PDE-Based Optimal Strategy for Unconstrained Online Learning

Zhiyu Zhang
Boston University
zhiyuz@bu.edu

Ashok Cutkosky
Boston University
ashok@cutkosky.com

Ioannis Ch. Paschalidis
Boston University
yannisp@bu.edu

Abstract

Unconstrained Online Linear Optimization (OLO) is a practical problem setting to study the training of machine learning models. Existing works proposed a number of potential-based algorithms, but in general the design of such potential functions is ad hoc and heavily relies on guessing. In this paper, we present a framework that generates time-varying potential functions by solving a Partial Differential Equation (PDE). Our framework recovers some classical potentials, and more importantly provides a systematic approach to design new ones.

The power of our framework is demonstrated through a concrete example. When losses are 1-Lipschitz, we design a novel OLO algorithm with anytime regret upper bound $C\sqrt{T} + \|u\|\sqrt{2T}[\sqrt{\log(1 + \|u\|/C)} + 2]$, where C is a user-specified constant and u is any comparator whose norm is unknown and unbounded a priori. By constructing a matching lower bound, we further show that the leading order term, *including* the constant multiplier $\sqrt{2}$, is tight. To our knowledge, this is the first parameter-free algorithm with optimal leading constant. The obtained theoretical benefits are validated by experiments.

1 Introduction

Advances in online learning have brought deeper understanding and better algorithms to the training of machine learning models. Among all the problem settings therein, unconstrained online learning has received special attention since the parameter of the model is often unrestricted before seeing any data. Compared to conventional settings with a bounded domain, the unconstrained setting poses an additional challenge: starting from a bad initialization, how can an algorithm quickly find the optimal parameter that may be *far-away*? With the growing popularity of high-dimensional machine learning models, such an issue becomes increasingly important.

In this paper, we address this issue by studying a theoretical problem called *unconstrained Online Linear Optimization* (OLO). Given an unbounded domain $\mathbb{R}^d$, we need to design an algorithm such that in each round it makes a prediction $x_t \in \mathbb{R}^d$, observes a loss gradient $g_t \in \mathbb{R}^d$ and suffers a loss $\langle g_t, x_t\rangle$, where g_t is adversarial (can arbitrarily depend on $x_1, \dots, x_t$) and satisfies $\|g_t\| \leq 1$. The considered performance metric is the regret

$$\text{Regret}_T(u) = \sum_{t=1}^{T} \langle g_t, x_t\rangle - \sum_{t=1}^{T} \langle g_t, u\rangle ,$$

and the goal of the algorithm is to achieve low regret for all comparator $u \in \mathbb{R}^d$, time horizon $T \in \mathbb{N}_+$ and loss gradients $g_1, \dots, g_T$. In particular, we are interested in the dependence of $\text{Regret}_T(u)$ on $\|u\|$, as it captures how well the algorithm *adapts* to a far-away optimal comparator. It is well-known that OLO algorithms can be used to solve *Online Convex Optimization* (OCO) problems [Zin03] with Lipschitz losses, and the latter is a powerful model with numerous real-world applications. We refer the readers to general expositions on this topic, such as [Haz19, Ora19].

For designing online learning algorithms, one of the main approaches is the potential method [CBL03, CBL06]. Given a potential function $V_t(\cdot)$, the key idea is to accumulate the history of the decision process into a "sufficient statistic" S_t[1] and predict the gradient of $V_t(\cdot)$ at this point, i.e., $x_t = \nabla V_t(S_t)$. Through this procedure, designing

[1] For unconstrained OLO, typically $S_t = -\sum_{i=1}^{t-1} g_i$.

new algorithms is translated into a more tangible task of finding good potentials. Specifically for unconstrained OLO, existing works (e.g., [MO14, Ora14, OP16, MK20]) adopted the one-dimensional potential

$$V_t(S_t) = Ct^{-1/2} \exp[S_t^2/(2t)] \tag{1}$$

and its variants to achieve regret bound

$$\text{Regret}_T(u) \leq C + \|u\| O\left(\sqrt{T \log \frac{\|u\|T}{C}}\right), \tag{2}$$

where C is an arbitrary constant. Among all the achievable upper bounds with at most constant $\text{Regret}_T(0)$, the order of $\|u\|$ and T in (2) is optimal up to multiplicative constants [SM12, Ora13, Ora19]. Practically, those algorithms are often called *parameter-free* as they nearly match the performance of the optimally-tuned gradient descent (in hindsight) without any tuning.

Despite its popularity, the above result is not a conclusive solution to our problem though. In many cases, there is no absolute need to suffer only constant $\text{Regret}_T(0)$ all the time. More generally, the RHS of (2) should be viewed as a trade-off between the values of $\text{Regret}_T(u)$ at small $\|u\|$ and large $\|u\|$: if the *cumulative loss* $\text{Regret}_T(0)$ is allowed to slowly increase with T, then one may expect smaller regret with respect to far-away comparators. Such a behavior is favorable in high-dimensional problems, as accurate initialization is hard and the distance to the optimal comparator can be very large. However, it remains unclear whether this *loss-regret trade-off* can be achieved in a practical and optimal way.

Towards this end, we go back and rethink the design of potential functions in unconstrained OLO. The classical workflow is heuristic: one first guesses the form of a good potential, and then verifies its quality using existing techniques [MO14, OP16]. The first step can be very challenging, especially when the suitable potential is not an elementary function (e.g., when involving complicated integrals or series). Our goal is to propose a systematic approach for this task, which (*i*) reduces the amount of guessing; and (*ii*) produces a novel algorithm with an optimal loss-regret trade-off.

1.1 Result and contribution

Following the above discussion, our contributions are twofold.

- We propose a framework that uses solutions of a specific *Partial Differential Equation* (PDE) as potential functions for unconstrained OLO. To this end, we characterize a class of minimax optimal potentials via a backward recursion, and our PDE naturally arises in its continuous-time limit. Solutions of this PDE approximately solve the discrete-time recursion. Therefore, one may search for suitable potentials within such solutions and their variants, which is a much more structured procedure than direct guessing.

