Parameter-Free Online Linear Optimization with Side Information via Universal Coin Betting

J. Jon Ryu
UC San Diego

Alankrita Bhatt
UC San Diego

Young-Han Kim
UC San Diego/Gauss Labs Inc.

Abstract

A class of parameter-free online linear optimization algorithms is proposed that harnesses the structure of an adversarial sequence by adapting to some side information. These algorithms combine the reduction technique of Orabona and Pál (2016) for adapting coin betting algorithms for online linear optimization with universal compression techniques in information theory for incorporating sequential side information to coin betting. Concrete examples are studied in which the side information has a tree structure and consists of quantized values of the previous symbols of the adversarial sequence, including fixed-order and variable-order Markov cases. By modifying the context-tree weighting technique of Willems, Shtarkov, and Tjalkens (1995), the proposed algorithm is further refined to achieve the best performance over all adaptive algorithms with tree-structured side information of a given maximum order in a computationally efficient manner.

1 INTRODUCTION

In this paper, we consider the problem of online linear optimization (OLO) in a Hilbert space V with norm $\|\cdot\|$. In each round $t = 1, 2, \ldots$, a learner picks an action $\mathbf{x}_t \in V$, receives a vector $\mathbf{g}_t \in V$ with $\|\mathbf{g}_t\| \le 1$, and suffers loss $\langle \mathbf{g}_t, \mathbf{x}_t \rangle$. In this repeated game, the goal of the learner is to keep her *cumulative regret* small with respect to any competitor $\mathbf{u}$ for any adversarial sequence $\mathbf{g}^T := \mathbf{g}_1, \ldots, \mathbf{g}_T$, where the cumulative regret is defined as the difference between the cumulative losses of the learner and $\mathbf{u} \in V$, i.e.,

$$\mathsf{Reg}_T(\mathbf{u}) := \mathsf{Reg}(\mathbf{u}; \mathbf{g}^T) := \sum_{t=1}^{T} \langle \mathbf{g}_t, \mathbf{x}_t \rangle - \sum_{t=1}^{T} \langle \mathbf{g}_t, \mathbf{u} \rangle.$$

Albeit simple in nature, an OLO algorithm serves as a versatile building block in machine learning algorithms (Shalev-Shwartz, 2011); for example, it can be used to solve online convex optimization.

While there exist standard algorithms such as online gradient descent (OGD) that achieve optimal regret of order $\mathsf{Reg}_T(\mathbf{u}) = O(\|\mathbf{u}\|\sqrt{T})$, these algorithms typically require tuning parameters with unknowns such as the norm $\|\mathbf{u}\|$ of a target competitor $\mathbf{u}$. For example, OGD with step size $\eta = 1/\sqrt{T}$ achieves $\mathsf{Reg}_T(\mathbf{u}) = O((1 + \|\mathbf{u}\|^2)\sqrt{T})$ for any $\mathbf{u} \in V$, while OGD with $\eta = U/\sqrt{T}$ achieves $\mathsf{Reg}_T(\mathbf{u}) = O(U\sqrt{T})$ for any $\mathbf{u} \in V$ such that $\|\mathbf{u}\| \le U$; see, e.g., (Shalev-Shwartz, 2011). To avoid tuning parameters, several *parameter-free* algorithms have been proposed in the last decade, aiming to achieve cumulative regret of order $\tilde{O}(\|\mathbf{u}\|\sqrt{T})$ for any $\mathbf{u} \in V$ without knowing $\|\mathbf{u}\|$ a priori (Orabona, 2013; McMahan and Abernethy, 2013; Orabona, 2014; McMahan and Orabona, 2014; Orabona and Pál, 2016), where $\tilde{O}(\cdot)$ hides any polylogarithmic factor in the big O notation; the extra polylogarithimic factor is known to be necessary (Orabona, 2013; McMahan and Abernethy, 2013).

While these optimality guarantees on regret seem sufficient, they may not be satisfactory in bounding the incurred loss of the algorithm, due to the limited power of the class of static competitors $\mathbf{u}$ as a benchmark. For example, consider the adversarial sequence $\mathbf{g}, -\mathbf{g}, \mathbf{g}, -\mathbf{g}, \ldots$ for a fixed vector $\mathbf{g} \in \mathbb{B} := \{\mathbf{x} \in V : \|\mathbf{x}\| \le 1\}$. Despite the apparent structure (or predictability) in the sequence, the best achievable reward of any static competitor $\mathbf{u} \in V$ is zero for any even T. In general, the cumulative loss of a static competitor $\mathbf{u}$ is $\sum_{t=1}^{T} \langle \mathbf{g}_t, \mathbf{u} \rangle = \langle \sum_{t=1}^{T} \mathbf{g}_t, \mathbf{u} \rangle$, and can be large if and only if the norm $\|\sum_{t=1}^{T} \mathbf{g}_t\|$ is large, or equivalently, when $\mathbf{g}_1, \ldots, \mathbf{g}_T$ are well *aligned*. It is not only a theoretical issue, since, for example, when we consider a

Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151.

practical scenario such as weather forecasting, the sequence $(\mathbf{g}_t)$ may have such a *temporal structure* that can be exploited in optimization, rather than being completely adversarial.

One remedy for this issue is to consider a larger class of competitors, which may *adapt* to the history $\mathbf{g}^{t-1} := \mathbf{g}_1, \dots, \mathbf{g}_{t-1}$. Hereafter, we use x_t^s to denote the sequence $x_t, \dots, x_s$ for $t \le s$ and $x^t := x_1^t$ by convention. For instance, in the previous example, consider a competitor which can play two different actions $\mathbf{u}_{+1}$ and $\mathbf{u}_{-1}$ based on the quantization $Q(\mathbf{g}_{t-1}) = \mathrm{sgn}(\langle \mathbf{f}, \mathbf{g}_{t-1} \rangle)$ for some fixed $\mathbf{f} \in V$; for example, we chose standard vectors $\mathbf{e}_i$ for a Euclidean space V in our experiments; see Section 4. Then the best loss achieved by the competitor class on this sequence becomes $-(T/2)\|\mathbf{g}\|(\|\mathbf{u}_{+1}\| + \|\mathbf{u}_{-1}\|)$, which could be much smaller than 0. We remark that, from the view of binary prediction, this example can be thought of a first-order Markov prediction, which takes only the previous time step into consideration. Hence, it is natural to consider a k-th order extension of the previous example, i.e., a competitor that adapts to the length-k sequence $Q(\mathbf{g}_{t-k}^{t-1}) := Q(\mathbf{g}_{t-k}) \dots Q(\mathbf{g}_{t-1}) \in \{1, \bar{1}\}^k$, where we define $\bar{1} := -1$.

We can even further sophisticate a competitor's dependence structure by allowing it to adapt to a *tree structure* (also known as a *variable-order Markov structure*) of the quantization sequence, which is widely deployed structure in sequence prediction; see, e.g., (Begleiter et al., 2004). For example, for the depth-2 quantization sequence $Q(\mathbf{g}_{t-2}^{t-1})$, rather than adapting to the all four possible states, a competitor may adapt to the suffix falls into a set of suffixes $\mathbf{T} = \{*1, 1\bar{1}, \bar{1}\bar{1}\}$ of one fewer states; here, $*$ denotes that any symbol from $\{1, \bar{1}\}$ is possible in that position. As depicted in Figure 1 for $\mathbf{T}$, in general, a suffix set has a one-to-one correspondence between a full binary tree, and is thus often identified as a tree; see Section 3.3.2 for the formal definition and further justification of the tree side information.

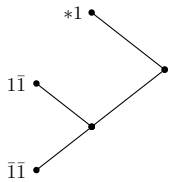

Figure 1: $\mathbf{T} = \{*1, 1\bar{1}, \bar{1}\bar{1}\}$.

Since we do not know a priori which tree structure is best to adapt to, we ultimately aim to design an OLO algorithm that achieves the performance of the best tree competitor of given maximum depth $D \ge 1$. Since there are $O(2^{2^D})$ possible trees of depth at most D, it becomes challenging even for a moderate size of D. We remark that the problem of following the best tree structure in hindsight, the *tree problem* in short, is a classical problem which has been studied in multiple areas such as information theory (Willems et al., 1995) and online learning (Freund et al., 1997), but an application of this framework to the OLO problem has not been considered in the literature.

To address this problem, we combine two technical components from online learning and information theory. Namely, we apply an information theoretic technique of following the best tree structure for universal compression, called the *context tree weighting* (CTW) algorithm invented by Willems et al. (1995), to generalize a parameter-free OLO algorithm called the *KT OLO algorithm* proposed by Orabona and Pál (2016), which is designed based on universal coin betting. Consequently, as the main result, we propose the *CTW OLO algorithm* that efficiently solves the problem with only $O(D)$ updates per round achieving nearly minimax optimal regret; see Section 3.3.

We motivate the proposed approach by solving two intermediate, abstract OLO problems, the one with (single) side information (Section 3.1) and the other with multiple side information (Section 3.2), and propose information theoretic OLO algorithms (i.e., product KT and mixture KT) respectively, which might be of independent interest. We remark, however, that it is not hard to convert any parameter-free algorithm to solve the abstract problems with same guarantees and complexity of the proposed solutions, using existing meta techniques such as a black-box aggregation scheme by Cutkosky (2019) with per-state extension of a base OLO algorithm; hence, the contribution of the intermediate solutions is rather purely of intellectual merit.

In Section 4, we experimentally demonstrate the power of the CTW OLO algorithm with real-world temporal datasets. We conclude with some remarks in Section 5. All proofs and discussion with related work are deferred to Appendix due to the space constraint.

Notation Given a tuple $\mathbf{a} = (a_1, \dots, a_m)$, we use $\sum \mathbf{a} := \sum_{i=1}^m a_i$ to denote the sum of all entries in a tuple $\mathbf{a}$. For example, we write $\sum g^{t-1}$ to denote the sum of $g_1, \dots, g_{t-1}$ by identifying g^{t-1} as a tuple $(g_1, \dots, g_{t-1})$. For the empty tuple $()$, we define $\sum() := 0$ by convention. We use $|\mathbf{a}|$ to denote the number of entries of a tuple $\mathbf{a}$. For a tuple of vectors $\mathbf{u}_{1:S} := (\mathbf{u}_1, \dots, \mathbf{u}_S) \in V \times \dots \times V$, we use $\|\mathbf{u}\|_{1:S} := (\|\mathbf{u}_1\|, \dots, \|\mathbf{u}_S\|) \in \mathbb{R}^S_{\ge 0}$ to denote the tuple of norms of each entry.

2 PRELIMINARIES

We review the coin betting based OLO algorithm of Orabona and Pál (2016). From this point, we will describe all algorithms in the reward maximization framework, which is philosophically consistent with

the goal of gambling, to avoid any confusion, but we will keep using the conventional naming OGD even though it is actually gradient *ascent*.[1]

2.1 Continuous Coin Betting and 1D OLO

Consider the following repeated gambling. Starting with an initial wealth W_0, at each round t, a player picks a *signed relative bet* $b_t \in [-1, 1]$. At the end of the round, a real number $g_t \in [-1, 1]$ is revealed as an outcome of the "continuous coin toss" and the player gains the reward $g_t b_t \mathsf{W}_{t-1}$. This game leads to the cumulative wealth

$$\mathsf{W}_t(g^t) = \mathsf{W}_0 \prod_{i=1}^{t} (1 + g_i b_i).$$

When $g_t \in \{\pm 1\}$, this game boils down to the standard coin betting, where the player splits her wealth into $\frac{1+b_t}{2}\mathsf{W}_{t-1}$ and $\frac{1-b_t}{2}\mathsf{W}_{t-1}$, and bets the amounts on the binary outcomes $+1$ and -1, respectively. It is well known that the standard coin betting game is equivalent to the binary compression, or binary log-loss prediction, which have been extensively studied in information theory; see, e.g., (Cover and Thomas, 2006, Chapter 6).

Even when the outcomes g_t are allowed to take continuous values, many interesting connections remain to hold. For example, the Krichevsky and Trofimov (1981)'s (KT) probability assignment, which is competitive against i.i.d. Bernoulli models, can be translated into a betting strategy

$$b^{\mathsf{KT}}(g^{t-1}) := b_t^{\mathsf{KT}}(\textstyle\sum g^{t-1}),$$

where $b_t^{\mathsf{KT}}(x) := \frac{x}{t}$ for $x \in [-t+1, t-1]$. As a natural continuous extension of the KT probability assignment, we define the *KT coin betting potential*

$$\psi^{\mathsf{KT}}(g^t) := \psi_t^{\mathsf{KT}}(\textstyle\sum g^t) := 2^t \tilde{q}_t^{\mathsf{KT}}(\sum g^t),$$

where

$$\tilde{q}_t^{\mathsf{KT}}(x) := B\Big(\frac{t+x+1}{2}, \frac{t-x+1}{2}\Big) \Big/ B\Big(\frac{1}{2}, \frac{1}{2}\Big)$$

for $x \in [-t, t]$ and $B(x, y) := \Gamma(x)\Gamma(y)/\Gamma(x+y)$ and $\Gamma(x)$ denote the Beta function and Gamma function, respectively. We remark that the interpolation for continuous values is naturally defined via the Gamma functions. This simple KT betting scheme guarantees that the cumulative wealth satisfies

$$\mathsf{W}_T(g^T) \geq \mathsf{W}_0 \psi^{\mathsf{KT}}(g^T) = \mathsf{W}_0 2^T \tilde{q}_T^{\mathsf{KT}}(\textstyle\sum g^T) \qquad (2.1)$$

for any $T \geq 1$ and $g_1, \ldots, g_T \in [-1, 1]$; see the proof of Theorem 2.1 in Appendix. It can be easily shown that the wealth lower bound is near-optimal when compared to the best static bettor $b_t = b$ for some fixed $b \in [-1, 1]$ in hindsight, the so-called Kelly betting (Kelly Jr., 1956). This follows as a simple consequence of the fact that the KT probability assignment is a near-optimal probability assignment for universal compression of i.i.d. sequences. In this paper, going forward the interpretation of the coin betting potential as probability assignment in the parlance of compression will prove useful.

In their insightful work, Orabona and Pál (2016) demonstrated that the universal continuous coin betting algorithm can be directly translated to an OLO algorithm with a parameter-free guarantee. By defining an *absolute betting* $w_t := b_t \mathsf{W}_{t-1}$, we can write the cumulative wealth in an additive form

$$\mathsf{W}_t(g^t) = \mathsf{W}_0 + \sum_{i=1}^{t} g_t w_t,$$

whence we interpret $\sum_{i=1}^{t} g_i w_i$ as the cumulative reward in the 1D OLO with $g_1, \ldots, g_t \in [-1, 1]$. Now, if we define the KT coin betting OLO algorithm by the action

$$w_t^{\mathsf{KT}} := w^{\mathsf{KT}}(g^{t-1}) = b^{\mathsf{KT}}(g^{t-1}) \mathsf{W}_{t-1}(g^{t-1}),$$

then the "universal" wealth lower bound (2.1) with respect to any g^T can be translated to establish a "parameter-free" bound on the 1D regret

$$\mathsf{Reg}(u; g^T) := \sum_{t=1}^{T} g_t u - \sum_{t=1}^{T} g_t w_t^{\mathsf{KT}},$$

against static competitors $u \in \mathbb{R}$. Let $(\psi_T^{\mathsf{KT}})^\star \colon \mathbb{R} \to \mathbb{R}$ denote the Fenchel dual of the potential function $\psi_T^{\mathsf{KT}} \colon \mathbb{R} \to \mathbb{R}$, i.e.,

$$(\psi_T^{\mathsf{KT}})^\star(u) := \sup_{g \in \mathbb{R}} (gu - \psi_T^{\mathsf{KT}}(g)).$$

Theorem 2.1. *For any $g_1, \ldots, g_T \in [-1, 1]$, the 1D OLO algorithm $w_t^{\mathsf{KT}} = b^{\mathsf{KT}}(g^{t-1}) \mathsf{W}_{t-1}$ satisfies*

$$\sup_{u \in \mathbb{R}} \Big\{ \mathsf{Reg}(u; g^T) - \mathsf{W}_0 (\psi_T^{\mathsf{KT}})^\star\Big(\frac{u}{\mathsf{W}_0}\Big) \Big\} \leq \mathsf{W}_0.$$

In particular, for any $u \in \mathbb{R}$, we have

$$\mathsf{Reg}(u; g^T) \leq \sqrt{T u^2 \ln(T u^2 / (e\sqrt{\pi} \mathsf{W}_0^2) + 1)} + \mathsf{W}_0.$$

2.2 Reduction of OLO over a Hilbert Space to Continuous Coin Betting

This reduction can be extended for OLO over a Hilbert space V with norm $\|\cdot\|$, where we wish to maximize the

[1] Note that one can translate a reward maximization algorithm to an equivalent loss minimization algorithm by feeding $-\mathbf{g}_t$ instead of $\mathbf{g}_t$, and vice versa.

cumulative reward $\sum_{t=1}^{T}\langle \mathbf{g}_t, \mathbf{x}_t\rangle$ for $\mathbf{g}_1, \dots, \mathbf{g}_T \in \mathbb{B} := \{\mathbf{x} \in V \colon \|\mathbf{x}\| \le 1\}$. Orabona and Pál (2016) proposed the following OLO algorithm over Hilbert space based on the continuous coin betting. For an initial wealth $\mathsf{W}_0 > 0$, we define the *cumulative wealth*

$$\mathsf{W}_T(\mathbf{g}^T) := \mathsf{W}_0 + \sum_{t=1}^{T}\langle \mathbf{g}_t, \mathbf{x}_t\rangle$$

as the cumulative reward plus the initial wealth, analogously to the coin betting. If we define the *vectorial betting* given $\mathbf{g}^{t-1}$ as

$$\mathbf{v}^{\mathsf{KT}}(\mathbf{g}^{t-1}) := b_t^{\mathsf{KT}}(\|\textstyle\sum \mathbf{g}^{t-1}\|)\dfrac{\sum \mathbf{g}^{t-1}}{\|\sum \mathbf{g}^{t-1}\|} = \dfrac{1}{t}\sum \mathbf{g}^{t-1}$$

and define a *potential* function

$$\Psi^{\mathsf{KT}}(\mathbf{g}^t) := \psi_t^{\mathsf{KT}}(\|\textstyle\sum \mathbf{g}^t\|) = 2^t \tilde{q}_t^{\mathsf{KT}}(\|\sum \mathbf{g}^t\|),$$

then the corresponding OLO algorithm ensures the wealth lower bound $\mathsf{W}_t(\mathbf{g}^t) \ge \mathsf{W}_0\Psi^{\mathsf{KT}}(\mathbf{g}^t)$, and thus the corresponding regret upper bound in the same spirit of Theorem 2.1.

