applying EXP3.P to this problem results in regret that scales exponentially with the number of edges in the DAG, as the number of paths can be exponentially large. To overcome this, Awerbuch and Kleinberg (2004) designed an algorithm that achieves a pseudo-regret of $T^{2/3}$ that also scales polynomially with the number of edges. Later, György et al. (2007) extended this result, showing that a regret bound of $T^{2/3}$ can be achieved with high probability against adaptive adversaries, while maintaining polynomial dependence on the number of edges.

Motivated by research on DAGs, a series of works have explored combinatorial linear bandits, where $\mathcal{X} \subseteq \{0,1\}^d$ and $\mathbf{y}_t$ is a linear loss function. Notably, the problem on DAGs is a special case of combinatorial linear bandits. Several algorithms have been developed in this setting, including Geometric Hedge (Dani et al., 2007), ComBand (Cesa-Bianchi and Lugosi, 2012), and EXP2 with John's exploration (Bubeck et al., 2012a). The best known pseudo-regret bound, $\mathcal{O}(\sqrt{dT \log |\mathcal{X}|})$, is achieved by EXP2 with John's exploration. Later, Zimmert and Lattimore (2022) established a high-probability regret bound of $\mathcal{O}(\sqrt{dT \log |\mathcal{X}|})$ against adaptive adversaries for EXP3 with Kiefer-Wolfowitz exploration.

While the above algorithms achieve low regret, they can be computationally inefficient. A series of works have addressed this issue for continuous sets in $\mathbb{R}^d$. Abernethy et al. (2008) were the first to propose a computationally efficient algorithm that achieved a pseudo-regret of $\text{poly}(d) \cdot \sqrt{T}$. They also proposed a computationally efficient algorithm for the online shortest path problem on DAGs with the same pseudo-regret. The best known pseudo-regret for a computationally efficient algorithm is $\widetilde{O}(d\sqrt{T})$, attained by the algorithms in Hazan and Karnin (2016) and Ito et al. (2020). The efficient algorithm by Hazan and Karnin (2016) also matches the pseudo-regret of EXP2 with John's exploration for the online shortest path problem on DAGs.

For continuous sets in $\mathbb{R}^d$, Lee et al. (2020) proposed the first efficient algorithm achieving a *high-probability* regret of $\text{poly}(d) \cdot \sqrt{T}$ against an adaptive adversary. Later, Zimmert and Lattimore (2022) developed an efficient algorithm with a regret of $\widetilde{\mathcal{O}}(d^2\sqrt{T})$ with high probability against an adaptive adversary, which remains the best known result to date. For a more detailed discussion of all the related works, we refer the reader to Appendix A.

Reference	Regret	Efficient	Adaptive & high-prob.
Bubeck et al. (2012a)	$\sqrt{\lvert E\rvert T \log \lvert\mathcal{X}\rvert}$	✗	✗
Zimmert and Lattimore (2022)	$\sqrt{\lvert E\rvert T \log \lvert\mathcal{X}\rvert}$	✗	✓
Abernethy et al. (2008)	$\sqrt{\lvert E\rvert^3 T}$	✓	✗
Hazan and Karnin (2016)	$\sqrt{\lvert E\rvert T \log \lvert\mathcal{X}\rvert}$	✓	✗
Ito et al. (2020)	$\sqrt{\lvert E\rvert^2 T}$	✓	✗
Lee et al. (2020)	$\sqrt{\lvert E\rvert^7 T}$	✓	✓‡
Zimmert and Lattimore (2022)	$\sqrt{\lvert E\rvert^4 T}$	✓	✓‡
This paper (Theorem 7)	$\sqrt{\lvert E\rvert T \log \lvert\mathcal{X}\rvert}$	✓	✓

Table 1: Summary of regret guarantees for the online shortest path problem on a directed acyclic graph (DAG) $G = (V, E)$, with the set of paths $\mathcal{X} \subseteq \{0,1\}^E$ from source to sink, ignoring constants and logarithmic factors in $|E|$ and T. ‡The high-probability guarantee was formally proved only for continuous sets. However, we believe that their analysis extends to discrete decision sets, such as paths in a DAG, using the same techniques as Abernethy et al. (2008).

In this paper, we revisit the online shortest path problem in DAGs—the motivation behind much of the prior work—and pose the following question:

> *Can we design a computationally efficient algorithm for the online shortest path problem in a directed acyclic graph that, under bandit feedback, achieves a minimax-optimal regret bound with high probability against adaptive adversaries, up to logarithmic factors in the number of edges?*

1.1. Contributions and Techniques

In this paper, we answer the above question in the affirmative. For any directed acyclic graph (DAG) $G = (V, E)$ with a set of paths $\mathcal{X} \subseteq \{0,1\}^E$ from source to sink, we design the first computationally efficient algorithm to achieve a high-probability regret bound of $\widetilde{\mathcal{O}}(\sqrt{|E|T \log |\mathcal{X}|})$ against an adaptive adversary under bandit feedback, where $\widetilde{\mathcal{O}}(\cdot)$ hides logarithmic factors in $|E|$. We refer the reader to Table 1 for a comparison of our result with previous algorithms. Moreover, for the class of DAGs with at most d edges and at most N paths, we establish a minimax lower bound of $\Omega\left(\sqrt{dT \log(N)/\log(d)}\right)$. Hence, our algorithm is minimax-optimal upto logarithmic factors.

We further apply our efficient algorithm to combinatorial domains such as hypercubes, multi-task multi-armed bandits (MAB), extensive-form games, walks in directed graphs, the Colonel Blotto game, and m-sets, all of which can be represented as DAGs. This results in improved high-probability regret bounds in each setting compared to those in Zimmert and Lattimore (2022). For a detailed discussion of these improvements, we refer the reader to Section 4.

Our main technical contribution is a novel algorithmic approach for regret minimization on DAGs. Prior works relied on variants of exponential weights or FTRL, requiring mixing with a fixed distribution before selecting a path. Instead, we use a novel importance-sampling-inspired

loss estimator to enable implicit exploration and apply centroid-based decomposition to modify the input graph, achieving a nearly minimax optimal bound.

Our algorithm proceeds in two steps. The first step is to design an algorithm for graphs with $|V|$ vertices, $|E|$ edges, and a longest path length of K. While a path can be represented using $|E|$ bits (one per edge), we introduce an extended representation with additional $\mathcal{O}(|V| + K)$ bits and denote the corresponding set of paths as $\mathcal{X}^\dagger$. We then solve the FTRL optimization problem:

$$\widetilde{\mathbf{x}}_t \leftarrow \operatorname*{arg\,min}_{\mathbf{x}\in \mathrm{co}(\mathcal{X}^\dagger)} \left(\eta \sum_{s=1}^{t-1} \langle \mathbf{x}, \widehat{\mathbf{y}}_s \rangle + F(\mathbf{x}) \right),$$

where $F(\cdot)$ is a Legendre function, and in our work, we use the Tsallis-1/2 entropy. We then efficiently sample a path $\mathbf{x}_t$ such that its expectation is $\widetilde{\mathbf{x}}_t$. We then introduce a novel importance-sampling-inspired loss estimator $\widetilde{\mathbf{y}}_t$, ensuring that the difference $\langle \mathbf{x}_1, \widetilde{\mathbf{y}}_t \rangle - \langle \mathbf{x}_2, \widetilde{\mathbf{y}}_t \rangle$ remains an unbiased estimate of the loss difference between any two paths encoded as $\mathbf{x}_1, \mathbf{x}_2$ in the DAG, even though $\widetilde{\mathbf{y}}_t$ itself is not an unbiased estimator of the actual loss vector. Building on this, we perform implicit exploration, similar to standard multi-armed bandits (Neu, 2015), by introducing a bias in $\widetilde{\mathbf{y}}_t$ to construct our final estimator $\widehat{\mathbf{y}}_t$, thereby achieving a high-probability regret bound of $\widetilde{\mathcal{O}}(\sqrt{K|E|T})$ against any adaptive adversary.

The second step of our algorithmic approach considers the problem on an arbitrary DAG $G = (V, E)$ with the set of all paths from source to sink denoted by $\mathcal{X}$ and reduces it to a problem on a newly constructed DAG $G^\dagger = (V^\dagger, E^\dagger)$ that satisfies several key properties. The length of any path from source to sink in $G^\dagger$ is $\mathcal{O}(\log |\mathcal{X}|)$, while the number of vertices and edges satisfy $|V^\dagger| = \mathcal{O}(|V|)$ and $|E^\dagger| = \widetilde{\mathcal{O}}(|E|)$, respectively. Additionally, there exists a bijective mapping between the paths in G and $G^\dagger$, which can be efficiently computed. To achieve this reduction, we introduce a novel centroid–based decomposition approach. Applying our FTRL method to $G^\dagger$, we obtain a high-probability regret bound of $\widetilde{\mathcal{O}}(\sqrt{|E|T \log |\mathcal{X}|})$ against any adaptive adversary.

2. Preliminaries

Let $G = (V, E)$ be a Directed Acyclic Graph (DAG), where V is the set of vertices and $E \subseteq V \times V$ is the set of directed edges. A path $P = (v_0, e_1, v_1, \ldots, e_k, v_k)$ of length $k > 0$ is an interleaved sequence of vertices and edges satisfying $v_i \in V$ for $i \in \{0, 1, \ldots, k\}$ and $e_i = (v_{i-1}, v_i) \in E$ for $i \in \{1, \ldots, k\}$. Since G is acyclic, no path P can exist with $v_0 = v_k$.

For a vertex $v \in V$, the set of incoming edges is denoted by $\delta^-(v) := \{(u, v) \in E\}$, and the set of outgoing edges is denoted by $\delta^+(v) := \{(v, u) \in E\}$. Given a weight function $w : E \to \mathbb{R}$ that assigns a weight to each edge, the shortest path problem seeks to find a path P from a source vertex v_s to a sink vertex v_t that minimizes the total weight of the edges along the path, given by

$$w(P) := \sum_{i=1}^{k} w(e_i).$$

Without loss of generality, we assume that every vertex v is reachable from v_s and can reach v_t.

We consider the online shortest path problem with bandit feedback. In each round t, an agent selects a path P_t from v_s to v_t, while an adversary simultaneously selects a weight function $w_t(\cdot)$. The agent then observes *only* the loss, which is the path weight $\ell_t := w_t(P_t)$. The objective is to

minimize the cumulative regret against the optimal path:

$$\mathrm{Regret}(T) := \sum_{t=1}^{T} w_t(P_t) - \min_{P \in \mathcal{P}} \sum_{t=1}^{T} w_t(P),$$

where $\mathcal{P}$ is the set of all paths from $v_{\mathtt{s}}$ to $v_{\mathtt{t}}$.

Denote by $\mathcal{X} \subseteq \{0,1\}^{V \cup E}$ the set of all paths in the graph G from $v_{\mathtt{s}}$ to $v_{\mathtt{t}}$, indexed by the vertices in V and the edges in E. Each vector $\mathbf{x} \in \mathcal{X}$ encodes a path in the graph, where $\mathbf{x}[v] = 1$ indicates that $v \in V$ appears in the path, and $\mathbf{x}[e] = 1$ indicates that $e \in E$ appears in the path. The convex hull of $\mathcal{X}$ forms the flow polytope:

$$\mathrm{co}(\mathcal{X}) = \left\{ \mathbf{x} \in [0,1]^{V \cup E} : \ \mathbf{x}[v_{\mathtt{s}}] = \mathbf{x}[v_{\mathtt{t}}] = 1, \text{ and } \mathbf{x}[v] = \sum_{e \in \delta^-(v)} \mathbf{x}[e] = \sum_{e \in \delta^+(v)} \mathbf{x}[e], \ \forall v \in V \right\}.$$

Correspondingly, the weight function $w_t(\cdot)$ can be encoded as a vector $\mathbf{y}_t \in \mathbb{R}^{V \cup E}$, where $\mathbf{y}_t[e] = w_t(e)$ for all edges $e \in E$ and $\mathbf{y}_t[v] = 0$ for all vertices $v \in V$. In this formulation, the total path weight can be expressed as the inner product $w_t(P_t) = \langle \mathbf{x}_t, \mathbf{y}_t \rangle$, allowing the regret to be rewritten as:

$$\mathrm{Regret}(T) = \sum_{t=1}^{T} \langle \mathbf{x}_t, \mathbf{y}_t \rangle - \min_{\mathbf{x} \in \mathcal{X}} \sum_{t=1}^{T} \langle \mathbf{x}, \mathbf{y}_t \rangle.$$

Finally, we denote by $\mathcal{F}_t := \{\mathbf{x}_\tau, \mathbf{y}_\tau\}_{\tau=1}^{t}$ the filtration generated by the first t rounds. We further use $\mathbb{P}_t[\cdot] := \mathbb{P}[\cdot|\mathcal{F}_{t-1}]$ as the conditional probability and $\mathbb{E}_t[\cdot] := \mathbb{E}[\cdot|\mathcal{F}_{t-1}]$ as the conditional expectation. The adversary is allowed to choose loss vector $\mathbf{y}_t$ that adapts to the past filtration and the agent's algorithm.

Throughout this paper, we impose the following standard assumption.

Assumption 1 *The adversary can only choose weight function w such that the absolute weight of any path is at most 1. That is, it can only choose $\mathbf{y}$ satisfying $\langle \mathbf{x}, \mathbf{y} \rangle \in [-1, 1]$ for all $\mathbf{x} \in \mathcal{X}$.*

General notations. We define $[\![k]\!] := \{1, 2, \ldots, k\}$ and $[\![a, b]\!] := \{a, a+1, \ldots, b\}$. Denote by 2^C the power set of set C. Let $\varnothing$ denote the empty set. The logarithm of x to base 2 is denoted as $\log x$. For any pair of tuples $A = (a_1, \ldots, a_n)$ and $B = (b_1, \ldots, b_m)$, let $A \circ B$ denote the tuple $(a_1, \ldots, a_n, b_1, \ldots, b_m)$. Similarly, vectors $\mathbf{x}$ and $\mathbf{y}$, let $\mathbf{z} = \mathbf{x} \circ \mathbf{y}$ denote the vector obtained by concatenating $\mathbf{y}$ to the end of $\mathbf{x}$. For any edge e and path P, $e \in P$ indicates e is part of P.

3. Algorithm for Online Shortest Paths in DAGs

In this section, we present our algorithm for the online shortest path problem in directed acyclic graphs (DAGs). Our approach differs from the standard method of using exponential weights combined with a fixed distribution, such as Kiefer-Wolfowitz exploration. We start by outlining an efficient algorithm for the case where all paths have equal lengths in Section 3.1. In Section 3.2, we introduce a method to relax this assumption. Finally, in Section 3.3, we show how to achieve a regret bound of $\widetilde{\mathcal{O}}(\sqrt{|E|T \log |\mathcal{X}|})$ while maintaining computational efficiency.

3.1. The Case of Equal Path Lengths

We first present an algorithm for a DAG $G = (V, E)$, where *every* path from the source v_{s} to the sink v_{t} contains exactly K edges. In each round t, the algorithm chooses a strategy in the flow polytope $\mathrm{co}(\mathcal{X})$ by solving the following optimization problem:

$$\widetilde{\mathbf{x}}_t \leftarrow \operatorname*{arg\,min}_{\mathbf{x} \in \mathrm{co}(\mathcal{X})} \left(\eta \sum_{\tau=1}^{t-1} \langle \mathbf{x}, \widehat{\mathbf{y}}_\tau \rangle + F(\mathbf{x}) \right), \tag{1}$$

where $F(\mathbf{x})$ is some Legendre function, η is some learning rate and $\widehat{\mathbf{y}}_\tau$ is some loss estimator that we define later. The actual path $\mathbf{x}_t$ is then sampled as follows: Starting from the source v_{s}, we traverse to a node v and select an edge $e \in \delta^+(v)$ among the outgoing edges of v with probability proportional to $\widetilde{\mathbf{x}}_t[e]$, moving to the endpoint of edge e. This process repeats until we reach the sink v_{t}. We denote the path traversed as P_t and choose the corresponding vector in $\mathcal{X}$ as $\mathbf{x}_t$. It can be easily verified that $\mathbb{E}_t[\mathbf{x}_t] = \widetilde{\mathbf{x}}_t$. We then observe the loss $\ell_t := \langle \mathbf{x}_t, \mathbf{y}_t \rangle$, construct our loss estimator $\widehat{\mathbf{y}}_t$ as shown below, and proceed to the next round.

Recall that $\mathbb{E}_t[\cdot] := \mathbb{E}[\cdot|\mathcal{F}_{t-1}]$ and $\mathbb{P}_t[\cdot] := \mathbb{P}[\cdot|\mathcal{F}_{t-1}]$, where $\mathcal{F}_{t-1}$ is the past filtration. Let $\boldsymbol{\gamma} \in \mathbb{R}_{>0}^{V \cup E}$ be a positive-valued vector indexed by the elements of $V \cup E$. We start with defining our estimator $\widehat{\mathbf{y}}_t$ for the loss vector $\mathbf{y}_t$ upon receiving the loss $\ell_t := \langle \mathbf{x}_t, \mathbf{y}_t \rangle$:

$$\widehat{\mathbf{y}}_t[e] := \frac{(1+\ell_t)\,\mathbb{1}[\mathbf{x}_t[e]=1]}{\mathbb{P}_t[\mathbf{x}_t[e]=1] + \boldsymbol{\gamma}[e]}, \quad \forall e \in E,$$
$$\widehat{\mathbf{y}}_t[v] := \frac{(1-\ell_t)\,\mathbb{1}[\mathbf{x}_t[v]=1]}{\mathbb{P}_t[\mathbf{x}_t[v]=1] + \boldsymbol{\gamma}[v]}, \quad \forall v \in V \setminus \{v_{\mathsf{s}}, v_{\mathsf{t}}\}, \quad \widehat{\mathbf{y}}_t[v_{\mathsf{s}}] = \widehat{\mathbf{y}}_t[v_{\mathsf{t}}] := 0.$$

Note that even though the loss vector satisfies $\mathbf{y}_t[v] = 0$ for any vertex $v \in V$, the estimator is still designed to assign weights to it. Next, let us define another estimator $\widetilde{\mathbf{y}}_t$ as:

$$\widetilde{\mathbf{y}}_t[e] := \frac{(1+\ell_t)\,\mathbb{1}[\mathbf{x}_t[e]=1]}{\mathbb{P}_t[\mathbf{x}_t[e]=1]}, \quad \forall e \in E,$$
$$\widetilde{\mathbf{y}}_t[v] := \frac{(1-\ell_t)\,\mathbb{1}[\mathbf{x}_t[v]=1]}{\mathbb{P}_t[\mathbf{x}_t[v]=1]}, \quad \forall v \in V \setminus \{v_{\mathsf{s}}, v_{\mathsf{t}}\}, \quad \widetilde{\mathbf{y}}_t[v_{\mathsf{s}}] = \widetilde{\mathbf{y}}_t[v_{\mathsf{t}}] := 0.$$

Observe that $\widehat{\mathbf{y}}_t$ is the implicitly biased version of $\widetilde{\mathbf{y}}_t$. Although $\widetilde{\mathbf{y}}_t$ appears to be a biased estimator of $\mathbf{y}_t$, the next lemma shows that it can effectively compare the losses between different paths.

Lemma 1 *For any path with representation $\mathbf{x} \in \mathcal{X}$, it holds that $\mathbb{E}_t[\langle \mathbf{x}, \widetilde{\mathbf{y}}_t \rangle] = \langle \mathbf{x}, \mathbf{y}_t \rangle + \|\mathbf{x}\|_1 - 2$.*

Proof. For some vertex $v \in V$, we denote by $\mathcal{E}_{t,v}$ the event that vertex v is chosen in the path in round t, i,e, $\mathbb{1}[\mathbf{x}_t[v] = 1]$. Under event $\mathcal{E}_{t,v}$, chosen path $\mathbf{x}_t$ can be divided into two subpath: one from v_{s} to v, and other from v to v_{t}. Let $\ell_{t,v}^-$ and $\ell_{t,v}^+$ be the total weight of the path from v_{s} to v and the path from v to v_{t}, respectively. According to the linearity of expectation, it satisfies that

$$\mathbb{E}_t[\ell_t \mid \mathcal{E}_{t,v}] = \mathbb{E}_t[\ell_{t,v}^- \mid \mathcal{E}_{t,v}] + \mathbb{E}_t[\ell_{t,v}^+ \mid \mathcal{E}_{t,v}]. \tag{2}$$

Note $\ell_{t,v_{\mathsf{s}}}^- = \ell_{t,v_{\mathsf{t}}}^+ = 0$. For some edge $e = (v_-, v_+) \in E$, we similarly define by $\mathcal{E}_{t,e}$ the event that $\mathbf{x}_t[e] = 1$. The total weight of the path can also be decomposed into

$$\mathbb{E}_t[\ell_t \mid \mathcal{E}_{t,e}] = \mathbb{E}_t[\ell_{t,v_-}^- \mid \mathcal{E}_{t,e}] + \mathbf{y}_t[e] + \mathbb{E}_t[\ell_{t,v_+}^+ \mid \mathcal{E}_{t,e}]. \tag{3}$$

Observe that our edge sampling procedure is Markovian. That is, under the event $\mathcal{E}_{t,u}$, the probability of choosing an outgoing edge from node u does not depend on the path from v_{s} to u. This implies that:

$$\mathbb{E}_t[\ell^-_{t,v_-} \mid \mathcal{E}_{t,v_-}] = \mathbb{E}_t[\ell^-_{t,v_-} \mid \mathcal{E}_{t,v_-} \cap \mathcal{E}_{t,e}] = \mathbb{E}_t[\ell^-_{t,v_-} \mid \mathcal{E}_{t,e}] \tag{4}$$

where the last inequality is given by $\mathcal{E}_{t,e} \subseteq \mathcal{E}_{t,v_-}$. Similarly, we have that

$$\mathbb{E}_t[\ell^+_{t,v_+} \mid \mathcal{E}_{t,v_+}] = \mathbb{E}_t[\ell^+_{t,v_+} \mid \mathcal{E}_{t,e}] \tag{5}$$

Let $P = (v_0, e_1, v_1, \ldots, e_k, v_k)$ be the path that corresponds to the vector $\mathbf{x} \in \mathcal{X}$, where $v_0 = v_{\mathsf{s}}$ and $v_k = v_{\mathsf{t}}$. The expectation of the inner product $\langle \mathbf{x}, \widetilde{\mathbf{y}}_t \rangle$ can be computed as follows:

$$\begin{aligned}
\mathbb{E}_t[\langle \mathbf{x}, \widetilde{\mathbf{y}}_t \rangle] &= \sum_{i=0}^{k} \mathbb{E}_t\left[\widetilde{\mathbf{y}}_t[v_i]\right] + \sum_{i=1}^{k} \mathbb{E}_t\left[\widetilde{\mathbf{y}}_t[e_i]\right] \\
&= \sum_{i=1}^{k-1} \mathbb{E}_t\left[\frac{(1-\ell_t)\,\mathbb{1}[\mathbf{x}_t[v_i]=1]}{\mathbb{P}_t[\mathbf{x}_t[v_i]=1]}\right] + \sum_{i=1}^{k} \mathbb{E}_t\left[\frac{(1+\ell_t)\,\mathbb{1}[\mathbf{x}_t[e_i]=1]}{\mathbb{P}_t[\mathbf{x}_t[e_i]=1]}\right] \\
&= \sum_{i=1}^{k-1} \mathbb{E}_t[1-\ell_t \mid \mathcal{E}_{t,v_i}] + \sum_{i=1}^{k} \mathbb{E}_t[1+\ell_t \mid \mathcal{E}_{t,e_i}] \\
&= 2k-1 - \sum_{i=1}^{k-1} \Big(\mathbb{E}_t[\ell^-_{t,v_i} \mid \mathcal{E}_{t,v_i}] + \mathbb{E}_t[\ell^+_{t,v_i} \mid \mathcal{E}_{t,v_i}]\Big) \\
&\qquad + \sum_{i=1}^{k} \Big(\mathbb{E}_t[\ell^-_{t,v_{i-1}} \mid \mathcal{E}_{t,e_i}] + \mathbf{y}_t[e_i] + \mathbb{E}_t[\ell^+_{t,v_i} \mid \mathcal{E}_{t,e_i}]\Big) \\
&= 2k-1 - \sum_{i=1}^{k-1} \mathbb{E}_t[\ell^-_{t,v_i} \mid \mathcal{E}_{t,v_i}] - \sum_{i=1}^{k-1} \mathbb{E}_t[\ell^+_{t,v_i} \mid \mathcal{E}_{t,v_i}] \\
&\qquad + \sum_{i=1}^{k-1} \mathbb{E}_t[\ell^-_{t,v_i} \mid \mathcal{E}_{t,v_i}] + \sum_{i=1}^{k} \mathbf{y}_t[e_i] + \sum_{i=1}^{k-1} \mathbb{E}_t[\ell^+_{t,v_i} \mid \mathcal{E}_{t,v_i}] \\
&= \langle \mathbf{x}, \mathbf{y}_t \rangle + \|\mathbf{x}\|_1 - 2.
\end{aligned}$$

where the second equality follows from the definition of $\widehat{\mathbf{y}}_t$, the third equality follows from the definition of the events $\mathcal{E}_{t,v}$ and $\mathcal{E}_{t,e}$, the fourth equality follows from equations (2) and (3), and the fifth equality follows from equations (4), (5), and $\ell^-_{t,v_{\mathsf{s}}} = \ell^+_{t,v_{\mathsf{t}}} = 0$.

■

If all paths have the same length K, then $\mathbb{E}_t[\langle \mathbf{x} - \mathbf{x}', \widetilde{\mathbf{y}}_t \rangle] = \langle \mathbf{x} - \mathbf{x}', \mathbf{y}_t \rangle$ for all $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$. This equality is crucial for developing a framework in Appendix C, which enables implicit exploration—similar to the framework for standard multi-armed bandits by Neu (2015)—within an FTRL problem such as the one formulated in this section. The equality ensures the framework's correct application. Using $F(\mathbf{x}) = -\sum_{v \in V} \sqrt{\mathbf{x}[v]} - \sum_{e \in E} \sqrt{\mathbf{x}[e]}$ as our regularizer, we can apply this framework to achieve a regret bound of at most $\mathcal{O}(\sqrt{K|E|T\log(|E|/\delta)})$ with probability at least $1-\delta$ against any adaptive adversary. We refer the reader to Appendix D.1 for the omitted details.

3.2. Relaxing the Equal Path Length Assumption

In this section, we relax the assumption that all paths have the same length. Denote $K(v)$ as the length of the longest path from $v_{\mathtt{s}}$ to vertex v. Let $K = K(v_{\mathtt{t}})$ denote the length of the longest path within the DAG. Note that $K(v_{\mathtt{s}}) = 0$.