  Our framework recovers the classical choice (1). Based on the potential interpretation of *Follow the Regularized Leader* (FTRL), it can also generate nonstandard regularizers for unconstrained FTRL.

- Using our framework, we design a novel one-dimensional potential which is not elementary and hard to guess without the help of a PDE. The induced algorithm guarantees (for any user-specified C):

$$\text{Regret}_T(u) \leq C\sqrt{T} + \|u\| \sqrt{2T} \left[\sqrt{\log\left(1 + \frac{\|u\|}{\sqrt{2}C}\right)} + 2\right].$$

  Compared to (2), our bound captures a different loss-regret trade-off particularly useful when a good initialization is not available. By constructing a matching lower bound, we further show that the leading order term, *including* the constant multiplier $\sqrt{2}$, is tight. To our knowledge, our algorithm is the first parameter-free algorithm with optimal leading constant. The obtained theoretical benefits are validated by experiments.

1.2 Related work

Unconstrained OLO The study of unconstrained domain has a long and rich history within the optimization community. In both the offline and online setting, strong performance guarantees can be obtained assuming

certain curvature on the loss function. When no curvature is present, the problem becomes considerably harder but more practical. Specifically, the algorithm should only use the gradient in its update, which is the only available feedback in many large scale problems.

In unconstrained OLO, if the optimal learning rate in hindsight is known a priori, *Online Gradient Descent* (OGD) [Zin03] guarantees $\|u^*\|\sqrt{T}$ regret with respect to the optimal comparator u^*. Without that prior knowledge, the regret bound downgrades to $O(\|u^*\|^2\sqrt{T})$. Parameter-free algorithms aim at achieving $\tilde{O}(\|u^*\|\sqrt{T})$ regret in the latter setting. Specifically, [SM12] proposed the first algorithm with $O(\|u^*\|\sqrt{T}\log(\|u^*\|T))$ regret, which was later improved to the optimal [SM12, Ora13, Ora19] rate $O(\|u^*\|\sqrt{T\log(\|u^*\|T)})$ by a potential-based algorithm [MO14]. The analysis was streamlined in [OP16, CO18] through a *coin-betting game*, and in [FRS18] through the *Burkholder method*. The obtained algorithms find applications in differential privacy [JO19, vdH19], combining optimizers [Cut19, Cut20, ZCP21] and training neural networks [OT17].

Among all these results, a shared limitation is the focus on $\text{Regret}_T(0) \leq$ constant. Other forms of loss-regret trade-off are less well explored, both theoretically and practically. Moreover, the optimality of leading constants has not been established.

Differential equations for online learning Recently, applying differential equations in online learning has received growing interests. The first idea was proposed in [KP10], where a nonstandard potential function for *Learning with Expert Advice* (LEA) [LW94] was designed by solving an *Ordinary Differential Equation* (ODE). As a key benefit, the obtained regret bound achieves a novel trade-off with respect to different experts. Such techniques were later applied to the discounted setting [AP13] and the movement-constrained setting [DM19]. Interestingly, our prior work [ZCP21] used the coin-betting approach to achieve a similar goal as [DM19], suggesting intriguing connections between differential equations and parameter-free online learning.

An improved approach uses PDEs (rather than ODEs) to generate time-dependent potential functions. Still considering the LEA problem, such works can achieve the optimal regret bound that is nonasymptotic in the number of experts. [Zhu14] first derived such a PDE to characterize the continuous-time limit of LEA, whose arguments were streamlined in [DK20b]. Exact solutions were obtained in special cases [BEZ20a, BEZ20b, DK20b], and more generally, approximate solutions were derived in [Rok17, KKW20a, KKW20b]. Follow-up works considered history-dependent experts [DC20, DK20a] and malicious experts [BPZ20, BEZ21]. Furthermore, [HLPR20] extended this technique to the anytime setting with two experts.

Compared to these works on LEA, our use of PDE in unconstrained OLO has two major differences.

- Existing works considered settings that enforce a unique solution to the PDE, for example by requiring a fixed time horizon (e.g., [Zhu14, DK20b, KKW20a]) or imposing artificial boundary conditions [HLPR20]. In contrast, we directly consider a class of solutions which are generally not comparable to each other.
- In LEA, the goal of the PDE approach is to achieve optimal uniform regret (with respect to all experts). In contrast, we use a PDE to further achieve trade-offs on the performance metric. The trade-off between experts has been considered using ODEs (e.g., [KP10]). However, we focus on the anytime setting, and the trade-off in unconstrained OLO is with respect to all comparators in $\mathbb{R}^d$, which is much harder.

1.3 Notation

Let $\|\cdot\|$ be the Euclidean norm, and let B^d be the unit d-dimensional Euclidean norm ball. For a twice differentiable function $V(t,S)$ where t represents time and S represents a spatial variable, let $\nabla_t V$, $\nabla_{tt} V$, $\nabla_S V$ and $\nabla_{SS} V$ be the first and second order partial derivatives. Let $\lambda_{\max}(\cdot)$ be the largest eigenvalue of a real symmetric matrix. For a function f, let f^* be its Fenchel conjugate. For two integers $a \leq b$, $[a:b]$ is the set of all integers c such that $a \leq c \leq b$; the brackets are removed when on the subscript, denoting a finite sequence with indices in $[a:b]$. Finally, log denotes natural logarithm when the base is omitted.

2 OLO, betting and limiting PDE

In this section, we derive the important PDE for our framework. The first step is to define a coin-betting problem with infinite player budget, which generalizes the classical setting from [OP16]. Solving this problem is

equivalent to solving the unconstrained OLO problem we care about. However, the analysis of coin-betting does not involve comparators, and therefore is technically easier than OLO.