Theorem 2.2 (Orabona and Pál, 2016, Theorem 3)**.** *For any $\mathbf{g}_1, \dots, \mathbf{g}_T \in \mathbb{B}$, the OLO algorithm $\mathbf{w}_t^{\mathsf{KT}} = \mathbf{v}^{\mathsf{KT}}(\mathbf{g}^{t-1})\mathsf{W}_{t-1}$ based on the coin betting satisfies $\mathsf{W}_T \ge \mathsf{W}_0\Psi^{\mathsf{KT}}(\mathbf{g}^T)$, and moreover*

$$\sup_{\mathbf{u}\in V}\Big\{\mathsf{Reg}(\mathbf{u}; \mathbf{g}^T) - \mathsf{W}_0(\psi_T^{\mathsf{KT}})^\star\Big(\frac{\|\mathbf{u}\|}{\mathsf{W}_0}\Big)\Big\} \le \mathsf{W}_0.$$

In particular, for any $\mathbf{u} \in V$, we have

$$\mathsf{Reg}(\mathbf{u}; \mathbf{g}^T) \le \sqrt{T\|\mathbf{u}\|^2 \ln(T\|\mathbf{u}\|^2/(e\sqrt{\pi}\mathsf{W}_0^2) + 1)} + \mathsf{W}_0.$$

3 MAIN RESULTS

In what follows, we will illustrate how to incorporate (multiple) sequential side information based on coin betting algorithms in OLO over Hilbert space with an analogous guarantee by extending the aforementioned algorithmic reduction and guarantee translation. In doing so, we will leverage the connection between coin betting and compression, and adopt universal compression techniques beyond the KT strategy, namely per-state adaptation (Section 3.1), mixture (Section 3.2), and context tree weighting techniques (Section 3.3.2). For each case, we will first define a potential function and introduce a corresponding vectorial betting which guarantees the cumulative wealth to be at least the desired potential function.

3.1 OLO with Single Side Information via Product Potential

We consider the scenario when a (discrete) side information $H = (h_t \in [S])_{t\ge 1}$ is sequentially available for some $S \ge 1$. That is, at each round t, the side information h_t is revealed before the plays. As motivated in the introduction, the canonical example is a *causal* side information based on the history $\mathbf{g}^{t-1}$ such as a quantization of $\mathbf{g}_{t-D}^{t-1}$ for some $D \ge 1$. Yet another example is side information given by an oracle with foresight such as $h_t = \mathrm{sgn}(\langle \mathbf{g}_t, \mathbf{f}\rangle)$, i.e., the sign of the correlation between a fixed vector $\mathbf{f} \in V$ and the incoming symbol $\mathbf{g}_t$, as a rough hint to the future.

We define an *adaptive competitor with respect to the side information H*, denoted as $\mathbf{u}_{1:S}[H]$ for an S-tuple $\mathbf{u}_{1:S} := (\mathbf{u}_1, \dots, \mathbf{u}_S) \in V \times \cdots \times V$, to play $\mathbf{u}_{h_t}$ at time t, and let $\mathcal{C}[H] := \{\mathbf{u}_{1:S}[H] \colon \mathbf{u}_{1:S} \in V \times \cdots \times V\}$ denote the collection of all such adaptive competitors.

We first observe that the cumulative loss incurred by an adaptive competitor $\mathbf{u}_{1:S}[H] \in \mathcal{C}[H]$ can be decomposed with respect to the *states* defined by the side information symbols, i.e.,

$$\sum_{t=1}^{T}\langle \mathbf{g}_t, \mathbf{u}_{h_t}\rangle = \sum_{s=1}^{S}\Big\langle \sum_{t\in[T]:h_t=s} \mathbf{g}_t, \mathbf{u}_s\Big\rangle.$$

Hence, a naive solution is to run independent OGD algorithms for each subsequence $\mathbf{g}^t(s; h^t) := (\mathbf{g}_i \colon h_i = s, i \in [t])$ sharing the same side information $s \in [S]$; it is straightforward to show that the per-state OGD with optimal learning rates achieves the regret of order $O(\sum_{s=1}^{S}\|\mathbf{u}_s\|\sqrt{T_s})$ with knowing the competitor norms $\|\mathbf{u}\|_{1:S}$. Like the per-state OGD algorithm, we can also extend other parameter-free algorithms such as DFEG (Orabona, 2013) and AdaNormal (McMahan and Orabona, 2014) to adapt to side information; see Appendix B. This is what we call the *per-state extension* of an OLO algorithm.

Here, we propose a different type of parameter-free per-state algorithm based on coin betting. To compete against any adaptive competitor from $\mathcal{C}[H]$, we define a *product KT potential function*

$$\begin{aligned}\Psi^{\mathsf{KT}}(\mathbf{g}^t; h^t) &:= \prod_{s\in[S]} \Psi^{\mathsf{KT}}(\mathbf{g}^t(s; h^t)) \\ &= \prod_{s\in[S]} \psi_{t_s}^{\mathsf{KT}}(\|\textstyle\sum \mathbf{g}^t(s; h^t)\|),\end{aligned}$$

where $t_s := |\mathbf{g}^t(s; h^t)|$ for each $s \in [S]$. Note that $\Psi^{\mathsf{KT}}(\mathbf{g}^t; h^t)$ is a function of the summations of the subsequences $(\sum \mathbf{g}^t(1; h^t), \dots, \sum \mathbf{g}^t(S; h^t))$. For each time t, we then define the vectorial KT betting with side information h^t as the application of the vectorial KT betting onto the subsequence corresponding to the current side information symbol h_t, i.e.,

$$\mathbf{v}^{\mathsf{KT}}(\mathbf{g}^{t-1}; h^t) := \mathbf{v}^{\mathsf{KT}}(\mathbf{g}^{t-1}(h_t; h^{t-1})).$$

Unlike the other per-state extensions which play independent actions for each state thus allowing straightforward analyses, the per-state KT actions

$$\mathbf{w}_t^{\mathsf{KT}}(\mathbf{g}^{t-1}; h^t) = \mathbf{v}^{\mathsf{KT}}(\mathbf{g}^{t-1}; h^t)\mathsf{W}_{t-1} \tag{3.1}$$

depend on all previous history $\mathbf{g}^{t-1}$ due to the wealth factor W_{t-1}. We can establish the following guarantee with the same line of argument in the proof of Theorem 2.1, by analyzing the Fenchel dual of $\Psi^{\mathsf{KT}}(\mathbf{g}^t; h^t)$. Recall that for a multivariate function $\Psi\colon \mathbb{R}^d \to \mathbb{R}$, its Fenchel dual $\Psi^\star\colon \mathbb{R}^d \to \mathbb{R}$ is defined as

$$\Psi^\star(\mathbf{y}) := \sup_{\mathbf{x}\in\mathbb{R}^d} (\mathbf{y}^T\mathbf{x} - \Psi(\mathbf{x})).$$

Theorem 3.1. *For any side information $H = (h_t \in [S])_{t\ge1}$ and any $\mathbf{g}_1, \dots, \mathbf{g}_T \in \mathbb{B}$, let $\phi^{\mathsf{KT}}_{T_{1:S}}\colon \mathbb{R}^S \to \mathbb{R}$ be the Fenchel dual of the function*

$$(f_1, \dots, f_S) \mapsto \prod_{s\in[S]} \psi^{\mathsf{KT}}_{T_s}(f_s),$$

where $T_s := |\{t \in [T]\colon h_t = s\}|$. Then, the OLO algorithm $\mathbf{w}_t^{\mathsf{KT}}(\mathbf{g}^{t-1}; h^t) := \mathbf{v}^{\mathsf{KT}}(\mathbf{g}^{t-1}; h^t)\mathsf{W}_{t-1}$ satisfies $\mathsf{W}_T \ge \mathsf{W}_0\Psi^{\mathsf{KT}}(\mathbf{g}^T; h^T)$, and moreover

$$\sup_{\mathbf{u}_{1:S}}\left\{\mathsf{Reg}(\mathbf{u}_{1:S}[H]; \mathbf{g}^T) - \mathsf{W}_0\phi^{\mathsf{KT}}_{T_{1:S}}\left(\frac{\|\mathbf{u}\|_{1:S}}{\mathsf{W}_0}\right)\right\} \le \mathsf{W}_0.$$

In particular, for any $\mathbf{u}_{1:S}[H] \in \mathcal{C}[H]$,

$$\mathsf{Reg}(\mathbf{u}_{1:S}[H]; \mathbf{g}^T) = \mathsf{W}_0 + \tilde{O}\left(\sqrt{\sum_{s=1}^S T_s\|\mathbf{u}_s\|^2}\right). \tag{3.2}$$

Example 3.1. Recall the "easy" adversarial sequence $\mathbf{g}^T = (\mathbf{g}, -\mathbf{g}, \mathbf{g}, \dots, -\mathbf{g})$ for some $\mathbf{g} \in \mathbb{B}$ previously considered in the introduction. For a side information $h_t = \mathrm{sgn}(\langle \mathbf{g}_t, \mathbf{f}\rangle)$ with some $\mathbf{f} \in V$, Theorem 3.1 states that $\mathsf{Reg}((\mathbf{u}_+, \mathbf{u}_-); \mathbf{g}^T) = \tilde{O}((\|\mathbf{u}_+\| + \|\mathbf{u}_-\|)\sqrt{T})$, matching the regret guarantee of the optimally tuned per-state OGD up to logarithmic factors. Overall, the regret guarantee against adaptive competitors for the per-state KT method implies a much larger overall reward than was achieved by an algorithm competing against static competitors.

Remark 3.1 (Cost of noninformative side information). Consider a scenario where competitors of the form $\mathbf{u}_{1:S} = (\mathbf{u}, \dots, \mathbf{u})$ with some vector $\mathbf{u} \in V$ perform best; in this case, an algorithm without adapting to side information may suffice for optimal regret guarantees. Even in such cases with *noninformative* side information, the dominant factor in the regret remains the same as the regret guarantee with respect to the static competitor class, since $\sum_{s=1}^S T_s\|\mathbf{u}_s\|^2 = T\|\mathbf{u}\|^2$.

Remark 3.2 (Effect of large S). While side information with larger S may provide more levels of granularity, too large S may degrade the performance of the per-state algorithms. Intuitively, if $S \gg 1$, it is likely that we will see each state only few times, which results in poor convergence for almost every state. These are also captured in the regret guarantee; we note that the hidden logarithmic factor of the regret bound (3.2) might incur a multiplicative factor of at most $O(\sqrt{S})$. Similarly, in the optimal regret attained by the per-state OGD, we have $O(\sum_{s=1}^S \|\mathbf{u}_s\|\sqrt{T_s}) \le O(\max_{s\in[S]}\|\mathbf{u}_s\|\sqrt{ST})$.

3.2 OLO with Multiple Side Information via Mixture of Product Potentials

Now suppose that multiple side information sequences $\{H^{(m)} = (h_t^{(m)} \in S^{(m)})_{t\ge1}\colon m \in [M]\}$ are sequentially available; for example, each $H^{(m)}$ can be either constructed based on a different quantizer $Q_m\colon V \to \{1, \bar{1}\}$ and/or based on the history $\mathbf{g}_{t-D_m}^{t-1}$ of different lengths $D_m \ge 0$, each of which aims to capture a different structure of $(\mathbf{g}_t)$. In this setting, we aim to minimize the *worst* regret among all possible side information, i.e.,

$$\begin{aligned}&\max_{m\in[M]} \mathsf{Reg}(\mathbf{u}_{1:S^{(m)}}[H_m]; \mathbf{g}^T)\\ &\quad= \sum_{t=1}^T \langle \mathbf{g}_t, \mathbf{w}_t\rangle - \min_{m\in[M]} \sum_{t=1}^T \langle \mathbf{g}_t, \mathbf{u}^{(H)}_{h_{mt}}\rangle,\end{aligned} \tag{3.3}$$

which is equivalent to aiming to follow the best side information in hindsight.

We first remark that Cutkosky (2019) recently proposed a simple black-box meta algorithm that combines multiple OLO algorithms achieving the best regret guarantee, which can also be applied to solving this multiple side information problem. For example, for algorithms $(\mathcal{A}_m)_{m\in[M]}$ each of which play an action $\mathbf{w}_t^{(m)}$, the meta algorithm $\mathcal{A}$ which we refer to the *addition* plays $\mathbf{w}_t = \sum_{m=1}^M \mathbf{w}_t^{(m)}$ and guarantees the regret

$$\mathsf{Reg}_T^{\mathcal{A}}(\mathbf{u}) \le \varepsilon + \min_{m\in[M]} \mathsf{Reg}_T^{\mathcal{A}_m}(\mathbf{u}),$$

provided that $\mathcal{A}_m$'s suffer at most constant regret ε against $\mathbf{u} = 0$; the same guarantee also hold for adaptive competitors.

Rather, we propose the following information theoretic solution. For each side information sequence $H^{(m)}$, we can apply the per-state KT algorithm from the previous section, which guarantees the wealth lower bound $\mathsf{W}_0\Psi^{\mathsf{KT}}(\mathbf{g}^t; (h^{(m)})^t)$. To achieve the best among the per-state KT algorithms, we consider the *mixture potential*

$$\Psi^{\mathsf{mix}}(\mathbf{g}^t; \mathbf{h}^t) = \sum_{m=1}^M w_m \Psi^{\mathsf{KT}}(\mathbf{g}^t; (h^{(m)})^t)$$

for some $w_1, \dots, w_M > 0$ such that $\sum_{m=1}^M w_m = 1$. Here, $\mathbf{h}_t := (h_t^{(1)}, \dots, h_t^{(M)})$ denotes the side information vector revealed at time t. When there exists no prior belief on how useful each side information is, one can choose the uniform weight $w_1 = \dots = w_M = 1/M$ by default. Now, define the *vectorial mixture betting* given $\mathbf{g}^{t-1}$ and $\mathbf{h}^t$ as

$$\mathbf{v}^{\mathsf{mix}}(\mathbf{g}^{t-1}; \mathbf{h}^t) := \frac{\mathbf{u}^{\mathsf{mix}}(\mathbf{g}^{t-1}; \mathbf{h}^t)}{\Psi^{\mathsf{mix}}(\mathbf{g}^{t-1}; \mathbf{h}^{t-1})}, \quad \text{where}$$

$$\mathbf{u}^{\mathsf{mix}}(\mathbf{g}^{t-1}; \mathbf{h}^t) := \sum_{m=1}^M w_m \Psi^{\mathsf{KT}}(\mathbf{g}^{t-1}; (h^{(m)})^{t-1}) \mathbf{v}^{\mathsf{KT}}(\mathbf{g}^{t-1}; (h^{(m)})^t),$$

and finally define the *mixture OLO* algorithm by the action

$$\mathbf{w}_t^{\mathsf{mix}}(\mathbf{g}^{t-1}; \mathbf{h}^t) := \mathbf{v}^{\mathsf{mix}}(\mathbf{g}^{t-1}; \mathbf{h}^t) \mathsf{W}_{t-1}. \tag{3.4}$$

In the language of gambling, the mixture strategy bets by distributing her wealth based on the weights w_m's to strategies, each of which is tailored to a side information sequence, and thus can guarantee at least w_m times the cumulative wealth attained by the m-th strategy following $H^{(m)}$ for any $m \in [M]$.