Construct the augmented vector as $\mathbf{x}^\dagger := \mathbf{x} \circ b(\mathbf{x})$, where $b(\mathbf{x}) \in \{0,1\}^{K-1}$ is given by

$$b(\mathbf{x})[i] := \mathbb{1}\left[\exists (u,v) \in E, \mathbf{x}[(u,v)] = 1, K(u) < i < K(v)\right].$$

We denote $\mathcal{X}^\dagger := \{\mathbf{x}^\dagger \mid \mathbf{x} \in \mathcal{X}\}$ as the augmented decision space. Let $\widehat{\boldsymbol{\gamma}} \in \mathbb{R}_{>0}^{K-1}$ be a positive-valued vector. Correspondingly, we construct the augmented loss estimator $\widehat{\mathbf{y}}_t^\dagger := \widehat{\mathbf{y}}_t \circ \widehat{\mathbf{c}}_t$, where $\widehat{\mathbf{c}}_t \in \mathbb{R}_{\geq 0}^{K-1}$ is defined as:

$$\widehat{\mathbf{c}}_t[i] := \frac{2 \cdot \mathbb{1}[b(\mathbf{x}_t)[i] = 1]}{\mathbb{P}_t[b(\mathbf{x}_t)[i] = 1] + \widehat{\boldsymbol{\gamma}}[i]},$$

and $\mathbf{x}_t$ is the path chosen according to the selection procedure from the previous section. We also define $\widetilde{\mathbf{c}}_t \in \mathbb{R}_{\geq 0}^{K-1}$ as:

$$\widetilde{\mathbf{c}}_t[i] := \frac{2 \cdot \mathbb{1}[b(\mathbf{x}_t)[i] = 1]}{\mathbb{P}_t[b(\mathbf{x}_t)[i] = 1]},$$

Observe that $\widehat{\mathbf{c}}_t[i]$ is the implicitly biased version of $\widetilde{\mathbf{c}}_t[i]$. The next lemma establishes a key property of the loss estimator $\widetilde{\mathbf{y}}_t^\dagger := \widetilde{\mathbf{y}}_t \circ \widetilde{\mathbf{c}}_t$, in which the implicit biasing is absent.

Lemma 2 *For any path with representation $\mathbf{x} \in \mathcal{X}$, it holds that $\mathbb{E}_t[\langle \mathbf{x}^\dagger, \widetilde{\mathbf{y}}_t^\dagger \rangle] = \langle \mathbf{x}, \mathbf{y}_t \rangle + 2K - 1$.*

Proof. Consider any $\mathbf{x} \in \mathcal{X}$, and its corresponding path $P = (v_0, e_1, v_1, \ldots, e_k, v_k)$, where $v_0 = v_{\mathtt{s}}$ and $v_k = v_{\mathtt{t}}$. The expectation of the inner product of the auxiliary bits, $\langle b(\mathbf{x}), \widetilde{\mathbf{c}}_t \rangle$, satisfies

$$\mathbb{E}_t[\langle b(\mathbf{x}), \widetilde{\mathbf{c}}_t \rangle] = \sum_{i=1}^{K-1} b(\mathbf{x})[i] \cdot \mathbb{E}_t[\widetilde{\mathbf{c}}_t[i]].$$

By construction, $\mathbb{E}_t[\widetilde{\mathbf{c}}_t[i]] = 2$. Since $K(v_{j-1}) < K(v_j)$ for all $j \in [\![k]\!]$, for any given index $i \in [\![K-1]\!]$, there is at most one index $j_i \in [\![k]\!]$ such that $K(v_{j_i - 1}) < i < K(v_{j_i})$. Consequently,

$$\begin{aligned}
\sum_{i=1}^{K-1} b(\mathbf{x})[i] &= \sum_{i=1}^{K-1} \sum_{j=1}^{k} \mathbb{1}[K(v_{j-1}) < i < K(v_j)] \\
&= \sum_{j=1}^{k} \sum_{i=1}^{K-1} \mathbb{1}[K(v_{j-1}) < i < K(v_j)] \\
&= \sum_{j=1}^{k} \big(K(v_j) - K(v_{j-1}) - 1\big) = K(v_{\mathtt{t}}) - K(v_{\mathtt{s}}) - k.
\end{aligned}$$

Hence, using the fact that $\|\mathbf{x}\|_1 = 2k + 1$, which follows from the mapping between $\mathbf{x}$ and P,

$$\mathbb{E}_t[\langle b(\mathbf{x}), \widetilde{\mathbf{c}}_t \rangle] = 2(K(v_{\mathtt{t}}) - K(v_{\mathtt{s}}) - k) = 2K - \|\mathbf{x}\|_1 + 1. \tag{6}$$

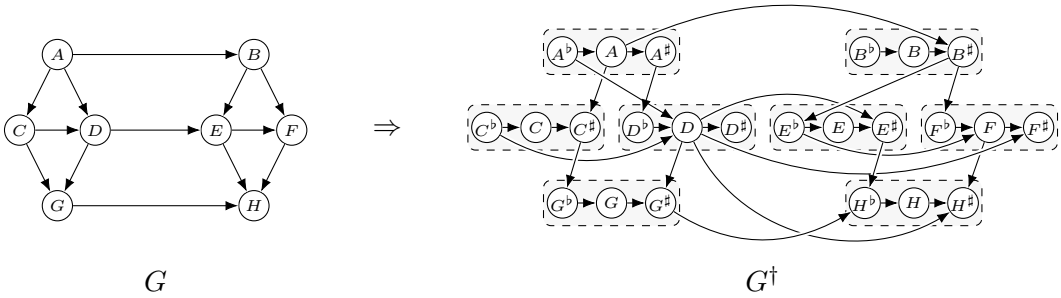

Figure 1: Example G and $G^\dagger$ according to conversion in Section 3.3. The longest path from source to sink in $G^\dagger$ is upper bounded by $\mathcal{O}(\log|\mathcal{X}|)$. See Figure 2 in Appendix for more details.

As a result,

$$\begin{aligned}\mathbb{E}_t[\langle \mathbf{x}^\dagger, \widetilde{\mathbf{y}}_t^\dagger\rangle] = \mathbb{E}_t[\langle \mathbf{x}, \widetilde{\mathbf{y}}_t\rangle + \langle b(\mathbf{x}), \widetilde{\mathbf{c}}_t\rangle] &= \langle \mathbf{x}, \mathbf{y}_t\rangle + \|\mathbf{x}\|_1 - 2 + 2K - \|\mathbf{x}\|_1 + 1 \\ &= \langle \mathbf{x}, \mathbf{y}_t\rangle + 2K - 1,\end{aligned}$$

where the second equality follows from Equation (6) and Lemma 2. ■

We can appropriately modify our FTRL algorithm from the previous section to work with the augmented decision space $\mathcal{X}^\dagger := \{\mathbf{x}^\dagger \mid \mathbf{x} \in \mathcal{X}\}$, augmented loss estimator $\widehat{\mathbf{y}}_t^\dagger$ and augmented regularizer $F(\mathbf{x}) = -\sum_{v\in V}\sqrt{\mathbf{x}[v]} - \sum_{e\in E}\sqrt{\mathbf{x}[e]} - \sum_{i\in[\![K-1]\!]}\sqrt{\mathbf{x}[i]}$ for any $\mathbf{x} \in [0,1]^{V\cup E\cup[\![K-1]\!]}$. Thus, we can apply our FTRL framework for implicitly biased estimators from Appendix C to obtain a regret bound of at most $\mathcal{O}(\sqrt{K|E|T\log(|E|/\delta)})$ with probability at least $1-\delta$ against any adaptive adversary. Furthermore, we assert that our FTRL approach can be implemented efficiently, as it can be easily shown that the set $\mathrm{co}(\mathcal{X}^\dagger)$ can be represented using a polynomial number of linear constraints. We refer the reader to Appendix D.2 for the omitted details of this section.

3.3. Achieving a Regret Upper Bound of $\widetilde{\mathcal{O}}(\sqrt{|E|T\log|\mathcal{X}|})$

In this section, we transform the input DAG G into a new DAG $G^\dagger$ with an equivalent decision space but reduced complexity. The core idea is to introduce "express" edges that compress long paths in G, ensuring the longest path in $G^\dagger$ is bounded by $\mathcal{O}(\log|\mathcal{X}|)$ while minimally increasing the number of edges and vertices. Consider a long path $P = (v_0, e_1, \ldots, e_k, v_k)$ in G. We concisely represent all subpaths of P with the help of the middle vertex $v_{\lfloor k/2\rfloor}$. For each $i < \lfloor k/2\rfloor$, we add an edge $(v_i, v_{\lfloor k/2\rfloor})$ to represent the subpath from v_i to $v_{\lfloor k/2\rfloor}$. Similarly, for each $j > \lfloor k/2\rfloor$, we add an edge $(v_{\lfloor k/2\rfloor}, v_j)$. Thus, any subpath from v_i to v_j (where $i < \lfloor k/2\rfloor < j$) can be represented with just two edges, $(v_i, v_{\lfloor k/2\rfloor})$ and $(v_{\lfloor k/2\rfloor}, v_j)$. Recursively applying this method creates a hierarchical structure where every subpath of P requires only $\mathcal{O}(1)$ edges. Extending this concept using centroid-based decomposition for the spanning tree ensures that the longest path in $G^\dagger$ remains bounded by $\mathcal{O}(\log|\mathcal{X}|)$, with only a logarithmic increase in edges and vertices. Consequently, the online shortest path problem in G reduces to $G^\dagger$, allowing our algorithm from Section 3.2 to achieve a high-probability regret bound of $\widetilde{\mathcal{O}}(\sqrt{|E|T\log|\mathcal{X}|})$ against any adaptive adversary. Further details, including omitted proofs, are provided in Appendix D.3.

We formally begin our transformation. Let $C : V \to \mathbb{N}$ denote the number of distinct paths from the source v_{s} to any vertex v. It holds that $C(v_{\mathsf{s}}) = 1$ and $C(v) := \sum_{(u,v)\in\delta^-(v)} C(u)$ for any $v \neq v_{\mathsf{s}}$. According to the definition, it satisfies that $C(v_{\mathsf{t}}) = |\mathcal{X}|$. Let $h(v) := \arg\max_{(u,v)\in\delta^-(v)} C(u)$ be the incoming edge that brings the maximum number of paths to vertex v, with ties broken arbitrarily. Let $E^{\clubsuit} := \{h(v) \mid v \in V \setminus \{v_{\mathsf{s}}\}\}$ be the set of all such edges. The underlying subgraph $S := (V, E^{\clubsuit})$ forms a directed spanning tree of G. It can be easily shown that the number of non-tree edges (edges not in $E^{\clubsuit}$) on any path from v_{s} to v_{t} in G is at most $\log |\mathcal{X}|$.

We now introduce the *centroid-based decomposition*: Given a directed tree $S = (V, E^{\clubsuit})$, we identify a vertex $c \in V$ such that the connected components $\widehat{S}_1, \ldots, \widehat{S}_k$ resulting from its removal satisfy $|\widehat{V}_i| \leq |V|/2$ for all $i \in [\![k]\!]$, where $\widehat{V}_i$ is the set of vertices in the subtree $\widehat{S}_i$. Such a vertex c, known as the *centroid*, always exists in any tree (Jordan, 1869; Della Giustina et al., 2019). We associate the centroid c with the tree S by defining $S_c := S$. The above procedure is then applied recursively to each component $\widehat{S}_i$ for $i \in [\![k]\!]$. If a component reduces to a single vertex c, we designate c as its centroid and terminate the recursion.

Since the sets $\widehat{V}_i$ resulting from the removal of c form a partition of $V \setminus \{c\}$, each vertex $v \in V$ will eventually be assigned as the centroid of some subtree $S_v = (V_v, E_v^{\clubsuit})$. Consequently, this procedure generates a collection of subtrees $\mathcal{T} := \{S_v : v \in V\}$, where each vertex v is uniquely associated with a subtree of S in which it serves as the centroid. Furthermore, we define $\mathcal{T}(S_v) := \{S_w : w \in V_v\}$ as the centroid-based decomposition of the subtree S_v.

We now state the construction for a new graph $G^\dagger = (V^\dagger, E^\dagger)$ using $\mathcal{T}$ as follows:

1. Initialize $V^\dagger \leftarrow \varnothing$ and $E^\dagger \leftarrow \varnothing$.
2. For each vertex $c \in V$:
   (a) $V^\dagger \leftarrow V^\dagger \cup \{c^\flat, c, c^\sharp\}$.
   (b) For each vertex $v \in V_c$:
      i. If there is a directed path from v to c in S_c, or if v is c, update $E^\dagger \leftarrow E^\dagger \cup \{(v^\flat, c)\}$.
      ii. If there is a directed path from c to v in S_c, or if v is c, update $E^\dagger \leftarrow E^\dagger \cup \{(c, v^\sharp)\}$.
3. For each non-tree edge $(u, v) \in E \setminus E^{\clubsuit}$, update $E^\dagger \leftarrow E^\dagger \cup \{(u^\sharp, v^\flat)\}$.

We refer to Figure 1 for one example of our conversion. It is easy to verify that the graph $G^\dagger$ is a Directed Acyclic Graph with source node $v_{\mathsf{s}}^\flat$ and sink node $v_{\mathsf{t}}^\sharp$. We now demonstrate that the converted graph $G^\dagger$ is essentially equivalent to G. We define a mapping $\sigma : E^\dagger \to 2^E$ as follows:

- For $e^\dagger = (v^\flat, c)$, $\sigma(e^\dagger)$ consists of all edges on the unique path from v to c in the tree S.
- For $e^\dagger = (c, v^\sharp)$, $\sigma(e^\dagger)$ consists of all edges on the unique path from c to v in the tree S.
- For $e^\dagger = (u^\sharp, v^\flat)$, $\sigma(e^\dagger) = \{(u, v)\} \subseteq E \setminus E^{\clubsuit}$ contains the corresponding edge.

The above mapping assigns each edge $e^\dagger = (u^\dagger, v^\dagger) \in E^\dagger$ a path from u to v (which may be empty), as specified by $\sigma(e^\dagger)$, where $w^\dagger \in \{w^\flat, w, w^\sharp\}$ for $w \in \{u, v\}$. Denote by $\mathcal{P}^\dagger$ the set of paths from $v_{\mathsf{s}}^\flat$ to $v_{\mathsf{t}}^\sharp$ in $G^\dagger$. The following lemma establishes an important property of $\sigma(e^\dagger)$.

Lemma 3 *For any path $P^\dagger \in \mathcal{P}^\dagger$, $\sigma(e_1^\dagger) \cap \sigma(e_2^\dagger) = \varnothing$ for any distinct edges $e_1^\dagger, e_2^\dagger \in P^\dagger$.*

The next lemma establishes that this mapping defines a bijection between the paths from v_{s} to v_{t} in G and the paths $P^\dagger$ from $v_{\mathsf{s}}^\flat$ to $v_{\mathsf{t}}^\sharp$ in $G^\dagger$. We slightly abuse notation for σ.

Lemma 4 *There exists an efficiently computable bijection $\sigma : \mathcal{P}^\dagger \to \mathcal{P}$ such that an edge $e \in E$ belongs to $\sigma(P^\dagger)$ if and only if there exists an edge $e^\dagger \in P^\dagger$ with $e \in \sigma(e^\dagger)$.*

Let $w : E \to \mathbb{R}$ be a weight function in the graph $G = (V, E)$. Define $w^\dagger : E^\dagger \to \mathbb{R}$ as the weight function for the converted graph $G^\dagger = (V^\dagger, E^\dagger)$:

$$w^\dagger(e^\dagger) := \mathbb{1}[|\sigma(e^\dagger)| \geq 1] \cdot \sum_{e \in \sigma(e^\dagger)} w(e). \tag{7}$$

Using this mapping, we can convert a decision problem on G to a decision problem on $G^\dagger$ as follows.

Lemma 5 *The online shortest path problem on $G = (V, E)$ can be efficiently reduced to the online shortest path problem on $G^\dagger = (V^\dagger, E^\dagger)$.*

Proof. First, we can efficiently construct the DAG $G^\dagger$ using the DAG G. Next, given any weight function w_t encoded as $\mathbf{y}_t \in \mathbb{R}^{V \cup E}$ in the graph G, we can convert it into a weight function $\mathbf{y}_t^\dagger \in \mathbb{R}^{V^\dagger \cup E^\dagger}$ corresponding to $w_t^\dagger$ for $G^\dagger$ according to (7). For any chosen path $P_t^\dagger$ in $G^\dagger$ (encoded as $\mathbf{x}_t^\dagger$) , we can efficiently choose $\mathbf{x}_t \in \mathcal{X}$ corresponding to the path $\sigma(P^\dagger)$ in G following the bijective mapping g in Lemma 4. Due to Lemma 3 and Lemma 4, we have:

$$\langle \mathbf{x}^\dagger, \mathbf{y}_t^\dagger \rangle = \sum_{e^\dagger \in P^\dagger} \mathbb{1}[|\sigma(e^\dagger)| \geq 1] \cdot \sum_{e \in \sigma(e^\dagger)} w(e) = \sum_{e \in P} w(e) = \langle \mathbf{x}, \mathbf{y}_t \rangle.$$

The second equality follows from the fact that the set of edges in $\sigma(P^\dagger)$ is given by $\bigcup_{e^\dagger \in P^\dagger} \sigma(e^\dagger)$, and that $\sigma(e_1^\dagger) \cap \sigma(e_2^\dagger) = \varnothing$ for any distinct edges $e_1^\dagger, e_2^\dagger \in P^\dagger$. Hence, we can efficiently reduce the online shortest path problem on G to the online shortest path problem on $G^\dagger$. ■

Finally, the graph $G^\dagger$ satisfies the required size constraints, as stated in the following lemma.

Lemma 6 *The graph $G^\dagger = (V^\dagger, E^\dagger)$ contains $|V^\dagger| \leq \mathcal{O}(|V|)$ vertices and $|E^\dagger| \leq \mathcal{O}(|V| \log |V| + |E|)$ edges. Moreover, The number of edges on the longest path from $v_{\mathsf{s}}^\flat$ to $v_{\mathsf{t}}^\sharp$ is upper bounded by $\mathcal{O}(\log |\mathcal{X}|)$.*

By combining Lemmas 5 and 6, and applying our FTRL algorithm from the previous section on the DAG $G^\dagger$, we establish the main theorem:

Theorem 7 *There exists an computationally efficient algorithm that incurs a regret bound of at most $\widetilde{\mathcal{O}}(\sqrt{|E|T \log(|\mathcal{X}|/\delta)})$ with probability at least $1 - \delta$ against any adaptive adversary, where $\widetilde{\mathcal{O}}(\cdot)$ only hides logarithmic factors in $|E|$.*

We refer the reader to Appendix D.3 for the omitted details of this section. Moreover, our algorithm is nearly minimax-optimal as for the class of DAGs with at most d edges and at most N paths, we establish a minimax lower bound of $\Omega(\sqrt{dT \log(N)/\log(d)})$ in Appendix F.4.

Remark 8 *The key steps of our algorithm are computationally efficient, as outlined below:*

1. *We transform the DAG to reduce the maximum path length using a centroid-based decomposition, which is computable in polynomial time. The bijection between paths in the original and transformed DAGs can also be computed efficiently, as shown in Lemma 4.*
2. *We augment the path vector with additional bits, which are computed based on the longest path from the source to each node—a task that can be performed in polynomial time.*
3. *In each round, we perform an FTRL update with a strongly convex regularizer over the convex polytope $\mathrm{co}(\mathcal{X}^\dagger)$, which is defined by a polynomial number of linear constraints and supports efficient optimization using standard optimization methods.*

Combinatorial set	**Best known regret**[‡]	**Our improved regret**
Hypercube	$d^2\sqrt{T}$	$d\sqrt{T}$
Multi-task MAB	$(\sum_{i=1}^m d_i)^2\sqrt{T}$	$\sum_{i=1}^m \sqrt{d_i T}$
m-sets	$d^2\sqrt{T}$	$\sqrt{md^2T}$
Shortest walk	$\lvert E\rvert^2\sqrt{T}$	$\sqrt{K^2\lvert E\rvert T}$
Extensive-form games	$\lvert\mathcal{Z}\rvert^2\sqrt{T}$	$\sqrt{\lvert\mathcal{Z}\rvert T\log(N)}$
Colonel Blotto game	$K^2N^2\sqrt{T}$	$\sqrt{K^3NT}$

Table 2: Summary of high-probability regret guarantees for efficient algorithms across various combinatorial sets, ignoring constants and logarithmic factors. [‡]The best-known high-probability regret guarantee for efficient algorithms was formally proven only for continuous sets by Zimmert and Lattimore (2022). However, we believe their analysis extends to discrete decision sets, such as the combinatorial sets considered, using the same techniques as Abernethy et al. (2008).

4. Applications

While it might not be apparent at first glance, learning in several structured domains $\mathcal{X} \subseteq \{0,1\}^d$ can be efficiently reduced to online shortest paths in suitably-defined DAGs.[1] These include at least the following examples.

- *Hypercube:* $\mathcal{X} := \{0,1\}^d$.
- *Multi-task MAB:* $\mathcal{X} := \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_m$, where $\mathcal{X}_i = \{e_1, \ldots, e_{d_i}\}$ is a set of unit vectors.
- $\mathbf{m}$*-sets:* $\mathcal{X} := \{\mathbf{x} \in \{0,1\}^d : \|\mathbf{x}\|_1 = m\}$.
- *Shortest walk in directed graph.* We consider the online shortest walk problem in a directed graph $G = (V, E)$, where walks can have a length of at most $K \leq |E|$.
- *Extensive-form games.* The game consists of decision nodes $\mathcal{X}$, observation nodes $\mathcal{Y}$, and terminal nodes $\mathcal{Z}$. We choose one of the $N \leq 2^{|\mathcal{Z}|}$ possible strategies at the decision nodes and aim to minimize the total loss incurred.
- *Colonel Blotto games.* In this game, the goal is to assign N soldiers across K battlefields while minimizing the total loss incurred.

We refer the reader to Appendix E for a detailed discussion of each setting and its DAG reduction. The important point is that in light of the connection to DAGs, our method applies directly to the above settings as well. We summarize the results we obtain for these settings in Table 2, comparing the regret guarantees enjoyed by our method compared to the prior known high-probability regret guarantees achieved by efficient algorithms. We remark that our high-probability regret bound matches that of EXP3 with Kiefer-Wolfowitz exploration for the Hypercube and Extensive-form games. For Multi-task MAB, our high-probability regret bound significantly improves upon that of EXP3 with Kiefer-Wolfowitz exploration, and we also establish a matching lower bound, up to logarithmic factors. More details on previous approaches and implementation details of our methods in these settings are available in Appendix E.

1. To our knowledge, we are the first to point out this fact in the case of m-sets and, more importantly, extensive-form games, for which the reduction is not immediate.

5. Conclusion and Future Work

In this paper, we studied the online shortest path problem on DAGs. We designed the first computationally efficient algorithm to achieve a high-probability nearly minimax-optimal regret bound of $\widetilde{\mathcal{O}}(\sqrt{|E|T\log|\mathcal{X}|})$ against any adaptive adversary, where $\widetilde{\mathcal{O}}(\cdot)$ hides logarithmic factors in $|E|$. Beyond shortest paths, our algorithm can be applied to various combinatorial sets in $\{0,1\}^d$, and we provided improved high-probability regret bounds for them.

Our work raises several interesting open questions in combinatorial bandits. First, can our approach be further generalized to achieve high-probability regret bounds for any combinatorial set in $\{0,1\}^d$? Second, is there an efficient algorithm that achieves a high-probability minimax-optimal regret bound of $\mathcal{O}(\sqrt{dT\log|\mathcal{X}|})$ for any combinatorial set $\mathcal{X} \subseteq \{0,1\}^d$? Finally, given a fixed combinatorial set $\mathcal{X} \subseteq \{0,1\}^d$, what are the tight upper and lower bounds on regret relative to $\mathcal{X}$?

Acknowledgments

The authors are grateful to Haipeng Luo for helpful discussion regarding the prior work (Lee et al., 2020).

This work was supported in part by NSF TRIPODS CCF Award #2023166, a Northrop Grumman University Research Award, ONR YIP award # N00014-20-1-2571, NSF award #1844729, and NSF award # CCF-2443068.

References

Jacob D Abernethy, Elad Hazan, and Alexander Rakhlin. Competing in the dark: An efficient algorithm for bandit linear optimization. In *COLT*, pages 263–274. Citeseer, 2008.

Jean-Yves Audibert and Sébastien Bubeck. Regret bounds and minimax policies under partial monitoring. *The Journal of Machine Learning Research*, 11:2785–2836, 2010.

Jean-Yves Audibert, Sébastien Bubeck, and Gábor Lugosi. Regret in online combinatorial optimization. *Mathematics of Operations Research*, 39(1):31–45, 2014.

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

Baruch Awerbuch and Robert D Kleinberg. Adaptive routing with end-to-end feedback: Distributed learning and geometric approaches. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 45–53, 2004.

Peter Bartlett, Varsha Dani, Thomas Hayes, Sham Kakade, Alexander Rakhlin, and Ambuj Tewari. High-probability regret bounds for bandit online linear optimization. In *Proceedings of the 21st Annual Conference on Learning Theory-COLT 2008*, pages 335–342. Omnipress, 2008.

Soheil Behnezhad, Sina Dehghani, Mahsa Derakhshan, Mohammedtaghi Hajiaghayi, and Saeed Seddighin. Fast and simple solutions of blotto games. *Operations Research*, 71(2):506–516, 2023.

Sébastien Bubeck and Aleksandrs Slivkins. The best of both worlds: Stochastic and adversarial bandits. In *Conference on Learning Theory*, pages 42–1. JMLR Workshop and Conference Proceedings, 2012.

Sébastien Bubeck, Nicolo Cesa-Bianchi, and Sham M Kakade. Towards minimax policies for online linear optimization with bandit feedback. In *Conference on Learning Theory*, pages 41–1. JMLR Workshop and Conference Proceedings, 2012a.

Sébastien Bubeck, Nicolo Cesa-Bianchi, et al. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012b.

Nicolo Cesa-Bianchi and Gábor Lugosi. Combinatorial bandits. *Journal of Computer and System Sciences*, 78(5):1404–1422, 2012.

Tianyi Chen, Qing Ling, and Georgios B Giannakis. An online convex optimization approach to proactive network resource allocation. *IEEE Transactions on Signal Processing*, 65(24):6350–6364, 2017.

Alon Cohen, Tamir Hazan, and Tomer Koren. Tight bounds for bandit combinatorial optimization. In *Conference on Learning Theory*, pages 629–642. PMLR, 2017.

Varsha Dani and Thomas P Hayes. Robbing the bandit: Less regret in online geometric optimization against an adaptive adversary. In *SODA*, volume 6, pages 937–943, 2006.

Varsha Dani, Sham M Kakade, and Thomas Hayes. The price of bandit information for online optimization. *Advances in Neural Information Processing Systems*, 20, 2007.