Next, we characterize good coin-betting strategies through a minimax analysis. Analogous to potential functions in unconstrained OLO, these coin-betting strategies rely on value functions defined by a backward recursion. Directly solving this recursion is infeasible due to the lack of a terminal condition. Instead, we derive a PDE in a scaling limit whose solutions can approximately solve this recursion, thus suggesting good OLO algorithms.

2.1 Unconstrained coin-betting and duality

Unconstrained coin-betting is a two-person zero-sum game, with $\mathcal{X} = \mathbb{R}^d$ and $\mathcal{C} = \mathsf{B}^d$ being the action space of the player and the adversary respectively. The player's policy $\boldsymbol{p}$ contains an initial bet $x_1 \in \mathcal{X}$ and a collection of functions $\{p_2, p_3, \ldots\}$, with $p_t : \mathcal{C}^{t-1} \to \mathcal{X}$. Similarly, the adversary's policy $\boldsymbol{a}$ is defined as a collection of functions $\{a_1, a_2, \ldots\}$, with $a_t : \mathcal{X}^t \to \mathcal{C}$.

Fixing policies $\boldsymbol{p}$ and $\boldsymbol{a}$ on both sides, the game runs as follows. In the t-th round, the player makes a bet $x_t = p_t(c_{1:t-1})$ based on past coin outcomes. Then, the adversary decides a new coin outcome $c_t = a_t(x_{1:t})$, reveals it to the player, and the player gains $\langle c_t, x_t \rangle$ amount of money (effectively, the player loses money if $\langle c_t, x_t \rangle$ is negative). The performance metric for the player is the total gained wealth

$$\text{Wealth}_T = \sum_{t=1}^{T} \langle c_t, x_t \rangle ,$$

where T is not pre-specified. In other words, the player aims to ensure an *anytime* wealth lower bound against all possible adversaries. This is fundamentally different from the fixed T setting [Cov66] where all achievable lower bounds can be characterized via dynamic programming.

Due to a classical dual relation [MO14], solving this coin-betting game is equivalent to solving unconstrained OLO. To see this, we can construct a unique OLO algorithm (Algorithm 1) from any coin-betting algorithm $\mathcal{A}$ and characterize its performance in Lemma 2.1.

Algorithm 1 From coin-betting to OLO.

Require: An algorithm $\mathcal{A}$ for unconstrained coin-betting.
1: **for** $t = 1, 2, \ldots$ **do**
2: Query $\mathcal{A}$ for its t-th bet x_t and predict it *exactly* in OLO.
3: Observe loss gradient g_t and suffer $\langle g_t, x_t \rangle$.
4: Let $c_t = -g_t$ and send it to $\mathcal{A}$ as the t-th coin outcome.
5: **end for**

Lemma 2.1 (Theorem 9.6 of [Ora19]). *Let Ψ be any proper, closed and convex function. For all $T \in \mathbb{N}_+$, the following two statements are equivalent:*

1. *The unconstrained coin-betting algorithm $\mathcal{A}$ guarantees $\text{Wealth}_T \geq \Psi(\sum_{t=1}^{T} c_t)$ against any adversary.*
2. *The unconstrained OLO algorithm constructed from $\mathcal{A}$ guarantees $\text{Regret}_T(u) \leq \Psi^*(u)$ for all $u \in \mathbb{R}^d$, against any adversarial loss sequence. Ψ^* is the Fenchel conjugate of Ψ.*

Note that this coin-betting game strictly generalizes the existing one for unconstrained OLO analysis [MA13, OP16]. In the latter, the player is assigned an initial wealth C and only bets a fraction of his wealth in each round; i.e., by choosing a *betting fraction* $\beta_t \in \mathsf{B}^d$, the player's bet is $x_t = \beta_t(C + \sum_{i=1}^{t-1} \langle c_i, g_i \rangle)$. A budget constraint of this form faithfully models many real-world investment problems, but since our ultimate goal is OLO rather than betting, such a constraint is not necessary for our purpose. In fact, relaxing it gives us greater flexibility to achieve general forms of regret trade-offs beyond (2). Intuitively speaking, the player in our setting can make decisions solely based on the perceived risk-gain trade-off, without being constrained by its budget.

2.2 Minimax coin-betting

Next, we characterize the unconstrained coin-betting game from a minimax perspective. Traditionally the value of the game is considered. However, as the adversary can choose $c_t = -x_t/\|x_t\|$ to sabotage any non-zero bet, the value of the game is 0 which is not informative. To replace it, we consider a more refined quantity called *value function*.

Definition 2.1 (Value function)**.** *A function $V : \mathbb{N} \times \mathbb{R}^d \to \mathbb{R}$ is a value function of the unconstrained coin-betting game if*

1. *$V(0, 0) = 0$.*
2. *For all $t \in \mathbb{N}$, $V(t, \cdot)$ is continuous on $\mathbb{R}^d$.*
3. *For all $t \in \mathbb{N}$ and $S \in t \cdot \mathcal{C}$,*

$$V(t, S) = \min_{x \in \mathcal{X}} \max_{c \in \mathcal{C}} \left[V(t + 1, S + c) - \langle c, x \rangle \right]. \tag{3}$$

Remark 2.1. *Even though $\mathcal{X}$ is not compact, the minimization on the RHS of (3) is well-posed since $\max_{c \in \mathcal{C}} \left[V(t + 1, S + c) - \langle c, x \rangle \right]$ as a function of x is coercive.*

The recursive relation in Definition 2.1 is reminiscent of the *conditional value function* previously studied in online learning (e.g., [RSS12, MA13, DK20b]) and minimax dynamic programming (e.g., [Ber12]). The key difference is that we care about anytime performance, therefore a terminal condition to initiate the backward recursion (3) is missing. Rather than the *value-to-go*, we model the *value-so-far*. This makes the analysis of (3) a lot harder: specifically, its solution is not unique (e.g., $V(t, S) = kS$, where k is an arbitrary constant). In general, similar to the concept of *Pareto optimality*, different value functions are not comparable as they represent different trade-offs on the *shape* of the wealth lower bound (ultimately, the associated regret upper bound due to Lemma 2.1).