Theorem 3.2. *For any side information $H^{(1)}, \dots, H^{(M)}$ and any $\mathbf{g}_1, \dots, \mathbf{g}_T \in \mathbb{B}$, the mixture OLO algorithm* (3.4) *satisfies $\mathsf{W}_T \geq \mathsf{W}_0 \Psi^{\mathsf{mix}}(\mathbf{g}^T; \mathbf{h}^T)$, and moreover for any $m \in [M]$, we have*

$$\sup_{\mathbf{u}_{1:S^{(m)}}} \Big\{ \mathsf{Reg}(\mathbf{u}_{1:S^{(m)}}[H^{(m)}]); \mathbf{g}^T) - w_m \mathsf{W}_0 \phi^{\mathsf{KT}}_{T_{1:S^{(m)}}} \Big(\frac{\|\mathbf{u}\|_{1:S^{(m)}}}{w_m \mathsf{W}_0} \Big) \Big\} \leq w_m \mathsf{W}_0.$$

In other words, for any m and any $\mathbf{u}_{1:S^{(m)}}$, we have

$$\mathsf{Reg}(\mathbf{u}_{1:S^{(m)}}[H_m]; \mathbf{g}^T) = w_m \mathsf{W}_0 + \tilde{O}\left(\sqrt{ \Big(\ln \frac{1}{w_m} \Big) \sum_{s_m=1}^{S_m} T_{s_m}^{(H_m)} \|\mathbf{u}_{s_m}^{(H_m)}\|^2 } \right).$$

Remark 3.3 (Cost of mixture). A mixture strategy adapts to any available side information with the cost of replacing W_0 with $w_m \mathsf{W}_0$ in the regret guarantee for each $m \in [M]$. Since the dependence of regret on W_0 scales as $O(\sqrt{\ln(1 + 1/\mathsf{W}_0)} + \mathsf{W}_0)$ from Theorem 3.1, a small w_m may degrade the quality of the regret guarantee by only a small multiplicative factor $O(\sqrt{\ln(1/w_m)})$.

Remark 3.4 (Comparison to the addition technique). While the mixture algorithm attains a similar guarantee to the addition technique (Cutkosky, 2019), it is only applicable to coin betting based algorithms and requires a rather sophisticated aggregation step. Thus, if there are only moderate number of side information sequences, the addition of per-state parameter-free algorithms suffices. The merit of mixture will become clear in the next section in the tree side information problem of combining $O(2^{2^D})$ many components for a depth parameter $D \geq 1$, while a naive application of the addition technique to the tree problem is not feasible due to the number of side information; see Section 5 for an alternative solution with the addition technique.

3.3 OLO with Tree Side Information

In this section, we formally define and study a tree-structured side information H, which was illustrated in the introduction. We suppose that there exists an auxiliary binary sequence $\Omega = (\omega_t \in \{\pm 1\})_{t \geq 1}$, which is revealed one-by-one at the *end* of each round; hence, a learner has access to ω^{t-1} when deciding an action at round t. In the motivating problem in the introduction, such an auxiliary sequence was constructed as $\omega_t := Q(\mathbf{g}_t)$ with a fixed binary quantizer $Q \colon V \to \{\pm 1\}$.

3.3.1 Markov Side Information

Given $\Omega = (\omega_t)_{t \geq 1}$, the most natural form of side information is the *depth-D Markov side information* $h_t := \omega_{t-D}^{t-1} \in \{\pm 1\}^D$, i.e., the last D bits of $(\omega_t)_{t \geq 1}$—note that it can be mapped into a perfect binary tree of depth D with 2^D possible states.

Example 3.2. As an illustrative application of the mixture algorithm and a precursor to the tree side information problem, suppose that we wish to compete with any Markov side information of depth $\leq D$. Then, there are $D+1$ different side information, one for each depth $d = 0, \dots, D$; for simplicity, assume uniform weights $w_d = 1/(D+1)$ for each depth d. Then, Theorem 3.2 guarantees that the mixture OLO algorithm (3.4) satisfies, for any depth $d = 0, \dots, D$,

$$\mathsf{Reg}(\mathbf{u}_{1:2^d}^{(d)}; \mathbf{g}^T) = \frac{\mathsf{W}_0}{D+1} + \tilde{O}\left(\sqrt{ \ln(D+1) \sum_{s=1}^{2^d} T_s^{(d)} \|\mathbf{u}_s^{(d)}\|^2 } \right)$$

for any competitor $\mathbf{u}_{1:2^d}^{(d)} \in V^{2^d}$, where we identify 2^d possible states by $1, \dots, 2^d$ and $T_s^{(d)}$ is the number of time steps with s as side information.

While a larger D can capture a longer dependence in the sequence, however, the performance of a per-state algorithm could significantly degrade due to the exponential number of states as pointed out in Remark 3.2.

3.3.2 Tree-Structured Side Information

The limitation of Markov side information motivates a general *tree-structured side information* (or tree side information in short). Informally, we say that a sequence has a *depth-D tree structure* if the state at time t depends on at most D of the previous occurrences, corresponding to a full binary tree of depth D; see Figure 1. This degree of freedom allows to consider different lengths of history for each state, leading to the terminology *variable-order Markov structure*, as opposed to the previous *fixed-order Markov structure*. If an underlying structure is approximately captured by a tree structure of depth D with the number of leaves far fewer than 2^D, the corresponding per-state algorithm can enjoy a much lower regret guarantee.

We now formally define a tree side information. We say that a string $\omega_{1-l}\omega_{2-l}\dots\omega_0$ is a *suffix* of a string $\omega'_{1-l'}\omega'_{2-l'}\dots\omega'_0$, if $l \le l'$ and $\omega_{-i} = \omega'_{-i}$ for all $i \in \{0,\dots,l-1\}$. Let λ denote the empty string. We define a *(binary) suffix set* $\mathbf{T}$ as a set of binary strings that satisfies the following two properties (Willems et al., 1995): (1) Properness: no string in $\mathbf{T}$ is a suffix of any other string in $\mathbf{T}$; (2) Completeness: every semi-infinite binary string $\dots h_{t-2}h_{t-1}h_t$ has a suffix from $\mathbf{T}$. Since there exists an one-to-one correspondence between a binary suffix set and a full binary tree, we also call $\mathbf{T}$ a *suffix tree*. Given $D \ge 0$, let $\mathscr{T}_{\le D}$ denote the set of all suffix trees of depth at most D.

For a suffix tree $\mathbf{T} \in \mathscr{T}_{\le D}$, we define a *tree side information* $H_{\mathbf{T};\Omega}$ with respect to $\mathbf{T}$ and $\Omega = (\omega_t)_{t\ge 1}$ as the matching suffix from the auxiliary sequence. We can also identify h_t, the tree side information defined by $\mathbf{T}$ at time t, with a unique leaf node $s_t^{\mathbf{T}} \in \mathbf{T}$. For example, if a suffix set $\mathbf{T}$ consists of all possible 2^D binary strings of length $D \ge 1$, then it boils down to the fixed-order Markov case $h_t = \omega_{t-D}^{t-1}$.

For a single tree $\mathbf{T}$, the goal is to keep the regret

$$\mathsf{Reg}(\mathbf{u}[\mathbf{T}];\mathbf{g}^T) := \sum_{t=1}^{T} \langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{u}^{\mathbf{T}}_{s_t^{\mathbf{T}}} \rangle$$

small for any competitor $\mathbf{u}[\mathbf{T}] := (\mathbf{u}_s^{\mathbf{T}})_{s\in\mathbf{T}}$. In the next two subsections, we aim to follow the performance of the *best suffix tree* of depth at most D, or equivalently, to keep the worst regret $\max_{\mathbf{T}\in\mathscr{T}_{\le D}} \mathsf{Reg}_{\mathcal{A}}(\mathbf{u}[\mathbf{T}];\mathbf{g}^t)$ small for any collection of competitors $(\mathbf{u}[\mathbf{T}])_{\mathbf{T}\in\mathscr{T}_{\le D}}$.

Remark 3.5 (Matching Lower Bound). When the auxiliary sequence Ω is constructed from a binary quantizer Q with the history $\mathbf{g}^{t-1}$ as mentioned earlier, we can show an optimality of the per-state KT algorithm in Section 3 for a single tree by establishing a matching regret lower bound extending the technique of Orabona (2019, Theorem 5.12); see Appendix C.2.3.

Below, we will use the *tree potential* with respect to $\mathbf{T}$ and Ω defined as

$$\Psi^{\mathsf{KT}}(\mathbf{g}^t;\mathbf{T},\Omega) := \prod_{s\in\mathbf{T}} \Psi^{\mathsf{KT}}(\mathbf{g}^t(s;\Omega)),$$

where we write $s \in \mathbf{T}$ for any leaf node s of the tree $\mathbf{T}$ with a slight abuse of notation and we define

$$\mathbf{g}^t(s;\Omega) := (\mathbf{g}_i \colon s \text{ is a suffix of } \omega_{i-D}^{i-1},\ 1 \le i \le t).$$

From now on, we will hide any dependence on Ω whenever the omission does not incur confusion.

3.3.3 Context Tree Weighting for OLO with Tree Side Information

To compete against the best competitor adaptive to *any* tree side information of depth $\le D$, a natural solution is to consider a mixture of all tree potentials; note, however, that there are doubly-exponentially many $O(2^{2^D})$ possible suffix trees of depth $\le D$, and thus it is not computationally feasible to compute such a mixture naively. Instead, inspired by the context tree weighting (CTW) probability assignment of Willems et al. (1995), we analogously define the CTW potential as $\Psi^{\mathsf{CTW}}(\mathbf{g}^t) := \Psi^{\mathsf{CTW}}_{\lambda}(\mathbf{g}^t)$ with a recursive formula

$$\Psi^{\mathsf{CTW}}_s(\mathbf{g}^t) := \begin{cases} \frac{1}{2}\Psi^{\mathsf{KT}}_s(\mathbf{g}^t) + \frac{1}{2}\Psi^{\mathsf{CTW}}_{1s}(\mathbf{g}^t)\Psi^{\mathsf{CTW}}_{\bar{1}s}(\mathbf{g}^t) & \text{if } |s| < D \\ \Psi^{\mathsf{KT}}_s(\mathbf{g}^t) & \text{if } |s| = D \end{cases} \tag{3.5}$$

for any binary string s of length $\le D$ and $\Psi^{\mathsf{KT}}_s(\mathbf{g}^t) := \Psi^{\mathsf{KT}}(\mathbf{g}^t(s))$. Conceptually, this recursion can be performed over the perfect suffix tree of depth D, which we denote by $\mathcal{T}_D$ and call the context tree of depth D; see Figure 2 for the context tree of depth $D = 2$. Following the same logic of Willems et al. (1995), one can easily show that

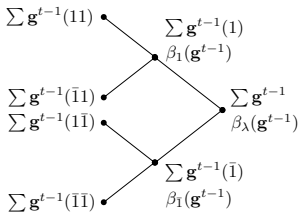

Figure 2: A context tree of depth 2.

$$\Psi^{\mathsf{CTW}}(\mathbf{g}^t) = \sum_{\mathbf{T}\in\mathscr{T}_{\le D}} w(\mathbf{T})\Psi^{\mathsf{KT}}(\mathbf{g}^t;\mathbf{T})$$

for $w(\mathbf{T}) = 2^{-\Gamma_D(\mathbf{T})}$, where $\Gamma_D(\mathbf{T}) := 2|\mathbf{T}| - 1 - |\{s \in \mathbf{T} \colon |s| = D\}|$ is a complexity measure of a full binary tree $\mathbf{T}$ of depth $\le D$, $|\mathbf{T}|$ denotes the number of leaf nodes of a full binary tree $\mathbf{T}$, and $\mathscr{T}_{\le D}$ denotes the set of all suffix trees of depth $\le D$.

For a path ρ from the root to a leaf node of $\mathcal{T}_D$ and a full binary tree $\mathbf{T}$, we let $s_{\mathbf{T}}(\rho)$ denote the unique leaf node of $\mathbf{T}$ that intersects with the path ρ. We also

define $\mathbf{v}^{\mathsf{KT}}(\mathbf{g}^{t-1};\mathbf{T}) := \mathbf{v}^{\mathsf{KT}}(\mathbf{g}^{t-1}(s_{\mathbf{T}}(\omega_{t-D}^{t-1})))$. Then, based on the construction of the vectorial betting for a mixture potential in Section 3.2, we define the vectorial CTW betting

$$\mathbf{v}^{\mathsf{CTW}}(\mathbf{g}^{t-1}) := \frac{\mathbf{u}^{\mathsf{CTW}}(\mathbf{g}^{t-1})}{\Psi^{\mathsf{CTW}}(\mathbf{g}^{t-1})}, \qquad \text{where} \tag{3.6}$$

$$\mathbf{u}^{\mathsf{CTW}}(\mathbf{g}^{t-1}) := \sum_{\mathbf{T}\in\mathscr{T}_{\le D}} w(\mathbf{T})\Psi^{\mathsf{KT}}(\mathbf{g}^{t-1};\mathbf{T})\mathbf{v}^{\mathsf{KT}}(\mathbf{g}^{t-1};\mathbf{T}),$$

then we define the CTW OLO algorithm as the action

$$\mathbf{w}^{\mathsf{CTW}}(\mathbf{g}^{t-1}) := \mathbf{v}^{\mathsf{CTW}}(\mathbf{g}^{t-1})\mathsf{W}_{t-1}(\mathbf{g}^{t-1}). \tag{3.7}$$

By Theorem 3.2, we readily have the regret guarantee of the CTW OLO algorithm as follows:

Corollary 3.3. *Let $D \ge 0$ be fixed. For any $\mathbf{g}_1,\dots,\mathbf{g}_T \in \mathbb{B}$, the CTW OLO algorithm (3.7) satisfies $\mathsf{W}_T \ge \mathsf{W}_0\Psi^{\mathsf{CTW}}(\mathbf{g}^T)$. Moreover, we have*

$$\begin{aligned}&\mathsf{Reg}(\mathbf{u}[\mathbf{T}];\mathbf{g}^T)\\ &= w(\mathbf{T})\mathsf{W}_0 + \tilde{O}\left(\sqrt{\left(\ln\frac{1}{w(\mathbf{T})}\right)\sum_{s\in\mathbf{T}} T_s^{\mathbf{T}}\|\mathbf{u}_s^{\mathbf{T}}\|^2}\right)\end{aligned}$$

for any tree $\mathbf{T}\in\mathscr{T}_{\le D}$, where $T_s^{\mathbf{T}}$ denotes the number of occurrences of a side information symbol $s\in\mathbf{T}$ with respect to the tree side information $H_{\mathbf{T};\Omega}$.

Hence, the CTW OLO algorithm (3.7) can tailor to the best tree side information in hindsight. Now, the remaining question is: can we *efficiently* compute the vectorial CTW betting (3.6)? As a first attempt, the summation over the trees $\mathbf{T}\in\mathscr{T}_{\le D}$ in (3.6) can be naively computed via a similar recursive formula as (3.5). We define

$$\rho(\omega_{t-D}^{t-1}) := \{\lambda, \omega_{t-1},\dots,\omega_{t-D}^{t-1}\}$$

and call the *active nodes* given the side information suffix ω_{t-D}^{t-1}.