Puja Das. Online convex optimization and its application to online portfolio selection. 2014.

Davide Della Giustina, Nicola Prezza, and Rossano Venturini. A new linear-time algorithm for centroid decomposition. In *International Symposium on String Processing and Information Retrieval*, pages 274–282. Springer, 2019.

Gabriele Farina, Robin Schmucker, and Tuomas Sandholm. Bandit linear optimization for sequential decision making and extensive-form games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 5372–5380, 2021.

Côme Fiegel, Pierre Ménard, Tadashi Kozuno, Rémi Munos, Vianney Perchet, and Michal Valko. Adapting to game trees in zero-sum imperfect information games. In *International Conference on Machine Learning*, pages 10093–10135. PMLR, 2023.

Erzsébet Frigó and Levente Kocsis. Online convex combination of ranking models. *User Modeling and User-Adapted Interaction*, 32(4):649–683, 2022.

Geoffrey J Gordon. No-regret algorithms for online convex programs. *Advances in Neural Information Processing Systems*, 19, 2006.

András György, Tamás Linder, Gábor Lugosi, and György Ottucsák. The on-line shortest path problem under partial monitoring. *Journal of Machine Learning Research*, 8(10), 2007.

Elad Hazan and Zohar Karnin. Volumetric spanners: an efficient exploration basis for learning. *Journal of Machine Learning Research*, 2016.

Shinji Ito, Daisuke Hatano, Hanna Sumita, Kei Takemura, Takuro Fukunaga, Naonori Kakimura, and Ken-Ichi Kawarabayashi. Improved regret bounds for bandit combinatorial optimization. *Advances in Neural Information Processing Systems*, 32, 2019.

Shinji Ito, Shuichi Hirahara, Tasuku Soma, and Yuichi Yoshida. Tight first-and second-order regret bounds for adversarial linear bandits. *Advances in Neural Information Processing Systems*, 33: 2028–2038, 2020.

Camille Jordan. Sur les assemblages de lignes. 1869.

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

Chung-Wei Lee, Haipeng Luo, Chen-Yu Wei, and Mengxiao Zhang. Bias no more: high-probability data-dependent regret bounds for adversarial bandits and mdps. *Advances in neural information processing systems*, 33:15522–15533, 2020.

Jinjiao Lin, Yibin Li, and Jian Lian. A novel recommendation system via l0-regularized convex optimization. *Neural Computing and Applications*, 32:1649–1663, 2020.

H Brendan McMahan and Avrim Blum. Online geometric optimization in the bandit setting against an adaptive adversary. In *Learning Theory: 17th Annual Conference on Learning Theory, COLT 2004, Banff, Canada, July 1-4, 2004. Proceedings 17*, pages 109–123. Springer, 2004.

Gergely Neu. Explore no more: Improved high-probability regret bounds for non-stochastic bandits. *Advances in Neural Information Processing Systems*, 28, 2015.

Aleksandrs Slivkins et al. Introduction to multi-armed bandits. *Foundations and Trends® in Machine Learning*, 12(1-2):1–286, 2019.

Julian Zimmert and Tor Lattimore. Return of the bias: Almost minimax optimal high probability bounds for adversarial linear bandits. In *Conference on Learning Theory*, pages 3285–3312. PMLR, 2022.

Julian Zimmert and Yevgeny Seldin. Tsallis-inf: An optimal algorithm for stochastic and adversarial bandits. *Journal of Machine Learning Research*, 22(28):1–49, 2021.

Julian Zimmert, Haipeng Luo, and Chen-Yu Wei. Beating stochastic and adversarial semi-bandits optimally and simultaneously. In *International Conference on Machine Learning*, pages 7683–7692. PMLR, 2019.

Appendix A. Related Works

Multi-Armed Bandits. For non-stochastic Multi-Armed Bandits, Auer et al. (2002) introduced the EXP3 algorithm (short for "exponential-weight algorithm for exploration and exploitation"), which achieves a pseudo-regret of $\mathcal{O}(\sqrt{KT \log K})$. They also established a regret lower bound of $\Omega(\sqrt{KT})$. Subsequently, Audibert and Bubeck (2010) introduced the implicitly normalized forecaster, achieving a pseudo-regret bound of $\mathcal{O}(\sqrt{KT})$. Building on this, Bubeck and Slivkins (2012) initiated the study of "best of both worlds" algorithms, which attain near-optimal pseudo-regret bounds in both stochastic and non-stochastic settings. Finally, Zimmert and Seldin (2021) demonstrated that Tsallis-1/2-INF achieves optimal pseudo-regret bounds for the best of both worlds problem.

Auer et al. (2002) also introduced a variant of EXP3, called EXP3.P, which incorporates explicit exploration and achieves a regret of $\mathcal{O}(\sqrt{KT \log(KT/\delta)})$ with probability at least $1-\delta$. Bubeck et al. (2012b) later analyzed a version of EXP3.P that attains a regret of $5.15\sqrt{KT \log(K/\delta)}$ with the same probability guarantee. Building on this, Neu (2015) proposed EXP3-IX (EXP3 with Implicit Exploration), which leverages implicit exploration to achieve a regret of $2\sqrt{2KT \log(K/\delta)}$ with probability at least $1-\delta$.

Adversarial Linear Bandits. For a bounded arm set $\mathcal{X} \subset \mathbb{R}^d$ and loss values in $[-1, 1]$ for any arm, McMahan and Blum (2004) were the first to design a sublinear regret algorithm, achieving an expected regret of $T^{3/4}$. A later, improved analysis by Dani and Hayes (2006) reduced this bound to $T^{2/3}$, while maintaining polynomial dependence on d. This result holds even against an adaptive adversary.

For the special case of DAGs, Awerbuch and Kleinberg (2004) designed the first algorithm with a pseudo-regret of $T^{2/3}$ and polynomial dependence on d. Subsequently, György et al. (2007) extended this result by developing an algorithm that achieves a high-probability regret bound of $T^{2/3}$, also with polynomial dependence on d, even against an adaptive adversary.

Dani et al. (2007) were the first to design an algorithm called Geometric Hedge, which achieves a regret of $T^{1/2}$ with polynomial dependence on d. Later, Bartlett et al. (2008) introduced a variant of Geometric Hedge that incurs a high-probability regret bound of $\widetilde{\mathcal{O}}(d^{3/2}\sqrt{T})$.

Cesa-Bianchi and Lugosi (2012) followed up by designing an algorithm called Comband, which achieves a pseudo-regret of $\mathcal{O}(\sqrt{dT \log |\mathcal{X}|})$ for various combinatorial sets $\mathcal{X} \subseteq \{0, 1\}^d$. Subsequently, Bubeck et al. (2012a) showed that EXP2 with John's exploration incurs a pseudo-regret of $\mathcal{O}(\sqrt{dT \log |\mathcal{X}|})$ for any finite set $\mathcal{X} \subseteq \mathbb{R}^d$. For a general regret analysis of a similar algorithm, EXP3 for Linear Bandits with any fixed exploration distribution, we refer the reader to Lattimore and Szepesvári (2020). Later, Zimmert and Lattimore (2022) designed a high-probability version called EXP3 with Kiefer-Wolfowitz exploration, which achieves a regret of $\mathcal{O}(\sqrt{dT \log(|\mathcal{X}|/\delta)})$ with probability at least $1-\delta$ for any finite set $\mathcal{X} \subseteq \mathbb{R}^d$.

For the combinatorial setting, where the loss of each individual coordinate is bounded between -1 and 1, Audibert et al. (2014) provided near-optimal worst-case upper bounds for both semi-bandit and bandit feedback. For combinatorial sets such as m-sets, DAGs, multi-task MAB, and maximum matching in bipartite graphs, Cohen et al. (2017); Ito et al. (2019) established tight worst-case lower bounds under bandit feedback. The techniques used in these works can be appropriately adapted to derive tight lower bounds for the standard bandit setting, where the loss value of any arm lies within $[-1, 1]$.

Computationally Efficient Algorithms. For compact convex sets $\mathcal{X} \subset \mathbb{R}^d$, Abernethy et al. (2008) were the first to propose a computationally efficient algorithm that achieves a pseudo-regret of $\text{poly}(d) \cdot \sqrt{T}$. Their approach leveraged efficient self-concordant barriers. They also analyzed the online shortest path problem in DAGs, providing an efficient algorithm with a pseudo-regret of $\widetilde{\mathcal{O}}(\sqrt{|E|^3 T})$.

Cesa-Bianchi and Lugosi (2012) demonstrated computationally efficient implementations of ComBand for certain combinatorial sets. For general convex decision sets, Hazan and Karnin (2016) designed a computationally efficient algorithm with $\widetilde{\mathcal{O}}(d\sqrt{T})$ pseudo-regret using volumetric spanners. Their approach extends to the online shortest path problem in DAGs, where their efficient algorithm achieves a pseudo-regret of $\widetilde{\mathcal{O}}(\sqrt{|E|T \log |\mathcal{X}|})$.

Given access to an efficient linear optimization oracle, Ito et al. (2020) proposed a computationally efficient algorithm based on continuous multiplicative weight updates, which achieves $\widetilde{\mathcal{O}}(d\sqrt{T})$ pseudo-regret while also providing tight first- and second-order guarantees.

For compact convex sets $\mathcal{X} \subset \mathbb{R}^d$, Lee et al. (2020) proposed the first efficient algorithm achieving a high-probability regret of $\text{poly}(d) \cdot \sqrt{T}$ against an adaptive adversary. Their approach leveraged an efficient self-concordant barrier and yielded a worst-case high-probability regret of $\widetilde{\mathcal{O}}(\sqrt{d^7 T})$. Subsequently, Zimmert and Lattimore (2022) developed an improved efficient algorithm with a regret bound of $\widetilde{\mathcal{O}}(d^2\sqrt{T})$ with high probability against an adaptive adversary, which remains the best known result to date. Their method relied on the entropic barrier. Notably, both high-probability guarantees were formally established only for continuous decision sets. However, we believe their analysis extends to discrete decision sets, such as paths in a DAG, using the same techniques as Abernethy et al. (2008).

Appendix B. Technical Lemmas

Lemma 9 (Slivkins et al. (2019)) *Fix $\varepsilon \in (0, \frac{1}{4})$. Let RC_ε denote a random coin with bias ε, i.e., a distribution over $\{0,1\}$ with expectation $\frac{1}{2} + \varepsilon$. Then $KL(RC_\varepsilon, RC_0) \le 8\varepsilon^2$ and $KL(RC_0, RC_\varepsilon) \le 4\varepsilon^2$.*

Lemma 10 (Chain Rule) *Let $f(x_1, x_2, \ldots, x_n)$ and $g(x_1, x_2, \ldots, x_n)$ be two joint PMFs for a tuple of random variables $(X_i)_{i\in[n]}$. Let the sample space be $\Omega = \{0,1\}^n$. Then we have the following:*

$$KL(f, g) = \sum_{\omega \in \Omega} f(\omega) \left(KL(f(X_1), g(X_1)) + \sum_{i=2}^{n} KL(f(X_i | X_{-i} = \omega_{-i}), g(X_i | X_{-i} = \omega_{-i})) \right)$$

where $X_{-i} = (X_1, \ldots, X_{i-1})$, $\omega_{-i} = (\omega_1, \ldots, \omega_{i-1})$.

Proof. Let $\Omega^i = \{0,1\}^i$. Now we have the following:

$$\begin{aligned}
\mathrm{KL}(f,g) &= \sum_{\omega\in\Omega} f(\omega)\log\left(\frac{f(\omega)}{g(\omega)}\right)\\
&= \sum_{\omega\in\Omega} f(\omega)\log\left(\frac{f(\omega_1)\prod_{i=2}^n f(\omega_i|\omega_{-i})}{g(\omega_1)\prod_{i=2}^n g(\omega_i|\omega_{-i})}\right)\\
&= \sum_{\omega\in\Omega} f(\omega)\left(\log\left(\frac{f(\omega_1)}{g(\omega_1)}\right)+\sum_{i=2}^n\log\left(\frac{f(\omega_i|\omega_{-i})}{g(\omega_i|\omega_{-i})}\right)\right)\\
&= \sum_{\omega\in\Omega} f(\omega)\log\left(\frac{f(\omega_1)}{g(\omega_1)}\right) + \sum_{i=2}^n\sum_{\omega\in\Omega} f(\omega)\log\left(\frac{f(\omega_i|\omega_{-i})}{g(\omega_i|\omega_{-i})}\right)\\
&= \sum_{\omega_1\in\mathbb{R}} f(\omega_1)\log\left(\frac{f(\omega_1)}{g(\omega_1)}\right) + \sum_{i=2}^n\sum_{\omega\in\Omega^i} f(\omega)\log\left(\frac{f(\omega_i|\omega_{-i})}{g(\omega_i|\omega_{-i})}\right)\\
&= \mathrm{KL}(f(X_1),g(X_1)) + \sum_{i=2}^n\sum_{\omega_{-i}\in\Omega^{i-1}} f(\omega_{-i})\sum_{\omega_i\in\Omega^1} f(\omega_i)\log\left(\frac{f(\omega_i|\omega_{-i})}{g(\omega_i|\omega_{-i})}\right)\\
&= \mathrm{KL}(f(X_1),g(X_1)) + \sum_{i=2}^n\sum_{\omega_{-i}\in\Omega^{i-1}} f(\omega_{-i})\mathrm{KL}(f(X_i|X_{-i}=\omega_{-i}),g(X_i|X_{-i}=\omega_{-i}))\\
&= \sum_{\omega\in\Omega} f(\omega)\mathrm{KL}(f(X_1),g(X_1)) + \sum_{i=2}^n\sum_{\omega\in\Omega} f(\omega)\mathrm{KL}(f(X_i|X_{-i}=\omega_{-i}),g(X_i|X_{-i}=\omega_{-i}))\\
&= \sum_{\omega\in\Omega} f(\omega)\left(\mathrm{KL}(f(X_1),g(X_1)) + \sum_{i=2}^n\mathrm{KL}(f(X_i|X_{-i}=\omega_{-i}),g(X_i|X_{-i}=\omega_{-i}))\right)
\end{aligned}$$

■

Lemma 11 ((Fiegel et al., 2023)) *Let $(u_t)_{t\in[\![T]\!]}$ be a random process adapted to the filtration $(\mathcal{F}_t)_{t\in[T]}$ such that $0 \le u_t \le H$ for all $t \in [\![T]\!]$. Then, with probability at least $1-\delta$, we have*

$$\sum_{t=1}^T [u_t - \mathbb{E}[u_t|\mathcal{F}_{t-1}]] \le H\sqrt{2T\log(1/\delta)}$$

Similarly, with probability at least $1-\delta$, we have

$$\sum_{t=1}^T [\mathbb{E}[u_t|\mathcal{F}_{t-1}] - u_t] \le H\sqrt{2T\log(1/\delta)}$$

Corollary 12 ((Fiegel et al., 2023)) *Let $(u_t)_{t\in[\![T]\!]}$ be a random process adapted to the filtration $(\mathcal{F}_t)_{t\in[T]}$ such that $-H \le u_t \le H$ for all $t \in [\![T]\!]$. Then, with probability at least $1-\delta$, we have*

$$\sum_{t=1}^T [u_t - \mathbb{E}[u_t|\mathcal{F}_{t-1}]] \le H\sqrt{8T\log(1/\delta)}$$

Similarly, with probability at least $1-\delta$*, we have*

$$\sum_{t=1}^{T}[\mathbb{E}[u_t|\mathcal{F}_{t-1}]-u_t] \le H\sqrt{8T\log(1/\delta)}$$

Lemma 13 (Lattimore and Szepesvári (2020)) *Let* $\eta > 0$ *and* f *be Legendre and twice differentiable with positive definite Hessian in* $A = int(dom(f))$*. Then for all* $x, y \in A$*, there exists a* $z \in [x,y] = \{(1-\alpha)x + \alpha y : \alpha \in [0,1]\}$ *such that*

$$\langle x-y, u\rangle - \frac{D_f(x,y)}{\eta} \le \frac{\eta}{2}\|u\|^2_{(\nabla^2 f(z))^{-1}}.$$

where $D_f(x,y)$ *with respect to* f*.*

Appendix C. General Framework for FTRL with Implicit Exploration

In this section, we extend the ideas from Neu (2015) to provide a general framework for using FTRL with implicit exploration in combinatorial bandits. In this setting, we are given a combinatorial set $\mathcal{X} \subseteq \{0,1\}^d$. Define the ℓ_1-norm of the set as $m := \max_{\mathbf{x}\in\mathcal{X}} \|\mathbf{x}\|_1$.

In each round t, the algorithm selects $\mathbf{x}_t \in \mathcal{X}$ and incurs a loss given by $\ell_t := \langle \mathbf{x}_t, \mathbf{y}_t\rangle \in [-1,1]$ where $\mathbf{y}_t$ is the loss vector chosen adaptively by an adversary based on the past filtration $\mathcal{F}_{t-1}$ and the algorithm. The regret with respect to a fixed element $\mathbf{x} \in \mathcal{X}$ is defined as

$$R_T(\mathbf{x}) := \sum_{t=1}^{T}\langle \mathbf{x}_t, \mathbf{y}_t\rangle - \sum_{t=1}^{T}\langle \mathbf{x}, \mathbf{y}_t\rangle.$$

The goal is to provide a high-probability regret guarantee on $\max_{\mathbf{x}\in\mathcal{X}} R_T(\mathbf{x})$.

Let $\widetilde{\mathbf{y}}_t \in \mathbb{R}^d$ be a *relatively unbiased* estimator defined as

$$\widetilde{\mathbf{y}}_t[i] := \frac{\mathbb{1}[\mathbf{x}_t[i]=1]\cdot \ell_{t,i}}{\mathbb{P}_t[\mathbf{x}_t[i]=1]},$$

where $\ell_{t,i}$ is some random variable based on ℓ_t. Assume there is an absolute constant b such that $\ell_{t,i} \in [0,b]$ for every $t \in [\![T]\!]$ and $i \in [\![d]\!]$. An estimator is *relatively unbiased* when it satisfies

$$\mathbb{E}_t[\langle \mathbf{x}-\mathbf{x}', \widetilde{\mathbf{y}}_t\rangle] = \langle \mathbf{x}-\mathbf{x}', \mathbf{y}_t\rangle$$

for any two elements $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$, preserving their differences.

We analyze the FTRL algorithm, which follows the update rule

$$\widetilde{\mathbf{x}}_t \leftarrow \arg\min_{\mathbf{x}\in \mathrm{co}(\mathcal{X})} \left(\eta \sum_{\tau=1}^{t-1}\langle \mathbf{x}, \widehat{\mathbf{y}}_\tau\rangle + F(\mathbf{x})\right),$$

where $F(\cdot)$ is a Legendre function such that $\nabla^2 F(\cdot)$ is always a diagonal matrix with positive diagonal entries for any point on the chord $[\widetilde{x}_t, \widetilde{x}_{t+1}]$. Moreover, $\widehat{\mathbf{y}}_t \in \mathbb{R}^d$ is the loss estimator given by

$$\widehat{\mathbf{y}}_t[i] := \frac{\mathbb{1}[\mathbf{x}_t[i]=1]\cdot \ell_{t,i}}{\mathbb{P}_t[\mathbf{x}_t[i]=1]+\gamma_i}.$$

The algorithm then samples $\mathbf{x}_t \in \mathcal{X}$ such that $\mathbb{E}_t[\mathbf{x}_t] = \widetilde{\mathbf{x}}_t$. Note that it holds $\mathbb{P}_t[\mathbf{x}_t[i] = 1] = \widetilde{\mathbf{x}}_t[i]$ for every $i \in [\![d]\!]$.

Now, we begin our regret analysis.

Lemma 14 *Denote by $\mathcal{E}_1$ the event that*

$$\sum_{t=1}^{T} \langle \mathbf{x}_t - \widetilde{\mathbf{x}}_t, \mathbf{y}_t \rangle \leq \sqrt{8T \log(1/\delta_0)}.$$

It satisfies that $\mathbb{P}[\mathcal{E}_1] \geq 1 - \delta_0$.

Proof. First observe that

$$\sum_{t=1}^{T} \langle \mathbf{x}_t, \mathbf{y}_t \rangle = \sum_{t=1}^{T} \langle \widetilde{\mathbf{x}}_t, \mathbf{y}_t \rangle + \sum_{t=1}^{T} (\langle \mathbf{x}_t, \mathbf{y}_t \rangle - \mathbb{E}_t[\langle \mathbf{x}_t, \mathbf{y}_t \rangle]).$$

As $\langle \mathbf{x}_t, \mathbf{y}_t \rangle \in [-1, 1]$, according to Corollary 12, with probability at least $1 - \delta_0$, we have

$$\sum_{t=1}^{T} (\langle \mathbf{x}_t, \mathbf{y}_t \rangle - \mathbb{E}_t[\langle \mathbf{x}_t, \mathbf{y}_t \rangle]) \leq \sqrt{8T \log(1/\delta_0)},$$

which concludes the proof. ■

Lemma 15 *Denote by $\mathcal{E}_2$ the event that*

$$\sum_{t=1}^{T} \langle \widetilde{\mathbf{x}}_t, \mathbb{E}_t[\widehat{\mathbf{y}}_t] - \widehat{\mathbf{y}}_t \rangle \leq b \cdot m \sqrt{2T \log(1/\delta_0)}.$$

It satisfies that $\mathbb{P}[\mathcal{E}_2] \geq 1 - \delta_0$.

Proof. The lemma can be proved directly by applying Lemma 11 and the fact that $\langle \widetilde{\mathbf{x}}_t, \widehat{\mathbf{y}}_t \rangle \in [0, b \cdot m]$. ■

Lemma 16 *It always holds that*

$$\sum_{t=1}^{T} \langle \widetilde{\mathbf{x}}_t, \mathbb{E}_t[\widetilde{\mathbf{y}}_t] - \mathbb{E}_t[\widehat{\mathbf{y}}_t] \rangle \leq b \cdot T \cdot \sum_{i=1}^{d} \gamma_i.$$

Proof. From definition,

$$\begin{aligned}
\sum_{t=1}^{T} \langle \widetilde{\mathbf{x}}_t, \mathbb{E}_t[\widetilde{\mathbf{y}}_t] - \mathbb{E}_t[\widehat{\mathbf{y}}_t] \rangle &= \sum_{t=1}^{T} \sum_{i=1}^{d} \widetilde{\mathbf{x}}_t[i] \cdot \mathbb{E}_t[\widetilde{\mathbf{y}}_t[i]] \cdot \left(1 - \frac{\widetilde{\mathbf{x}}_t[i]}{\widetilde{\mathbf{x}}_t[i] + \gamma_i}\right) \\
&\leq b \cdot \sum_{t=1}^{T} \sum_{i=1}^{d} \frac{\widetilde{\mathbf{x}}_t[i] \cdot \gamma_i}{\widetilde{\mathbf{x}}_t[i] + \gamma_i} \qquad (\widetilde{\mathbf{x}}_t[i] \leq 1, \mathbb{E}_t[\widetilde{\mathbf{y}}_t[i]] \leq b) \\
&\leq b \cdot \sum_{t=1}^{T} \sum_{i=1}^{d} \gamma_i \\
&= b \cdot T \cdot \sum_{i=1}^{d} \gamma_i
\end{aligned}$$

■

Lemma 17 *Define $\beta_i := 2\gamma_i/b$. Let $\mathcal{E}_3$ be the event that simultaneously for all $i \in [\![d]\!]$,*

$$\sum_{t=1}^{T}(\widehat{\mathbf{y}}_t[i] - \mathbb{E}_t[\widetilde{\mathbf{y}}_t[i]]) \le \frac{\log(d/\delta_0)}{\beta_i}$$

It satisfies that $\mathbb{P}[\mathcal{E}_3] \ge 1 - \delta_0$. Furthermore, under event $\mathcal{E}_3$, it satisfies that

$$\sum_{t=1}^{T}\langle \mathbf{x}, \widehat{\mathbf{y}}_t - \mathbb{E}_t[\widetilde{\mathbf{y}}_t]\rangle \le \sum_{i=1}^{d}\mathbf{x}[i]\cdot\frac{\log(d/\delta_0)}{\beta_i}, \qquad \forall \mathbf{x} \in \mathcal{X}.$$

Proof. Fix $i \in [\![d]\!]$. First we have the following:

$$\begin{aligned}
\widehat{\mathbf{y}}_t[i] &= \frac{\mathbf{x}_t[i]\cdot\ell_{t,i}}{\widetilde{\mathbf{x}}_t[i] + \gamma_i} \\
&\le \frac{\mathbf{x}_t[i]\cdot\ell_{t,i}}{\widetilde{\mathbf{x}}_t[i] + (\gamma_i/b)\cdot\ell_{t,i}} && (\text{as } \ell_{t,i} \in [0,b]) \\
&= \frac{1}{\beta_i}\cdot\frac{\beta_i\cdot\mathbf{x}_t[i]\cdot\ell_{t,i}}{\widetilde{\mathbf{x}}_t[i] + (\beta_i/2)\cdot\ell_{t,i}} && (\text{as } \beta_i = \tfrac{2\gamma_i}{b}) \\
&\le \frac{1}{\beta_i}\cdot\frac{\beta_i\cdot\mathbf{x}_t[i]\cdot\ell_{t,i}}{\widetilde{\mathbf{x}}_t[i] + (\beta_i/2)\cdot\mathbf{x}_t[i]\cdot\ell_{t,i}} && (\text{as } \mathbf{x}_t[i] \in \{0,1\}) \\
&= \frac{1}{\beta_i}\cdot\frac{\beta_i\cdot\widetilde{\mathbf{y}}_t[i]}{1 + (\beta_i/2)\cdot\widetilde{\mathbf{y}}_t[i]} && (\text{as } \widetilde{\mathbf{y}}_t[i] = \mathbf{x}_t[i]\cdot\ell_{t,i}/\widetilde{\mathbf{x}}_t[i]) \\
&\le \frac{1}{\beta_i}\cdot\ln(1 + \beta_i\cdot\widetilde{\mathbf{y}}_t[i]) && (\text{as } \tfrac{z}{1+z/2} \le \ln(1+z) \text{ for all } z \ge 0)
\end{aligned}$$

Next, we have the following:

$$\begin{aligned}
\mathbb{E}_t[\exp(\beta_i\widehat{\mathbf{y}}_t[i])] &\le \mathbb{E}_t[(1 + \beta_i\widetilde{\mathbf{y}}_t[i])] \\
&= 1 + \beta_i\mathbb{E}_t[\widetilde{\mathbf{y}}_t[i]] \\
&\le \exp(\beta_i\mathbb{E}_t[\widetilde{\mathbf{y}}_t[i]]) && (\text{as } 1 + z \le \exp(z) \text{ for all } z \in \mathbb{R})
\end{aligned}$$

Hence, the process $Z_0 = 1$ and $Z_t = \exp(\beta_i\sum_{\tau=1}^{t}(\widehat{\mathbf{y}}_\tau[i] - \mathbb{E}_t[\widetilde{\mathbf{y}}_\tau[i]]))$ for all $t \ge 1$ is a supermartingale with respect to $(\mathcal{F}_t)$ as $\mathbb{E}_t[Z_t] \le Z_{t-1}$. Hence, we have $\mathbb{E}[Z_t] \le \mathbb{E}[Z_{t-1}] \le \ldots \le 1$. Therefore, by Markov inequality we have,

$$\mathbb{P}\left[\sum_{t=1}^{T}\widehat{\mathbf{y}}_t[i] - \mathbb{E}_t[\widetilde{\mathbf{y}}_t[i]] > \frac{\log(d/\delta_0)}{\beta_i}\right] \le \mathbb{E}\left[\exp\left(\beta_i\cdot\sum_{t=1}^{T}(\widehat{\mathbf{y}}_t[i] - \mathbb{E}_t[\widetilde{\mathbf{y}}_t[i]])\right)\right]\cdot\exp\left(\log(d/\delta_0)\right) \le \frac{\delta_0}{d}$$

By union bound over $i \in [\![d]\!]$, we get that the event $\mathcal{E}_3$ holds with probability at least $1 - \delta_0$.