On the bright side, any value function can lead to a pair of player-adversary strategies with tight wealth lower and upper bounds. If we can find a good value function (or more generally, its approximation), then a good betting algorithm can be naturally induced. The proof is deferred to Appendix A.1.

Lemma 2.2. *Given any value function V satisfying Definition 2.1,*

1. *We can construct a player policy $\boldsymbol{p}^*$ such that for all $\boldsymbol{a}$ and $T \in \mathbb{N}_+$,*

$$\mathit{Wealth}_T \geq V\left(T, \sum_{t=1}^{T} c_t\right).$$

   In addition, for all t, the player's bet $p_t^*(c_{1:t-1})$ depends on the past coin outcomes only through their sum $\sum_{i=1}^{t-1} c_i$.

2. *We can construct an adversary policy $\boldsymbol{a}^*$ such that for all $\boldsymbol{p}$ and $T \in \mathbb{N}_+$,*

$$\mathit{Wealth}_T \leq V\left(T, \sum_{t=1}^{T} c_t\right).$$

2.3 The scaled game and limiting PDE

Let us define the *unit time* as the time interval between consecutive rounds in the coin betting game, and assign it to 1. In this way, the game can be formulated on the real time axis $t \in \mathbb{R}_+$. Intuitively, solving the backward recursion (3) is difficult due to its discrete formulation. If we adopt a finer discretization on the time axis, then the recursion becomes "smoother" which is easier to describe using continuous-time analysis. To this end, let us define a scaled coin-betting game.

Definition 2.2 (Scaled game)**.** *Given a constant $\varepsilon > 0$, the ε-scaled game is defined as the unconstrained coin-betting game with unit time ε^2 and adversary action space $\varepsilon \cdot \mathcal{C}$. That is, actions are taken every ε^2 original unit time, and the adversary chooses the coin outcome in a scaled set $\varepsilon \cdot \mathcal{C}$ instead of $\mathcal{C}$.*

Remark 2.2. *The scaling factors are motivated by results in the existing coin-betting setting with budget constraints. See Appendix A.2 for a detailed discussion.*

Similar to Definition 2.1, we can define *ε-scaled value functions* V_ε on the scaled game. Moreover, we extend its domain and assume it is twice-differentiable on $\mathbb{R}_{>0} \times \mathbb{R}^d$. The backward recursion on V_ε is

$$V_\varepsilon(t, S) = \min_{x \in \mathcal{X}} \max_{c \in \mathcal{C}} \left[V_\varepsilon(t + \varepsilon^2, S + \varepsilon c) - \langle \varepsilon c, x \rangle \right].$$

Borrowing the idea from [Zhu14, DK20b], we take a second-order Taylor approximation on the RHS,

$$V_\varepsilon(t + \varepsilon^2, S + \varepsilon c) = V_\varepsilon(t, S) + \varepsilon^2 \nabla_t V_\varepsilon(t, S) + \varepsilon \langle c, \nabla_S V_\varepsilon(t, S) \rangle + \frac{\varepsilon^2}{2} \langle \nabla_{SS} V_\varepsilon(t, S) \cdot c, c \rangle + o(\varepsilon^2),$$

which leads to

$$0 = \min_{x \in \mathcal{X}} \max_{c \in \mathcal{C}} \left[\langle c, \nabla_S V_\varepsilon(t, S) - x \rangle + \varepsilon \nabla_t V_\varepsilon(t, S) + \frac{\varepsilon}{2} \langle \nabla_{SS} V_\varepsilon(t, S) \cdot c, c \rangle + o(\varepsilon) \right].$$

As $\varepsilon \to 0$, the dominant term on the RHS is $\min_{x \in \mathcal{X}} \max_{c \in \mathcal{C}} \langle c, \nabla_S V_\varepsilon(t, S) - x \rangle$, therefore the outer minimizing argument becomes $x = \nabla_S V_\varepsilon(t, S)$. Using this argument, taking $\varepsilon \to 0$ and plugging in $\mathcal{C} = \mathsf{B}^d$, we obtain a second order nonlinear PDE for a *limiting value function*.

Definition 2.3 (Limiting value function)**.** *A function $\bar{V} : \mathbb{R}_{>0} \times \mathbb{R}^d \to \mathbb{R}$ is a limiting value function of the unconstrained coin-betting game if*

$$\nabla_t \bar{V} = -\frac{1}{2} \max\{\lambda_{\max}(\nabla_{SS} \bar{V}), 0\}. \tag{4}$$

To proceed, the key idea is that the PDE (4) is a continuous-time approximation of the backward recursion (3). Therefore, we can invoke a perturbed analysis of Lemma 2.2 and obtain tight wealth lower bounds.

3 One-dimensional analysis

It is challenging to solve the nonlinear PDE (4) in its full generality. Instead, we focus on the one-dimensional convex case where the nonlinear equation becomes linear. Despite this restriction, our approach is still able to handle the general d-dimensional unconstrained OLO problem due to a standard extension technique [CO18] reviewed in Appendix C.

For now let us consider $d = 1$. To further comply with the duality lemma (Lemma 2.1), we will only consider $\bar{V}$ that are convex with respect to the second argument. Then, the PDE (4) reduces to the one-dimensional *backward heat equation*

$$\nabla_t \bar{V} = -\frac{1}{2} \nabla_{SS} \bar{V}. \tag{5}$$

Such a linear PDE has been recently studied in [HLPR20], but both the setting and the derivation are different from ours. Nonetheless, since we care about the trade-offs on our performance bounds rather than a single-value (the regret uniformly with respect to all comparators in LEA), we need to perform a more comprehensive search for solutions $\bar{V}$.