Proposition 3.4. *For each node s of $\mathcal{T}_D$, define*

$$\mathbf{u}_s^{\mathsf{CTW}}(\mathbf{g}^{t-1}) := \begin{cases}\frac{1}{2}\Psi_s^{\mathsf{KT}}(\mathbf{g}^{t-1})\mathbf{v}_s^{\mathsf{KT}}(\mathbf{g}^{t-1})\\ \quad+\frac{1}{2}\mathbf{u}_{1s}^{\mathsf{CTW}}(\mathbf{g}^{t-1})\mathbf{u}_{1s}^{\mathsf{CTW}}(\mathbf{g}^{t-1}) & \text{if } |s|<D,\\ \Psi_s^{\mathsf{KT}}(\mathbf{g}^{t-1})\mathbf{v}_s^{\mathsf{KT}}(\mathbf{g}^{t-1}) & \text{if } |s|=D,\end{cases}$$

$$\mathbf{v}_s^{\mathsf{KT}}(\mathbf{g}^{t-1}) := \begin{cases}\mathbf{v}^{\mathsf{KT}}(\mathbf{g}^{t-1}(s)) & \text{if } s\in\rho(\omega_{t-D}^{t-1})\\ 1 & \text{otherwise.}\end{cases} \tag{3.8}$$

Then, the recursion is well-defined, and $\mathbf{u}_\lambda^{\mathsf{CTW}}(\mathbf{g}^{t-1}) = \mathbf{u}^{\mathsf{CTW}}(\mathbf{g}^{t-1})$.

While the recursions (3.5) and (3.8) take $O(2^D)$ steps for computing a mixture of $O(2^{2^D})$ many tree potentials, they are still not feasible as an online algorithm even for a moderate D. In the next section, we show that the per-round time complexity $O(2^D)$ can be significantly improved to $O(D)$ by exploiting the tree structure further.

3.3.4 The Efficient CTW OLO Algorithm with $O(D)$ Steps Per Round

(1) Compute $\mathbf{v}^{\mathsf{CTW}}$ in $O(D)$ steps The key idea is that, given the suffix ω_{t-D}^{t-1}, the vector betting $\mathbf{v}^{\mathsf{CTW}} = \mathbf{u}^{\mathsf{CTW}}/\Psi^{\mathsf{CTW}}$ can be computed efficiently via the recursive formulas (3.5) and (3.8), by only traversing the active nodes $\rho(\omega_{t-D}^{t-1}) = \{\lambda,\omega_{t-1},\dots,\omega_{t-D}^{t-1}\}$ in the context tree $\mathcal{T}_D$. In order to do so, we define

$$\beta_s(\mathbf{g}^{t-1}) := \frac{\Psi_s^{\mathsf{KT}}(\mathbf{g}^{t-1})}{\Psi_{\bar{1}s}^{\mathsf{CTW}}(\mathbf{g}^{t-1})\Psi_{1s}^{\mathsf{CTW}}(\mathbf{g}^{t-1})} \tag{3.9}$$

for every *internal* node s of $\mathcal{T}_D$.

Proposition 3.5. *Define*

$$\mathbf{v}_{s_d}^{\mathsf{CTW}}(\mathbf{g}^{t-1}) := \begin{cases}\frac{\beta_{s_d}(\mathbf{g}^{t-1})}{\beta_{s_d}(\mathbf{g}^{t-1})+1}\mathbf{v}_{s_d}^{\mathsf{KT}}(\mathbf{g}^{t-1})\\ \quad+\frac{1}{\beta_{s_d}(\mathbf{g}^{t-1})+1}\mathbf{v}_{s_{d+1}}^{\mathsf{CTW}}(\mathbf{g}^{t-1}) & \text{if } d<D\\ \mathbf{v}_{s_D}^{\mathsf{KT}}(\mathbf{g}^{t-1}) & \text{if } d=D\end{cases} \tag{3.10}$$

for $s_d = \omega_{t-d}^{t-1}\in\mathcal{T}_D$, $d=0,\dots,D$. Then, $\mathbf{v}^{\mathsf{CTW}}(\mathbf{g}^{t-1}) = \mathbf{v}_\lambda^{\mathsf{CTW}}(\mathbf{g}^{t-1})$.

Hence, if we can store $\sum\mathbf{g}^{t-1}(s)$ and the value $\beta_s(\mathbf{g}^{t-1})$ as defined in (3.9) for every node s of $\mathcal{T}_D$, we can compute $\mathbf{v}^{\mathsf{CTW}}$ in $O(D)$.

(2) Update β_s in $O(D)$ steps Upon receiving $\mathbf{g}_t$, we need to update $\beta_{s_d}(\mathbf{g}^{t-1})$ as

$$\beta_{s_d}(\mathbf{g}^t) = \beta_{s_d}(\mathbf{g}^{t-1})\frac{\Psi_{s_d}^{\mathsf{KT}}(\mathbf{g}^t)}{\Psi_{s_d}^{\mathsf{KT}}(\mathbf{g}^{t-1})}\frac{\Psi_{s_{d+1}}^{\mathsf{CTW}}(\mathbf{g}^{t-1})}{\Psi_{s_{d+1}}^{\mathsf{CTW}}(\mathbf{g}^t)} \tag{3.11}$$

for each $s_d = \omega_{t-d}^{t-1}\in\mathcal{T}_D$. Here, the ratio $\Psi_{s_d}^{\mathsf{CTW}}(\mathbf{g}^t)/\Psi_{s_d}^{\mathsf{CTW}}(\mathbf{g}^{t-1})$ can be also computed efficiently while traversing the path $\rho(\omega_{t-D}^{t-1})$ from the leaf node s_D to the root $s_0=\lambda$, based on the following recursion:

Proposition 3.6. *For each node $s_d = \omega_{t-d}^{t-1}\in\mathcal{T}_D$, $d=0,\dots,D$,*

$$\frac{\Psi_{s_d}^{\mathsf{CTW}}(\mathbf{g}^t)}{\Psi_{s_d}^{\mathsf{CTW}}(\mathbf{g}^{t-1})} = \begin{cases}\frac{\beta_{s_d}(\mathbf{g}^{t-1})}{\beta_{s_d}(\mathbf{g}^{t-1})+1}\frac{\Psi_{s_d}^{\mathsf{KT}}(\mathbf{g}^t)}{\Psi_{s_d}^{\mathsf{KT}}(\mathbf{g}^{t-1})}\\ \quad+\frac{1}{\beta_{s_d}(\mathbf{g}^{t-1})+1}\frac{\Psi_{s_{d+1}}^{\mathsf{CTW}}(\mathbf{g}^t)}{\Psi_{s_{d+1}}^{\mathsf{CTW}}(\mathbf{g}^{t-1})} & \text{if } d<D\,.\\ \frac{\Psi_{s_D}^{\mathsf{KT}}(\mathbf{g}^t)}{\Psi_{s_D}^{\mathsf{KT}}(\mathbf{g}^{t-1})} & \text{if } d=D\end{cases} \tag{3.12}$$

Hence, updating β_s's can be also performed efficiently in $O(D)$ time. The space complexity of this algorithm is $O(DT)$, since there can be at most D nodes activated for the first time at each round. The complete algorithm is summarized in Algorithm D.3 in Appendix.

4 EXPERIMENTS

To validate the motivation of this work and demonstrate the power of the proposed algorithms in online convex optimization, we performed online linear regression with absolute loss following Orabona and Pál (2016). We observed, however, that the datasets considered therein do not contain any temporal dependence and thus the proposed algorithms did not prove useful (data not shown). Instead, we chose two real-world temporal datasets (Beijing PM2.5 (Liang et al., 2015) and Metro Interstate Traffic Volume (Hogue, 2019)) from the UCI machine learning repository (Dua and Graff, 2019). All details including data preprocessing can be found in Appendix E and the code that fully reproduce the results is available at `https://github.com/jongharyu/olo-with-side-information`.

To construct auxiliary sequences, we used the *canonical binary quantizers* $Q_{\mathbf{e}_i}$, where $\mathbf{e}_i$ denotes the i-th standard vector. We first ran the per-state versions of OGD, AdaNormal (McMahan and Orabona, 2014), DFEG (Orabona, 2013), and KT with Markov side information of different depths and ran the CTW algorithm for the maximum depth ranging $0, 1, 3 \ldots, 11$. We optimally tuned the per-state OGD using only a single rate for all states due to the prohibitively large complexity of the optimal grid search; see Figures E.4(a) and E.5(a) in Appendix. While the per-state KT consistently showed the best performance, the performance degraded as we used too deep Markov side information beyond some threshold for all algorithms. In Figures E.4(b) and E.5(b) in Appendix, CTW often achieved even better performance than the best performance achieved by KT across the different choices of quantizer, also being robust to the choice of the maximum depth.

In practice, however, we do not know which dimension to quantize a priori. Hence, we showed the performance of the combined CTW algorithms over all d quantizers aggregated by either the mixture or the addition—conceptually, the mixture of CTWs can be viewed as a *context forest weighting*. As a benchmark, we also ran the combined KT algorithms over all d quantizers for each depth. In Figure 3, we summarized the per-coordinate results by taking the best performance over all quantizers; see the first five dashed lines in the legend. While these are only hypothetical which were not attained by an algorithm, surprisingly, the combined CTW algorithms over different quantizers, either by the mixture or the addition of Cutkosky (2019), achieved the hypothetically best performance (plotted solid).

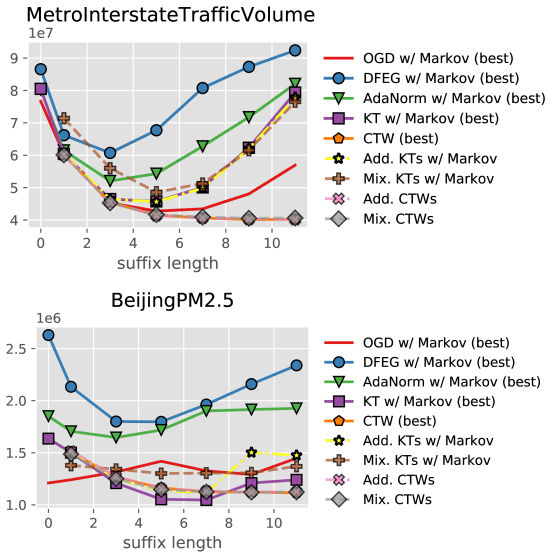

Figure 3: Summary of the experiments.

5 CONCLUDING REMARKS

Aiming to leverage a temporal structure in the sequence $\mathbf{g}^n$, we developed the CTW OLO algorithm that can efficiently adapt to the best tree side information in hindsight by combining a universal coin betting based OLO algorithm and universal compression (or prediction) techniques from information theory. Experimental results demonstrate that the proposed framework can be effective in solving real-life online convex optimization problems.

The key technical contribution of the paper is to consider the product and mixture potentials, motivated from information theory, and to adapt the CTW algorithm of Willems et al. (2006) to online linear optimization in Hilbert spaces. Main technical difficulties lie in analyzing the product potential (Proposition C.14) and properly invoking Rissanen's lower bound in Theorem C.7 to establish the optimality.

We remark that an anonymous reader of an earlier version of this manuscript proposed a simpler alternative approach based on a meta algorithm that recasts any parameter-free OLO algorithm for tree-structured side information. The idea is to combine the specialist framework of Freund et al. (1997) and apply the addition technique of Cutkosky (2019). Running a base OLO algorithm at each node of a context tree as a specialist, the meta algorithm adds up the outputs of the specialists on the active path at each round and updates them at the end of the round. This approach achieves a similar regret guarantee of the CTW OLO (Corllary 3.3) with the same complexity. A detailed study is beyond the scope of this paper and thus left as future work.

Acknowledgements

This work was supported in part by the National Science Foundation under Grant CCF-1911238. The authors appreciate insightful feedback from anonymous reviewers to improve earlier versions of the manuscript.

References

Bauschke, H. H. and Combettes, P. L. (2011). *Convex analysis and monotone operator theory in Hilbert spaces*, volume 408. Springer.

Begleiter, R., El-Yaniv, R., and Yona, G. (2004). On prediction using variable order Markov models. *J. Artif. Intell. Res.*, 22:385–421.

Bhaskara, A., Cutkosky, A., Kumar, R., and Purohit, M. (2020a). Online learning with imperfect hints. In *Proc. Int. Conf. Mach. Learn.*, pages 822–831. PMLR.

Bhaskara, A., Cutkosky, A., Kumar, R., and Purohit, M. (2020b). Online linear optimization with many hints. *arXiv preprint arXiv:2010.03082*.

Cesa-Bianchi, N. and Lugosi, G. (2006). *Prediction, learning, and games*. Cambridge University Press.

Chaudhuri, K., Freund, Y., and Hsu, D. (2009). A parameter-free hedging algorithm. In *Adv. Neural Inf. Proc. Syst.*, volume 22. Curran Associates, Inc.

Chen, L., Luo, H., and Wei, C.-Y. (2021). Impossible tuning made possible: A new expert algorithm and its applications. *arXiv preprint arXiv:2102.01046*.

Chernov, A. and Vovk, V. (2010). Prediction with advice of unknown number of experts. In *Proc. Uncertain. Artif. Intell.*

Cover, T. M. and Thomas, J. A. (2006). *Elements of information theory*. John Wiley & Sons.

Cutkosky, A. (2019). Combining online learning guarantees. In *Conf. Learn. Theory*, pages 895–913. PMLR.

Cutkosky, A. and Boahen, K. (2017). Online learning without prior information. In *Conf. Learn. Theory*, pages 643–677. PMLR.

Dekel, O., Flajolet, A., Haghtalab, N., and Jaillet, P. (2017). Online learning with a hint. In *Adv. Neural Inf. Proc. Syst.*, volume 30, pages 5299–5308. Curran Associates, Inc.

Dua, D. and Graff, C. (2019). UCI Machine Learning Repository.

Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12(7).

Foster, D. J., Rakhlin, A., and Sridharan, K. (2015). Adaptive online learning. In *Adv. Neural Inf. Proc. Syst.*, volume 28, pages 3375–3383. Curran Associates, Inc.

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

Freund, Y., Schapire, R. E., Singer, Y., and Warmuth, M. K. (1997). Using and combining predictors that specialize. In *Proc. Annu. ACM Symp. Theory Comput.*, pages 334–343.

Hogue, J. (2019). Metro interstate traffic volume data set.

Jiao, J., Permuter, H. H., Zhao, L., Kim, Y.-H., and Weissman, T. (2013). Universal estimation of directed information. *IEEE Trans. Inf. Theory*, 59(10):6220–6242.

Jun, K.-S. and Orabona, F. (2019). Parameter-free online convex optimization with sub-exponential noise. In *Conf. Learn. Theory*, pages 1802–1823. PMLR.

Jun, K.-S., Orabona, F., Wright, S., and Willett, R. (2017). Online learning for changing environments using coin betting. *Electron. J. Stat.*, 11(2):5282–5310.

Kelly Jr., J. L. (1956). A new interpretation of information rate. *IRE Trans. Inf. Theory*, 3(2):185–189.

Koolen, W. M. and Van Erven, T. (2015). Second-order quantile methods for experts and combinatorial games. In *Conf. Learn. Theory*, pages 1155–1175. PMLR.

Kozat, S. S., Singer, A. C., and Bean, A. J. (2008). Universal portfolios via context trees. In *Proc. IEEE Int. Conf. Acoust. Speech. Signal Process.*, pages 2093–2096. IEEE.

Krichevsky, R. and Trofimov, V. (1981). The performance of universal encoding. *IEEE Trans. Inf. Theory*, 27(2):199–207.

Kuzborskij, I. and Cesa-Bianchi, N. (2020). Locally-adaptive nonparametric online learning. In *Adv. Neural Inf. Proc. Syst.*, volume 33.

Liang, X., Zou, T., Guo, B., Li, S., Zhang, H., Zhang, S., Huang, H., and Chen, S. X. (2015). Assessing Beijing's PM2.5 pollution: Severity, weather impact, APEC and winter heating. *Proc. R. Soc. A*, 471(2182):20150257.

Luo, H. and Schapire, R. E. (2015). Achieving all with no parameters: AdaNormalHedge. In *Conf. Learn. Theory*, pages 1286–1304. PMLR.

McMahan, H. B. and Abernethy, J. (2013). Minimax optimal algorithms for unconstrained linear opti-

mization. In *Adv. Neural Inf. Proc. Syst.*, volume 26. Curran Associates, Inc.

McMahan, H. B. and Orabona, F. (2014). Unconstrained online linear learning in Hilbert spaces: Minimax algorithms and normal approximations. In *Conf. Learn. Theory*, pages 1020–1039. PMLR.

Messias, J. V. and Whiteson, S. (2018). Dynamic-depth context tree weighting. In *Adv. Neural Inf. Proc. Syst.*, volume 31. Curran Associates, Inc.

Orabona, F. (2013). Dimension-free exponentiated gradient. In *Adv. Neural Inf. Proc. Syst.*, volume 26, pages 1806–1814. Curran Associates, Inc.