Finally, under event $\mathcal{E}_3$,

$$\sum_{t=1}^{T}\langle \mathbf{x}, \widehat{\mathbf{y}}_t - \mathbb{E}_t[\widetilde{\mathbf{y}}_t]\rangle = \sum_{i=1}^{d}\mathbf{x}[i]\cdot\sum_{t=1}^{T}(\widehat{\mathbf{y}}_t[i] - \mathbb{E}_t[\widetilde{\mathbf{y}}_t[i]]) \le \sum_{i=1}^{d}\mathbf{x}[i]\cdot\frac{\log(d/\delta_0)}{\beta_i}$$

■

Lemma 18 *Given Bregman divergence $\mathcal{D}_F(p,q) := F(p) - F(q) - \langle \nabla F(q), p-q\rangle$, let*

$$\texttt{VAR}_t := \langle \widetilde{\mathbf{x}}_t - \widetilde{\mathbf{x}}_{t+1}, \widehat{\mathbf{y}}_t^+\rangle - \frac{1}{\eta}\cdot\mathcal{D}_F(\widetilde{\mathbf{x}}_{t+1}, \widetilde{\mathbf{x}}_t).$$

Denote by $\mathcal{E}_4$ the event that

$$\sum_{t=1}^{T}\texttt{VAR}_t \le \sum_{t=1}^{T}\mathbb{E}_t\Big[\frac{\eta}{2}||\widehat{\mathbf{y}}_t^+||^2_{(\nabla^2 F(\mathbf{z}_t))^{-1}}\Big] + b\cdot m\cdot\sqrt{2T\log(1/\delta_0)}$$

where $\widehat{\mathbf{y}}_t^+ \in \mathbb{R}^d$ is a vector defined as

$$\widehat{\mathbf{y}}_t^+[i] := \widehat{\mathbf{y}}_t[i]\cdot\mathbb{1}[\widetilde{\mathbf{x}}_{t+1}[i] \le \widetilde{\mathbf{x}}_t[i]].$$

It satisfies that $\mathbb{P}[\mathcal{E}_4] \ge 1-\delta_0$.

Proof. Let $\texttt{VAR}_t^+ := \max\{\texttt{VAR}_t, 0\}$. Since $\mathcal{D}_F(\widetilde{\mathbf{x}}_{t+1}, \widetilde{\mathbf{x}}_t) \ge 0$ and $\widehat{\mathbf{y}}_t^+[i] \ge 0$ for all $i \in [\![d]\!]$, we obtain the following:

$$\texttt{VAR}_t^+ \le \langle \widetilde{\mathbf{x}}_t, \widehat{\mathbf{y}}_t^+\rangle \le \langle \widetilde{\mathbf{x}}_t, \widehat{\mathbf{y}}_t\rangle \le b\cdot m.$$

According to Lemma 11, with probability at least $1-\delta_0$, we have

$$\sum_{t=1}^{T}\texttt{VAR}_t^+ \le \sum_{t=1}^{T}\mathbb{E}_t[\texttt{VAR}_t^+] + b\cdot m\cdot\sqrt{2T\log(1/\delta_0)}.$$

Next, due to Lemma 13, we have $\texttt{VAR}_t \le \frac{\eta}{2}||\widehat{\mathbf{y}}_t^+||^2_{(\nabla^2 F(\mathbf{z}_t))^{-1}}$, where $\mathbf{z}_t$ is some point on the chord $[\widetilde{\mathbf{x}}_t, \widetilde{\mathbf{x}}_{t+1}]$. Since $\frac{\eta}{2}||\widehat{\mathbf{y}}_t^+||^2_{(\nabla^2 F(\mathbf{z}_t))^{-1}} \ge 0$, it follows that $\texttt{VAR}_t^+ \le \frac{\eta}{2}||\widehat{\mathbf{y}}_t^+||^2_{(\nabla^2 F(\mathbf{z}_t))^{-1}}$.

Since $\texttt{VAR}_t \le \texttt{VAR}_t^+$ for all $t \in [\![T]\!]$, the event $\mathcal{E}_4$ holds with probability at least $1-\delta_0$. ■

Let $\mathcal{E} := \bigcup_{i=1}^{4}\mathcal{E}_i$ be our good event. Due to union bound, the event $\mathcal{E}$ holds with probability at least $1-5\delta_0$. Let us assume that the good event $\mathcal{E}$ holds and let us fix $\mathbf{x} \in \mathcal{X}$. First we have the

following:

$$\begin{aligned}
R_T(\mathbf{x}) &= \sum_{t=1}^{T} \langle \mathbf{x}_t - \mathbf{x}, \mathbf{y}_t \rangle \\
&\leq \sum_{t=1}^{T} \langle \widetilde{\mathbf{x}}_t - \mathbf{x}, \mathbf{y}_t \rangle + \sqrt{8T \log(1/\delta_0)} && \text{(as event } \mathcal{E}_1 \text{ holds)} \\
&= \sum_{t=1}^{T} \langle \widetilde{\mathbf{x}}_t - \mathbf{x}, \mathbb{E}_t[\widetilde{\mathbf{y}}_t] \rangle + \sqrt{8T \log(1/\delta_0)} \\
&= \sum_{t=1}^{T} \langle \widetilde{\mathbf{x}}_t - \mathbf{x}, \widehat{\mathbf{y}}_t \rangle + \sum_{t=1}^{T} \langle \widetilde{\mathbf{x}}_t, \mathbb{E}_t[\widehat{\mathbf{y}}_t] - \widehat{\mathbf{y}}_t \rangle + \sum_{t=1}^{T} \langle \widetilde{\mathbf{x}}_t, \mathbb{E}_t[\widetilde{\mathbf{y}}_t] - \mathbb{E}_t[\widehat{\mathbf{y}}_t] \rangle \\
&\quad + \sum_{t=1}^{T} \langle \mathbf{x}, \widehat{\mathbf{y}}_t - \mathbb{E}_t[\widetilde{\mathbf{y}}_t] \rangle + \sqrt{8T \log(1/\delta_0)} \\
&\leq \sum_{t=1}^{T} \langle \widetilde{\mathbf{x}}_t - \mathbf{x}, \widehat{\mathbf{y}}_t \rangle + \sqrt{8T \log(1/\delta_0)} + b \cdot m \sqrt{2T \log(1/\delta_0)} \\
&\quad + b \cdot T \sum_{i=1}^{d} \gamma_i + b \cdot \sum_{i=1}^{d} \mathbf{x}[i] \cdot \frac{\log(d/\delta_0)}{2\gamma_i} && \text{(as events } \mathcal{E}_2, \mathcal{E}_3 \text{ hold)}
\end{aligned}$$

Observe that if $\gamma_i = \gamma$ for all $i \in [\![d]\!]$, we also get the following as $||x||_1 \leq m$:

$$R_T(\mathbf{x}) \leq \sum_{t=1}^{T} \langle \widetilde{\mathbf{x}}_t - \mathbf{x}, \widehat{\mathbf{y}}_t \rangle + \sqrt{8T \log(1/\delta_0)} + b \cdot m \sqrt{2T \log(1/\delta_0)} + b \cdot T \cdot d \cdot \gamma + b \cdot m \cdot \frac{\log(d/\delta_0)}{2\gamma}$$

As $\widetilde{\mathbf{x}}_t$ is the solution to our FTRL equation above, we get the following using the standard FTRL analysis from Lattimore and Szepesvári (2020) and the fact that event $\mathcal{E}_4$ holds:

$$\sum_{t=1}^{T} \langle \widetilde{\mathbf{x}}_t - \mathbf{x}, \widehat{\mathbf{y}}_t \rangle \leq \frac{\text{diam}_F}{\eta} + \sum_{t=1}^{T} \texttt{VAR}_t \leq \frac{\text{diam}_F}{\eta} + \sum_{t=1}^{T} \mathbb{E}_t \left[\frac{\eta}{2} ||\widehat{\mathbf{y}}_t^+||^2_{(\nabla^2 F(\mathbf{z}_t))^{-1}} \right] + b \cdot m \cdot \sqrt{2T \log(1/\delta_0)}$$

where $\mathbf{z}_t$ is some point on the chord $[\widetilde{\mathbf{x}}_t, \widetilde{\mathbf{x}}_{t+1}]$ and $\text{diam}_F := \max_{\mathbf{x}, \mathbf{x}' \in \text{co}(\mathcal{X})} F(\mathbf{x}) - F(\mathbf{x}')$. Now we have the following theorem under our assumptions.

Theorem 19 *Let $\delta_0 = \frac{\delta}{5}$ and $\gamma_i = \sqrt{\frac{m \log(5d/\delta)}{dT}}$ for all $i \in [\![d]\!]$. Given an FTRL algorithm satisfying the assumptions of this section, we have the following guarantee on regret with probability at least $1 - \delta$ against any adaptive adversary:*

$$\max_{\mathbf{x} \in \mathcal{X}} R_T(\mathbf{x}) \leq \frac{diam_F}{\eta} + \sum_{t=1}^{T} \mathbb{E}_t \left[\frac{\eta}{2} ||\widehat{\mathbf{y}}_t^+||^2_{(\nabla^2 F(\mathbf{z}_t))^{-1}} \right] + c \cdot b \cdot \sqrt{mdT \log(d/\delta)}$$

where c is some absolute constant and $\mathbf{z}_t$ is some point on the chord $[\widetilde{\mathbf{x}}_t, \widetilde{\mathbf{x}}_{t+1}]$.

Algorithm 1: FTRL for online shortest path in a DAG $G = (V, E)$ with equal path lengths

Let K be the length of every path in the DAG G from source to sink.
for $t = 1$ **to** T **do**
  Compute $\widetilde{\mathbf{x}}_t \leftarrow \arg\min_{\mathbf{x} \in \mathrm{co}(\mathcal{X})} \left(\eta \sum_{\tau=1}^{t-1} \langle \mathbf{x}, \widehat{\mathbf{y}}_\tau \rangle + F(\mathbf{x}) \right)$.
  Initialize path $P_t \leftarrow (v_{\mathsf{s}})$ and reset $v_0 \leftarrow v_{\mathsf{s}}$.
  for $i = 1$ **to** K **do**
    Sample outgoing edge $e_i = (v_{i-1}, v_i) \in \delta^+(v_{i-1})$ with probability proportional to $\widetilde{\mathbf{x}}_t[e_i]$.
    Update $P_t \leftarrow P_t \circ (e_i, v_i)$
  end
  Choose the path P_t and observe loss its ℓ_t.
  Determine $\mathbf{x}_t \in \mathcal{X}$ corresponding to the path P_t and construct the loss estimator $\widehat{\mathbf{y}}_t$.
end

Appendix D. DAGs: Additional Details

D.1. Omitted Details from Section 3.1

First, we prove the following proposition.

Proposition 20 $\mathbb{E}_t[\mathbf{x}_t] = \widetilde{\mathbf{x}}_t$

Proof. Let $v_1, v_2, \ldots, v_{|V|}$ be the topological order of the vertices in the DAG G, where $v_1 = v_{\mathsf{s}}$ and $v_{|V|} = v_{\mathsf{t}}$. We now prove our proposition using mathematical induction. Let $P(i)$ be the statement that for all $j \in [\![i]\!]$, we have $\mathbb{E}_t[\mathbf{x}_t[v_j]] = \widetilde{\mathbf{x}}_t[v_j]$ and $\mathbb{E}_t[\mathbf{x}_t[e]] = \widetilde{\mathbf{x}}_t[e]$ for any outgoing edge e from v_j.

Consider the base case of $i = 1$. First, observe that $\mathbb{E}_t[\mathbf{x}_t[v_1]] = 1 = \widetilde{\mathbf{x}}_t[v_1]$. Next, due to our sampling procedure, we have

$$\mathbb{E}_t[\mathbf{x}_t[e]] = \frac{\widetilde{\mathbf{x}}_t[e]}{\sum_{e' \in \delta^+(v_1)} \widetilde{\mathbf{x}}_t[e]} = \widetilde{\mathbf{x}}_t[e]$$

for any outgoing edge e from v_1. Hence, $P(1)$ is true.

Next, let us make the inductive hypothesis that $P(k)$ is true. Now we show that $P(k+1)$ is true. First, observe that

$$\begin{aligned}
\mathbb{E}_t[\mathbf{x}_t[v_{k+1}]] &= \mathbb{E}_t\Big[\sum_{e \in \delta^-(v_{k+1})} \mathbf{x}_t[e] \Big] \\
&= \sum_{e \in \delta^-(v_{k+1})} \widetilde{\mathbf{x}}_t[e] && \text{(due to inductive hypothesis)} \\
&= \widetilde{\mathbf{x}}_t[v_{k+1}] && \text{(due to flow constraints)}
\end{aligned}$$

Next, observe that for any outgoing edge e from v_{k+1}, we have:

$$\begin{aligned}
\mathbb{E}_t[\mathbf{x}_t[e]] &= \mathbb{E}_t[\mathbf{x}_t[e] \mid \mathbf{x}_t[v_{k+1}] = 1] \cdot \mathbb{P}_t[\mathbf{x}_t[v_{k+1}] = 1] \\
&= \frac{\widetilde{\mathbf{x}}_t[e]}{\sum_{e' \in \delta^+(v_{k+1})} \widetilde{\mathbf{x}}_t[e']} \cdot \widetilde{\mathbf{x}}_t[v_{k+1}] && \text{(due to our sampling procedure)} \\
&= \frac{\widetilde{\mathbf{x}}_t[e]}{\widetilde{\mathbf{x}}_t[v_{k+1}]} \cdot \widetilde{\mathbf{x}}_t[v_{k+1}] && \text{(due to flow constraints)} \\
&= \widetilde{\mathbf{x}}_t[e]
\end{aligned}$$

Hence, $P(k+1)$ is true. Hence, due to principle of mathematical induction, we have $\mathbb{E}_t[\mathbf{x}_t] = \widetilde{\mathbf{x}}_t$. ■

Next, recall that if all the paths have equal length of K, then we have $\mathbb{E}_t[\langle \mathbf{x} - \mathbf{x}', \widetilde{\mathbf{y}}_t \rangle] = \langle \mathbf{x} - \mathbf{x}', \mathbf{y}_t \rangle$ for all $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$. If we use $F(\mathbf{x}) = -\sum_{v \in V} \sqrt{\mathbf{x}[v]} - \sum_{e \in E} \sqrt{\mathbf{x}[e]}$ as our regularizer, it easily follows that Equation (1) has a unique minimizer $\widetilde{\mathbf{x}}_t$ and $\widetilde{\mathbf{x}}_t[v] > 0$ for all $v \in V$ and $\widetilde{\mathbf{x}}_t[e] > 0$ for all $e \in E$. Therefore, $\nabla^2 F(z)$ is a diagonal matrix for any point z on the chord $[\widetilde{x}_t, \widetilde{x}_{t+1}]$. Let $\widehat{\mathbf{y}}_t^+ \in \mathbb{R}_{\geq 0}^{V \cup E}$ be a vector indexed by the elements in $V \cup E$ such that $\widehat{\mathbf{y}}_t^+[v] = \widehat{\mathbf{y}}_t[v] \cdot \mathbb{1}[\widetilde{\mathbf{x}}_{t+1}[v] \leq \widetilde{\mathbf{x}}_t[v]]$ for all $v \in V$ and $\widehat{\mathbf{y}}_t^+[e] = \widehat{\mathbf{y}}_t[e] \cdot \mathbb{1}[\widetilde{\mathbf{x}}_{t+1}[e] \leq \widetilde{\mathbf{x}}_t[e]]$ for all $e \in E$. Let us set every entry of $\boldsymbol{\gamma}$ to $\sqrt{\frac{K \log(5(|V|+|E|)/\delta)}{|E|T}}$. Now we apply Lemma 19 from Appendix C to get the following:

Theorem 21 *Algorithm 1 has the following guarantee on regret with probability at least* $1 - \delta$ *against any adaptive adversary:*

$$\max_{\mathbf{x} \in \mathcal{X}} R_T(\mathbf{x}) \leq \frac{\operatorname{diam}_F(\operatorname{co}(\mathcal{X}))}{\eta} + \sum_{t=1}^{T} \mathbb{E}_t \left[\frac{\eta}{2} ||\widehat{\mathbf{y}}_t^+||^2_{(\nabla^2 F(z_t))^{-1}} \right] + c \cdot \sqrt{K|E|T \log(|E|/\delta)}$$

where c *is some absolute constant,* $\operatorname{diam}_F(\operatorname{co}(\mathcal{X})) := \max_{\mathbf{x}, \mathbf{x}' \in \operatorname{co}(\mathcal{X})} F(\mathbf{x}) - F(\mathbf{x}')$*, and* z_t *is some point on the chord* $[\widetilde{\mathbf{x}}_t, \widetilde{\mathbf{x}}_{t+1}]$.

First we upper bound $\sum_{v \in V} \sqrt{\mathbf{x}[v]} + \sum_{e \in E} \sqrt{\mathbf{x}[e]}$ for any $\mathbf{x} \in \operatorname{co}(\mathcal{X})$ as follows:

$$\begin{aligned}
\sum_{v \in V} \sqrt{\mathbf{x}[v]} + \sum_{e \in E} \sqrt{\mathbf{x}[e]} &\leq \sqrt{(|V| + |E|) \cdot \left(\sum_{v \in V} \mathbf{x}[v] + \sum_{e \in E} \mathbf{x}[e]\right)} && \text{(Cauchy-Schwarz inequality)} \\
&\leq \sqrt{(2|E| + 1) \cdot \left(\sum_{v \in V} \mathbf{x}[v] + \sum_{e \in E} \mathbf{x}[e]\right)} && \text{(as } |V| \leq |E| + 1\text{)} \\
&\leq \sqrt{4(K+1)(|E|+1)}
\end{aligned}$$

We get the last inequality due to the following. First by definition any point in $\mathbf{x} \in \operatorname{co}(\mathcal{X})$ is a convex combination of points in $\mathcal{X}$. Therefore an upper bound on $\max_{\mathbf{x}' \in \mathcal{X}} \sum_{v \in V} \mathbf{x}'[v] + \sum_{e \in E} \mathbf{x}'[e]$ is also an upper bound on $\sum_{v \in V} \mathbf{x}[v] + \sum_{e \in E} \mathbf{x}[e]$. As any path in the DAG G has exactly K edges, we have $\sum_{v \in V} \mathbf{x}'[v] = K + 1$ and $\sum_{e \in E} \mathbf{x}'[e] = K$ for any $\mathbf{x}' \in \mathcal{X}$.

Now we upper bound the diameter as follows:

$$\mathrm{diam}_F(\mathrm{co}(\mathcal{X})) \leq \max_{\mathbf{x}\in\mathrm{co}(\mathcal{X})} \sum_{v\in V} \sqrt{\mathbf{x}[v]} + \sum_{e\in E} \sqrt{\mathbf{x}[e]} \leq \sqrt{4(K+1)(|E|+1)}$$

Next, we upper bound the second term of the regret upper bound above. Observe that $\nabla^2 F(\mathbf{z}) = \mathrm{diag}(1/(4\mathbf{z}^{3/2}))$. Due to the definition of $\widehat{\mathbf{y}}_t^+$, it follows that the term $||\widehat{\mathbf{y}}_t^+||^2_{\nabla^2 F(\mathbf{z}_t)^{-1}}$ is maximized when $\mathbf{z}_t = \widetilde{\mathbf{x}}_t$. Hence, we have $\mathbb{E}_t\left[||\widehat{\mathbf{y}}_t^+||^2_{\nabla^2 F(z_t)^{-1}}\right] \leq 16\sum_{v\in V}\sqrt{\widetilde{\mathbf{x}}_t[v]} + 16\sum_{e\in E}\sqrt{\widetilde{\mathbf{x}}_t[e]} \leq 32\sqrt{(K+1)(|E|+1)}$. Hence, we have the following by setting $\eta = \frac{1}{\sqrt{T}}$:

$$\begin{aligned}
\mathrm{Regret}(T) &\leq \frac{\mathrm{diam}_F(\mathrm{co}(\mathcal{X}))}{\eta} + \sum_{t=1}^{T} \mathbb{E}_t\left[\frac{\eta}{2}||\widehat{\mathbf{y}}_t^+||^2_{(\nabla^2 F(\mathbf{z}_t))^{-1}}\right] + c\cdot\sqrt{K|E|T\log(|E|/\delta)} \\
&\leq \frac{2\sqrt{(K+1)|E|}}{\eta} + \frac{32T\eta}{2}\sqrt{(K+1)|E|} + c\cdot\sqrt{K|E|T\log(|E|/\delta)} \\
&\leq 18\sqrt{(K+1)(|E|+1)T} + c\cdot\sqrt{K|E|T\log(|E|/\delta)}
\end{aligned}$$

where we get the last inequality by setting $\eta = \frac{1}{\sqrt{T}}$.

Theorem 22 *Under the assumption that every path from the source to the sink has length K, Algorithm 1 incurs a regret of at most $\mathcal{O}(\sqrt{K|E|T\log(|E|/\delta)})$ against any adaptive adversary with probability at least $1-\delta$.*

D.2. Omitted Details from Section 3.2

Algorithm 2: FTRL for online shortest path in a DAG $G = (V, E)$

Let K be the longest path in the DAG G from source to sink.
for $t = 1$ **to** T **do**
  Compute $\widetilde{\mathbf{x}}_t \leftarrow \arg\min_{\mathbf{x}\in\mathrm{co}(\mathcal{X}^\dagger)} \left(\eta\sum_{\tau=1}^{t-1}\langle\mathbf{x}, \widehat{\mathbf{y}}_\tau^\dagger\rangle + F(\mathbf{x})\right)$.
  Initialize path $P_t \leftarrow (v_\mathsf{s})$ and reset $v_0 \leftarrow v_\mathsf{s}$.
  for $i = 1$ **to** K **do**
    Sample outgoing edge $e_i = (v_{i-1}, v_i) \in \delta^+(v_{i-1})$ with probability proportional to $\widetilde{\mathbf{x}}_t[e_i]$.
    Update $P_t \leftarrow P_t \circ (e_i, v_i)$
  end
  Choose the path P_t and observe loss its ℓ_t.
  Determine $\mathbf{x}_t^\dagger \in \mathcal{X}^\dagger$ corresponding to the path P_t and construct the loss estimator $\widehat{\mathbf{y}}_t^\dagger$.
end

Recall that $K(v)$ is the length of the longest path from v_s to v and $K(v_\mathsf{s}) = 0$. For any edge $(u, v) \in E$, define $\mathcal{I}((u, v)) := \{i \in [\![K-1]\!] : K(u) < i < K(v)\}$. Now we make the following assumption.

Assumption 2 *For any* $i \in [\![K-1]\!]$, $|e \in E : i \in \mathcal{I}(e)| \geq 1$.

Note that if there exists an index $i \in [\![K-1]\!]$ such that $|e \in E : i \in \mathcal{I}(e)| = 0$, then we have $b(\mathbf{x})[i] = 0$ for all $\mathbf{x} \in \mathcal{X}$. Consequently, this bit can be excluded from our representation.

Next, we prove the following proposition.

Proposition 23 $\mathbb{E}_t[\mathbf{x}_t^\dagger] = \widetilde{\mathbf{x}}_t$

Proof. Due to Proposition 20, we have $\mathbb{E}_t[\mathbf{x}_t^\dagger[v]] = \widetilde{\mathbf{x}}_t[v]$ for all $v \in V$ and $\mathbb{E}_t[\mathbf{x}_t^\dagger[e]] = \widetilde{\mathbf{x}}_t[e]$ for all $e \in E$. Fix $i \in [\![K-1]\!]$. Later in this section, we prove that for any $\mathbf{x} \in \mathrm{co}(\mathcal{X}^\dagger)$, we have $\mathbf{x}[i] = \sum_{e \in E : i \in \mathcal{I}(e)} \mathbf{x}[e]$. Hence, we have $\widetilde{\mathbf{x}}_t[i] = \sum_{e \in E : i \in \mathcal{I}(e)} \widetilde{\mathbf{x}}_t[e] = \mathbb{E}_t[\sum_{e \in E : i \in \mathcal{I}(e)} \mathbf{x}_t^\dagger[e]] = \mathbb{E}_t[\mathbf{x}_t^\dagger[i]]$. ■

Recall the definition of the loss estimators $\widehat{\mathbf{y}}_t^\dagger$ and $\widetilde{\mathbf{y}}_t^\dagger$ from Section 3.2. Next, recall that we have $\mathbb{E}_t[\langle \mathbf{x}_{(1)}^\dagger - \mathbf{x}_{(2)}^\dagger, \widetilde{\mathbf{y}}_t^\dagger \rangle] = \langle \mathbf{x}_{(1)} - \mathbf{x}_{(2)}, \mathbf{y}_t \rangle$ for all $\mathbf{x}_{(1)}, \mathbf{x}_{(2)} \in \mathcal{X}$. If we use $F(\mathbf{x}) = -\sum_{v \in V} \sqrt{\mathbf{x}[v]} - \sum_{e \in E} \sqrt{\mathbf{x}[e]} - \sum_{i \in [\![K-1]\!]} \sqrt{\mathbf{x}[i]}$ as our regularizer, it easily follows that our FTRL equation has a unique minimizer $\widetilde{\mathbf{x}}_t$ and $\widetilde{\mathbf{x}}_t[v] > 0$ for all $v \in V$, $\widetilde{\mathbf{x}}_t[e] > 0$ for all $e \in E$ and $\widetilde{\mathbf{x}}_t[i] > 0$ for all $i \in [\![K-1]\!]$. Therefore, $\nabla^2 F(z)$ is a diagonal matrix for any point z on the chord $[\widetilde{x}_t, \widetilde{x}_{t+1}]$. Let $\widehat{\mathbf{y}}_t^+ \in \mathbb{R}_{\geq 0}^{V \cup E \cup [\![K-1]\!]}$ be a vector indexed by the elements in $V \cup E \cup [\![K-1]\!]$ such that $\widehat{\mathbf{y}}_t^+[v] = \widehat{\mathbf{y}}_t^\dagger[v] \cdot \mathbb{1}[\widetilde{\mathbf{x}}_{t+1}[v] \leq \widetilde{\mathbf{x}}_t[v]]$ for all $v \in V$, $\widehat{\mathbf{y}}_t^+[e] = \widehat{\mathbf{y}}_t[e] \cdot \mathbb{1}[\widetilde{\mathbf{x}}_{t+1}[e] \leq \widetilde{\mathbf{x}}_t[e]]$ for all $e \in E$, and $\widehat{\mathbf{y}}_t^+[i] = \widehat{\mathbf{y}}_t[i] \cdot \mathbb{1}[\widetilde{\mathbf{x}}_{t+1}[i] \leq \widetilde{\mathbf{x}}_t[i]]$ for all $i \in [\![K-1]\!]$. Let us set every entry of $\boldsymbol{\gamma}$ and $\widehat{\boldsymbol{\gamma}}$ to $\sqrt{\frac{K \log(5(|V|+|E|+K)/\delta)}{|E|T}}$. Now we apply Lemma 19 from Appendix C to get the following:

Theorem 24 *Algorithm 1 has the following guarantee on regret with probability at least* $1-\delta$ *against any adaptive adversary:*

$$\max_{\mathbf{x} \in \mathcal{X}} R_T(\mathbf{x}) \leq \frac{\mathrm{diam}_F(\mathrm{co}(\mathcal{X}^\dagger))}{\eta} + \sum_{t=1}^{T} \mathbb{E}_t \left[\frac{\eta}{2} ||\widehat{\mathbf{y}}_t^+||^2_{(\nabla^2 F(z_t))^{-1}} \right] + c \cdot \sqrt{K|E|T \log(|E|/\delta)}$$

where c *is some absolute constant,* $\mathrm{diam}_F(\mathrm{co}(\mathcal{X}^\dagger)) := \max_{\mathbf{x}, \mathbf{x}' \in \mathrm{co}(\mathcal{X}^\dagger)} F(\mathbf{x}) - F(\mathbf{x}')$*, and* z_t *is some point on the chord* $[\widetilde{\mathbf{x}}_t, \widetilde{\mathbf{x}}_{t+1}]$.