3.1 PDE-based policy class

We first present a class of solutions to the backward heat equation (5). Due to linearity, any linear combination of two solutions is also a solution, allowing the user to interpolate their induced behavior. Motivated by the form of the classical potential (1), let us consider the ansatz

$$\bar{V}(t, S) = t^\alpha g\left(c \cdot t^\beta S\right), \tag{6}$$

where α, β and c are constants, and $g : \mathbb{R} \to \mathbb{R}$ is a one-dimensional function to be determined. For simplicity we omit shifting on S, t and the function value. In other words, once we find appropriate (α, β, c) and g, we immediately obtain a more general solution

$$\bar{V}(t, S) = C_0 + (t + \tau)^\alpha g\left(c \cdot t^\beta (S + S_0)\right),$$

with shifting constants C_0, τ and S_0.

Plugging in (6) and letting $z = c \cdot t^\beta S$, our PDE reduces to a second order linear ODE for the function g:

$$\alpha g(z) + \beta z g'(z) + \frac{1}{2} c^2 t^{2\beta+1} g''(z) = 0.$$

Letting $\beta = -1/2$ and $c = 1/\sqrt{2}$, we convert the ODE into the standard *Hermite* type

$$g''(z) - 2z g'(z) + 4\alpha g(z) = 0, \tag{7}$$

whose general solutions can be expressed in power series [AWH13, Chapter 7]. By varying the parameter α and repeating this procedure, we obtain a rich class of limiting value functions $\bar{V}$.

The next step is to construct coin-betting policies from limiting value functions. Our key idea is to use $\bar{V}$ as a surrogate for the actual value function V (Definition 2.1) and apply the same argument as in Lemma 2.2. Specifically, the adversary should pick the coin outcome that maximizes the RHS of the backward recursion (3), which is

$$c_t \in \arg\max_{c \in \mathcal{C}} \left[\bar{V}\left(t, \sum_{i=1}^{t-1} c_i + c\right) - \langle c, x_t \rangle \right]. \tag{8}$$

Note that since $\bar{V}$ is convex and $\mathcal{C} = [-1, 1]$, the adversary can simply focus on the boundary coins $\{-1, 1\}$; the resulting policy is presented in Algorithm 2.

Algorithm 2 PDE-based adversary policy.

Require: A limiting value function $\bar{V}$ for 1d unconstrained coin-betting.

1: **for** $t = 1, 2, \ldots$ **do**

2: Receive the player's bet x_t and choose the coin outcome as

$$c_t \in \arg\max_{c \in \{-1,1\}} \left[\bar{V}\left(t, \sum_{i=1}^{t-1} c_i + c\right) - \langle c, x_t \rangle \right]. \tag{9}$$

3: **end for**

As for the player, the optimal bet is the one that minimizes the objective function in (8), which is equivalent to the discrete derivative shown in Algorithm 3. Intuitively, the discrete derivative serves as an approximation of the standard derivative in classical potential methods. Therefore, Algorithm 3 essentially has a potential-based structure, with the potential function $\bar{V}$ generated from a PDE. Another interpretation is that, Algorithm 3 is a discrete approximation of *Follow the Regularized Leader* (FTRL) [AHR08] whose regularizer is the Fenchel conjugate of $\bar{V}(t, \cdot)$. The equivalence of potential functions and regularizers can be found in [Ora19, Section 7.3].

Algorithm 3 PDE-based player policy.

Require: A limiting value function $\bar{V}$ for 1d unconstrained coin-betting.

1: **for** $t = 1, 2, \ldots$ **do**

2: Choose the bet

$$x_t = \frac{1}{2} \left[\bar{V}\left(t, \sum_{i=1}^{t-1} c_i + 1\right) - \bar{V}\left(t, \sum_{i=1}^{t-1} c_i - 1\right) \right]. \tag{10}$$

3: Observe the coin outcome c_t and store it.

4: **end for**

3.2 Example

Before analyzing the performance of Algorithm 3, let us demonstrate the generality of our framework through a few examples. We show how classical algorithms can be derived from our framework, and more importantly, we present a potential function which permits a novel trade-off. For any α, let $\bar{V}_\alpha$ be a limiting value function obtained from the previous subsection. Let $C > 0$ be any positive scaling constant.

Warm up: $\alpha = 1$. The Hermite ODE (7) has a solution $g(z) = C(2z^2 - 1)$, resulting in $\bar{V}_1(t, S) = C(S^2 - t)$. The associated bet in Algorithm 3 is $x_t = 2C \sum_{i=1}^{t-1} c_i = x_{t-1} + 2Cc_{t-1}$, which is equivalent to *Online Gradient Descent* (OGD) with learning rate $2C$. Notably, $\bar{V}_1$ also satisfies Definition 2.1; that is, $\bar{V}_1$ is not only a limiting value function, but *also a value function* for the discrete-time game. Consequently, the performance of both the induced player policy (Algorithm 3) and adversary policy (Algorithm 2) can be guaranteed through Lemma 2.2. Details are presented in Appendix B.1.

Recovering existing potentials: $\alpha = -1/2$. The Hermite ODE can be solved by $g(z) = C \exp(z^2)$, resulting in $\bar{V}_{-1/2}(t, S) = C \cdot t^{-1/2} \exp[S^2/(2t)]$. Such a potential recovers the existing popular choice (1), and its time shifted version $C \cdot (t+\tau)^{-1/2} \exp[S^2/(2(t+\tau))]$ naturally recovers the *shifted potential* [OP16] with minimum effort. Different from the previous example, $\bar{V}_{-1/2}$ does not satisfy Definition 2.1. Therefore, we should characterize its approximation error on the backward recursion (3) in order to quantify the performance of the induced player policy. This procedure will be demonstrated in the next subsection.