Orabona, F. (2014). Simultaneous model selection and optimization through parameter-free stochastic learning. *arXiv preprint arXiv:1406.3816*.

Orabona, F. (2019). A modern introduction to online learning. *arXiv preprint arXiv:1912.13213*.

Orabona, F. and Cutkosky, A. (2020). ICML 2020 tutorial on parameter-free online optimization. Websites: `https://parameterfree.com/icml-tutorial/`, `https://icml.cc/Conferences/2020/Schedule?showEvent=5753`.

Orabona, F. and Pál, D. (2016). Coin betting and parameter-free online learning. In *Adv. Neural Inf. Proc. Syst.*, volume 29. Curran Associates, Inc.

Orabona, F. and Tommasi, T. (2017). Training deep networks without learning rates through coin betting. In *Adv. Neural Inf. Proc. Syst.*, volume 30. Curran Associates, Inc.

Rakhlin, A. and Sridharan, K. (2013). Online learning with predictable sequences. In *Conf. Learn. Theory*, pages 993–1019. PMLR.

Rissanen, J. (1984). Universal coding, information, prediction, and estimation. *IEEE Trans. Inf. Theory*, 30(4):629–636.

Rissanen, J. J. (1996). Fisher information and stochastic complexity. *IEEE Trans. Inf. Theory*, 42(1):40–47.

Shalev-Shwartz, S. (2011). Online learning and online convex optimization. *Found. Trends Mach. Learn.*, 4(2):107–194.

Van der Hoeven, D., van Erven, T., and Kotłowski, W. (2018). The many faces of exponential weights in online learning. In *Conf. Learn. Theory*, pages 2067–2092. PMLR.

Willems, F. M., Shtarkov, Y. M., and Tjalkens, T. J. (1995). The context-tree weighting method: Basic properties. *IEEE Trans. Inf. Theory*, 41(3):653–664.

Willems, F. M., Tjalkens, T. J., and Ignatenko, T. (2006). Context-tree weighting and maximizing: Processing betas. In *Proc. UCSD Inf. Theory Appl. Workshop*.

Xie, Q. and Barron, A. R. (1997). Minimax redundancy for the class of memoryless sources. *IEEE Trans. Inf. Theory*, 43(2):646–657.

Zhang, L., Wang, G., Yi, J., and Yang, T. (2021). A simple yet universal strategy for online convex optimization. *arXiv preprint arXiv:2105.03681*.

Ziv, J. and Lempel, A. (1977). A universal algorithm for sequential data compression. *IEEE Trans. Inf. Theory*, 23(3):337–343.

Supplementary Material: Parameter-Free Online Linear Optimization with Side Information via Universal Coin Betting

A RELATED WORK

There have been several parameter-free methods proposed for OLO in Hilbert space (Orabona, 2013, 2014; McMahan and Orabona, 2014; Orabona and Pál, 2016) as well as learning with expert advice (LEA) (Freund and Schapire, 1997; Chaudhuri et al., 2009; Chernov and Vovk, 2010; Luo and Schapire, 2015; Foster et al., 2015; Koolen and Van Erven, 2015; Orabona and Pál, 2016); see also (Orabona, 2019, Chapter 9) and the references therein. A parallel line of work on parameter-free methods considers the case when the maximum norm of $\mathbf{g}_t$ (often referred to as the *Lipschitz constant*), which is assumed to be 1 throughout in this paper, is unknown but the competitor norm $\|\mathbf{u}\|$ is known (Duchi et al., 2011; Cutkosky and Boahen, 2017). Recently, Zhang et al. (2021); Chen et al. (2021) studied a similar setting in this paper, albeit establishing guarantees only for bounded domains. We remark that AdaNormalHedge (Luo and Schapire, 2015) is a parameter-free LEA algorithm which can compete with mixtures of forcasters with side information, in particular tree experts via mixtures of sleeping experts; for example, Kuzborskij and Cesa-Bianchi (2020) used AdaNormalHedge with tree experts for binary classification with absolute loss. For a comprehensive overview of these parameter-free methods, see the tutorial (Orabona and Cutkosky, 2020).

The connection between OLO and gambling was shown by Orabona and Pál (2016), where they also described a reduction for LEA. This idea was also applied to training deep neural networks (Orabona and Tommasi, 2017). While the proposed algorithms in this paper are against *stationary* competitors, Jun et al. (2017) proposed a coin betting based OLO algorithm against nonstationary competitors characterized by a sequence of vectors $\mathbf{u}_1, \dots, \mathbf{u}_T$ such that have at most m change points. Van der Hoeven et al. (2018, Section 5 and particularly Theorem 9) establishes a connection between the exponential weights (EW) algorithm and the coin-betting scheme. Earlier on in the paper, in Section 2 the interpretation of compression as a special case of EW with $\eta = 1$ is provided as well. Similarly, Jun and Orabona (2019) utilize such a connection as well. To the best of our knowledge, however, we did not find a clear bridge constructed between compression and coin-betting methods in either, even though a careful examination of the mathematical details may hint toward this connection.

Universal compression, which is a classical topic in information theory, aims to compress sequences with no (or very little) statistical assumptions. In the last century, there have been several techniques proposed that can compete against the best i.i.d. compressor (Krichevsky and Trofimov, 1981; Rissanen, 1984; Xie and Barron, 1997), finite state compressor (Ziv and Lempel, 1977) and tree compressor (Willems et al., 1995). The CTW probability assignment invented by (Willems et al., 1995) has been one of the most successful and widely used universal compression techniques. Beyond compression, this technique has been applied to estimation of directed information (Jiao et al., 2013), universal portfolios (Kozat et al., 2008), and reinforcement learning (Messias and Whiteson, 2018), to name a few. The efficient CTW OLO algorithm presented in Section 3.3.4 is in the spirit of the processing betas algorithm proposed by Willems et al. (2006) for computing the predictive conditional probability induced by the CTW probability assignment (Willems et al., 1995). Cesa-Bianchi and Lugosi (2006, Section 5.3) also presented a CTW-based Hedge algorithm for LEA; see bibliographic remarks therein for other applications of CTW to learning problems.

A related line of recent work on online learning with hints (Dekel et al., 2017; Bhaskara et al., 2020a,b) considers a scenario where the learner receives a vector $\mathbf{h}_t$ with $\|\mathbf{h}_t\| = 1$ such that $\langle \mathbf{h}_t, \mathbf{g}_t/\|\mathbf{g}_t\| \rangle \geq \alpha > 0$ as a "hint" to the future. However, our setting is not directly comparable, since we only consider a finite side information and this line of work aims to establish small regret $o(\sqrt{T})$ measured with respect to static competitors. We also remark that Rakhlin and Sridharan (2013) studied the problem of OLO when $\mathbf{g}_t$ is modelled as a "predictable" sequence, in the sense that $\mathbf{g}_t = M(\mathbf{g}^{t-1}) + \mathbf{n}_t$ with some adversarial noise $\mathbf{n}_t$ with a (possibly randomized) function M; yet, they considered static competitors unlike this work.

B PER-STATE EXTENSIONS OF EXISTING ALGORITHMS

Here we present per-state versions of OGD and two existing parameter-free OLO algorithms: the dimension-free exponentiated gradient algorithm (DFEG) (Orabona, 2013) and the adaptive normal algorithm (AdaNormal) (McMahan and Orabona, 2014).

Following the original problem setting in (Orabona, 2013), we describe the per-state DFEG only for online linear regression. Consider a loss function $\ell(\hat{y}, y)$, which is convex and L-Lipschitz in its first argument. At each round t, a learner picks $\mathbf{w}_t \in V$. A nature then reveals $(\mathbf{x}_t, y_t) \in V \times \mathbb{R}$, and the learner suffers loss $\ell_t(\mathbf{w}_t) := \ell(\hat{y}_t, y_t)$, where $\hat{y}_t := \langle \mathbf{w}_t, \mathbf{x}_t \rangle$. Note that the DFEG algorithm requires a norm of the instance $\|\mathbf{x}_t\|$ to form an action $\mathbf{w}_t$.

Algorithm B.1 Per-state Dimension-free Exponentiated Gradient (Orabona, 2013) for online regression

procedure PERSTATEDFEG($L, \delta, 0.882 \leq a \leq 1.109$)
  Initialize $\boldsymbol{\theta}^{(s)} \leftarrow 0 \in V, H^{(s)} \leftarrow \delta$ for each $s \in [S]$
  for $1 \leq t \leq T$ **do**
    Receive $h_t \in [S]$ and $\|\mathbf{x}_t\|$
    Update $H^{(h_t)} \leftarrow H^{(h_t)} + L^2 \max\{\|\mathbf{x}_t\|, \|\mathbf{x}_t\|^2\}$
    Set $\alpha_t \leftarrow a(H^{(h_t)})^{1/2}, \beta_t \leftarrow (H^{(h_t)})^{3/2}$
    if $\|\boldsymbol{\theta}^{(h_t)}\| = 0$ **then**
      Set $\mathbf{w}_t \leftarrow 0$
    else
      Set $\mathbf{w}_t \leftarrow \frac{\boldsymbol{\theta}^{(h_t)}}{\beta_t \|\boldsymbol{\theta}^{(h_t)}\|} \exp(\frac{\|\boldsymbol{\theta}^{(h_t)}\|}{\alpha_t})$
    end if
    Receive $(\mathbf{x}_t, y_t)$ and incur loss $\ell_t(\mathbf{w}_t)$
    Update $\boldsymbol{\theta}^{(h_t)} \leftarrow \boldsymbol{\theta}^{(h_t)} - \partial \ell_t(\langle \mathbf{w}_t, \mathbf{x}_t \rangle) \mathbf{x}_t$
  end for
end procedure

Algorithm B.2 Per-state AdaptiveNormal (McMahan and Orabona, 2014) for OLO with side information

procedure PERSTATEADANORMAL($L, a \geq \frac{3L^2\pi}{4}, \epsilon$)
  Initialize $\boldsymbol{\theta}^{(s)} \leftarrow 0 \in V$ for each $s \in [S]$
  for $1 \leq t \leq T$ **do**
    Receive $h_t \in [S]$
    if $\|\boldsymbol{\theta}^{(h_t)}\| = 0$ **then**
      Set $\mathbf{w}_t \leftarrow 0$
    else
      Set $\mathbf{w}_t \leftarrow \epsilon \frac{\boldsymbol{\theta}^{(h_t)}}{\|\boldsymbol{\theta}^{(h_t)}\|} \frac{1}{2L \ln^2(t+1)} \{\exp(\frac{(\|\boldsymbol{\theta}^{(h_t)}\|+L)^2}{2at}) - \exp(\frac{(\|\boldsymbol{\theta}^{(h_t)}\|-L)^2}{2at})\}$
    end if
    Receive $\mathbf{g}_t$ and incur loss $\langle \mathbf{g}_t, \mathbf{w}_t \rangle$
    Update $\boldsymbol{\theta}^{(h_t)} \leftarrow \boldsymbol{\theta}^{(h_t)} - \mathbf{g}_t$
  end for
end procedure

We remark that these two algorithms are also guaranteed to incur essentially the same order of regret without tuning learning rate. Also, while the per-state KT OLO algorithm serves as a base algorithm in the CTW OLO algorithm, to be a fair comparison, the two algorithms can be also used as a base in the specialist framework to solve the tree side information problem, as noted in Section 5. There are, however, two minor disadvantages we can observe. First of all, the DFEG algorithm is tailored to the online linear regression problem, while the per-state KT OLO and AdaptiveNormal algorithms can be applied to a general OLO problem. Second, while the KT OLO has only one hyperparameter, the initial wealth W_0, the above two per-state algorithms have two hyperparameters (except the Lipschitz constant), which may need to be chosen or tuned in practice.

C DEFERRED TECHNICAL MATERIALS

C.1 Proofs for Section 2

C.1.1 Proof of Theorem 2.1

We note that all statements in Section 2 originally appeared in (Orabona and Pál, 2016). The proofs given here are rephrased and simplified from (Orabona and Pál, 2016).

Before we prove Theorem 2.1, we state some key properties of the KT potential function ψ^{KT}.

Proposition C.1. *For each $t \geq 1$ and any $g_1, \dots, g_t \in [-1, 1]$, the followings hold:*

(a) (Coordinatewise convexity) $g \mapsto \psi^{\mathsf{KT}}(g^{t-1}g)$ is convex for $g \in [-1, 1]$.

(b) (Consistency) $\psi^{\mathsf{KT}}(g^{t-1}) = \frac{1}{2}(\psi^{\mathsf{KT}}(g^{t-1}1) + \psi^{\mathsf{KT}}(g^{t-1}\bar{1}))$.

(c) (The relation of signed betting and potential)

$$b^{\mathsf{KT}}(g^{t-1}) = \frac{\psi^{\mathsf{KT}}(g^{t-1}1) - \psi^{\mathsf{KT}}(g^{t-1}\bar{1})}{\psi^{\mathsf{KT}}(g^{t-1}1) + \psi^{\mathsf{KT}}(g^{t-1}\bar{1})} = \frac{\psi^{\mathsf{KT}}(g^{t-1}1) - \psi^{\mathsf{KT}}(g^{t-1}\bar{1})}{\psi^{\mathsf{KT}}(g^{t-1})}.$$

(d) For any $x \in [0, t)$, $x(\psi_t^{\mathsf{KT}})''(x) \geq (\psi_t^{\mathsf{KT}})'(x)$.

Proof. Recall $\tilde{q}_t^{\mathsf{KT}}(x) := B(\frac{t+x+1}{2}, \frac{t-x+1}{2})/B(\frac{1}{2}, \frac{1}{2})$ and $\psi^{\mathsf{KT}}(g^t) := \psi_t^{\mathsf{KT}}(\sum g^t) := 2^t \tilde{q}_t^{\mathsf{KT}}(\sum g^t)$. (a) and (d) follow from the properties of the Gamma function $\Gamma(\cdot)$; for details, see (Orabona and Pál, 2016, Lemma 12) and the proof therein. (b) and (c) can be easily verified by the definition of the KT potential ψ^{KT}. □

We remark that the relation (b) can be understood as a continuous extension of the consistency of $\tilde{q}^{\mathsf{KT}}$ as a joint probability over a binary sequence $g^t \in \{-1, 1\}^t$. Further, in view of the relation (c), the signed bet b^{KT} is a continuous extension of the prequential probability $\tilde{q}^{\mathsf{KT}}(\cdot|g^{t-1})$ induced by the joint probability assignment $\tilde{q}^{\mathsf{KT}}(g^t)$.

We now show the following single round bound.

Lemma C.2. *For any $t \geq 1$ and $g_1, \dots, g_t \in [-1, 1]$, we have*

$$(1 + g_t b_t^{\mathsf{KT}}(g^{t-1}))\psi^{\mathsf{KT}}(g^{t-1}) \geq \psi^{\mathsf{KT}}(g^t).$$

Proof. By the definition of coin betting potentials, we have

$$\begin{aligned}
(1 + g_t b^{\mathsf{KT}}(g^{t-1}))\psi^{\mathsf{KT}}(g^{t-1}) &\overset{(i)}{\geq} (1 + g_t b^{\mathsf{KT}}(g^{t-1}))\frac{1}{2}(\psi^{\mathsf{KT}}(g^{t-1}1) + \psi^{\mathsf{KT}}(g^{t-1}\bar{1})) \\
&\overset{(ii)}{=} \Big(1 + g_t \frac{\psi^{\mathsf{KT}}(g^{t-1}1) - \psi^{\mathsf{KT}}(g^{t-1}\bar{1})}{\psi^{\mathsf{KT}}(g^{t-1}1) + \psi^{\mathsf{KT}}(g^{t-1}\bar{1})}\Big)\frac{1}{2}(\psi^{\mathsf{KT}}(g^{t-1}1) + \psi^{\mathsf{KT}}(g^{t-1}\bar{1})) \\
&= \frac{1 + g_t}{2}\psi^{\mathsf{KT}}(g^{t-1}1) + \frac{1 - g_t}{2}\psi^{\mathsf{KT}}(g^{t-1}\bar{1}) \\
&\overset{(iii)}{\geq} \psi^{\mathsf{KT}}(g^t).
\end{aligned}$$

where (i), (ii), and (iii) follow from (b), (c), and (a) in Proposition C.1, respectively. □

While the above lemma establishes the lower bound on the cumulative wealth, we then need the following statement that connects regret and wealth via convex duality. We remark that this relation is the key statement that motivates all coin betting based algorithms.