First we upper bound $\sum_{v \in V} \sqrt{\mathbf{x}[v]} + \sum_{e \in E} \sqrt{\mathbf{x}[e]} + \sum_{i=1}^{K-1} \sqrt{\mathbf{x}[i]}$ for any $\mathbf{x} \in \mathrm{co}(\mathcal{X}^\dagger)$ as follows:

$$\begin{aligned}
&\sum_{v \in V} \sqrt{\mathbf{x}[v]} + \sum_{e \in E} \sqrt{\mathbf{x}[e]} + \sum_{i=1}^{K-1} \sqrt{\mathbf{x}[i]} \\
&\leq \sqrt{(|V| + |E| + K - 1) \cdot \Big(\sum_{v \in V} \mathbf{x}[v] + \sum_{e \in E} \mathbf{x}[e] + \sum_{i=1}^{K-1} \mathbf{x}[i] \Big)} && \text{(Cauchy-Schwarz inequality)} \\
&\leq \sqrt{(3|E|) \cdot \Big(\sum_{v \in V} \mathbf{x}[v] + \sum_{e \in E} \mathbf{x}[e] + \sum_{i=1}^{K-1} \mathbf{x}[\ell + i] \Big)} && \text{(as } |V| \leq |E| + 1,\ K \leq |E|) \\
&\leq \sqrt{9K|E|}
\end{aligned}$$

We get the last inequality due to the following. First by definition any point in $\mathbf{x} \in \text{co}(\mathcal{X}^\dagger)$ is a convex combination of points in $\mathcal{X}$. Therefore, an upper bound on $\max_{x' \in \mathcal{X}^\dagger} \sum_{v \in V} \mathbf{x}'[v] + \sum_{e \in E} \mathbf{x}'[e] + \sum_{i=1}^{K-1} \mathbf{x}'[i]$ is also an upper bound on $\sum_{v \in V} \mathbf{x}[v] + \sum_{e \in E} \mathbf{x}[e] + \sum_{i=1}^{K-1} \mathbf{x}[\ell + i]$. As any path in the DAG G has at most K edges, we have $\sum_{v \in V} \mathbf{x}'[v] \leq K+1$ and $\sum_{e \in E} \mathbf{x}'[e] \leq K$. We also have $\sum_{i=1}^{K-1} \mathbf{x}'[i] \leq K-1$. Now we upper bound the diameter as follows:

$$\text{diam}_F(\text{co}(\mathcal{X}^\dagger)) \leq \max_{\mathbf{x} \in \text{co}(\mathcal{X}^\dagger)} \sum_{v \in V} \sqrt{\mathbf{x}[v]} + \sum_{e \in E} \sqrt{\mathbf{x}[e]} + \sum_{i=1}^{K-1} \sqrt{\mathbf{x}[i]} \leq \sqrt{9K|E|}$$

Next, we upper bound the second term of the regret upper bound above. Next observe that $\nabla^2 F(z) = \text{diag}(1/(4z^{3/2}))$. Due to the definition of $\widehat{\mathbf{y}}_t^+$, it follows that the term $||\widehat{\mathbf{y}}_t^+||^2_{\nabla^2 F(z_t)^{-1}}$ is maximized when $z_t = \widetilde{\mathbf{x}}_t$. Hence, we have $\mathbb{E}_t \left[||\mathbf{y}_t'||^2_{\nabla^2 F(z_t)^{-1}} \right] \leq 16 \sum_{v \in V} \sqrt{\widetilde{\mathbf{x}}_t[v]} + 16 \sum_{e \in E} \sqrt{\widetilde{\mathbf{x}}_t[e]} + 16 \sum_{i=1}^{K-1} \sqrt{\widetilde{\mathbf{x}}_t[i]} \leq 48\sqrt{K|E|}$.

Hence, we have the following by setting $\eta = \frac{1}{\sqrt{T}}$:

$$\begin{aligned}
\text{Regret}(T) &\leq \frac{\text{diam}_F(\text{co}(\mathcal{X}^\dagger))}{\eta} + \sum_{t=1}^{T} \mathbb{E}_t \left[\frac{\eta}{2} ||\widehat{\mathbf{y}}_t^+||^2_{(\nabla^2 F(z_t))^{-1}} \right] + c \cdot \sqrt{K|E|T \log(|E|/\delta)} \\
&\leq \frac{3\sqrt{K|E|}}{\eta} + \frac{48T\eta}{2} \sqrt{K|E|} + c \cdot \sqrt{K|E|T \log(|E|/\delta)} \\
&\leq 27\sqrt{K|E|T} + c \cdot \sqrt{K|E|T \log(|E|/\delta)}
\end{aligned}$$

Our analysis leads to the following main theorem.

Theorem 25 *Under the assumption that every path from the source to the sink has length at most K, Algorithm 2 incurs a regret of at most $\mathcal{O}(\sqrt{K|E|T \log(|E|/\delta)})$ against any adaptive adversary with probability at least $1 - \delta$.*

We now show that $\text{co}(\mathcal{X}^\dagger)$ can be represented using a polynomial number of linear constraints. Recall that $\mathcal{X}^\dagger$ represents the set of all paths, including the appended $K-1$ bits. We now claim that, given a point $\mathbf{x} \in [0,1]^{V \cup E \cup [\![K-1]\!]}$, we can efficiently determine whether $\mathbf{x}$ lies within the convex hull of $\mathcal{X}^\dagger$. The coordinate values corresponding to the edges and vertices must satisfy the flow constraints, which can be verified efficiently.

Recall that for any edge $(u,v) \in E$, we defined $\mathcal{I}((u,v)) := \{i \in [\![K-1]\!] : K(u) < i < K(v)\}$. For any two edges e_1, e_2 on the same path, observe that $\mathcal{I}(e_1) \cap \mathcal{I}(e_2) = \varnothing$. Consequently, the coordinate values of $\mathbf{x}$ corresponding to the indices in $[\![K-1]\!]$ must satisfy the following condition:

$$\mathbf{x}[i] = \sum_{e \in E : i \in \mathcal{I}(e)} \mathbf{x}[e] \quad \forall i \in [\![K-1]\!].$$

If this condition fails for some $i \in [\![K-1]\!]$, then $\mathbf{x}$ does not lie in the convex hull of $\mathcal{X}^\dagger$. Conversely, if all flow constraints and the above condition hold, then $\mathbf{x}$ belongs to the convex hull. This verification is efficient, as $\mathcal{I}(e)$ can be computed efficiently.

Remark: One does not need to compute the exact minimizer $\widetilde{\mathbf{x}}_t$ of the FTRL equation in each round t. Our analysis remains valid if we instead compute an approximate minimizer $\widehat{\mathbf{x}}_t$ satisfying $\|\widehat{\mathbf{x}}_t - \widetilde{\mathbf{x}}_t\|_\infty \leq \frac{1}{T^2}$ and $\|\widehat{\mathbf{x}}_t\|_\infty > 0$. Such an approximate minimizer can be efficiently computed using standard optimization methods, such as the Ellipsoid method, in $\text{poly}(|E|, \log T)$ time steps, since our regularizer is both Legendre and strongly convex over $\text{co}(\mathcal{X}^\dagger)$, and $\text{co}(\mathcal{X}^\dagger)$ can be represented using a polynomial number of linear constraints.

D.3. Omitted Details from Section 3.3

Here, we present the full version of Section 3.3, including the omitted details.

Recall that $\delta^-(v)$ denotes the incoming edges and $\delta^+(v)$ denotes the outgoing edges of vertex v. Let $C : V \to \mathbb{N}$ denote the number of distinct paths from the source v_s to any vertex v. It holds that $C(v_\mathsf{s}) = 1$ and $C(v) := \sum_{(u,v)\in\delta^-(v)} C(u)$ for any $v \neq v_\mathsf{s}$. According to the definition, it satisfies that $C(v_\mathsf{t}) = |\mathcal{X}|$.

Let $h(v) := \arg\max_{(u,v)\in\delta^-(v)} C(u)$ be the incoming edge that brings the maximum number of paths to vertex v, with ties broken arbitrarily. Let

$$E^{\clubsuit} := \{h(v) \mid v \in V \setminus \{v_\mathsf{s}\}\}$$

be the set of all such edges. The underlying subgraph $S := (V, E^{\clubsuit})$ forms a directed spanning tree of G. The next lemma shows that the number of non-tree edges (edges not in $E^{\clubsuit}$) on any path from v_s to v_t in G is at most $\log|\mathcal{X}|$.

Lemma 26 *Let $P = (v_0 = v_\mathsf{s}, e_1, v_1, \ldots, e_k, v_k = v_\mathsf{t})$ be a path from the source to sink. We have that the number of non-tree edges on the path P is upper bounded by*

$$\sum_{i=1}^{k} \mathbb{1}[e_i \notin E^{\clubsuit}] \leq \log\left(|\mathcal{X}|\right).$$

Proof. Since the number of distinct paths is always non-negative, for any $i \in [\![k]\!]$, we have

$$C(v_i) = \sum_{(u,v_i)\in\delta^-(v_i)} C(u) \geq C(v_{i-1}),$$

where $v_{i-1} \in \delta^-(v_i)$. Moreover, for any non-tree edge $e_i = (v_{i-1}, v_i) \notin E^{\clubsuit}$, consider the tree edge $h(v_i) = (u_{i-1}, v_i)$, which is an incoming edge to vertex v_i. The number of distinct paths from v_s to v_i can then be lower bounded by:

$$C(v_i) = \sum_{(u,v_i)\in\delta^-(v_i)} C(u) \geq C(u_{i-1}) + C(v_{i-1}) \geq 2C(v_{i-1}),$$

where the last inequality follows from the selection criteria of the tree edge $h(v_i)$. More generally, we have that

$$\mathbb{1}[e_i \notin E^{\clubsuit}] \leq \log\left(\frac{C(v_i)}{C(v_{i-1})}\right).$$

Since $C(v_k) = C(v_{\mathsf{t}}) = |\mathcal{X}|$ and $C(v_0) = C(v_{\mathsf{s}}) = 1$, summing these inequalities yields

$$\sum_{i=1}^{k} \mathbb{1}[e_i \notin E^{\clubsuit}] \leq \log\left(\frac{C(v_{\mathsf{t}})}{C(v_0)}\right) = \log\left(|\mathcal{X}|\right).$$

■

We now introduce the *centroid-based decomposition*: Given a directed tree $S = (V, E^{\clubsuit})$, we identify a vertex $c \in V$ such that the connected components $\widehat{S}_1, \ldots, \widehat{S}_k$ resulting from its removal satisfy $|\widehat{V}_i| \leq |V|/2$ for all $i \in [\![k]\!]$, where $\widehat{V}_i$ is the set of vertices in the subtree $\widehat{S}_i$. Such a vertex c, known as the *centroid*, always exists in any tree (see (Jordan, 1869; Della Giustina et al., 2019)).

We associate the centroid c with the tree S by defining $S_c := S$. The above procedure is then applied recursively to each component $\widehat{S}_i$ for $i \in [\![k]\!]$. If a component reduces to a single vertex c, we designate c as its centroid and terminate the recursion.

Since the sets $\widehat{V}_i$ resulting from the removal of c form a partition of $V \setminus \{c\}$, each vertex $v \in V$ will eventually be assigned as the centroid of some subtree $S_v = (V_v, E_v^{\clubsuit})$. Consequently, this procedure generates a collection of subtrees $\mathcal{T} := \{S_v : v \in V\}$, where each vertex v is uniquely associated with a subtree of S in which it serves as the centroid. Furthermore, we define $\mathcal{T}(S_v) := \{S_w : w \in V_v\}$ as the centroid-based decomposition of the subtree S_v.

The above construction transforms S into a hierarchy of subtrees. The following folklore lemma establishes that the centroid-based decomposition systematically organizes every path in S.

Lemma 27 *Let $\mathcal{T}$ be the centroid-based decomposition of the directed tree S. For any pair of vertices $(u, v) \in V \times V$, there exists a unique subtree $S_w \in \mathcal{T}$ with centroid w such that:*

- *Both u and v belong to the subtree S_w, i.e., $u, v \in V_w$, where V_w is the vertex set of S_w.*
- *The path from u to v in the underlying undirected graph of S passes through w.*

Proof. We will prove the statement by induction, showing that it holds for any tree $S = (V, E^{\clubsuit})$ with at most k vertices, along with its corresponding centroid-based decomposition $\mathcal{T}$.

Base Case: When the tree $S = (\{c\}, \varnothing)$ contains only a single vertex $v = c$, the statement holds trivially by S_c. The only valid pair is (v, v), and the path from v to itself contains c by definition.

Inductive Step: Assume the statement holds for all trees with at most k vertices. Consider a tree $S = (V, E^{\clubsuit})$ with $|V| = k + 1$ vertices and a pair of vertices $(u, v) \in V \times V$. Let c be the centroid of S, and consider the path from u to v in the underlying undirected graph of S. We distinguish two cases:

1. **If the path from u to v contains c:** Then S_c is the desired subtree. Since the undirected path from u to v passes through c, u and v must either lie in different connected components formed after removing c from S, or one of them is c itself. In both cases, no other subtree $S_{w'} \in \mathcal{T}$ can contain both u and v. Hence, S_c is the only subtree satisfies the condition.

2. **If the path from u to v does not contain c:** In this case, both u and v lie entirely within one of the connected components $\widehat{S}_i$ formed by removing c from S. By the induction hypothesis, there exists one subtree $S_w \in \mathcal{T}(\widehat{S}_i)$ with the statement holds. For any other subtree $S_{w'} \in \mathcal{T} \setminus \mathcal{T}(\widehat{S}_i)$, we have neither u nor v contained in $S_{w'}$. Thus, S_w is the only subtree satisfies the condition.

Conclusion: By mathematical induction, the statement holds for any tree $S = (V, E^{\clubsuit})$. Thus, for any pair of vertices $(u, v) \in V \times V$, there exists a unique vertex $c \in V$ such that $u, v \in V_c$ and the path from u to v in the underlying undirected graph of S passes through c. ■

The following folklore lemma demonstrates that the total number of vertices introduced by the centroid-based decomposition is nearly linear in the number of vertices of the original tree:

Lemma 28 *Let $\mathcal{T}$ be the centroid-based decomposition of the directed tree S. The total number of vertices among $S_v = (V_v, E_v^{\clubsuit}) \in \mathcal{T}$ in centroid-based decomposition is upper-bounded by*

$$\sum_{S_v \in \mathcal{T}} |V_v| \leq (1 + \log |V|)|V|.$$

Proof. We will prove the statement by induction, showing that it holds for any tree $S = (V, E^{\clubsuit})$ with at most k vertices, along with its corresponding centroid-based decomposition $\mathcal{T}$.

Base Case: When the tree $S = (\{c\}, \varnothing)$ contains only a single vertex $v = c$, we have:

$$\sum_{S_v \in \mathcal{T}} |V_v| = |V| = 1 \leq (1 + \log |V|)|V|,$$

so the inequality holds.

Inductive Step: Assume the statement holds for all trees with at most k vertices. Consider a tree $S = (V, E^{\clubsuit})$ with $|V| = k + 1$ vertices. Denote by $\widehat{S}_1, \ldots, \widehat{S}_k$ the subtrees after the removal of centroid c from S. Let $\widehat{V}_i$ be the set of vertices of $\widehat{S}_i$. By the induction hypothesis, we have

$$\sum_{v \in \widehat{V}_i} |V_v| \leq (1 + \log |\widehat{V}_i|)|\widehat{V}_i| \leq |\widehat{V}_i| \log |V|,$$

where the second inequality follows from $|\widehat{V}_i| \leq |V_c|/2$ which is the property of the centroid.

Therefore, the summation $\sum_{v \in V} |V_v|$ can be upper-bounded via:

$$\sum_{S_v \in \mathcal{T}} |V_v| = |V_c| + \sum_{i=1}^{k} \sum_{v \in \widehat{V}_i} |V_v| \leq |V| + \sum_{i=1}^{k} |\widehat{V}_i| \log |V| \leq (1 + \log |V|)|V|.$$

where the last equality holds as $\sum_{i=1}^{k} |\widehat{V}_i| = |V| - 1$.

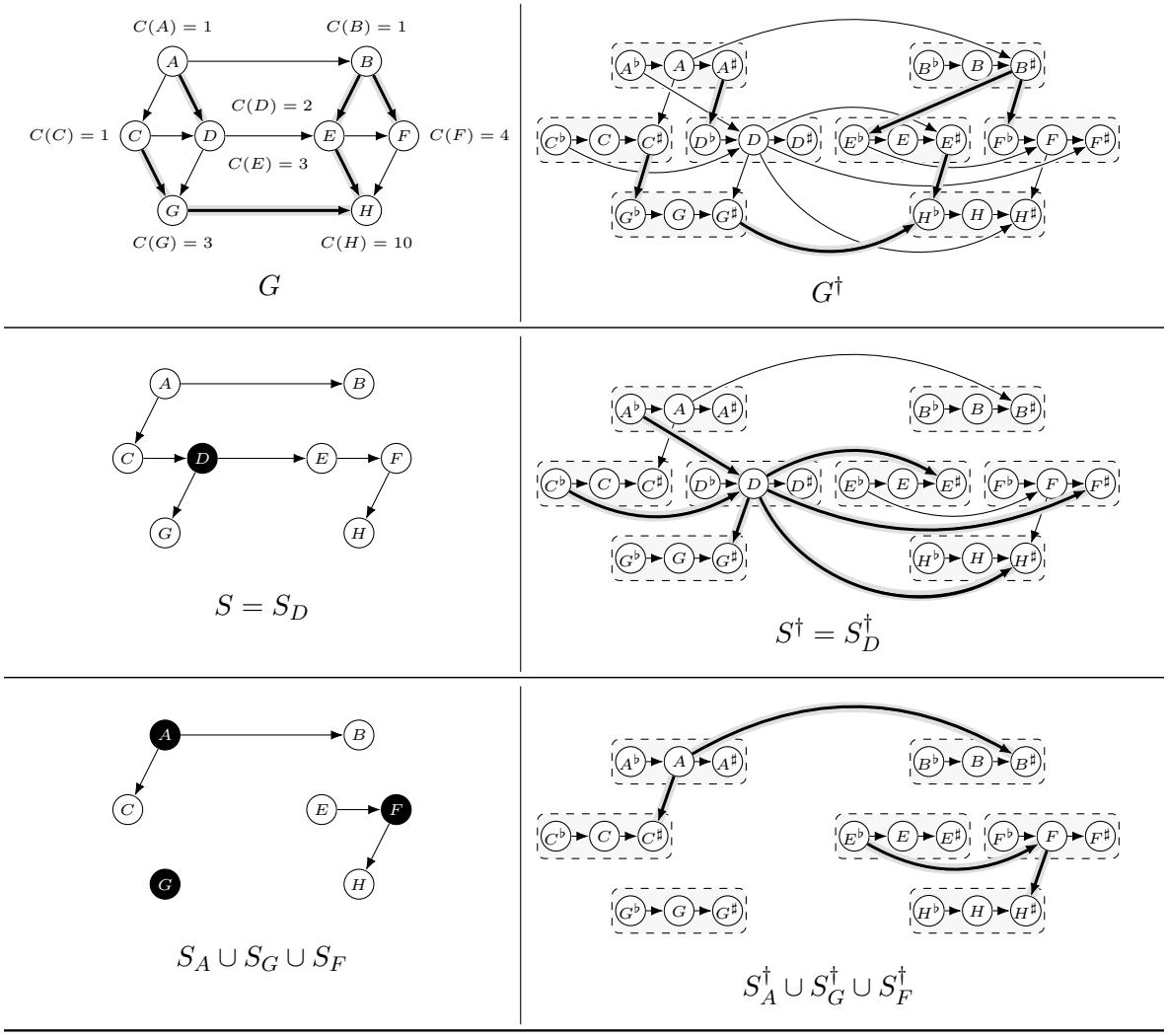

Figure 2: An example graph conversion from G to $G^\dagger$ is shown. The non-tree edges $E \setminus E^{\clubsuit}$ are shaded in G, and they correspond to the shaded edges in $G^\dagger$. The graph $S = (V, E^{\clubsuit})$ has a centroid vertex D. Removing D from S results in three subtrees: S_A, S_G, and S_F. The new linked edges for the corresponding centroids are shaded in the graphs on the right. Recall $C(\cdot)$ is the number of distinct path from source to the vertex.

Conclusion: By mathematical induction on the size of the tree, we have that

$$\sum_{S_v \in \mathcal{T}} |V_v| \leq (1 + \log |V|)|V|.$$

■

Starting from the tree $S = (V, E^{\clubsuit})$, we start by transforming a selected tree $S_c = (V_c, E_c^{\clubsuit})$ with centroid vertex c into an equivalent graph $S_c^\dagger = (V_c^\dagger, E_c^\dagger)$ in a recursive way, using the centroid-

based decomposition. Let $\widehat{S}_1, \ldots, \widehat{S}_k$ be the subtrees obtained after removing the centroid c from the tree S_c. Now, we do the following in a recursive way:

1. Initialize $S_c^\dagger \leftarrow \bigcup_{i=1}^k \widehat{S}_i^\dagger$, where $\widehat{S}_i^\dagger$ is the transformed graph of $\widehat{S}_i$.
2. Update $V_c^\dagger \leftarrow V_c^\dagger \cup \{c^\flat, c, c^\sharp\}$.
3. For each vertex $v \in V_c$:
   (a) If there is a directed path from v to c in S_c, or if v is c, update $E_c^\dagger \leftarrow E_c^\dagger \cup \{(v^\flat, c)\}$.
   (b) If there is a directed path from c to v in S_c, or if v is c, update $E_c^\dagger \leftarrow E_c^\dagger \cup \{(c, v^\sharp)\}$.

Finally, the graph $G^\dagger = (V^\dagger, E^\dagger)$ is generated as follows:

1. Initialize $G^\dagger \leftarrow S^\dagger$.
2. For each non-tree edge $(u, v) \in E \setminus E^\clubsuit$, update $E^\dagger \leftarrow E^\dagger \cup \{(u^\sharp, v^\flat)\}$.

We refer the reader to Figure 2 for one example of such a conversion. We now demonstrate that the converted graph $G^\dagger$ is essentially equivalent to G. We define a mapping $\sigma : E^\dagger \to 2^E$ as follows:

- For $e^\dagger = (v^\flat, c)$, $\sigma(e^\dagger)$ consists of all edges on the unique path from v to c in the tree S.
- For $e^\dagger = (c, v^\sharp)$, $\sigma(e^\dagger)$ consists of all edges on the unique path from c to v in the tree S.
- For $e^\dagger = (u^\sharp, v^\flat)$, $\sigma(e^\dagger) = \{(u, v)\} \subseteq E \setminus E^\clubsuit$ contains the corresponding edge.

The above mapping assigns each edge $e^\dagger = (u^\dagger, v^\dagger) \in E^\dagger$ a path from u to v (which may be empty), as specified by $\sigma(e^\dagger)$, where $w^\dagger \in \{w^\flat, w, w^\sharp\}$ for $w \in \{u, v\}$. Denote by $\mathcal{P}^\dagger$ the set of paths from $v_\mathsf{s}^\flat$ to $v_\mathsf{t}^\sharp$ in $G^\dagger$. The following lemma establishes an important property of $\sigma(e^\dagger)$.

Lemma 3 (restatement) *For any path $P^\dagger \in \mathcal{P}^\dagger$, $\sigma(e_1^\dagger) \cap \sigma(e_2^\dagger) = \varnothing$ for any distinct edges $e_1^\dagger, e_2^\dagger \in P^\dagger$.*

Proof. Consider a path $P^\dagger$ in $G^\dagger$. Suppose, for the sake of contradiction, that there exist two distinct edges $e_1^\dagger$ and $e_2^\dagger$ in $P^\dagger$ such that $\sigma(e_1^\dagger) \cap \sigma(e_2^\dagger) \neq \varnothing$. Let (u, v) be an edge in the intersection. This implies that there is a sub-path in G that starts at u, passes through v, and ends at u, contradicting our assumption that G is a DAG. ■

The next lemma shows that $G^\dagger$ is a DAG:

Lemma 29 *The graph $G^\dagger$ is a Directed Acyclic Graph with source node $v_\mathsf{s}^\flat$ and sink node $v_\mathsf{t}^\sharp$.*

Proof. First, we show that $G^\dagger$ is a directed acyclic graph. Note that by construction, there is no directed cycle involving only the three vertices $c^\flat, c, c^\sharp$ for any $c \in V$. Suppose, for the sake of contradiction, that there exists a path $P^\dagger = (v_0^\dagger, e_1^\dagger, v_1^\dagger, \ldots, v_{\ell-1}^\dagger, e_\ell^\dagger, v_\ell^\dagger)$ in $G^\dagger$ such that $v_0^\dagger = v_\ell^\dagger$ and $w^\dagger \in \{w^\flat, w, w^\sharp\}$ for all $w \in V$. Let i be the smallest index such that $v_0 \neq v_i$. This implies that there is a path in G from v_0 to v_i using the edges $\bigcup_{j \in [\![i]\!]} \sigma(e_j^\dagger)$, and a path from v_i back to v_0 using the edges $\bigcup_{j \in [\![i+1, \ell]\!]} \sigma(e_j^\dagger)$, which contradicts our assumption that G is a DAG.