A novel potential: $\alpha = 1/2$. The two linearly independent solutions of the Hermite ODE are both useful. First, $g(z) = \sqrt{2}Cz$ and $\bar{V}(t, S) = CS$. Such a potential leads to betting a fixed amount in coin-betting and shifting the coordinate system in unconstrained OLO. The idea is simple, and we will apply it in our experiments. For now, let us focus on the other solution which is more interesting.

$$g(z) = C\left[2z \cdot \int_0^z \exp(x^2)dx - \exp(z^2)\right] = C\left[2\int_0^z \left(\int_0^u \exp(x^2)dx\right) du - 1\right],$$

and the corresponding potential is

$$\bar{V}_{1/2}(t, S) = C\sqrt{t}\left[2\int_0^{S/\sqrt{2t}} \left(\int_0^u \exp(x^2)dx\right) du - 1\right]. \tag{11}$$

We will see that $\bar{V}_{1/2}$ often leads to superior performance compared to $\bar{V}_{-1/2}$, both in theory and in practice. However, without the help of a PDE, such a potential has not been discovered in the context of unconstrained OLO. This emphasizes the value of our PDE-based framework: it is a systematic way to generate potential function candidates, therefore significantly reduces the amount of guessing. Of course, not all of these candidates are good; their quality can be checked using the approach introduced in the sequel.

Before any rigorous analysis, let us compare the two player policies constructed from $\bar{V}_{-1/2}$ and $\bar{V}_{1/2}$. The bet (10) is roughly the derivative which is $\nabla_S \bar{V}_{-1/2}(t, S) = CSt^{-3/2} \exp[S^2/(2t)]$ for the former and $\nabla_S \bar{V}_{1/2}(t, S) = \sqrt{2}C \int_0^{S/\sqrt{2t}} \exp(x^2)dx$ for the latter, both evaluated at $S = \sum_{i=1}^{t-1} c_i$. As we show in Appendix B.3, for all $t \in \mathbb{N}_+$ and $|S| \leq t - 1$ we have $|\nabla_S \bar{V}_{1/2}(t, S)| \geq |\nabla_S \bar{V}_{-1/2}(t, S)|$. Therefore, given the same information, $\bar{V}_{1/2}$ intuitively induces a more aggressive betting behavior. Such a behavior cannot be achieved by simply scaling $\bar{V}_{-1/2}$.

3.3 Analysis of Algorithm 3

Next, we provide rigorous performance guarantees for the PDE-based player policy (Algorithm 3). To begin with, define discrete derivatives of a limiting value function $\bar{V}$ as

$$\bar{\nabla}_t \bar{V}(t, S) = \bar{V}(t, S) - \bar{V}(t-1, S),$$

$$\bar{\nabla}_{SS} \bar{V}(t, S) = \bar{V}(t, S+1) + \bar{V}(t, S-1) - 2\bar{V}(t, S).$$

When doing this we extend the domain of $\bar{V}(t, S)$ to $t = 0$ and assign $\bar{V}(0, 0) = 0$.

The key component of our analysis is the *Discrete Itô formula* [HLPR20, Kle13]. We modify it for our coin-betting problem, and the proof is provided in Appendix B.2.

Lemma 3.1 (Lemma D.3 and D.4 of [HLPR20], adapted)**.** *Consider applying Algorithm 3 against any adversary coin-betting policy $\boldsymbol{a}$. For all $t \in \mathbb{N}$,*

$$\bar{V}\left(t+1, \sum_{i=1}^{t+1} c_i\right) - \bar{V}\left(t, \sum_{i=1}^{t} c_i\right) \le c_{t+1}x_{t+1} + \underbrace{\left[\bar{\nabla}_t \bar{V}\left(t+1, \sum_{i=1}^{t} c_i\right) + \frac{1}{2}\bar{\nabla}_{SS}\bar{V}\left(t+1, \sum_{i=1}^{t} c_i\right)\right]}_{\lozenge}. \tag{12}$$

Moreover, equality is achieved when $c_{t+1} \in \{-1, 1\}$.

Summing (12) over $t \in [0 : T-1]$, the LHS becomes a telescopic sum which returns $\bar{V}(T, \sum_{i=1}^{T} c_i)$, and the RHS contains $\text{Wealth}_T = \sum_{t=1}^{T} c_t x_t$, which we aim to bound. Therefore, the remaining task is to quantify the sum $\lozenge$ in the bracket. Comparing $\lozenge$ to the backward heat equation (5), we can see that $\lozenge$ represents the "discrete approximation error" on the PDE. Bounding this error is case-dependent: we will only consider $\bar{V}_{1/2}$ in the following, and the analysis of $\bar{V}_{-1/2}$ is deferred to Appendix B.5.

Lemma 3.2. *For all $t \in \mathbb{N}_+$ and $S \in [1-t, t-1]$, $\bar{V}_{1/2}$ with any parameter $C > 0$ satisfies*

$$0 \ge \bar{\nabla}_t \bar{V}_{1/2}(t, S) + \bar{\nabla}_{SS}\bar{V}_{1/2}(t, S)/2 \ge \begin{cases} -C, & t = 1, \\ -\frac{C}{8}(t-1)^{-3/2} \exp\left(\frac{S^2}{2(t-1)}\right)\left(\frac{S^2}{t-1} + 1\right), & t > 1. \end{cases}$$

Combining the above, we can immediately obtain a wealth lower bound (Theorem 1) for the player policy constructed from $\bar{V}_{1/2}$. Its proof follows from a simple telescopic sum therefore omitted.

Theorem 1. *For all $T \in \mathbb{N}_+$, Algorithm 3 constructed from $\bar{V}_{1/2}$ guarantees a wealth lower bound*

$$\text{Wealth}_T \ge \bar{V}_{1/2}\left(T, \sum_{t=1}^{T} c_t\right),$$

against any adversary policy $\boldsymbol{a}$.