Proposition C.3 (McMahan and Orabona, 2014, (Orabona and Pál, 2016, Lemma 1))**.** *Let $\Phi\colon V \to \mathbb{R}$ be a convex function and let $\Phi^\star\colon V \to \mathbb{R} \cup \{+\infty\}$ denote its Fenchel conjugate function. For any $\mathbf{g}_1, \dots, \mathbf{g}_T \in V^\star$ and any $\mathbf{w}_t, \dots, \mathbf{w}_T \in V$, we have*

$$\sup_{\mathbf{u} \in V}\{\mathsf{Reg}(\mathbf{u}; \mathbf{g}^T) - \Phi(\mathbf{u})\} = -\sum_{t=1}^{T}\langle \mathbf{g}_t, \mathbf{w}_t\rangle + \Phi^\star\Big(\sum_{t=1}^{T}\mathbf{g}_t\Big),$$

where $\mathsf{Reg}(\mathbf{u}; \mathbf{g}^T) := \sum_{t=1}^T \langle \mathbf{g}_t, \mathbf{u} - \mathbf{w}_t \rangle$.

Proof. By definition of Fenchel dual, we have

$$\begin{aligned}
\sup_{\mathbf{u} \in V} \{\mathsf{Reg}(\mathbf{u}; \mathbf{g}^T) - \Phi(\mathbf{u})\} &= \sup_{\mathbf{u} \in V} \Big\{ \sum_{t=1}^T \langle \mathbf{g}_t, \mathbf{u} - \mathbf{w}_t \rangle - \Phi(\mathbf{u}) \Big\} \\
&= -\sum_{t=1}^T \langle \mathbf{g}_t, \mathbf{w}_t \rangle + \sup_{\mathbf{u} \in V} \Big\{ \Big\langle \sum_{t=1}^T \mathbf{g}_t, \mathbf{u} \Big\rangle - \Phi(\mathbf{u}) \Big\} \\
&= -\sum_{t=1}^T \langle \mathbf{g}_t, \mathbf{w}_t \rangle + \Phi^\star \Big(\sum_{t=1}^T \mathbf{g}_t \Big).
\end{aligned}$$

□

Now we are ready to prove Theorem 2.1.

Proof of Theorem 2.1. We first show the wealth lower bound $\mathsf{W}_t \geq \mathsf{W}_0 \psi^{\mathsf{KT}}(g^t)$ stated in (2.1) by induction on t. Suppose that $\mathsf{W}_{t-1} \geq \mathsf{W}_0 \psi^{\mathsf{KT}}(g^{t-1})$. Then,

$$\begin{aligned}
\mathsf{W}_t &= \mathsf{W}_{t-1} + g_t w_t \\
&= (1 + b^{\mathsf{KT}}(g^{t-1}) g_t) \mathsf{W}_{t-1} \\
&\overset{(a)}{\geq} (1 + b^{\mathsf{KT}}(g^{t-1}) g_t) \mathsf{W}_0 \psi^{\mathsf{KT}}(g^{t-1}) \\
&\overset{(b)}{\geq} \mathsf{W}_0 \psi^{\mathsf{KT}}(g^t),
\end{aligned}$$

where (a) follows from the induction hypothesis and (b) follows from Lemma C.2.

The wealth lower bound can be converted into the desired regret bound by Proposition C.3. That is, we have

$$\sup_{u \in \mathbb{R}} \{\mathsf{Reg}(u; g^T) - \phi(u)\} = -\sum_{t=1}^T g_t w_t + \mathsf{W}_0 \psi^{\mathsf{KT}}(g^T) \leq \mathsf{W}_0,$$

where $\phi \colon \mathbb{R} \to \mathbb{R}$ is a convex function such that its conjugate function $\phi^\star \colon \mathbb{R} \to \mathbb{R} \cup \{+\infty\}$ is equal to $\mathsf{W}_0 \psi_T^{\mathsf{KT}}(\sum g^t)$. Since $x \mapsto \psi_T^{\mathsf{KT}}(x)$ is a convex, proper, closed function, one can check that $\phi(u) = \mathsf{W}_0 (\psi_T^{\mathsf{KT}})^\star(\frac{u}{\mathsf{W}_0})$ using Lemma C.10. □

C.1.2 Proof of Theorem 2.2

As in 1D OLO case, we first show the following single round bound.

Lemma C.4. *For any* $\mathbf{g}_1, \dots, \mathbf{g}_t \in \mathbb{B}$*, we have*

$$(1 + \langle \mathbf{g}_t, \mathbf{v}^{\mathsf{KT}}(\mathbf{g}^{t-1}) \rangle) \Psi^{\mathsf{KT}}(\mathbf{g}^{t-1}) \geq \Psi^{\mathsf{KT}}(\mathbf{g}^t).$$

Proof. Let $\mathbf{f}_{t-1} := \sum \mathbf{g}^{t-1}$. Consider

$$\begin{aligned}
&(1 + \langle \mathbf{g}_t, \mathbf{v}^{\mathsf{KT}}(\mathbf{g}^{t-1}) \rangle) \Psi^{\mathsf{KT}}(\mathbf{g}^{t-1}) - \Psi^{\mathsf{KT}}(\mathbf{g}^t) \\
&\quad = \Psi^{\mathsf{KT}}(\mathbf{g}^{t-1}) + \langle \mathbf{g}_t, \mathbf{v}^{\mathsf{KT}}(\mathbf{g}^{t-1}) \rangle \Psi^{\mathsf{KT}}(\mathbf{g}^{t-1}) - \Psi^{\mathsf{KT}}(\mathbf{g}^t) \\
&\quad = \psi_{t-1}^{\mathsf{KT}}(\|\mathbf{f}_{t-1}\|) + \Big\langle \mathbf{g}_t, b_t^{\mathsf{KT}}(\|\mathbf{f}_{t-1}\|) \frac{\mathbf{f}_{t-1}}{\|\mathbf{f}_{t-1}\|} \Big\rangle \psi_{t-1}^{\mathsf{KT}}(\|\mathbf{f}_{t-1}\|) - \psi_t^{\mathsf{KT}}(\|\mathbf{f}_{t-1} + \mathbf{g}_t\|) \\
&\quad \overset{(a)}{\geq} \psi_{t-1}^{\mathsf{KT}}(\|\mathbf{f}_{t-1}\|) + \min_{r \in \{\pm 1\}} \{ r \|\mathbf{g}_t\| b_t^{\mathsf{KT}}(\|\mathbf{f}_{t-1}\|) \psi_{t-1}^{\mathsf{KT}}(\|\mathbf{f}_{t-1}\|) - \psi_t^{\mathsf{KT}}(\|\mathbf{f}_{t-1}\| + r\|\mathbf{g}_t\|) \} \\
&\quad = \min_{r \in \{\pm 1\}} \{ (1 + r\|\mathbf{g}_t\| b_t^{\mathsf{KT}}(\|\mathbf{f}_{t-1}\|)) \psi_{t-1}^{\mathsf{KT}}(\|\mathbf{f}_{t-1}\|) - \psi_t^{\mathsf{KT}}(\|\mathbf{f}_{t-1}\| + r\|\mathbf{g}_t\|) \} \\
&\quad \geq \min_{g \in [-1,1]} \{ (1 + g b_t^{\mathsf{KT}}(\|\mathbf{f}_{t-1}\|)) \psi_{t-1}^{\mathsf{KT}}(\|\mathbf{f}_{t-1}\|) - \psi_t^{\mathsf{KT}}(\|\mathbf{f}_{t-1}\| + g) \}
\end{aligned}$$

$\overset{(b)}{\geq} 0.$

Here, we apply Lemma C.8 since ψ_t^{KT} satisfies $x(\psi_t^{\mathsf{KT}})''(x) \geq (\psi_t^{\mathsf{KT}})'(x)$ for all $x \in [0,t)$, to have (a) by plugging in $\mathbf{u} \leftarrow \mathbf{g}_t$, $\mathbf{v} \leftarrow \mathbf{f}_{t-1}$, $c(\|\mathbf{u}\|, \|\mathbf{v}\|) \leftarrow \frac{b_t^{\mathsf{KT}}(\|\mathbf{f}_{t-1}\|)}{\|\mathbf{f}_{t-1}\|}\psi_{t-1}^{\mathsf{KT}}(\|\mathbf{f}_{t-1}\|)$, and $h(\cdot) \leftarrow \psi_t^{\mathsf{KT}}(\cdot)$. (b) follows from the single round bound for 1D case established in Lemma C.2. □

The proof of Theorem 2.2 now follows similarly to that of Theorem 2.1.

Proof of Theorem 2.2. We show $\mathsf{W}_t \geq \mathsf{W}_0\Psi^{\mathsf{KT}}(\mathbf{g}^t)$ by induction on t. For $t = 0$, it trivially holds. For $t \geq 1$, assume that $\mathsf{W}_{t-1} \geq \mathsf{W}_0\Psi^{\mathsf{KT}}(\mathbf{g}^{t-1})$ holds. Then, we have

$$\begin{aligned}
\mathsf{W}_t &= \langle \mathbf{g}_t, \mathbf{w}_t^{\mathsf{KT}} \rangle + \mathsf{W}_{t-1} \\
&= (1 + \langle \mathbf{g}_t, \mathbf{v}^{\mathsf{KT}}(\mathbf{g}^{t-1}) \rangle)\mathsf{W}_{t-1} \\
&\overset{(a)}{\geq} (1 + \langle \mathbf{g}_t, \mathbf{v}^{\mathsf{KT}}(\mathbf{g}^{t-1}) \rangle)\mathsf{W}_0\Psi^{\mathsf{KT}}(\mathbf{g}^{t-1}) \\
&\overset{(b)}{\geq} \mathsf{W}_0\Psi^{\mathsf{KT}}(\mathbf{g}^t).
\end{aligned}$$

Here, (a) follows from the induction hypothesis and (b) follows from the above lemma. The regret bound follows by the same logic of the 1D case using Proposition C.3 with the additional application of Lemma C.9, which implies that $(\psi_t^{\mathsf{KT}})^\star(\mathbf{u}) = (\psi_t^{\mathsf{KT}})^\star(\|\mathbf{u}\|)$. □

C.2 Proofs for Section 3

C.2.1 Proof of Theorem 3.1

The following statement generalizes Proposition C.3 for static competitors to adaptive competitors.

Proposition C.5. *Let $\Phi\colon V \times \cdots \times V \to \mathbb{R}$ be a convex function and let $\Phi^\star\colon V \times \cdots \times V \to \mathbb{R} \cup \{+\infty\}$. For any side information sequence $H = (h_t)_{t\geq 1}$, any $\mathbf{g}_1, \ldots, \mathbf{g}_T \in V^\star$, and any $\mathbf{w}_t, \ldots, \mathbf{w}_T \in V$, we have*

$$\sup_{\mathbf{u}_{1:S} \in V \times \cdots V} \{\mathsf{Reg}(\mathbf{u}_{1:S}[H]; \mathbf{g}^T) - \Phi(\mathbf{u}_{1:S})\} = -\sum_{t=1}^T \langle \mathbf{g}_t, \mathbf{w}_t \rangle + \Phi^\star\Big(\sum_{t\in[T]:h_t=1} \mathbf{g}_t, \ldots, \sum_{t\in[T]:h_t=S} \mathbf{g}_t\Big),$$

where $\mathsf{Reg}(\mathbf{u}_{1:S}[H]; \mathbf{g}^T) := \sum_{s=1}^S \sum_{t\in[T]:h_t=S} \langle \mathbf{g}_t, \mathbf{u}_s - \mathbf{w}_t \rangle$.

Proof. By definition of Fenchel dual, we have

$$\begin{aligned}
\sup_{\mathbf{u}_{1:S} \in V \times \cdots V} \{\mathsf{Reg}(\mathbf{u}_{1:S}[H]; \mathbf{g}^T) - \Phi(\mathbf{u}_{1:S})\} &= \sup_{\mathbf{u}_{1:S} \in V \times \cdots V} \Big\{\sum_{s=1}^S \sum_{t\in[T]:h_t=s} \langle \mathbf{g}_t, \mathbf{u}_s - \mathbf{w}_t \rangle - \Phi(\mathbf{u}_{1:S})\Big\} \\
&= -\sum_{t=1}^T \langle \mathbf{g}_t, \mathbf{w}_t \rangle + \sup_{\mathbf{u}_{1:S} \in V \times \cdots V} \Big\{\sum_{s=1}^S \Big\langle \sum_{t\in[T]:h_t=s} \mathbf{g}_t, \mathbf{u}_s \Big\rangle - \Phi(\mathbf{u}_{1:S})\Big\} \\
&= -\sum_{t=1}^T \langle \mathbf{g}_t, \mathbf{w}_t \rangle + \Phi^\star\Big(\sum_{t\in[T]:h_t=1} \mathbf{g}_t, \ldots, \sum_{t\in[T]:h_t=S} \mathbf{g}_t\Big).
\end{aligned}$$

□

We are now ready to prove Theorem 3.1.

Proof of Theorem 3.1. Since the vectorial betting $\mathbf{v}^{\mathsf{KT}}(\mathbf{g}^{t-1}; h^t)$ only affects the component potential $\Psi^{\mathsf{KT}}(\mathbf{g}^t(h_t; h^{t-1}))$ by construction, the wealth lower bound readily follows from the same argument in the proof of Theorem 2.2. Now, we observe that

$$\Psi^{\mathsf{KT}}(\mathbf{g}^T; h^T) = 2^T \prod_{s\in[S]} \tilde{q}_{T_s}^{\mathsf{KT}}(\|\textstyle\sum \mathbf{g}^T(s; h^T)\|),$$

where $T_s := |\{t \in [T] \colon h_t = s\}|$. Since $\tilde{q}_T^{\mathsf{KT}}(x) \geq \frac{1}{2^T e\sqrt{\pi}} \frac{1}{\sqrt{T}} e^{\frac{2x^2}{T}}$ for $T \geq 1$ by (Orabona and Pál, 2016, Lemma 14), we have

$$\Psi^{\mathsf{KT}}(\mathbf{g}^T; h^T) \geq \Big(\frac{1}{e\sqrt{\pi}}\Big)^{S'} \frac{1}{\sqrt{T'_1 \cdots T'_S}} \exp\Big(\sum_{s=1}^{S} \frac{2\|\sum \mathbf{g}^T(s; h^T)\|^2}{T'_s}\Big),$$

where $S' := \sum_{s=1}^{S} 1\{T_s \geq 1\}$ and $T'_s := T_s \vee 1$. Applying Propositions C.5 and C.14 then establishes the regret upper bound. □

C.2.2 Proof of Theorem 3.2

We show $\mathsf{W}_t \geq \mathsf{W}_0 \Psi^{\mathsf{mix}}(\mathbf{g}^t; \mathbf{h}^t)$ by induction on t. For $t = 0$, it trivially holds. For $t \geq 1$, assume that $\mathsf{W}_{t-1} \geq \mathsf{W}_0 \Psi^{\mathsf{mix}}(\mathbf{g}^{t-1}; \mathbf{h}^{t-1})$ holds. Then, we have

$$\begin{aligned}
\mathsf{W}_t &= \langle \mathbf{g}_t, \mathbf{w}_t^{\mathsf{mix}}(\mathbf{g}^{t-1}; \mathbf{h}^t) \rangle + \mathsf{W}_{t-1} \\
&= (1 + \langle \mathbf{g}_t, \mathbf{v}^{\mathsf{mix}}(\mathbf{g}^{t-1}; \mathbf{h}^t) \rangle) \mathsf{W}_{t-1} \\
&\overset{(a)}{\geq} (1 + \langle \mathbf{g}_t, \mathbf{v}^{\mathsf{mix}}(\mathbf{g}^{t-1}; \mathbf{h}^t) \rangle) \mathsf{W}_0 \Psi^{\mathsf{mix}}(\mathbf{g}^{t-1}; \mathbf{h}^{t-1}) \\
&\overset{(b)}{\geq} \mathsf{W}_0 \Psi^{\mathsf{mix}}(\mathbf{g}^t; \mathbf{h}^t).
\end{aligned}$$

Here, (a) follows from the induction hypothesis, and (b) follows from the construction of $\mathbf{v}^{\mathsf{mix}}(\mathbf{g}^{t-1}; \mathbf{h}^t)$. The regret guarantee for $m \in [M]$ readily follows from the construction of the mixture potential, which guarantees $\mathsf{W}_T \geq w_m \mathsf{W}_0 \Psi^{\mathsf{KT}}(g^T; (h^{(m)})^T)$. □

C.2.3 Matching lower bounds for tree side information

We first require the following theorem from (Orabona, 2019).