By construction, there is no incoming edges to $v_{\mathsf{s}}^{\flat}$ in $G^{\dagger}$ as there are no incoming edges to v_{s} in $G^{\dagger}$. Similarly, there is no outgoing edges from $v_{\mathsf{t}}^{\flat}$ in $G^{\dagger}$ as there are no outgoing edges from v_{t} in $G^{\dagger}$. ■

The next lemma demonstrates that this mapping establishes a bijection between the paths from v_{s} to v_{t} in G and the paths $P^{\dagger}$ from $v_{\mathsf{s}}^{\flat}$ to $v_{\mathsf{t}}^{\sharp}$ in $G^{\dagger}$. Note that we slightly abuse the notation σ.

Lemma 4 (restatement) *There exists an efficiently computable bijection $\sigma : \mathcal{P}^{\dagger} \to \mathcal{P}$ such that an edge $e \in E$ belongs to $\sigma(P^{\dagger})$ if and only if there exists an edge $e^{\dagger} \in P^{\dagger}$ with $e \in \sigma(e^{\dagger})$.*

Proof. First, observe that for a path $P^{\dagger}$ in $G^{\dagger}$ from $v_{\mathsf{s}}^{\flat}$ to $v_{\mathsf{t}}^{\sharp}$, the set of edges $\bigcup_{e^{\dagger} \in P^{\dagger}} \sigma(e^{\dagger})$ forms a path P from v_{s} to v_{t}, where the union is over all the edges in $P^{\dagger}$. This is because any edge $(u^{\dagger}, v^{\dagger}) \in E^{\dagger}$ corresponds to a path from u to v in G. As $\sigma(e^{\dagger})$ can be computed efficiently for any edge $e^{\dagger}$, $\sigma(P^{\dagger})$ can be computed efficiently for any path $P^{\dagger}$. Let g denote this mapping from $P^{\dagger}$ to P. We now show that the mapping g is a bijection.

Consider a path $P = (v_0, e_1, v_1, \ldots, e_k, v_k)$ in G, where $v_0 = v_{\mathsf{s}}$ and $v_k = v_{\mathsf{t}}$. Let us assume that there is at least one non-tree edges. An analogous proof exists if there are no non-tree edges. Let $e_{i_1}, e_{i_2}, \ldots, e_{i_t}$ be the sequence of non-tree edges in the path, that is, $e_{i_j} \in E \setminus E^{\clubsuit}$ for each $j \in [\![t]\!]$. These edges partition the path P into several segments:

$$(v_0, e_1, \ldots, v_{i_1-1}), e_{i_1}, (v_{i_1}, e_{i_1+1}, \ldots, v_{i_2-1}), \ldots, e_{i_t}, (v_{i_t}, e_{i_t+1}, \ldots, v_k),$$

where each segment $(v_{i_j}, e_{i_j+1}, \ldots, v_{i_{j+1}-1})$ is a path in G that consists only of edges from the directed tree $S = (V, E^{\clubsuit})$.

Let $i_0 = 0$ and $i_{t+1} = k + 1$. By Theorem 27, for any $j \in [\![0, t]\!]$, there exists a unique subtree $S_{c_j} \in \mathcal{T}$ with centroid c_j such that S_{c_j} contains the path from v_{i_j} to $v_{i_{j+1}-1}$, and this path contains c_j. By the construction of the graph $G^{\dagger}$, the edges

$$e_j^{\flat} := (v_{i_j}^{\flat}, c_j) \in E^{\dagger} \quad \text{and} \quad e_j^{\sharp} := (c_j, v_{i_{j+1}-1}^{\sharp}) \in E^{\dagger}$$

are present in $G^{\dagger}$. Furthermore, for any $j \in [\![t]\!]$, the graph $G^{\dagger}$ also contains the edge $e_{i_j}^{\dagger} := (v_{i_j-1}^{\sharp}, v_{i_j}^{\flat}) \in E^{\dagger}$, since $e_{i_j} = (v_{i_j-1}, v_{i_j}) \in E \setminus E^{\clubsuit}$ is a non-tree edge. As a result, $G^{\dagger}$ contains the following path from $v_{i_0}^{\flat} = v_{\mathsf{s}}^{\flat}$ to $v_{i_{t+1}-1}^{\sharp} = v_{\mathsf{t}}^{\sharp}$:

$$P^{\dagger} := (v_{i_0}^{\flat}, e_{i_0}^{\flat}, c_{i_0}, e_{i_0}^{\sharp}, v_{i_1-1}^{\sharp}, e_{i_1}^{\dagger}, v_{i_1}^{\flat}, \ldots, e_{i_t}^{\sharp}, v_{i_{t+1}-1}^{\sharp}).$$

■

The previous lemma shows that the decision problem for the shortest path in G can be converted to the shortest path problem in $G^{\dagger}$, and vice versa. Let $w : E \to \mathbb{R}$ be a weight function in the graph $G = (V, E)$. Define $w^{\dagger} : E^{\dagger} \to \mathbb{R}$ as the weight function for the converted graph $G^{\dagger} = (V^{\dagger}, E^{\dagger})$:

$$w^{\dagger}(e^{\dagger}) := \mathbb{1}[|\sigma(e^{\dagger})| \geq 1] \cdot \sum_{e \in \sigma(e^{\dagger})} w(e). \tag{8}$$

Using this mapping, we can convert a decision problem on G to a decision problem on $G^{\dagger}$:

Lemma 5 (restatement) *The online shortest path problem on $G = (V, E)$ can be efficiently reduced to the online shortest path problem on $G^\dagger = (V^\dagger, E^\dagger)$.*

The proof of the above lemma can be found in the main body. Finally, we need to show that the graph $G^\dagger$ satisfies the required size constraints:

Lemma 6 (restatement) *The graph $G^\dagger = (V^\dagger, E^\dagger)$ contains $|V^\dagger| \leq \mathcal{O}(|V|)$ vertices and $|E^\dagger| \leq \mathcal{O}(|V| \log |V| + |E|)$ edges. Moreover, The number of edges on the longest path from $v_{\mathsf{s}}^\flat$ to $v_{\mathsf{t}}^\sharp$ is upper bounded by $\mathcal{O}(\log |\mathcal{X}|)$.*

Proof. Each vertex $v \in V$ corresponds to three vertices $v^\flat$, v, and $v^\sharp$ in the graph $G^\dagger$. Thus, $G^\dagger$ contains a total of $3|V|$ vertices. Moreover, for each node $c \in V$, we add at most $|V_c| + 1$ edges of the form $(v^\flat, c)$ or $(c, v^\sharp)$. For each non-tree edge $e \in E \setminus E^{\clubsuit}$, we add one additional edge to $E^\dagger$. Therefore, the total number of edges can be bounded by:

$$|E^\dagger| \leq \sum_{c \in V} (|V_c| + 1) + |E \setminus E^\ddagger| \leq |V| \log |V| + 2|V| + |E|,$$

where the last inequality follows from Lemma 28.

Recall from the proof of Lemma 4 that any path $P^\dagger$ can be represented as

$$P^\dagger := (v_0^\flat, e_0^\flat, c_0, e_0^\sharp, v_{i_1-1}^\sharp, e_{i_1}^\dagger, v_{i_1}^\flat, \ldots, e_t^\sharp, v_{i_t}^\sharp),$$

where $e_{i_j}^\dagger$ corresponds to some non-tree edge $e_{i_j} \in E \setminus E^{\clubsuit}$. According to Lemma 26, we have $t \leq \log |\mathcal{X}|$. Since there are exactly $3t + 2$ edges in $P^\dagger$, the longest path in $G^\dagger$ is upper bounded by $\mathcal{O}(\log |\mathcal{X}|)$. ■

By combining Lemma 5, Lemma 6, and applying the regret guarantee of our FTRL algorithm from the previous section, we establish the main theorem:

Theorem 7 (restatement) *There exists an computationally efficient algorithm that incurs a regret bound of at most $\widetilde{\mathcal{O}}(\sqrt{|E|T \log(|\mathcal{X}|/\delta)})$ with probability at least $1 - \delta$ against any adaptive adversary, where $\widetilde{\mathcal{O}}(\cdot)$ only hides logarithmic factors in $|E|$.*

Appendix E. Applications

In this section, we demonstrate the application of our FTRL approach to various well-known combinatorial sets $\mathcal{X} \subseteq \{0, 1\}^d$. The core idea is to efficiently reduce problems involving these combinatorial sets to a problem on a directed acyclic graph (DAG) and establish the corresponding regret bound. Our method can be seen as a computationally efficient FTRL approach for these sets.

In certain cases, we either match or improve upon the $\mathcal{O}(\sqrt{dT \log |\mathcal{X}|})$ regret bound achieved by EXP3 with Kiefer-Wolfowitz exploration. In other cases, we demonstrate improvements over the best-known high-probability regret guarantees achieved by an efficient algorithm, specifically that of (Zimmert and Lattimore, 2022). We note that the high-probability regret guarantee was formally proven only for continuous sets by Zimmert and Lattimore (2022). However, we believe their analysis extends to discrete decision sets, such as the combinatorial sets considered, using the same techniques as Abernethy et al. (2008).

E.1. Hypercube

Let $\mathcal{X} = \{0,1\}^d$ denote the combinatorial hypercube. We construct a DAG G as follows. The vertex set and edge set are

$$V := \{v_0\} \cup \{v_i^\dagger, v_i\}_{i=1}^d, \quad E := \{(v_{i-1}, v_i), (v_{i-1}, v_i^\dagger), (v_i^\dagger, v_i)\}_{i=1}^d$$

respectively. In the graph G, $v_{\mathtt{s}} = v_0$ is the source vertex, and $v_{\mathtt{t}} = v_d$ is the sink vertex.

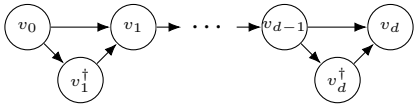

Figure 3: Conversion of hypercube to DAG

Next, for any loss function $\mathbf{y}_t : [\![d]\!] \to \mathbb{R}$, we define a weight function $w_t : E \to \mathbb{R}$ as follows:

$$w_t(e) = \begin{cases} \mathbf{y}_t[i] & \text{if } e = (v_{i-1}, v_i^\dagger) \text{ for some } i \in [\![d]\!] \\ 0 & \text{otherwise} \end{cases}.$$

We now apply our FTRL algorithm from Section 3.1 to the DAG G. At each round t, if the FTRL algorithm selects a path P_t in G, we choose $\mathbf{x}_t \in \mathcal{X}$ such that for any $i \in [\![d]\!]$, $\mathbf{x}_t[i] = 1$ if the edge $(v_{i-1}, v_i^\dagger)$ is part of the path P_t, and $\mathbf{x}_t[i] = 0$ otherwise. By the construction of w_t, it follows that $\langle \mathbf{x}_t, \mathbf{y}_t \rangle = w_t(P_t)$. Consequently, we provide $\langle \mathbf{x}_t, \mathbf{y}_t \rangle$ as the bandit feedback for the path P_t to the FTRL algorithm. The way we choose $\mathbf{x}_t$ induces a bijective mapping between the set of vectors in $\mathcal{X}$ and the set of paths in G, ensuring correctness. Moreover, our algorithm is computationally efficient.

Finally, observe that each path in G has a length of d. Thus, we incur a high-probability regret of $\widetilde{\mathcal{O}}(d\sqrt{T})$ against an adaptive adversary, which is near-optimal for the hypercube. This also improves upon the best-known high-probability regret bound of $\widetilde{\mathcal{O}}(d^2\sqrt{T})$ achieved by (Zimmert and Lattimore, 2022).

E.2. Multi-Task Multi-Armed Bandits

In the Multi-task Multi-Armed Bandit problem, we are given a set of m MAB problems, where in the i-th MAB problem there are d_i arms. In each round, we choose one arm from each MAB problem simultaneously and receive the sum of the losses of the arms chosen as the loss feedback. The goal is to do regret minimization w.r.t best arm in each MAB problem in hindsight.

The multi-task MAB problem is formally formulated as follows. Let $d = \sum_{i=1}^m d_i$. Let $d_{1:i} = \sum_{j=1}^i d_j$ and let $d_{1:0} = 0$. The set $\mathcal{X}$ of arms is defined as follows:

$$\mathcal{X} = \left\{ \mathbf{x} \in \{0,1\}^d : \forall j \in [\![m]\!] \sum_{i=d_{1:j-1}+1}^{d_{1:j}} \mathbf{x}[i] = 1 \right\}$$

For each round t, the loss function $\mathbf{y}_t : [\![d]\!] \to \mathbb{R}$ is chosen by an adversary. In each round the agent draw $\mathbf{x}_t \in \mathcal{X}$ and observe loss $\langle \mathbf{x}_t, \mathbf{y}_t \rangle \in [-1, 1]$. The goal is to minimize the following regret:

$$\text{Regret}(T) := \sum_{t=1}^T \langle \mathbf{x}_t, \mathbf{y}_t \rangle - \min_{\mathbf{x} \in \mathcal{X}} \sum_{t=1}^T \langle \mathbf{x}, \mathbf{y}_t \rangle$$

We now reduce the problem to online shortest path on DAG. We first construct a DAG G as follow: The vertex set and edge set are

$$V := \{v_i\}_{i=0}^{m} \cup \{v_i^j \mid i \in [\![m]\!], j \in [\![d_i]\!]\}, \qquad E := \{(v_{i-1}, v_i^j), (v_i^j, v_i) \mid i \in [\![m]\!], j \in [\![d_i]\!]\}$$

respectively. In graph G, $v_{\mathsf{s}} = v_0$ is the source vertex, and $v_{\mathsf{t}} = v_m$ is the sink vertex.

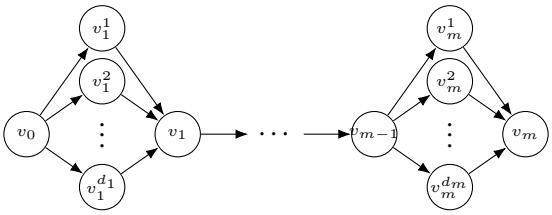

Figure 4: Conversion of Multi-task MAB to DAG

Next, for any loss vector $\mathbf{y}_t : [\![d]\!] \to \mathbb{R}$, we define a weight function $w_t : E \to \mathbb{R}$ as follows:

$$w_t(e) = \begin{cases} \mathbf{y}_t[d_{1:i-1} + j] & \text{if } e = (v_{i-1}, v_i^j) \text{ for some } i \in [\![m]\!], j \in [\![d_i]\!] \\ 0 & \text{otherwise} \end{cases}.$$

We now apply our FTRL algorithm from Section 3.1 to the DAG G. At each round t, if the FTRL algorithm selects a path P_t in G, we choose $\mathbf{x}_t \in \mathcal{X}$ such that for any $i \in [\![m]\!]$ and any $j \in [\![d_i]\!]$, $\mathbf{x}_t[d_{1:i-1} + j] = 1$ if the edge (v_{i-1}, v_i^j) is part of the path P_t, and $\mathbf{x}_t[d_{1:i-1} + j] = 0$ otherwise. By the construction of w_t, it follows that $\langle \mathbf{x}_t, \mathbf{y}_t \rangle = w_t(P_t)$. Consequently, we provide $\langle \mathbf{x}_t, \mathbf{y}_t \rangle$ as the bandit feedback for the path P_t to the FTRL algorithm. The way we choose $\mathbf{x}_t$ induces a bijective mapping between the set of vectors in $\mathcal{X}$ and the set of paths in G, ensuring correctness. Moreover, our algorithm is computationally efficient.

For our FTRL approach in Section 3.1, instead of equating all the coordinates of $\boldsymbol{\gamma}$ to the same fixed value, we assign it differently. Then using our analysis in Appendix D.1, one can easily show that the following holds with probability at least $1 - \delta$:

$$R_T(\mathbf{x}) \le c_1 \cdot d_\star \sqrt{T} + c_2 \cdot \left(T \sum_{v \in V} \boldsymbol{\gamma}[v] + T \sum_{e \in E} \boldsymbol{\gamma}[e] + \sum_{v \in V} \mathbf{x}[v] \cdot \frac{\log(d/\delta)}{\boldsymbol{\gamma}[v]} + \sum_{e \in E} \mathbf{x}[e] \cdot \frac{\log(d/\delta)}{\boldsymbol{\gamma}[e]} \right)$$

where $d_\star := \max_{\mathbf{x} \in \mathrm{co}(\mathcal{X})} \sum_{e \in E} \sqrt{\mathbf{x}[e]} + \sum_{v \in V} \sqrt{\mathbf{x}[v_i]}$

We now assign values to each coordinate of $\boldsymbol{\gamma}$ as follows. For all $i \in [\![0, m]\!]$, we have:

$$\boldsymbol{\gamma}[v_i] = \sqrt{\frac{\log(d/\delta)}{T}}.$$

Next for all $i \in [\![m]\!]$ and all $j \in [\![d_i]\!]$, we have:

$$\boldsymbol{\gamma}[v_i^j] = \boldsymbol{\gamma}[(v_{i-1}, v_i^j)] = \boldsymbol{\gamma}[(v_i^j, v_i)] = \sqrt{\frac{\log(d/\delta)}{d_i T}}.$$

Having defined the vector $\boldsymbol{\gamma}$, we now upper bound the second term in the regret above. First, we have:

$$T\sum_{i=1}^{m}\boldsymbol{\gamma}[v_i]+\sum_{i=1}^{m}\frac{\log(d/\delta)}{\boldsymbol{\gamma}[v_i]}\leq 2m\sqrt{T\log(d/\delta)}\leq 2\sum_{i=1}^{m}\sqrt{d_iT\log(d/\delta)}$$

where we get the inequality due to the fact that $d_i \geq 1$ for all $i \in [\![m]\!]$.

Next, we have

$$\begin{aligned}
&T\sum_{i=1}^{m}\sum_{j=1}^{d_i}\Big(\boldsymbol{\gamma}[v_i^j]+\boldsymbol{\gamma}[(v_{i-1},v_i^j)]+\boldsymbol{\gamma}[(v_i^j,v_i)]\Big)\\
&+\sum_{i=1}^{m}\sum_{j=1}^{d_i}\left(\frac{\mathbf{x}[v_i^j]\cdot\log(d/\delta)}{\boldsymbol{\gamma}[v_i^j]}+\frac{\mathbf{x}[(v_{i-1},v_i^j)]\cdot\log(d/\delta)}{\boldsymbol{\gamma}[(v_{i-1},v_i^j)]}+\frac{\mathbf{x}[(v_i^j,v_i)]\cdot\log(d/\delta)}{\boldsymbol{\gamma}[(v_i^j,v_i)]}\right)\\
&=T\sum_{i=1}^{m}\sum_{j=1}^{d_i}\Big(\boldsymbol{\gamma}[v_i^j]+\boldsymbol{\gamma}[(v_{i-1},v_i^j)]+\boldsymbol{\gamma}[(v_i^j,v_i)]\Big)\\
&+\sum_{i=1}^{m}\sum_{j=1}^{d_i}\left(\frac{\mathbf{x}[v_i^j]\cdot\log(d/\delta)}{\boldsymbol{\gamma}[v_i^j]}+\frac{\mathbf{x}[v_i^j]\cdot\log(d/\delta)}{\boldsymbol{\gamma}[(v_{i-1},v_i^j)]}+\frac{\mathbf{x}[v_i^j]\cdot\log(d/\delta)}{\boldsymbol{\gamma}[(v_i^j,v_i)]}\right)\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\text{(as }\mathbf{x}[v_i^j]=\mathbf{x}[(v_{i-1},v_i^j)]=\mathbf{x}[(v_i^j,v_i)]\text{)}\\
&=3T\sum_{i=1}^{m}\sum_{j=1}^{d_i}\sqrt{\frac{\log(d/\delta)}{d_iT}}+3\sum_{i=1}^{m}\sqrt{d_iT\log(d/\delta)}\\
&=6\sum_{i=1}^{m}\sqrt{d_iT\log(d/\delta)}
\end{aligned}$$

where we get the second equality due to the fact that for any $i \in [\![m]\!]$, there exists exactly one index $j \in [\![d_i]\!]$ such that $\mathbf{x}[v_i^j] = 1$.

Hence, the second term of the regret above is upper bounded by $8\sum_{i=1}^{m}\sqrt{d_iT\log(d/\delta)}$.

Next, we upper bound $d_\star$. Fix any flow $\mathbf{x} \in \mathrm{co}(\mathcal{X})$. First observe that $\mathbf{x}[v_i] = 1$ for all $i \in [\![0, m]\!]$. Due to the flow constraints, we also have $\sum_{j=1}^{d_i}\mathbf{x}[(v_{i-1},v_i^j)] = 1$ for any $i \in [\![m]\!]$. Next observe that for any $i \in [\![m]\!]$ and any $j \in [\![d_i]\!]$, $\mathbf{x}[(v_{i-1},v_i^j)] = \mathbf{x}[v_i^j] = \mathbf{x}[(v_i^j,v_i)]$. Now we have the following:

$$d_\star = 3\sum_{i=1}^{m}\sum_{j=1}^{d_i}\sqrt{\mathbf{x}[(v_{i-1},v_i^j)]}+m+1\leq 3\sum_{i=1}^{m}\sqrt{d_i}+m+1\leq 5\sum_{i=1}^{m}\sqrt{d_i}$$

where we get the first inequality due to Cauchy-Swartz and we get the second equality as $m \leq \sum_{i=1}^{m}\sqrt{d_i}$.

Hence, we obtain a high-probability regret upper bound of $\widetilde{\mathcal{O}}(\sum_{i=1}^{m}\sqrt{d_iT})$. In Appendix F.1, we show that this bound is nearly tight by proving a lower bound of $\Omega(\sum_{i=1}^{m}\sqrt{d_iT})$. Moreover, this bound can be significantly better than the $\mathcal{O}(\sqrt{dT\log|\mathcal{X}|})$ bound obtained using EXP3 with Kiefer-Wolfowitz exploration. For instance, if $d_i = 2$ for all $i \in [\![m-1]\!]$ and $d_m = m^2$, our regret

bound is $\widetilde{\mathcal{O}}(\sqrt{m^2T})$, whereas EXP3 with Kiefer-Wolfowitz exploration incurs a regret of at least $\Omega(\sqrt{m^3T})$. We refer the reader to Appendix F.2 for a detailed discussion of this lower bound for EXP3 with Kiefer-Wolfowitz exploration.

In Appendix F.3, we present a much simpler approach to solving the multi-task MAB problem. In Appendix F.4, we establish a minimax regret lower bound on DAGs using the reduction presented in this section.

E.3. Extensive-form games under Bandit feedback

Extensive-form games under Bandit feedback can be modeled as follows. There is a set of decision nodes $\mathcal{X}$, a set of observation nodes $\mathcal{Y}$, and a set of terminal nodes $\mathcal{Z}$. Each decision node $x \in \mathcal{X}$ is associated with a set of actions A_x, while each observation node $y \in \mathcal{Y}$ is associated with a set of actions B_y. W.l.o.g let us assume that $|A_x| > 1$ for all $\mathbf{x} \in \mathcal{X}$ and $|B_y| > 1$ for all $y \in \mathcal{Y}$. The non-terminal nodes are governed by injective transition functions: $\rho_x : A_x \to \mathcal{Y} \cup \mathcal{Z}$ for decision nodes, and $\rho_y : B_y \to \mathcal{X} \cup \mathcal{Z}$ for observation nodes. For any two distinct nodes v_1, v_2, the ranges of ρ_{v_1} and ρ_{v_2} have empty intersection.

At each round t, for every decision node $x \in \mathcal{X}$, we select an action $a_{x,t} \in A_x$. Similarly, for every observation node $y \in \mathcal{Y}$, an adversary selects an action $b_{y,t} \in B_y$. The adversary also specifies a loss function $\mathbf{y}_t : \mathcal{Z} \to [-1, 1]$ for the terminal nodes. Starting from the root node $x^{\mathsf{r}} \in \mathcal{X}$, the game proceeds as follows:

- If a decision node x is reached, the next node that is visited is $\rho_x[a_{x,t}]$.
- If an observation node y is reached, the next node that is visited is $\rho_y[b_{y,t}]$.
- If a terminal node z is reached, the process terminates, and we incur a loss of $\mathbf{y}_t[z]$.

Note that in an extensive-form game, no node is visited more than once.

Let $\mathbf{a}_t := \{a_{x,t}\}_{x\in\mathcal{X}}$ and $\mathbf{b}_t := \{b_{y,t}\}_{y\in\mathcal{Y}}$ represent the configurations of actions at decision and observation nodes, respectively. Define $z(\mathbf{a}_t, \mathbf{b}_t)$ as the terminal node reached when transitioning according to $\mathbf{a}_t$ and $\mathbf{b}_t$. As a decision maker, we observe only the loss $\mathbf{y}_t[z(\mathbf{a}_t, \mathbf{b}_t)]$ incurred at the terminal node $z(\mathbf{a}_t, \mathbf{b}_t)$; the sequence of nodes visited during the process remains unobserved. Let $\mathcal{A}$ denote the set of all possible configurations of actions at decision nodes. The objective is to minimize the regret:

$$\mathrm{Regret}(T) := \max_{\mathbf{a}\in\mathcal{A}} \sum_{t=1}^{T} \big(\mathbf{y}_t[z(\mathbf{a}_t, \mathbf{b}_t)] - \mathbf{y}_t[z(\mathbf{a}, \mathbf{b}_t)]\big).$$

One can reformulate this problem as an adversarial linear bandit problem and use EXP3 with Kiefer-Wolfowitz exploration to get a regret upper bound of $\mathcal{O}(\sqrt{|\mathcal{Z}|T\log(N)})$ where N is defined as follows. For each terminal node z we define $n(z) := 1$. For each decision node x, we define $n(x) = \sum_{a\in A_x} n(\rho_x[a])$, and for each observation node y, we define $n(y) = \prod_{a\in A_y} n(\rho_y[a])$. Now define $N := n(x^{\mathsf{r}})$ where x^{r} is the root node. We refer the reader to Appendix G.1 for more details. We now reduce extensive-form games to a problem on DAG and show that our approach incurs a regret of $\widetilde{\mathcal{O}}(\sqrt{|\mathcal{Z}|T\log(N)})$.