Furthermore, by applying the analysis on the opposite direction, the following theorem shows that Algorithm 2 is a strong adversary policy to confront Algorithm 3. That is, the pair of player-adversary policies induced by $\bar{V}_{1/2}$ has a "dual property". Note that when the player applies Algorithm 3, both $c_t = -1$ and $c_t = 1$ satisfy (9), therefore Algorithm 2 can freely choose from these two boundary coins. The proof is a bit more technical, deferred to Appendix B.4.

Theorem 2. *For all $T \in \mathbb{N}_+$ and $S \in [-T, T]$, we can construct $c_1 \in \mathcal{C}$ and $c_2, \ldots, c_T \in \{-1, 1\}$ such that*

1. *$\sum_{t=1}^{T} c_t = S$;*
2. *If the player applies Algorithm 3 constructed from $\bar{V}_{1/2}$ (with parameter C) and the adversary plays the aforementioned coin sequence $c_{1:T}$, then*

$$\text{Wealth}_T \le \bar{V}_{1/2}(T, S) + \frac{3C}{8}\exp\left(\frac{S^2}{2T}\right)\left(\frac{S^2}{T} + 1\right) + 2C.$$

Comparing Theorem 1 to Theorem 2, the lower and upper bound are separated by at most a constant when $\sum_{t=1}^{T} c_t = O(\sqrt{T})$. It means that *everywhere* on the set $\{(t, S) | S = O(\sqrt{t})\}$, the value of $\bar{V}_{1/2}(t, S)$ provides a tight performance guarantee for Algorithm 3.

3.4 Optimality of Algorithm 3

The previous wealth upper bound shows that Theorem 1 faithfully characterizes the performance of Algorithm 3, but does not address the optimality of this betting policy. To this end, we now present a wealth upper bound that holds for all betting policies. The proof is deferred to Appendix B.6.

Theorem 3. *For all $\lambda \geq \exp[(\sqrt{2}+1)/2]$, $T \geq 8\pi\lambda^2 \log \lambda$, and any player policy $\boldsymbol{p}$ that guarantees $\text{Wealth}_T \geq -C\sqrt{T}$ (e.g., Algorithm 3 constructed from $\bar{V}_{1/2}$), there exists an adversary policy $\boldsymbol{a}$ such that the following statement holds. In the coin-betting game induced by the policy pair $(\boldsymbol{p}, \boldsymbol{a})$,*

1. $|\sum_{t=1}^{T} c_t| \geq \sqrt{2T \log \lambda}$;
2. $\text{Wealth}_T \leq 2\sqrt{2\pi}\lambda\sqrt{\log \lambda} \cdot C\sqrt{T}$.

The proof of Theorem 3 is based on a stochastic adversary argument similar to [SM12, Ora13]. However, using an improved lower bound for the tail probability of random walks, our wealth upper bound is tight up to a poly-logarithmic factor. To see this, let us compare it to the wealth lower bound from Theorem 1: if $|\sum_{t=1}^{T} c_t| \geq \sqrt{2T \log \lambda}$, then Algorithm 3 guarantees (the last inequality due to Lemma B.2)

$$\text{Wealth}_T \geq \bar{V}_{1/2}\left(t, \sum_{t=1}^{T} c_t\right) \geq \bar{V}_{1/2}\left(t, \sqrt{2T \log \lambda}\right) \geq C\sqrt{T}\left[\frac{\lambda}{2 \log \lambda} - \frac{3}{2}\right].$$

For comparison, previous analysis [Ora13] only guarantees the suboptimal rate $\text{Wealth}_T \leq \tilde{O}(\lambda^{\log 4}\sqrt{T})$. Later we will see that matching the $\tilde{O}(\lambda\sqrt{T})$ factor in our wealth bounds leads to matching the leading term (*including* the multiplicative constant) in the regret of OLO.

4 Optimal unconstrained OLO

In this section we present our main results on unconstrained OLO. Notice that using the conversion from coin-betting to OLO (Algorithm 1), our PDE-based betting policy (Algorithm 3) can be directly converted into a one-dimensional unconstrained OLO algorithm with a potential structure. For clarity, we restate its pseudo-code as Algorithm 5 in Appendix C. To further extend it to $\mathbb{R}^d$, we use a classical polar decomposition technique [CO18]. Combining everything, our final product is a general unconstrained OLO algorithm (Algorithm 4) constructed from any solution of the one-dimensional PDE (5).

Algorithm 4 PDE-based unconstrained OLO algorithm.

Require: A one-dimensional limiting value function $\bar{V}$ which satisfies (5).

Define $\mathcal{A}_B$ as the standard Online Gradient Descent (OGD) on B^d with learning rate $\eta_t = 1/\sqrt{t}$, initialized at the origin.
Initialize a parameter ("sufficient statistic") $S_1 = 0$.
for $t = 1, 2, \ldots$ **do**
  Let $y_t = \left[\bar{V}\left(t, S_t + 1\right) - \bar{V}\left(t, S_t - 1\right)\right]/2$.
  Query $\mathcal{A}_B$ for its t-th prediction and assign it to z_t.
  Predict $x_t = y_t z_t \in \mathbb{R}^d$.
  Observe $g_t \in \mathbb{R}^d$ generated by an adversary (g_t can depend on $x_1, \ldots, x_t$).
  Return g_t as the t-th loss gradient to $\mathcal{A}_B$, and let $S_{t+1} = S_t - \langle g_t, z_t \rangle$.
end for

Next, let us specifically consider Algorithm 4 constructed from $\bar{V}_{1/2}$ (11). We leave proofs in this section to Appendix C.1. Recall that C is any positive scaling constant. Converting the coin-betting lower bound (Theorem 1) to OLO, we have

Theorem 4. *For all $T \in \mathbb{N}_+$ and $u \in \mathbb{R}^d$, against any adversary, Algorithm 4 constructed from $\bar{V}_{1/2}$ guarantees*

$$\text{Regret}_T(u) \leq C\sqrt{T} + \|u\|\sqrt{2T}\left[\sqrt{\log\left(1 + \frac{\|u\|}{\sqrt{2}C}\right)} + 2\right].$$

The obtained regret upper bound is *parameter-free* since the dependence on $\|u\|$ is $\tilde{O}(\|u\|)$. Moreover, compared to the existing bound (2), our algorithm has larger $\text{Regret}_T(0)$ ($C\sqrt{T}$ instead of C), but smaller regret

with respect to far-away comparators ($\|u\|\sqrt{T\log\|u\|}$ instead of $\|u\|\sqrt{T\log(\|u\|T)}$). One may use the doubling trick [SS+11] to set a time-varying C and turn special versions of (2) (e.g., [MO14, Theorem 11]) into our form, but that approach (i) is impractical; and (ii) introduces a multiplicative constant which is suboptimal (will be shown shortly).