Theorem C.6 (Orabona, 2019, Theorem 5.11)**.** *Suppose that an OLO algorithm satisfies that for each $t \geq 0$*

$$\sup_{\mathbf{g}^t \in \mathbb{B}^t} \mathsf{Reg}(\mathbf{0}; \mathbf{g}^t) = -\inf_{\mathbf{g}^t \in \mathbb{B}^t} \sum_{i=1}^{t} \langle \mathbf{g}_i, \mathbf{w}_i \rangle \leq \mathsf{W}_0^{(t)} \tag{C.1}$$

with some nondecreasing sequence $(\mathsf{W}_0^{(t)})_{t \geq 0}$. Then, for each $T \geq 1$, there exists $\mathbf{v}_1, \ldots, \mathbf{v}_T \in \mathbb{B}$ such that

$$\mathbf{w}_t = \mathbf{v}_t\Big(\mathsf{W}_0^{(T)} + \sum_{i=1}^{t-1} \langle \mathbf{g}_i, \mathbf{w}_i \rangle\Big) \quad \textit{for all } t \in [T].$$

For a binary quantizer $Q \colon \mathbb{B} \to \{\pm 1\}$, let $H_{\mathbf{T},Q}$ denote the tree side information with respect to a tree $\mathbf{T}$ and an auxiliary sequence $\Omega = (\omega_t)_{t \geq 1}$ with $\omega_t = Q(\mathbf{g}_t)$.

Theorem C.7. *Let $V = \mathbb{R}^d$ be the d-dimensional Euclidean space. Suppose that a binary quantizer $Q \colon \mathbb{B} \to \{\pm 1\}$ satisfies $Q(\mathbf{e}_j) = 1$ and $Q(-\mathbf{e}_j) = -1$ for some $j \in [d]$. For T sufficiently large, for any* causal *OLO algorithm that satisfies the condition* (C.1) *in Theorem C.6, for any binary suffix tree $\mathbf{T}$, there exist a sequence $\mathbf{g}_1, \ldots, \mathbf{g}_T \in \mathbb{B}$ and a competitor $(\mathbf{u}_s^*)_{s \in \mathbf{T}}[H_{\mathbf{T},Q}] \in \mathcal{M}(H_{\mathbf{T},Q})$ such that*

$$\mathsf{Reg}((\mathbf{u}_s^*)_{s \in \mathbf{T}}[H_{\mathbf{T},Q}]); \mathbf{g}^T) \geq \sqrt{\sum_{s \in \mathbf{T}} T_s \|\mathbf{u}_s^*\|_2^2 \ln\Big(\frac{(T/|\mathbf{T}|)^{|\mathbf{T}|}}{(\mathsf{W}_0^{(T)})^2} \sum_{s \in \mathbf{T}} T_s \|\mathbf{u}_s^*\|_2^2 + 1\Big)} + \mathsf{W}_0^{(T)}.$$

Proof. Without loss of generality, assume that the binary quantizer $Q \colon \mathbb{B} \to \{\pm 1\}$ satisfies $Q(\mathbf{e}_1) = 1$ and $Q(-\mathbf{e}_1) = -1$. For a binary sequence $c^T \in \{\pm 1\}^T$, we set $\mathbf{g}_t = (c_t, 0, \ldots, 0)$ for $c_t \in \{\pm 1\}$, so that $\langle \mathbf{g}_t, \mathbf{w}_t \rangle = c_t x_{t1}$. Then, by Theorem C.6, we can write

$$x_{t1} = v_{t1}\Big(\mathsf{W}_0^{(T)} + \sum_{i=1}^{t-1} \langle \mathbf{g}_i, \mathbf{w}_i \rangle\Big) = v_{t1}\Big(\mathsf{W}_0^{(T)} + \sum_{i=1}^{t-1} c_i x_{i1}\Big)$$

for some v_{t1} such that $|v_{t1}| \le 1$. Hence, the OLO problem with any causal algorithms satisfying (C.1) with respect to the 1D sequences $\mathbf{g}^T$ can be equivalently viewed as the 1D coin betting with initial wealth $\mathsf{W}_0 = \mathsf{W}_0^{(T)}$.

Now, we state the celebrated Rissanen's lower bound for universal compression in the form of the wealth upper bound for the coin betting. Rissanen (1996) showed that for any probability assignment $q(x^T)$ on a binary sequence $x^T \in \{0,1\}^T$, there exists a sequence $\tilde{x}^T \in \{0,1\}$ such that

$$q(\tilde{x}^T) \le e^{-\frac{|\mathbf{T}|}{2} \ln \frac{T}{|\mathbf{T}|}} \max_{p_{\mathbf{T}}} p_{\mathbf{T}}(\tilde{x}^T),$$

where the maximum is over all possible tree sources $p_{\mathbf{T}}$ with the underlying tree $\mathbf{T}$. This can be translated into the wealth upper bound for the standard coin betting with binary outcomes $c_t \in \{\pm 1\}$ thanks to the equivalence between the coin betting and universal compression: for any continuous coin betting algorithm which plays a relative bet $b_t \in [-1,1]$ at time t, there exists a binary sequence $\tilde{c}^T \in \{\pm 1\}^T$ such that

$$\begin{aligned}
\frac{\mathsf{W}_T}{\mathsf{W}_0} = \prod_{t=1}^T (1 + b_t \tilde{c}_t) &\le \Big(\frac{|\mathbf{T}|}{T}\Big)^{\frac{|\mathbf{T}|}{2}} \prod_{s \in \mathbf{T}} \max_{b_s \in [-1,1]} \prod_{t \in [T]: h_t = s} (1 + b_s \tilde{c}_t) \\
&\overset{(a)}{\le} \Big(\frac{|\mathbf{T}|}{T}\Big)^{\frac{|\mathbf{T}|}{2}} \prod_{s \in \mathbf{T}} \exp\Big(\frac{\ln 2}{T'_s} \Big(\sum_{t \in [T]: h_t = s} \tilde{c}_t \Big)^2 \Big), \\
&= f\big((\textstyle\sum \tilde{c}^T(s; H_{\mathbf{T},Q}))_{s \in \mathbf{T}}\big), \qquad \text{(C.2)}
\end{aligned}$$

where h_t denotes the suffix of the sequence c^{t-1} with respect to $\mathbf{T}$ at time t, $T'_s := T_s \vee 1$, $T_s := |\{t \in [T] \colon h_t = s\}|$, $f((x_s)_{s \in \mathbf{T}}) := \prod_{s \in \mathbf{T}} h_s(x_s)$, and $h_s(x_s) = \beta_s \exp(\frac{x_s^2}{2\alpha_s})$ with $\alpha_s = \frac{2T_{s'}}{\ln 2}$, and $\beta_s = \sqrt{|\mathbf{T}|/T}$. Here, (a) follows by Lemma C.15.

For the adversarial coin sequence $(\tilde{c}_t)_{t \ge 1}$ satisfying (C.2), define $\mathbf{g}_t := (\tilde{c}_t, 0, \dots, 0)$. Then, we have

$$\begin{aligned}
\mathsf{W}_0^{(T)} + \sum_{t=1}^T \langle \tilde{\mathbf{g}}_t, \mathbf{w}_t \rangle &= \mathsf{W}_0^{(T)} + \sum_{t=1}^T \tilde{c}_t x_{t1} \\
&\le \mathsf{W}_0^{(T)} f\big((\textstyle\sum \tilde{c}^T(s; H_{\mathbf{T},Q}))_{s \in \mathbf{T}}\big) \\
&= \sum_{s \in \mathbf{T}} \big(\textstyle\sum \tilde{c}^T(s; H_{\mathbf{T},Q})\big) u_s^* - \mathsf{W}_0^{(T)} f^\star\Big(\Big(\frac{|u_s^*|}{\mathsf{W}_0^{(T)}}\Big)_{s \in \mathbf{T}}\Big) \\
&= \sum_{t=1}^T \langle \mathbf{g}_t, \mathbf{u}_{h_t}^* \rangle - \mathsf{W}_0^{(T)} f^\star\Big(\Big(\frac{\|\mathbf{u}_s^*\|_2}{\mathsf{W}_0^{(T)}}\Big)_{s \in \mathbf{T}}\Big),
\end{aligned}$$

where $(u_s^*)_{s \in \mathbf{T}} = \mathsf{W}_0^{(T)} \nabla f((\sum \tilde{c}^T(s; H_{\mathbf{T},Q}))_{s \in \mathbf{T}})$ and $\mathbf{u}_s^* := (u_s^*, 0, \dots, 0)$ for each $s \in \mathbf{T}$. Rearranging the terms, we have

$$\begin{aligned}
\mathsf{Reg}((\mathbf{u}_s^*)_{s \in \mathbf{T}}[H_{\mathbf{T},Q}]); \mathbf{g}^T) &= \sum_{t=1}^T \langle \mathbf{g}_t, \mathbf{u}_{h_t}^* \rangle - \sum_{t=1}^T \langle \mathbf{g}_t, \mathbf{w}_t \rangle \\
&\ge \mathsf{W}_0^{(T)} + \mathsf{W}_0^{(T)} f^\star\Big(\Big(\frac{\|\mathbf{u}_s^*\|_2}{\mathsf{W}_0^{(T)}}\Big)_{s \in \mathbf{T}}\Big). \qquad \square
\end{aligned}$$

C.2.4 Proof of Proposition 3.4

We use a backward induction over the depth $|s|$ to show that the recursion is well-defined. First, if $|s| = D$, $\mathbf{u}_s^{\mathsf{CTW}}(\mathbf{g}^{t-1}) = \Psi_s^{\mathsf{KT}}(\mathbf{g}^{t-1}) \mathbf{v}_s^{\mathsf{KT}}(\mathbf{g}^{t-1})$. By definition of $\mathbf{v}_s^{\mathsf{KT}}(\mathbf{g}^{t-1})$, $\mathbf{u}_s^{\mathsf{CTW}}(\mathbf{g}^{t-1})$ is a vector if s is the active node at depth D, and a scalar otherwise. Now, for $d \le D-1$, assume that $\mathbf{u}_{s'}^{\mathsf{CTW}}(\mathbf{g}^{t-1})$ is a scalar if s' is an active node and a vector otherwise for any $|s'| = d+1$ (induction hypothesis). Consider any node s of $\mathcal{T}_D$ with $|s| = d$. If s is an active node, then $\mathbf{u}_{\bar{1}s}^{\mathsf{CTW}}(\mathbf{g}^{t-1}) \mathbf{u}_{1s}^{\mathsf{CTW}}(\mathbf{g}^{t-1})$ is a vector by the induction hypothesis, since exactly one of $\bar{1}s$ and $1s$ is active. Hence, $\mathbf{u}_s^{\mathsf{CTW}}(\mathbf{g}^{t-1})$ is a vector. If s is not an active node, then, $\mathbf{u}_{\bar{1}s}^{\mathsf{CTW}}(\mathbf{g}^{t-1}) \mathbf{u}_{1s}^{\mathsf{CTW}}(\mathbf{g}^{t-1})$ is a scalar by the induction hypothesis, since neither of $\bar{1}s$ and $1s$ is active. Hence, $\mathbf{u}_s^{\mathsf{CTW}}(\mathbf{g}^{t-1})$ is a scalar. This completes the induction and thus the recursion is well-defined for all nodes s.

The claim $\mathbf{u}_\lambda^{\mathsf{CTW}}(\mathbf{g}^{t-1}) = \mathbf{u}^{\mathsf{CTW}}(\mathbf{g}^{t-1})$ can be checked by a similar induction argument. $\square$

C.2.5 Proof of Proposition 3.5

We claim that $\mathbf{v}_s^{\mathsf{CTW}}(\mathbf{g}^{t-1}) = \frac{\mathbf{u}_s^{\mathsf{CTW}}(\mathbf{g}^{t-1})}{\Psi_s^{\mathsf{CTW}}(\mathbf{g}^{t-1})}$ for any $s = s_d = \omega_{t-d}^{t-1} \in \mathcal{T}_D$, $d = 0, \dots, D$. This trivially holds for the leaf node $s_D = \omega_{t-D}^{t-1}$. For the internal nodes s_d with $d < D$, by plugging in the recursive formulas of $\mathbf{u}^{\mathsf{CTW}}(\mathbf{g}^{t-1})$ and $\Psi^{\mathsf{CTW}}(\mathbf{g}^{t-1})$, we can write

$$\frac{\mathbf{u}^{\mathsf{CTW}}(\mathbf{g}^{t-1})}{\Psi^{\mathsf{CTW}}(\mathbf{g}^{t-1})} = \frac{\beta_s(\mathbf{g}^{t-1})}{\beta_s(\mathbf{g}^{t-1})+1}\mathbf{v}_s^{\mathsf{KT}}(\mathbf{g}^{t-1}) + \frac{1}{\beta_s(\mathbf{g}^{t-1})+1}\frac{\mathbf{u}_{1s}^{\mathsf{CTW}}(\mathbf{g}^{t-1})}{\Psi_{1s}^{\mathsf{CTW}}(\mathbf{g}^{t-1})}\frac{\mathbf{u}_{1s}^{\mathsf{CTW}}(\mathbf{g}^{t-1})}{\Psi_{1s}^{\mathsf{CTW}}(\mathbf{g}^{t-1})}.$$

It is now enough to show that

$$\frac{\mathbf{u}_{s'}^{\mathsf{CTW}}(\mathbf{g}^{t-1})}{\Psi_{s'}^{\mathsf{CTW}}(\mathbf{g}^{t-1})} = 1 \text{ for } s' = \overline{\omega_{t-1-|s|}}s.$$

This holds since $\mathbf{u}_s^{\mathsf{CTW}} = \Psi_s^{\mathsf{CTW}}$ for any off-path node $s \notin \rho(\omega_{t-D}^{t-1})$ by definition (3.8). □

C.2.6 Proof of Proposition 3.6

Similar to the processing betas algorithm (Willems et al., 2006), we only need to show that

$$\frac{\Psi_{\overline{\omega_{t-1-|s|}}s}^{\mathsf{CTW}}(\mathbf{g}^t)}{\Psi_{\overline{\omega_{t-1-|s|}}s}^{\mathsf{CTW}}(\mathbf{g}^{t-1})} = 1 \text{ for } s' = \overline{\omega_{t-1-|s|}}s \text{ for any } s \notin \rho(\omega_{t-D}^{t-1}).$$

Since the new symbol $\mathbf{g}_t$ is added to a node s if and only if $s \in \rho(\omega_{t-D}^{t-1})$, if $s \notin \rho(\omega_{t-D}^{t-1})$, then the CTW potential on the node s will not be updated. This proves the claim. □

C.3 Technical lemmas

Lemma C.8 (Orabona and Pál, 2016, Lemma 10)**.** *Let $h\colon (-a, a) \to \mathbb{R}$ be an even, twice differentiable function that satisfies $xh''(x) \geq h'(x)$ for all $x \in [0, a)$. Let $c\colon [0,\infty) \times [0,\infty) \to \mathbb{R}$ be an arbitrary function. If $u, v \in \mathcal{H}$ satisfy $\|u\| + \|v\| < a$, then*

$$c(\|u\|, \|v\|) \cdot \langle u, v\rangle - h(\|u + v\|) \geq \min_{r\in\{\pm 1\}} \{rc(\|u\|, \|v\|)\|u\|\|v\| - h(\|u\| + r\|v\|)\}.$$

Proof sketch. It is easy to check that the inequality holds if $u = 0$ or $v = 0$. Hence, we assume $u, v \neq 0$. With $\alpha := \langle u, v\rangle/(\|u\|\|v\|)$, we can write the left hand side of the desired inequality as

$$f(\alpha) := c(\|u\|, \|v\|)\|u\|\|v\|\alpha - h(\sqrt{\|u\|^2 + \|v\|^2 + 2\alpha\|u\|\|v\|}).$$

Since the function h is assumed to be even, it is equivalent to showing that

$$\inf_{\alpha\in[-1,1]} f(\alpha) = \min\{f(+1), f(-1)\}.$$

By using the condition $xh''(x) \geq h'(x)$, one can easily show that f is concave by checking $f''(\alpha) \leq 0$, which concludes the proof. □

Lemma C.9 (Bauschke and Combettes, 2011, Example 13.7)**.** *Let $\phi\colon \mathbb{R} \to (-\infty, +\infty]$ be even. Then $(\phi \circ \|\cdot\|)^\star = \phi^\star \circ \|\cdot\|$.*

Lemma C.10 (Orabona, 2019, Lemma 5.8)**.** *Let f be a function and let $f^\star$ be its Fenchel conjugate. For $a > 0$ and $b \in \mathbb{R}$, the Fenchel conjugate of $g(x) = af(x) + b$ is $g^\star(z) = af^\star(z/a) - b$.*

Lemma C.11 (Orabona, 2019, Theorem 5.8)**.** *For a convex, proper, closed function $h\colon \mathbb{R}^d \to (-\infty, +\infty]$, we have $\langle \theta, x\rangle \geq h(x) + h^\star(\theta)$, where the equality is attained if and only if $x \in \partial h^\star(\theta)$.*

Since $f(x) \geq h(x)$ for any $x \in \mathbb{R}$ implies $f^\star(u) \geq h^\star(u)$ for any $u \in \mathbb{R}$, it is enough to find the conjugate dual of a function $h(x) = \beta \exp(\frac{x^2}{2\alpha})$ for $\alpha, \beta > 0$.