We define a DAG $G = (V, E)$ as follows. Let $V = \{u_{\mathsf{s}}, u_{\mathsf{t}} : u \in \mathcal{X}\cup\mathcal{Y}\cup\mathcal{Z}\}$. For each terminal node $z \in \mathcal{Z}$, we add the edge $(z_{\mathsf{s}}, z_{\mathsf{t}})$ to E. Next for each decision node $\mathbf{x} \in \mathcal{X}$, we add the set of

edges $\{(x_{\mathsf{s}}, u_{\mathsf{s}}), (u_{\mathsf{t}}, x_{\mathsf{t}}) : u \in \{\rho_x[a] : a \in A_x\}\}$ to E. Finally for each observation node $y \in \mathcal{Y}$, let us index the nodes in $\{\rho_y[b] : b \in B_x\}$ as $\{u^{(1)}, u^{(2)}, \ldots, u^{(\ell)}\}$. We first add the set of edges $\{(y_{\mathsf{s}}, u_{\mathsf{s}}^{(1)}), (u_{\mathsf{t}}^{(\ell)}, y_{\mathsf{t}})\}$ to E. If $\ell > 1$, we then add the set of edges $\{(u_{\mathsf{t}}^{(i)}, u_{\mathsf{s}}^{(i+1)}) : i \in [\![\ell - 1]\!]\}$ to E. It is easy to observe that $x_{\mathsf{s}}^{\mathsf{r}}$ is the source node of G and $x_{\mathsf{t}}^{\mathsf{r}}$ is the sink node of G. We refer the reader to Figures 5 and 6 for two examples of this construction.

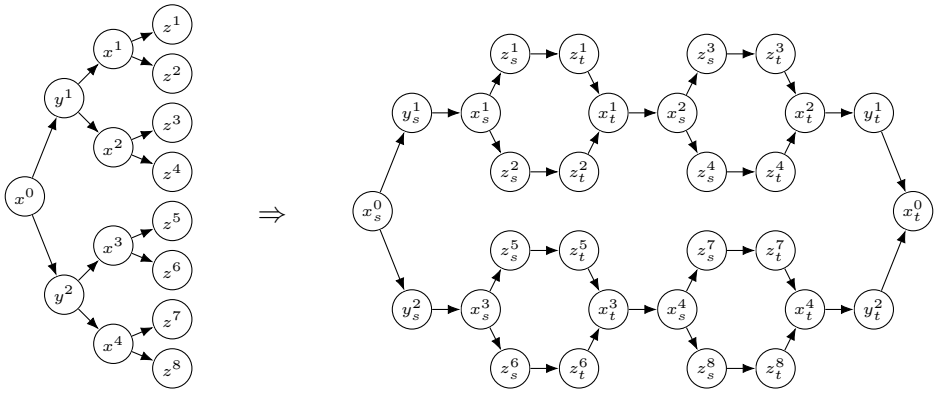

Figure 5: Example extensive-form game 1

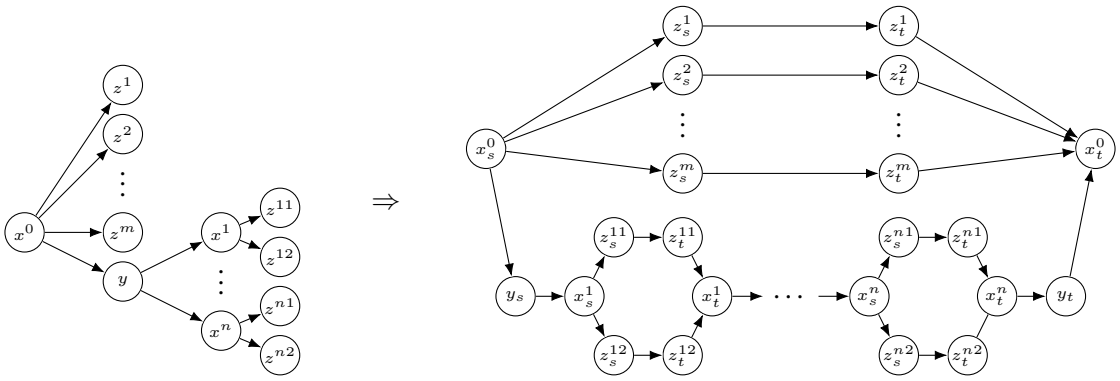

Figure 6: Example extensive-form game 2

Next, for any loss function $\mathbf{y}_t : \mathcal{Z} \to [-1, 1]$ and configuration of actions $\mathbf{b}_t$, we define a corresponding weight function $w_t : E \to \mathbb{R}$ for the edges of G as follows. For each edge $e = (v_{\mathsf{s}}, v_{\mathsf{t}})$ such that $\exists \mathbf{a} \in \mathcal{A}$ such that $v = z(\mathbf{a}, \mathbf{b}_t)$, we define $w_t(e) := \mathbf{y}_t[z(\mathbf{a}, \mathbf{b}_t)]$. For rest of the edges e, we define $w_t(e) := 0$.

We now apply our FTRL algorithm from Section 3.3 to the DAG G. At each round t, if the FTRL algorithm selects a path P_t in G, we choose $\mathbf{a}_t \in \mathcal{A}$ as follows. For each $\mathbf{x} \in \mathcal{X}$, if $(x_{\mathsf{s}}, u_{\mathsf{s}}) \in P_t$ for some $u \in \mathcal{Y} \cup \mathcal{Z}$, we assign $a_{x,t} = \rho_x^{-1}[u]$; otherwise we arbitrarily choose $a_{x,t}$ as the node x will not be reached in the current time-step by the decision process. By the construction of w_t, it follows that $\mathbf{y}_t[z(\mathbf{a}_t, \mathbf{b}_t)] = w_t(P_t)$. Consequently, we provide $\mathbf{y}_t[z(\mathbf{a}_t, \mathbf{b}_t)]$ as the bandit feedback for the path P_t to the FTRL algorithm. The way we choose $\mathbf{a}_t$ induces a bijective mapping between the set of configurations in $\mathcal{A}$ and the set of paths in G, ensuring correctness. Moreover, our algorithm is computationally efficient.

It can be shown that the number of edges in G is $\mathcal{O}(|\mathcal{Z}|)$ and number of paths from the source node to sink node is N. We refer the reader to Appendix G.2 for the proof of this fact. Hence, we incur a high-probability regret of $\widetilde{\mathcal{O}}(\sqrt{|\mathcal{Z}|T\log(N)})$ against an adaptive adversary. This is the first efficient algorithm to match the regret of EXP3 with Kiefer-Wolfowitz exploration, upto logarithmic factors. Our algorithm contributes to the long line of research on learning in extensive-form games under bandit feedback. See Farina et al. (2021) for learning in this setting without access to trajectory information (which matches our setting) and Fiegel et al. (2023) for learning with additional trajectory information (which differs from our setting).

E.4. Shortest Walk in Directed Graphs

Finding the shortest simple path in directed graphs with cycles when edges can have negative weights is NP-hard. This result extends to finding the shortest trail as well. Therefore in this section, we focus on the online shortest walk in directed graphs. In a walk, repeated vertices or edges are permitted. Given a weight function $w_t : E \to \mathbb{R}$, the weight of a walk $W = (v_0, e_1, v_1, \ldots, e_k, v_k)$ is defined as the sum of the weights of its edges:

$$w(W) := \sum_{i=1}^{k} w(e_i).$$

Let $G = (V, E)$ be a directed graph with source v_{s} and sink v_{t}. In each round t, an agent selects a walk W_t of length at most $K \le |E|$ from v_{s} to v_{t}, and the environment simultaneously chooses a weight function $w_t(\cdot)$. The agent's goal is to minimize cumulative regret relative to the optimal path:

$$\mathrm{Regret}(T) := \sum_{t=1}^{T} w_t(W_t) - \min_{W \in \mathcal{W}} \sum_{t=1}^{T} w_t(W),$$

where $\mathcal{W}$ is the set of all walks from v_{s} to v_{t} of length at most $K \le |E|$. This problem was first studied by Awerbuch and Kleinberg (2004) and can be represented as an adversarial linear bandit problem in $\mathbb{R}^{|E|}$.

We solve the online shortest walk problem in directed graph, by reducing the problem to online shortest path problem on DAG. Given the graph $G = (V, E)$, we construct a layered DAG $G^\dagger = (V^\dagger, E^\dagger)$. The vertex set is defined as:

$$V^\dagger := \{v^{(i)} : v \in V, i \in [\![0, K]\!]\}.$$

The edge set is defined as:

$$E^\dagger := \left\{(u^{(i-1)}, v^{(i)}) : (u, v) \in E \cup \{(v_{\mathsf{t}}, v_{\mathsf{t}})\}, i \in [\![K]\!]\right\}.$$

For the weight function $w_t : E \to \mathbb{R}$, we construct a weight assignment $w_t^\dagger : E^\dagger \to \mathbb{R}$ as follow:

$$w_t^\dagger\big((u^{(i-1)}, v^{(i)})\big) := \mathbb{1}[u \neq v] \cdot w_t\big((u, v)\big)$$

for all $(u, v) \in E \cup \{(v_{\mathsf{t}}, v_{\mathsf{t}})\}$ and $i \in [\![K]\!]$.

We now consider the online shortest path problem in the DAG $G^\dagger$, where the set of paths consists of those from the source node $v_{\mathsf{s}}^{(0)}$ to the sink node $v_{\mathsf{t}}^{(K)}$. We apply our FTRL algorithm from

Section 3.1 to $G^\dagger$, but first, we preprocess $G^\dagger$ to discard redundant nodes and edges that will never be part of any path from $v_{\mathsf{s}}^{(0)}$ to $v_{\mathsf{t}}^{(K)}$.

At each round t, if the FTRL algorithm selects a path P_t in $G^\dagger$, we choose $W_t \in \mathcal{W}$ as follows: for each edge $(u^{(i)}, v^{(i+1)}) \in P_t$ with $u \neq v$, we add (u, v) as the i-th edge of the walk W_t. By the construction of $w_t^\dagger$, it follows that $w_t[W_t] = w_t^\dagger[P_t]$. Consequently, we provide $w_t[W_t]$ as the bandit feedback for the path P_t to the FTRL algorithm. The way we choose W_t induces a bijective mapping between the set of walks in $\mathcal{W}$ and the set of paths in $G^\dagger$, ensuring correctness. Moreover, our algorithm is computationally efficient.

Observe that the number of edges in $G^\dagger$ is $\mathcal{O}(K|E|)$ and length of any path from the source $v_{\mathsf{s}}^{(0)}$ to sink $v_{\mathsf{t}}^{(K)}$ is K. Hence we incur a high-probability regret of at most $\widetilde{\mathcal{O}}(\sqrt{K^2|E|T})$ against an adversary. This improves upon the best-known high-probability regret bound of $\widetilde{\mathcal{O}}(|E|^2\sqrt{T})$ achieved by Zimmert and Lattimore (2022).

An open question remains: is there an efficient algorithm that matches the high-probability regret bound of $\widetilde{\mathcal{O}}(\sqrt{K|E|T})$ achieved by EXP3 with Kiefer-Wolfowitz exploration?

E.5. Colonel Blotto game

In a Colonel Blotto game, two players, A and B, have N and M soldiers, respectively, which they must allocate across K battlefields. In each round t, player A selects an allocation of N soldiers, denoted as $\mathbf{a}_t = (a_{t,1}, \ldots, a_{t,K})$, where $a_{t,i} \geq 0$ represents the number of soldiers assigned to battlefield i, and the total allocation satisfies $\sum_{i=1}^K a_{t,i} = N$. Simultaneously, player B chooses an allocation of M soldiers, given by $\mathbf{b}_t = (b_{t,1}, \ldots, b_{t,K})$.

At each battlefield i, player A incurs a loss of $\mathbf{y}_{t,i}[a_{t,i}, b_{t,i}]$. Our goal is to control player A and minimize the following regret:

$$\mathrm{Regret}(T) := \max_{\mathbf{a}\in\mathcal{A}} \sum_{t=1}^{T}\sum_{i=1}^{K} \left(\mathbf{y}_{t,i}[a_{t,i}, b_{t,i}] - \mathbf{y}_{t,i}[a_i, b_{t,i}]\right),$$

where $\mathcal{A}$ denotes the set of all possible allocations of N soldiers across the K battlefields by player A. We assume that for any $\mathbf{a} = (a_1, \ldots, a_K) \in \mathcal{A}$, the total loss satisfies $\sum_{i=1}^K \mathbf{y}_{t,i}[a_i, b_{t,i}] \in [-1, 1]$, and player A observes only $\sum_{i=1}^K \mathbf{y}_{t,i}[a_{t,i}, b_{t,i}]$ as bandit feedback at the end of round t. It is easy to see that this problem can be represented as a combinatorial bandit problem in $\{0,1\}^{KN}$.

This problem can be reduced to a problem on directed acyclic graphs (DAGs), as first shown by Behnezhad et al. (2023). For completeness, we provide the reduction here.

We construct a DAG $G = (V, E)$ as follows. Define the vertex set as:

$$V = \{v_0^0, v_K^N\} \cup \{v_i^j \mid i \in [\![K-1]\!], j \in [\![0, N]\!]\}.$$

Next, we add the following sets of edges to E:

- $\{(v_0^0, v_1^j) \mid j \in [\![0, N]\!]\}$,
- $\{(v_{i-1}^{j_0}, v_i^{j_1}) \mid i \in [\![1, K-1]\!], j_0 \in [\![0, N]\!], j_1 \in [\![j_0, N]\!]\}$,
- $\{(v_{K-1}^j, v_K^N) \mid j \in [\![0, N]\!]\}$.

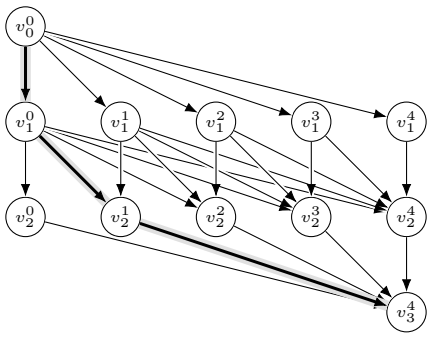

Figure 7: Dag for Blotto game with $N = 4$ soldiers and $K = 3$ battlefields. The shaded path corresponds to the allocation $\mathbf{a} = (0, 1, 3)$.

Observe that $v_{\mathsf{s}} = v_0^0$ serves as the source node and $v_{\mathsf{t}} = v_K^N$ as the sink node.

For any set of loss functions $\mathbf{y}_{t,i}$ and allocation $\mathbf{b}_t$, we define a corresponding weight function $w_t : E \to \mathbb{R}$ on the edges of G as follows. For any edge $e = (v_{i-1}^{j_0}, v_i^{j_1})$, we set

$$w_t(e) := \mathbf{y}_{t,i}[j_1 - j_0, b_{t,i}].$$

We now apply our FTRL algorithm from Section 3.1 to the DAG G. At each round t, if the FTRL algorithm selects a path P_t in G, we determine $\mathbf{a}_t \in \mathcal{A}$ as follows. Fix an index $i \in [\![K]\!]$ and consider the edge $e = (v_{i-1}^{j_0}, v_i^{j_1}) \in P_t$. We then set $a_{t,i} = j_1 - j_0$. By the construction of w_t, it follows that $\sum_{i=1}^K \mathbf{y}_{t,i}[a_{t,i}, b_{t,i}] = w_t(P_t)$. Thus, we provide $\sum_{i=1}^K \mathbf{y}_{t,i}[a_{t,i}, b_{t,i}]$ as the bandit feedback for the path P_t to the FTRL algorithm. The way we choose $\mathbf{a}_t$ induces a bijective mapping between the set of allocations in $\mathcal{A}$ and the set of paths in G, ensuring correctness. Moreover, our algorithm is computationally efficient.

Observe that the number of edges in G is $\mathcal{O}(K^2 N)$, and the length of any path from the source to the sink is K. Consequently, we incur a high-probability regret of at most $\widetilde{\mathcal{O}}(\sqrt{K^3 NT})$ against an adaptive adversary. This improves upon the best-known high-probability regret bound of $\widetilde{\mathcal{O}}(K^2 N^2 \sqrt{T})$ achieved by Zimmert and Lattimore (2022).

An open question remains: is there an efficient algorithm that matches the high-probability regret bound of $\widetilde{\mathcal{O}}(\sqrt{K^2 NT})$ achieved by EXP3 with Kiefer-Wolfowitz exploration?

E.6. m-sets

An m-set is the set of vectors $\mathcal{X} := \{x \in \{0,1\}^d : \|x\|_1 = m\}$. We construct a DAG G as follows. The vertex set of G is

$$V = \{v_i^j : i \in [\![0, d-m]\!], j \in [\![0, m]\!]\}.$$

The edge set is defined as

$$E = \{(v_i^{j-1}, v_i^j) : i \in [\![0, d-m]\!], j \in [\![m]\!]\} \cup \{(v_{i-1}^j, v_i^j) : i \in [\![d-m]\!], j \in [\![0, m]\!]\}.$$

Note that $v_{\mathsf{s}} = v_0^0$ is the source node of G, and $v_{\mathsf{t}} = v_{d-m}^m$ is the sink node.

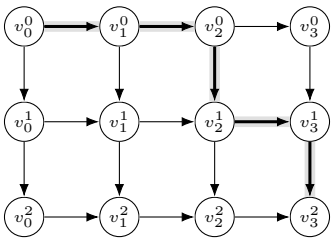

Figure 8: DAG for m-set with $m = 2$ and $d = 5$. The shaded path corresponds to the vector $\mathbf{x}$ such that $\mathbf{x}[3] = \mathbf{x}[5] = 1$ and $\mathbf{x}[1] = \mathbf{x}[2] = \mathbf{x}[4] = 0$.

Next, for any loss function $\mathbf{y}_t : [\![d]\!] \to \mathbb{R}$, we define a corresponding weight function $w_t : E \to \mathbb{R}$ for the edges of G as follows. For an edge $e = (v_i^j, v_{i+1}^j)$, we set $w_t(e) := 0$. For an edge $e = (v_i^{j-1}, v_i^j)$, we define $w_t(e) := \mathbf{y}_t[i + j]$.

We now apply our FTRL algorithm from Section 3.1 to the DAG G. At each round t, if the FTRL algorithm selects a path P_t in G, we choose $\mathbf{x}_t \in \mathcal{X}$ such that for any $k \in [\![d]\!]$, we set $\mathbf{x}_t[k] = 1$ if there exists an edge $(v_i^{j-1}, v_i^j) \in P_t$ with $k = i + j$, and $\mathbf{x}_t[k] = 0$ otherwise. By the construction of w_t, it follows that $\langle \mathbf{x}_t, \mathbf{y}_t \rangle = w_t(P_t)$. Consequently, we provide $\langle \mathbf{x}_t, \mathbf{y}_t \rangle$ as the bandit feedback for the path P_t to the FTRL algorithm. The way we choose $\mathbf{x}_t$ induces a bijective mapping between the set of vectors in $\mathcal{X}$ and the set of paths in G, ensuring correctness. Moreover, our algorithm is computationally efficient.

Finally, observe that each path in G has a length of d and the number of edges in G is $\mathcal{O}(md)$. Thus, we incur a high-probability regret of $\widetilde{\mathcal{O}}(\sqrt{md^2T})$ against an adaptive adversary. This improves upon the best-known high-probability regret bound of $\widetilde{\mathcal{O}}(d^2\sqrt{T})$ achieved by Zimmert and Lattimore (2022).

An open question remains: is there an efficient algorithm that matches the high-probability regret bound of $\widetilde{\mathcal{O}}(\sqrt{mdT})$ achieved by EXP3 with Kiefer-Wolfowitz exploration?

Appendix F. Multi-Task MAB: Additional Details

F.1. Multi-Task MAB Lower Bound

Consider the Multi-task MAB instance where the set of arms $\mathcal{X}$ is defined as follows:

$$\mathcal{X} = \left\{ \mathbf{x} \in \{0,1\}^d : \forall j \in [\![m]\!] \sum_{i=d_{1:j-1}+1}^{d_{1:j}} \mathbf{x}[i] = 1 \right\}$$

where $d_i \geq 2$, $d = \sum_{i=1}^m d_i$, $d_{1:j} = \sum_{i=1}^j d_i$ and $d_{1:0} = 0$ for all $j \in [\![m]\!]$.

Let us fix one regret minimizing algorithm, say $\mathcal{A}$ and assume that $\mathcal{A}$ is deterministic. Now we show that algorithm $\mathcal{A}$ incurs a regret of $\Omega(\sum_{i=1}^m \sqrt{d_i T})$. We later extend the result to randomized algorithms using Yao's lemma. For all $j \in [\![m]\!]$, let $\varepsilon_j > 0$ be a parameter that we fix later in the proof.

Let $\widetilde{\mathcal{X}} = \left\{ \mathbf{x} \in \{0,1\}^d : \forall j \in [\![m]\!] \sum_{i=d_{1:j-1}+1}^{d_{1:j}} \mathbf{x}[i] \leq 1 \right\}$. First we describe an instance $I_{\widetilde{\mathbf{x}}}$, where $\widetilde{\mathbf{x}} \in \widetilde{\mathcal{X}}$. In this instance, in each round t, we choose a loss function $\mathbf{y}_t : [\![d]\!] \to [-1, 1]$

as follows. First we choose an index $j \in [\![m]\!]$ uniformly at random. Now we define $\mathbf{y}_t[i] = 0$ if $i \notin [\![d_{1:j-1}+1, d_{1:j}]\!]$. Next for all $i \in [\![d_j]\!]$, we sample $v_i \sim \mathrm{Ber}(\frac{1}{2} - \varepsilon_j \cdot \mathbb{1}[\widetilde{\mathbf{x}}[d_{1:j-1}+i] = 1])$ and assign $\mathbf{y}_t[d_{1:j-1}+i] = v_i$. Now observe that expected loss of any arm $\mathbf{x} \in \mathcal{X}$ under the instance $I_{\widetilde{\mathbf{x}}}$ is $\frac{1}{m}\sum_{j=1}^{m}\sum_{i=d_{1:j-1}+1}^{d_{1:j}}(\frac{1}{2} - \varepsilon_j \cdot \mathbb{1}[\mathbf{x}[i] = \widetilde{\mathbf{x}}[i] = 1])$. For $\mathbf{x} \in \mathcal{X}$, $\mathbf{x}$ is the best arm for the instance $I_{\mathbf{x}}$ and its reward is $\mu^* := \frac{1}{2} - \frac{1}{m}\sum_{j=1}^{m}\varepsilon_j$.

In each round t, if $\mathcal{A}$ chooses an arm $\mathbf{x}_t \in \mathcal{X}$, then the regret $R_T(\mathbf{x})$ on an instance $I_{\mathbf{x}}$, where $\mathbf{x} \in \mathcal{X}$, is equal to:

$$R_T(\mathbf{x}) = \mathbb{E}_{I_{\mathbf{x}}}\left[\sum_{t=1}^{T}\langle \mathbf{x}_t, \mathbf{y}_t\rangle\right] - T\cdot\mu^* = \frac{1}{m}\sum_{j=1}^{m}\varepsilon_j T - \frac{\varepsilon_j}{m}\sum_{t=1}^{T}\sum_{j=1}^{m}\sum_{i=d_{1:j-1}+1}^{d_{1:j}}\mathbb{P}_{I_{\mathbf{x}}}[\mathbf{x}_t[i] = \mathbf{x}[i] = 1]$$

where $\mathbb{P}_{I_{\mathbf{x}}}$ is probability law under the instance $I_{\mathbf{x}}$.

Now for any instance $I_{\mathbf{x}}$ such that $\mathbf{x} \in \mathcal{X}$, the regret can be broken down as $R_T(\mathbf{x}) = \sum_{j=1}^{m} R_T^{(j)}(\mathbf{x})$ where

$$R_T^{(j)}(\mathbf{x}) := \frac{\varepsilon_j T}{m} - \frac{\varepsilon_j}{m}\sum_{t=1}^{T}\sum_{i=d_{1:j-1}+1}^{d_{1:j}}\mathbb{P}_{I_{\mathbf{x}}}[\mathbf{x}_t[i] = 1, \mathbf{x}[i] = 1]$$

Let $\mathcal{I} = \bigcup_{\mathbf{x}\in\mathcal{X}} I_{\mathbf{x}}$. Fix an index $j \in [\![m]\!]$. We now show that $\mathbb{E}_{I_{\mathbf{x}'}\sim\mathrm{Unif}(\mathcal{I})}[R_T^{(j)}(\mathbf{x}')] \geq c\sqrt{d_j T}$ where c is some absolute constant. Let

$$\mathcal{X}^{(j)} := \left\{\mathbf{x} \in \{0,1\}^d : \forall i \in [\![m]\!] \setminus \{j\} \sum_{s=d_{1:i-1}+1}^{d_{1:i}}\mathbf{x}[s] = 1, \sum_{s=d_{1:j-1}+1}^{d_{1:j}}\mathbf{x}[s] = 0\right\}.$$

For any $\mathbf{x} \in \mathcal{X}^{(j)}$, let $\mathbf{x}^{(i)}$ be the vector in $\mathcal{X}$ such that $\mathbf{x}^{(i)}[s] = \mathbf{x}[s]$ for all $s \notin [\![d_{1:j-1}+1, d_{1:j}]\!]$ and $\mathbf{x}^{(i)}[d_{1:j-1}+i] = 1$.

First, we consider the case where $d_j \geq 48$. Let us fix $\mathbf{x} \in \mathcal{X}^{(j)}$. Now we claim that there is a set $\mathcal{S}_{\mathbf{x}} \subseteq [\![d_j]\!]$ with at least $d_j/3$ indices such that for each $i \in \mathcal{S}_{\mathbf{x}}$, we have $R_T^{(j)}(\mathbf{x}^{(i)}) \geq c_0\sqrt{d_j T}$ where c_0 is some absolute constant.

Before we prove our claim, we first show that if our claim holds true, then we have that $\mathbb{E}_{I_{\mathbf{x}'} \sim \mathrm{Unif}(\mathcal{I})}[R_T^{(j)}(\mathbf{x}')] \geq c \cdot \sqrt{d_j T}$ where c is some absolute constant. Now we have the following:

$$\begin{aligned}
\mathbb{E}_{I_{\mathbf{x}'} \sim \mathrm{Unif}(\mathcal{I})}[R_T^{(j)}(\mathbf{x}')] &= \frac{1}{\prod_{s=1}^m d_s} \sum_{\mathbf{x} \in \mathcal{X}^{(j)}} \sum_{i=1}^{d_j} R_T^{(j)}(\mathbf{x}^{(i)}) \\
&\geq \frac{1}{\prod_{s=1}^m d_s} \sum_{\mathbf{x} \in \mathcal{X}^{(j)}} \sum_{i \in \mathcal{S}_{\mathbf{x}}} R_T^{(j)}(\mathbf{x}^{(i)}) \\
&\geq \frac{c_0}{\prod_{s=1}^m d_s} \sum_{\mathbf{x} \in \mathcal{X}^{(j)}} \sum_{i \in \mathcal{S}_{\mathbf{x}}} \sqrt{d_j T} \\
&\geq \frac{c_0}{\prod_{s=1}^m d_s} \sum_{\mathbf{x} \in \mathcal{X}^{(j)}} \frac{d_j}{3} \cdot \sqrt{d_j T} \\
&= \frac{c_0}{\prod_{s=1}^m d_s} \cdot \prod_{s \neq j} d_s \cdot \frac{d_j}{3} \cdot \sqrt{d_j T} \\
&= \frac{c \sqrt{d_j T}}{3}
\end{aligned}$$

Now we prove our claim. We use the following version of the chain rule in our analysis.