The *Lambert function* $W\colon (-1/e, \infty) \to [0, \infty)$ is defined by the equation $x = W(x)e^{W(x)}$ for $x \geq 0$.

Lemma C.12 (Orabona and Pál, 2016, Lemma 17)**.** *For $x \geq 0$,*

$$0.6321 \ln(x+1) \leq W(x) \leq \ln(x+1).$$

Remark C.1. Here, $0.6321\ldots \approx 1/b^*$, where b^* is the solution to the equation

$$\frac{eb}{(e+1)b+1} = \frac{b}{(b+1)\ln(b+1)}.$$

Proposition C.13 (Orabona and Pál, 2016, Lemma 18)**.** *For $h(x) = \beta \exp(\frac{x^2}{2\alpha})$ with $\alpha, \beta > 0$,*

$$h^\star(y) = y\sqrt{\alpha W\Big(\frac{\alpha y^2}{\beta^2}\Big)} - \beta \exp\Big(\frac{1}{2}W\Big(\frac{\alpha y^2}{\beta^2}\Big)\Big) = y\sqrt{\alpha}\Big(\sqrt{W\Big(\frac{\alpha y^2}{\beta^2}\Big)} - \sqrt{\frac{1}{W(\frac{\alpha y^2}{\beta^2})}}\Big).$$

In particular,

$$h^\star(y) \leq y\sqrt{\alpha \ln\Big(\frac{\alpha y^2}{\beta^2}+1\Big)} - \beta.$$

For a generalization with the product potential, we also have the following proposition.

Proposition C.14. *Define $f_i(y_i) = \beta_i \exp(\frac{y_i^2}{2\alpha_i})$ with $\alpha_i, \beta_i > 0$ for each $i \in S$, and define $f(y_1, \ldots, y_S) = f_1(y_1)\cdots f_S(y_S)$. Then, we have*

$$f^\star(y_1,\ldots,y_S) = \sqrt{\alpha_1 y_1^2 + \ldots + \alpha_S y_S^2}\Big(\sqrt{W\Big(\frac{\alpha_1 y_1^2 + \ldots + \alpha_S y_S^2}{\beta_1^2\cdots\beta_S^2}\Big)} - \frac{1}{\sqrt{W\Big(\frac{\alpha_1 y_1^2+\ldots+\alpha_S y_S^2}{\beta_1^2\cdots\beta_S^2}\Big)}}\Big).$$

In particular,

$$f^\star(y_1,\ldots,y_S) \leq \sqrt{(\alpha_1 y_1^2 + \ldots + \alpha_S y_S^2)\ln\Big(\frac{\alpha_1 y_1^2 + \ldots + \alpha_S y_S^2}{\beta_1^2\cdots\beta_S^2}+1\Big)} - \beta_1\cdots\beta_S$$

Proof. For the sake of simplicity, we prove only for $S = 2$. The proof can be generalized to any $S \geq 2$ with little modification. To find

$$f^\star(y_1, y_2) = \sup_{x_1, x_2}(y_1x_1 + y_2x_2 - f_1(x_1)f_2(x_2)),$$

we consider the stationarity conditions

$$\frac{\partial}{\partial x_i}(y_1x_1 + y_2x_2 - f_1(x_1)f_2(x_2)) = 0$$

for $i \in \{1, 2\}$, which leads to

$$\begin{cases} y_1 &= f_1'(x_1)f_2(x_2), \\ y_2 &= f_1(x_1)f_2'(x_2). \end{cases}$$

Since $f_i'(x) = \frac{x}{\alpha_i}f_i(x)$, we have

$$\begin{cases} y_1 &= \frac{x_1}{\alpha_1}f_1(x_1)f_2(x_2), \\ y_2 &= \frac{x_2}{\alpha_2}f_1(x_1)f_2(x_2). \end{cases}$$

Manipulating the equations, we have

$$\Big(\frac{x_1^2}{\alpha_1} + \frac{x_2^2}{\alpha_2}\Big)\exp\Big(\frac{x_1^2}{\alpha_1} + \frac{x_2^2}{\alpha_2}\Big) = \frac{\alpha_1 y_1^2 + \alpha_2 y_2^2}{\beta_1^2\beta_2^2},$$

which leads to

$$\frac{x_1^2}{\alpha_1} + \frac{x_2^2}{\alpha_2} = W\Big(\frac{\alpha_1 y_1^2 + \alpha_2 y_2^2}{\beta_1^2\beta_2^2}\Big).$$

Hence,

$$f(x_1^*, x_2^*) = \beta_1\beta_2 \exp\Big(\frac{1}{2} W\Big(\frac{\alpha_1 y_1^2 + \alpha_2 y_2^2}{\beta_1^2\beta_2^2}\Big)\Big) = \sqrt{\frac{\alpha_1 y_1^2 + \alpha_2 y_2^2}{W(\frac{\alpha_1 y_1^2 + \alpha_2 y_2^2}{\beta_1^2\beta_2^2})}}.$$

Finally, we can compute

$$y_1 x_1^* + y_2 x_2^* = \frac{\alpha_1 y_1^2 + \alpha_2 y_2^2}{f(x_1^*, x_2^*)} = \sqrt{(\alpha_1 y_1^2 + \alpha_2 y_2^2) W\Big(\frac{\alpha_1 y_1^2 + \alpha_2 y_2^2}{\beta_1^2\beta_2^2}\Big)},$$

whence

$$\begin{aligned} f^\star(y_1, y_2) &= y_1 x_1^* + y_2 x_2^* - f(x_1^*, x_2^*) \\ &= \sqrt{\alpha_1 y_1^2 + \alpha_2 y_2^2}\Big(\sqrt{W\Big(\frac{\alpha_1 y_1^2 + \alpha_2 y_2^2}{\beta_1^2\beta_2^2}\Big)} - \frac{1}{\sqrt{W(\frac{\alpha_1 y_1^2 + \alpha_2 y_2^2}{\beta_1^2\beta_2^2})}}\Big). \end{aligned}$$

□

Lemma C.15 (Orabona, 2019, Lemma 9.4)**.** *For any $T \geq 1$ and any $c^T \in [-1, 1]^T$, we have*

$$\max_{b \in [-1,1]} \prod_{t \in [T]} (1 + b c_t) \leq \exp\Big(\frac{\ln 2}{T} (\sum c^T)^2\Big).$$

D THE CTW OLO ALGORITHM

Algorithm D.3 CTW OLO algorithm

Parameters maximum depth $D \geq 1$, auxiliary sequence $\Omega = (\omega_t)_{t \geq 1}$, initial wealth $\mathsf{W}_0 > 0$.

procedure CTWOLO(D, Ω, W_0)
  Initialize a context tree $\mathcal{T}_D$ of depth D with $G_s \leftarrow \phi$ and $\beta_s \leftarrow 1$ for each $s \in \mathcal{T}_D$
  for each $t = 1, 2, \ldots$ **do**
    Compute $\mathbf{v}^{\mathsf{CTW}}(\mathbf{g}^{t-1}) = \mathbf{v}^{\mathsf{CTW}}_{\lambda}(\mathbf{g}^{t-1})$ by computing, for $s_0, \ldots, s_D \in \rho(\omega_{t-D}^{t-1})$,

$$\mathbf{v}^{\mathsf{CTW}}_{s_d}(\mathbf{g}^{t-1}) \leftarrow \begin{cases} \frac{\beta_{s_d}(\mathbf{g}^{t-1})}{\beta_{s_d}(\mathbf{g}^{t-1})+1} \mathbf{v}^{\mathsf{KT}}_{s_d}(\mathbf{g}^{t-1}) + \frac{1}{\beta_{s_d}(\mathbf{g}^{t-1})+1} \mathbf{v}^{\mathsf{CTW}}_{s_{d+1}}(\mathbf{g}^{t-1}) & \text{if } d < D \\ \mathbf{v}^{\mathsf{KT}}_{s_D}(\mathbf{g}^{t-1}) & \text{if } d = D \end{cases} \tag{3.10}$$

    Set $\mathbf{w}^{\mathsf{CTW}}_t(\mathbf{g}^{t-1}) \leftarrow \mathbf{v}^{\mathsf{CTW}}(\mathbf{g}^{t-1}) \mathsf{W}_{t-1}$
    Receive $\mathbf{g}_t$ and update the cumulative wealth $\mathsf{W}_t \leftarrow \mathsf{W}_{t-1} + \langle \mathbf{g}_t, \mathbf{w}^{\mathsf{CTW}}_t(\mathbf{g}^{t-1}) \rangle$
    Update $G_s \leftarrow G_s + \mathbf{g}_t$ and update β_s for $s_d = \omega_{t-d}^{t-1}$, $d = 0, \ldots, D-1$, as

$$\beta_{s_d}(\mathbf{g}^{t-1}) \leftarrow \beta_{s_d}(\mathbf{g}^t) = \beta_{s_d}(\mathbf{g}^{t-1}) \frac{\Psi^{\mathsf{KT}}_{s_d}(\mathbf{g}^t)}{\Psi^{\mathsf{KT}}_{s_d}(\mathbf{g}^{t-1})} \frac{\Psi^{\mathsf{CTW}}_{s_{d+1}}(\mathbf{g}^{t-1})}{\Psi^{\mathsf{CTW}}_{s_{d+1}}(\mathbf{g}^t)}, \tag{3.11}$$

    where

$$\frac{\Psi^{\mathsf{CTW}}_{s_d}(\mathbf{g}^t)}{\Psi^{\mathsf{CTW}}_{s_d}(\mathbf{g}^{t-1})} = \begin{cases} \frac{\beta_{s_d}(\mathbf{g}^{t-1})}{\beta_{s_d}(\mathbf{g}^{t-1})+1} \frac{\Psi^{\mathsf{KT}}_{s_d}(\mathbf{g}^t)}{\Psi^{\mathsf{KT}}_{s_d}(\mathbf{g}^{t-1})} + \frac{1}{\beta_{s_d}(\mathbf{g}^{t-1})+1} \frac{\Psi^{\mathsf{CTW}}_{s_{d+1}}(\mathbf{g}^t)}{\Psi^{\mathsf{CTW}}_{s_{d+1}}(\mathbf{g}^{t-1})} & \text{if } d < D \\ \frac{\Psi^{\mathsf{KT}}_{s_D}(\mathbf{g}^t)}{\Psi^{\mathsf{KT}}_{s_D}(\mathbf{g}^{t-1})} & \text{if } d = D \end{cases} \tag{3.12}$$

    for $s_d = \omega_{t-d}^{t-1}$, $d = 0, \ldots, D$
    Receive ω_t
  end for
end procedure

E EXPERIMENT DETAILS AND ADDITIONAL FIGURES

Problem setting We applied the proposed OLO algorithms to solve the online linear regression problem as described in Appendix B especially with absolute loss $\ell_t(\mathbf{w}_t) = |\langle \mathbf{w}_t, \mathbf{x}_t \rangle - y_t|$, where $\mathbf{w}_t$ denotes the action of an OLO algorithm and $\mathbf{x}_t$ denotes the feature vector. Hence, we linearized the convex loss and fed the subgradient $\partial \ell_t(\mathbf{w}_t) = \operatorname{sgn}(\langle \mathbf{w}_t, \mathbf{x}_t \rangle - y_t)\mathbf{x}_t$ to an OLO algorithm.

Data preprocessing For each dataset, we linearly interpolated any missing values. We discarded time stamps as well as some categorical features such as `cbwd` of Beijing PM2.5 and `weather_description` of Metro Inter State Traffic Volume, and binarized the others, if possible, such as `holiday`, `weather_main`, and `snow_1h` of Metro Inter State Traffic Volume. We also applied a logarithmic mapping $x \mapsto \ln(1+x)$ for the features `lws`, `ls`, `lr` of Beijing PM2.5 and applied another logarithmic mapping $x \mapsto \ln x$ to the feature `rain_1h`, to make the features more suitable for linear regression. We then normalized each feature $\tilde{\mathbf{x}}_t$ so that $\|\tilde{\mathbf{x}}_t\|_2 = 1$ and added all-one coordinates as the bias component with an additional scaling by $1/\sqrt{2}$. After this preprocessing step, we obtained 7-dimensional feature vectors for both datsets. See the attached Python code for the details in Supplementary Material.

Computing resource All experiments were run on a single laptop with a CPU Intel(R) Core(TM) i7-9750H CPU 2.60GHz with 12 (logical) cores and 16GB of RAM.

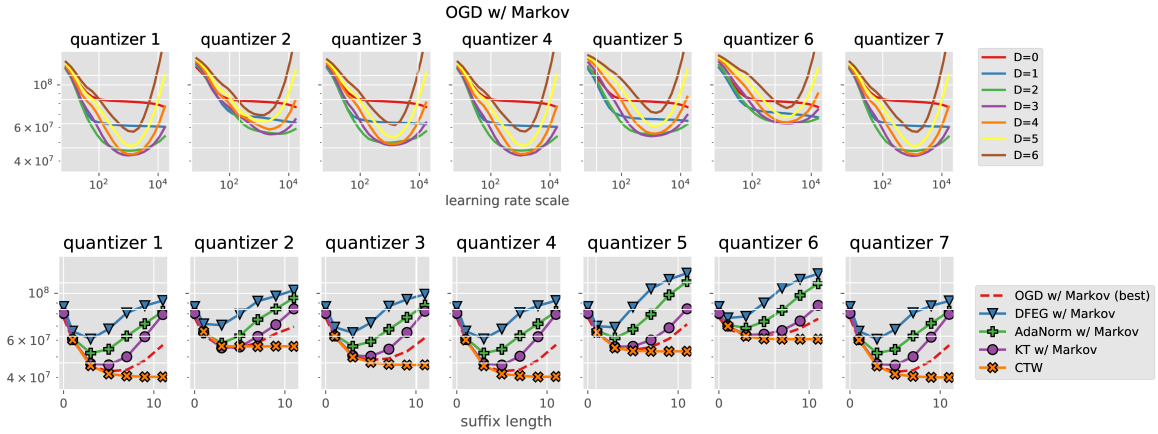

Figure E.4: Metro Inter State Traffic Volume dataset (Hogue, 2019). The y-axes represent cumulative losses. (a) Performance of per-state OGD adaptive to Markov side information with various learning rate scales. (b) Performance of parameter-free algorithms.

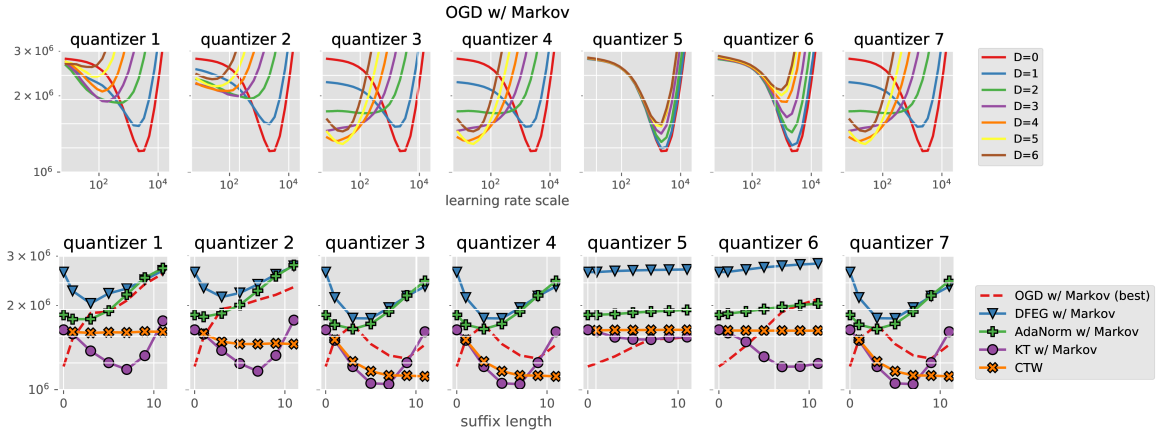

Figure E.5: Beijing PM2.5 dataset (Liang et al., 2015). See the caption of Figure E.4 for details.