Lemma 30 (Chain Rule) *Let $f(x_1, x_2, \ldots, x_n)$ and $g(x_1, x_2, \ldots, x_n)$ be two joint PMFs for a tuple of random variables $(X_i)_{i \in [n]}$. Let the sample space be $\Omega = \{0,1\}^n$. Then we have the following:*

$$KL(f, g) = \sum_{\omega \in \Omega} f(\omega) \left(KL(f(X_1), g(X_1)) + \sum_{i=2}^{n} KL(f(X_i | X_{-i} = \omega_{-i}), g(X_i | X_{-i} = \omega_{-i})) \right)$$

where $X_{-i} = (X_1, \ldots, X_{i-1})$, $\omega_{-i} = (\omega_1, \ldots, \omega_{i-1})$.

For an instance $I_{\mathbf{x}^{(i)}}$, let $f_i(\ell_1, \ldots, \ell_T)$ denote the joint PMF for the tuple of loss values observed by $\mathcal{A}$ in each round under the probability law $\mathbb{P}_{I_{\mathbf{x}^{(i)}}}$. Observe that our sample space is $\Omega = \{0,1\}^T$. This is a valid sample space as $\mathcal{A}$ is deterministic and the probability of it seeing a loss value of 1 in round t only depends on the loss values it observed in the previous rounds. Similarly for the alternate instance $I_{\mathbf{x}}$, let $f_0(\ell_1, \ldots, \ell_T)$ denote the joint PMF for the tuple of loss values observed by $\mathcal{A}$ in each round under the probability law $\mathbb{P}_{I_{\mathbf{x}}}$.

First observe that the instances $I_{\mathbf{x}^{(i)}}$ and $I_{\mathbf{x}}$ only differ at index $d_{1:j-1} + i$. For each $\omega \in \Omega$, let $\mathbf{x}_{1,\omega}, \mathbf{x}_{2,\omega}, \ldots, \mathbf{x}_{T,\omega}$ be the sequence of arms chosen by $\mathcal{A}$ on ω. Conditioning on a set of outcomes $X_1 = \omega_1, X_2 = \omega_2, \ldots, X_{t-1} = \omega_{t-1}$, we have $X_t \sim \mathrm{Ber}(\mu_i)$ for the instance $I_{\mathbf{x}^{(i)}}$ and $X_t \sim \mathrm{Ber}(\mu_0)$ for the instance $I_{\mathbf{x}}$ where $\mu_0 - \mu_i = \frac{\varepsilon_j}{m} \cdot \mathbf{x}_{t,\omega}[d_{1:j-1}+i]$. Let $T_i = \sum_{t=1}^T \mathbf{x}_t[d_{1:j-1}+i]$. For each $\omega \in \Omega$, let $T_{i,\omega} = \sum_{t=1}^T \mathbf{x}_{t,\omega}[d_{1:j-1} + i]$. Note that T_i is a random variable and $T_{i,\omega}$ is a

fixed value. Now we have the following:

$$\begin{aligned}
\mathrm{KL}(f_0, f_i) &= \sum_{\omega\in\Omega} f_0(\omega)\left(\mathrm{KL}(f_0(X_1), f_i(X_1)) + \sum_{t=2}^{T}\mathrm{KL}(f_0(X_t|X_{-t}=\omega_{-t}), f_i(X_t|X_{-t}=\omega_{-t}))\right)\\
&\le \frac{4\varepsilon_j^2}{m^2}\sum_{\omega\in\Omega} f_0(\omega)\sum_{t=1}^{T}\mathbf{x}_{t,\omega}[d_{1:j-1}+i]\\
&= \frac{4\varepsilon_j^2}{m^2}\sum_{\omega\in\Omega} f_0(\omega)T_{i,\omega}\\
&= \frac{4\varepsilon_j^2}{m^2}\cdot\mathbb{E}_{I_{\mathbf{x}}}[T_i]
\end{aligned}$$

Now observe that $\sum_{i=1}^{d_j}\mathbb{E}_{I_{\mathbf{x}}}[T_i] = T$. Hence, there exists a set $\mathcal{S}_{\mathbf{x}} \subseteq [\![d_j]\!]$ with at least $d_j/3$ indices such that for each $i \in \mathcal{S}_x$, we have $\mathbb{E}_{I_{\mathbf{x}}}[T_i] \le \frac{3T}{d_j}$. Fix $\varepsilon_j = \frac{m\cdot d_j^{1/2}}{10T^{1/2}}$. Now for each $i \in \mathcal{S}_{\mathbf{x}}$, we have $\mathrm{KL}(f_0, f_i) \le \frac{12\varepsilon_j^2 T}{m^2 d_j} = \frac{3}{25}$.

Fix $i \in \mathcal{S}_x$. Let A_i be the event that $T_i \le \frac{12T}{d_j}$. Due to Markov's inequality, we have $\mathbb{P}_{I_{\mathbf{x}}}(A_i) \ge \frac{3}{4}$. Now due to Pinsker's inequality we have the following:

$$\begin{aligned}
\mathbb{P}_{I_{\mathbf{x}^{(i)}}}(A_i) &\ge \mathbb{P}_{I_{\mathbf{x}}}(A_i) - \sqrt{\frac{\mathrm{KL}(f_0, f_j)}{2}}\\
&\ge \frac{3}{4} - \sqrt{\frac{3}{50}}\\
&> \frac{1}{2}
\end{aligned}$$

Using the above the inequality, we get $\mathbb{E}_{I_{\mathbf{x}^{(i)}}}[T_i] \le T\cdot\mathbb{P}_{I_{\mathbf{x}^{(i)}}}(A_i^c) + \frac{12T}{d_j} \le \frac{3T}{4}$ when $d_j \ge 48$. Now we have the following:

$$\begin{aligned}
R_T^{(j)}(\mathbf{x}^{(i)}) &= \frac{\varepsilon_j T}{m} - \frac{\varepsilon_j}{m}\sum_{t=1}^{T}\mathbb{P}_{I_{\mathbf{x}^{(i)}}}[\mathbf{x}_t[d_{1:j-1}+i]=1]\\
&= \frac{\varepsilon_j T}{m} - \frac{\varepsilon_j}{m}\mathbb{E}_{I_{\mathbf{x}^{(i)}}}\left[\sum_{t=1}^{T}\mathbf{x}_t[d_{1:j-1}+i]\right]\\
&= \frac{\varepsilon_j T}{m} - \frac{\varepsilon_j}{m}\mathbb{E}_{I_{\mathbf{x}^{(i)}}}[T_i]\\
&\ge \frac{\varepsilon_j T}{m} - \frac{3\varepsilon_j T}{4m}\\
&= \frac{\sqrt{d_j T}}{40}
\end{aligned}$$

Next we look at the case when $d_j \le 48$. For simplicity of presentation, let us assume that $d_j = 2$. Our analysis can be easily extended to any constant between 2 and 48.

For any $i \in \{1,2\}$ and $t \in [T]$, let $A_{i,t}$ be the event that $\mathbf{x}[d_{1:j-1}+i] = 1$. Note that $A_{1,t} = A_{2,t}^c$. Fix $\mathbf{x} \in \mathcal{X}^{(j)}$. Now we claim that there an index $i \in \{1,2\}$ such that $\mathbb{P}_{I_{\mathbf{x}^{(i)}}}(A_{i,t}) < \frac{3}{4}$. For

the sake of contradiction, let us assume that $\mathbb{P}_{I_{\mathbf{x}^{(i)}}}(A_{i,t}) \geq \frac{3}{4}$ for all $i \in \{1,2\}$. Then we have $\mathbb{P}_{I_{\mathbf{x}^{(1)}}}(A_{1,t}) - \mathbb{P}_{I_{\mathbf{x}^{(2)}}}(A_{1,t}) > \frac{1}{2}$.

For an instance $I_{\mathbf{x}^{(i)}}$, let $f_i(\ell_1, \ldots, \ell_T)$ denote the joint PMF for the tuple of loss values observed by $\mathcal{A}$ in each round under the probability law $\mathbb{P}_{I_{\mathbf{x}^{(i)}}}$. Our sample space is $\Omega = \{0,1\}^T$.

First observe that the instances $I_{\mathbf{x}^{(1)}}$ and $I_{\mathbf{x}^{(2)}}$ only differ at the indices $d_{1:j-1}+1$ and $d_{1:j-1}+2$. For each $\omega \in \Omega$, let $\mathbf{x}_{1,\omega}, \mathbf{x}_{2,\omega}, \ldots, \mathbf{x}_{T,\omega}$ be the sequence of arms chosen by $\mathcal{A}$ on ω. Conditioning on a set of outcomes $X_1 = \omega_1, X_2 = \omega_2, \ldots, X_{t-1} = \omega_{t-1}$, we have $X_t \sim \mathrm{Ber}(\mu_1)$ for the instance $I_{\mathbf{x}^{(1)}}$ and $X_t \sim \mathrm{Ber}(\mu_2)$ for the instance $I_{\mathbf{x}^{(2)}}$ where $|\mu_1 - \mu_2| = \frac{\varepsilon_j}{m}$. Now we have the following:

$$\begin{aligned}
\mathrm{KL}(f_1, f_2) &= \sum_{\omega \in \Omega} f_1(\omega) \left(\mathrm{KL}(f_1(X_1), f_2(X_1)) + \sum_{t=2}^{T} \mathrm{KL}(f_1(X_t | X_{-t} = \omega_{-t}), f_2(X_t | X_{-t} = \omega_{-t})) \right) \\
&\leq \frac{4\varepsilon_j^2 T}{m^2} \sum_{\omega \in \Omega} f_1(\omega) \\
&= \frac{4\varepsilon_j^2 T}{m^2}
\end{aligned}$$

Fix $\varepsilon_j = \frac{m}{4\sqrt{T}}$. Due to Pinsker's inequality we arrive at the following contradiction:

$$\begin{aligned}
\mathbb{P}_{I_{\mathbf{x}^{(1)}}}(A_{1,t}) - \mathbb{P}_{I_{\mathbf{x}^{(2)}}}(A_{1,t}) &\leq \sqrt{\frac{\mathrm{KL}(f_1, f_2)}{2}} \\
&\leq \sqrt{\frac{2\varepsilon_j^2 T}{m^2}} \\
&< \frac{1}{2}
\end{aligned}$$

Now we have the following:

$$\begin{aligned}
R_T^{(j)}(\mathbf{x}^{(1)}) + R_T^{(j)}(\mathbf{x}^{(2)}) &= \frac{2\varepsilon_j T}{m} - \frac{\varepsilon_j}{m} \sum_{t=1}^{T} \mathbb{P}_{I_{\mathbf{x}^{(1)}}}\left[\mathbf{x}_t[d_{1:j-1}+1] = 1\right] + \mathbb{P}_{I_{\mathbf{x}^{(2)}}}\left[\mathbf{x}_t[d_{1:j-1}+2] = 1\right] \\
&= \frac{2\varepsilon_j T}{m} - \frac{\varepsilon_j}{m} \sum_{t=1}^{T} \mathbb{P}_{I_{\mathbf{x}^{(1)}}}[A_{1,t}] + \mathbb{P}_{I_{\mathbf{x}^{(2)}}}[A_{2,T}] \\
&> \frac{2\varepsilon_j T}{m} - \frac{7\varepsilon_j}{4m} && (\mathbb{P}_{I_{\mathbf{x}^{(1)}}}[A_{1,t}] + \mathbb{P}_{I_{\mathbf{x}^{(2)}}}[A_{2,T}] < \tfrac{7}{4}) \\
&= \frac{\varepsilon_j T}{4m} \\
&= \frac{\sqrt{T}}{16}
\end{aligned}$$

Now we have the following:

$$\begin{aligned}
\mathbb{E}_{I_{\mathbf{x}'}\sim\mathrm{Unif}(\mathcal{I})}[R_T^{(j)}(\mathbf{x}')] &= \frac{1}{\prod_{s=1}^m d_s} \sum_{\mathbf{x}\in\mathcal{X}^{(j)}} \sum_{i\in\{1,2\}} R_T^{(j)}(\mathbf{x}^{(i)}) \\
&\geq \frac{1}{16\prod_{s=1}^m d_s} \sum_{\mathbf{x}\in\mathcal{X}^{(j)}} \sqrt{T} \\
&= \frac{1}{16\prod_{s=1}^m d_s} \cdot \prod_{s\neq j} d_s \cdot \sqrt{T} \\
&= \frac{\sqrt{T}}{32}
\end{aligned}$$

Hence, our claim holds and therefore we have

$$\mathbb{E}_{I_{\mathbf{x}'}\sim\mathrm{Unif}(\mathcal{I})}[R_T(\mathbf{x}')] = \sum_{j=1}^m \mathbb{E}_{I_{\mathbf{x}'}\sim\mathrm{Unif}(\mathcal{I})}[R_T^{(j)}(\mathbf{x}')] \geq c' \sum_{j=1}^m \sqrt{d_j T}$$

where c' is some absolute constant. Due to Yao's lemma we have that any randomized algorithm should also have a regret of at least $c'' \sum_{j=1}^m \sqrt{d_j T}$ where c'' is some absolute constant.

F.2. Lower Bound for EXP3 with Kiefer-Wolfowitz Exploration

For any set of arms $\mathcal{X} \subseteq \{0,1\}^d$ such that the dimension $\mathcal{X}$ is $\Theta(d)$, EXP3 with Kiefer-Wolfowitz exploration plays a fixed distribution π over the set of arms with probability at least $\sqrt{\frac{d\log|\mathcal{X}|}{cT}}$ where c is an absolute constant.

Consider the Multi-task MAB instance with set of arms $\mathcal{X}$ where $d_i = 2$ for all $i \in [\![m-1]\!]$ and $d_m = m^2$ where $m \geq 2$. Recall that $d = \sum_{j=1}^m d_j$ and $d_{1:i} = \sum_{j=1}^i d_j$. There exists an index $i_\star \in [\![d_{1:m-1}+1, d_m]\!]$, such that $\sum_{\mathbf{x}\in\mathcal{X}} \pi[\mathbf{x}[i_\star]] \leq \frac{1}{2}$. For all $t \in [\![T]\!]$, we choose a loss function $\mathbf{y}_t : [\![d]\!] \to [-1,1]$ such that $\mathbf{y}_t[i] = -1$ if $i = i_\star$ and $\mathbf{y}_t[i] = 0$ otherwise. It is easy to observe that EXP3 with Kiefer-Wolfowitz exploration incurs an expected regret of at least $\sqrt{\frac{d\log|\mathcal{X}|}{cT}} \cdot \frac{1}{2}$ in each round. Hence, EXP3 with Kiefer-Wolfowitz exploration incurs a regret of at least $\Omega(\sqrt{dT\log|\mathcal{X}|}) = \Omega(\sqrt{m^3 T})$.

F.3. A Simple, Efficient Algorithm for Multi-task MAB

Recall that in the Multi-task MAB problem, we are given a set of m multi-armed bandit (MAB) problems, where the i-th MAB problem has d_i arms. In each round, we simultaneously choose one arm from each MAB problem and receive the sum of the losses of the chosen arms as the loss feedback. The objective is to minimize regret with respect to the best arm in each MAB problem in hindsight.

Consider the following algorithm. For each $i \in [\![m]\!]$, we independently execute EXP3-IX (Neu, 2015) on the i-th MAB problem. Note that for any MAB problem with K arms and losses in $[-1,1]$, the version of EXP3-IX under consideration incurs a regret of at most $c\sqrt{KT\log(K/\delta')}$ with probability at least $1-\delta'$, where c is an absolute constant.

Let $\mathbf{y}_{t,i} : [\![d_i]\!] \to \mathbb{R}$ be the loss function for the arms in the i-th MAB. For each $i \in [\![m]\!]$, let $I_{t,i}$ be the arm selected by the EXP3-IX algorithm for the i-th MAB in round t. We choose these

recommended arms and observe the total loss $\ell_t = \sum_{i=1}^{m} \mathbf{y}_{t,i}[I_{t,i}]$, which satisfies $\ell_t \in [-1, 1]$. We then provide ℓ_t as the loss feedback to each EXP3-IX algorithm.

If we set $\delta' = \delta/m$, then with probability at least $1 - \delta$, the regret incurred in the multi-task MAB problem is upper-bounded as follows:

$$\begin{aligned}
\text{Regret}(T) &= \max_{(j_1,j_2,\ldots,j_m)\in[d_1]\times[d_2]\times\ldots\times[d_m]} \sum_{t=1}^{T}\sum_{i=1}^{m} \mathbf{y}_{t,i}[I_{t,i}] - \sum_{t=1}^{T}\sum_{i=1}^{m} \mathbf{y}_{t,i}[j_i] \\
&= \sum_{i=1}^{m} \max_{j_i\in[d_i]} \sum_{t=1}^{T} \mathbf{y}_{t,i}[I_{t,i}] - \sum_{t=1}^{T} \mathbf{y}_{t,i}[j_i] \\
&\leq c \cdot \sum_{i=1}^{m} \sqrt{d_i T \log(m d_i/\delta)}
\end{aligned}$$

We obtain the last inequality because the EXP3-IX algorithm running on the i-th MAB effectively operates on a bandit instance where the loss of the j-th arm is adaptively chosen as $\mathbf{y}_{t,i}[j] + \sum_{i'\neq i} \mathbf{y}_{t,i'}[I_{t,i'}] \in [-1, 1]$. Consequently, with probability at least $1 - \delta/m$, we have:

$$\begin{aligned}
&\max_{j_i\in[d_i]} \sum_{t=1}^{T} \mathbf{y}_{t,i}[I_{t,i}] - \sum_{t=1}^{T} \mathbf{y}_{t,i}[j_i] \\
&= \max_{j_i\in[d_i]} \sum_{t=1}^{T} \left(\mathbf{y}_{t,i}[I_{t,i}] + \sum_{i'\neq i} \mathbf{y}_{t,i'}[I_{t,i'}] \right) - \sum_{t=1}^{T} \left(\mathbf{y}_{t,i}[j_i] + \sum_{i'\neq i} \mathbf{y}_{t,i'}[I_{t,i'}] \right) \\
&\leq c \cdot \sqrt{d_i T \log(m d_i/\delta)}.
\end{aligned}$$

We then obtain the high-probability regret guarantee by applying the union bound.

Remark: Prior to our work, Zimmert et al. (2019) used a similar approach for Hypercube.

F.4. Minimax Lower Bound for DAGs

We prove the following theorem in this section.

Theorem 31 *Consider integers $d, N \geq 4$ satisfying $d \leq N \leq 2^{d/2}$. There exists a DAG G with at most d edges and at most N paths from the source to the sink such that the regret is lower bounded by $\Omega\left(\sqrt{dT\log(N)/\log(d)}\right)$.*

Proof. Consider the instance of the multi-task MAB problem where $m = \log(N)/\log(d)$ and $d_i = \frac{d}{2m}$. For simplicity, we assume that both m and $\frac{d}{2m}$ are integers. As shown in Appendix F.1, this instance has a regret lower bound of $\Omega\left(\sum_{i=1}^{m} \sqrt{d_i T}\right) = \Omega\left(\sqrt{dT\log(N)/\log(d)}\right)$.

Next, consider the reduction of this multi-task MAB problem to a directed acyclic graph (DAG) G, which is described in Appendix E.2. First, note that for the graph G, the regret lower bound remains $\Omega\left(\sqrt{dT\log(N)/\log(d)}\right)$. Furthermore, the graph G contains d edges and has at most $\left(\frac{d}{2m}\right)^m \leq N$ paths from the source to the sink. This follows from the fact that $m\log\left(\frac{d}{2m}\right) \leq m\log(d) = \log(N)$. Hence, DAG G is the required graph. ■

Appendix G. Extensive-Form Games: Additional Details

G.1. Linear Bandit Formulation of Extensive-Form Games

First, we prove the following lemma.

Lemma 32 *Consider a tree such that each non-leaf node has at least 2 children. Then the number of leaf nodes in the tree at least the number of non-leaf nodes in the tree.*

Proof. Let L_1 be the number of leaf nodes in the tree and L_2 be the number of non-leaf nodes in the tree. Let E be the total number of edges in the tree. Then we have $E = L_1 + L_2 - 1$. Also, we have $E \geq 2L_2$ as each non-leaf node contributes to at least two edges as they have at least 2 children each. Hence, we have $L_1 + L_2 - 1 \geq 2L_2$ which implies that $L_1 \geq L_2 + 1$. ■

Now we describe the linear bandit formulation of Extensive-form games. For a configuration $\mathbf{a} = \{a_x\}_{\mathbf{x}\in\mathcal{X}}$ of actions at the decision nodes, we describe a vector $s^{\mathbf{a}} \in \{0,1\}^{|\mathcal{X}|+|\mathcal{Y}|+|\mathcal{Z}|}$ indexed by the nodes in the game as follows:

$$\begin{aligned}
&s^{\mathbf{a}}[x^{\mathsf{r}}] = 1\\
&s^{\mathbf{a}}[x] = s^{\mathbf{a}}[\rho_x[a_x]] \quad \forall \mathbf{x} \in \mathcal{X}\\
&s^{\mathbf{a}}[\rho_x[a]] = 0 \quad \forall \mathbf{x} \in \mathcal{X}, \forall a \in A_x \setminus \{a_x\}\\
&s^{\mathbf{a}}[y] = s^{\mathbf{a}}[\rho_y[b_y]] \quad \forall y \in \mathcal{Y}, \forall b_y \in B_y
\end{aligned}$$

Let $\mathcal{S}_x$ be the set of all such vectors $s^{\mathbf{a}}$ corresponding to all possible configurations $\mathbf{a}$ of actions at the decision nodes. $\mathcal{S}_x$ now is the set of arms in our linear bandit formulation. Recall the definition of N. It is easy to observe that $|\mathcal{S}_x| = N$.

Next for a configuration $\mathbf{b} = \{b_y\}_{y\in\mathcal{Y}}$ of actions at the decision nodes, we describe a vector $s^{\mathbf{b}} \in \{0,1\}^{|\mathcal{X}|+|\mathcal{Y}|+|\mathcal{Z}|}$ indexed by the nodes in the game as follows:

$$\begin{aligned}
&s^{\mathbf{b}}[x^{\mathsf{r}}] = 1\\
&s^{\mathbf{b}}[y] = s^{\mathbf{b}}[\rho_y[b_y]] \quad \forall y \in \mathcal{Y}\\
&s^{\mathbf{b}}[\rho_y[b]] = 0 \quad \forall y \in \mathcal{Y}, \forall b \in B_y \setminus \{b_y\}\\
&s^{\mathbf{b}}[x] = s^{\mathbf{b}}[\rho_x[a_x]] \quad \forall \mathbf{x} \in \mathcal{X}, \forall a_x \in A_x
\end{aligned}$$

Next, for a loss function $\mathbf{y}_t : \mathcal{Z} \to [-1,1]$ over the terminal nodes and a configuration $\mathbf{b}_t := \{b_{y,t}\}_{y\in\mathcal{Y}}$ of actions at the observation nodes, we describe a loss function $\widehat{\mathbf{y}}_t : \mathcal{X}\cup\mathcal{Y}\cup\mathcal{Z} \to [-1,1]$ as follows. For all $v \in \mathcal{X}\cup\mathcal{Y}$, we have $\widehat{\mathbf{y}}_t[v] := 0$. For all $z \in \mathcal{Z}$, we have $\widehat{\mathbf{y}}_t[z] = \mathbf{y}_t[z] \cdot \mathbb{1}[s^{\mathbf{b}_t}[z] = 1]$. It is easy to observe that $\mathbf{y}_t[z(\mathbf{a}_t, \mathbf{b}_t)] = \langle s^{\mathbf{a}_t}, \widehat{\mathbf{y}}_t\rangle$.

Let $\mathcal{T} = (\mathcal{V}, \mathcal{E})$ be the extensive-form game tree where $\mathcal{V} = \mathcal{X}\cup\mathcal{Y}\cup\mathcal{Z}$ and $\mathcal{E} = \{(x, \rho_x[a_x]) : \mathbf{x} \in \mathcal{X}, a_x \in A_x\} \cup \{(y, \rho_y[b_y]) : y \in \mathcal{Y}, b_y \in B_y\}$. Due to Lemma 32, we have $|\mathcal{X}| + |\mathcal{Y}| \leq |\mathcal{Z}|$. Hence, by using the EXP3 algorithm, we get an upper bound of $\mathcal{O}(\sqrt{|\mathcal{Z}|T\log(N)})$.

G.2. Additional Details on Reduction to DAG

In this section, we show that the DAG $G = (V, E)$ that we constructed during the reduction from extensive-form games has $\mathcal{O}(|\mathcal{Z}|)$ nodes. Let $\mathcal{T} = (\mathcal{V}, \mathcal{E})$ be the extensive-form game tree where $\mathcal{V} = \mathcal{X} \cup \mathcal{Y} \cup \mathcal{Z}$ and $\mathcal{E} = \{(x, \rho_x[a_x]) : \mathbf{x} \in \mathcal{X}, a_x \in A_x\} \cup \{(y, \rho_y[b_y]) : y \in \mathcal{Y}, b_y \in B_y\}$. Due to Theorem 32, we have $|\mathcal{X}| + |\mathcal{Y}| \leq |\mathcal{Z}|$. Hence, there are $\mathcal{O}(|\mathcal{Z}|)$ edges in $\mathcal{T}$.

Recall the construction of G. Each terminal node z is associated with the edge $(z_{\mathsf{s}}, z_{\mathsf{t}})$. Next each edge of the type $(x, \rho_x[a])$ is associated with the edges $(x_{\mathsf{s}}, u_{\mathsf{s}})$ and $(u_{\mathsf{t}}, x_{\mathsf{t}})$. Similarly, each edge of the type $(y, u^{(i)})$ is associated with the edges $(v_1, u_{\mathsf{s}}^{(i)})$ and $(u_{\mathsf{s}}^{(i)}, v_2)$, where $u^{(i)} = \rho_y[b]$ for some $b \in B_y$, v_1 is either y_{s} or $u_{\mathsf{s}}^{(i-1)}$, and v_2 is either y_{t} or $u_{\mathsf{s}}^{(i+1)}$. Hence, G has $\mathcal{O}(|\mathcal{Z}|)$ edges.