Notably, the form of our bound corresponds to a largely unexplored trade-off on the shape of the regret function, which is particularly useful in high-dimensional problems. When parameter-free algorithms are applied in practice, the origin of the coordinate system is typically shifted to the initial guess of the optimal comparator u^*. It is hard to obtain accurate guesses in high-dimensional settings, therefore the effective distance to the optimal comparator can be large. Compared to existing approaches, our algorithm theoretically guarantees better performance in this situation, and our experiments will further support this finding.

Now we proceed to the regret lower bound derived from the wealth upper bound (Theorem 3). For clarity, we write $\text{Regret}_T^{\mathcal{A},adv}(u)$ as the regret induced by an algorithm $\mathcal{A}$ and an adversary adv.

Theorem 5. *Define $\mathcal{A}_{1/2}$ as Algorithm 4 constructed from $\bar{V}_{1/2}$, then Theorem 4 leads to*

$$\limsup_{U\to\infty}\limsup_{T\to\infty}\sup_{\|u\|=U,adv}\frac{\textit{Regret}_T^{\mathcal{A}_{1/2},adv}(u)}{\|u\|\sqrt{T\log\|u\|}}\le\sqrt{2}.$$

Conversely, for all C and any unconstrained OLO algorithm $\mathcal{A}$ (e.g., $\mathcal{A}_{1/2}$) that guarantees $\textit{Regret}_T^{\mathcal{A},adv}(0)\le C\sqrt{T}$ for all adv and T, we have

$$\liminf_{U\to\infty}\liminf_{T\to\infty}\sup_{\|u\|=U,adv}\frac{\textit{Regret}_T^{\mathcal{A},adv}(u)}{\|u\|\sqrt{T\log\|u\|}}\ge\sqrt{2}.$$

Theorem 5 shows that the leading term in the regret upper bound, including the multiplying constant $\sqrt{2}$, is tight. To the best of our knowledge, our algorithm is the first parameter-free algorithm with such optimality. In other words, it achieves the *optimal loss-regret trade-off* sketched in the Introduction.

Finally, we leave parallel results based on $\bar{V}_{-1/2}$ to Appendix C.2. In particular, we show that the leading terms in the upper and lower bounds are separated by a $\sqrt{2}$ multiplicative factor. Future works may consider closing this gap. We also convert the player-dependent coin-betting upper bound (Theorem 2) to OLO, presented in Appendix C.3. It estimates the performance of our one-dimensional OLO algorithm (Algorithm 5) up to a small error term that does not grow with time.

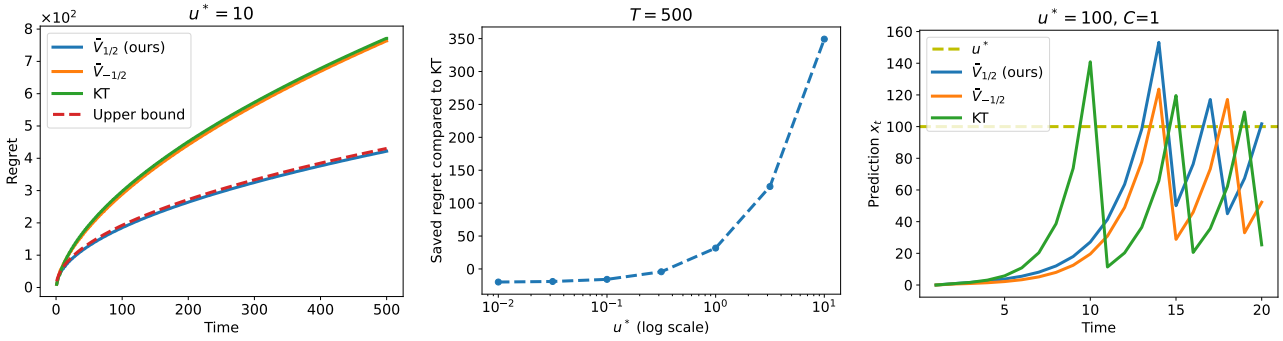
5 Experiment

Our theoretical results are supported by experiments. First, we test our one-dimensional unconstrained OLO algorithm (Algorithm 5) on a synthetic task. The simplicity of this setting allows us to directly compute the regret, thus clearly demonstrate the benefit of $\bar{V}_{1/2}$ over the existing potential $\bar{V}_{-1/2}$. Next, we test our high-dimensional algorithm (Algorithm 4) on an online regression task with real data.

5.1 One-dimensional synthetic task

To begin with, let us consider a simple one-dimensional OCO problem [Zin03] with time invariant loss function $|x_t-u^*|$, where $u^*\in\mathbb{R}$ is a constant hidden from the online learning algorithm. Translated into OLO, the adversary picks the loss gradient $g_t=1$ if $x_t\ge u^*$, while $g_t=-1$ otherwise. The most natural comparator is the hidden constant u^*, and the induced regret of OLO can be nicely interpreted as the *cumulative loss* of OCO. That is, $\text{Regret}_T(u^*)=\sum_{t=1}^T g_t(x_t-u^*)=\sum_{t=1}^T|x_t-u^*|$. We will test three algorithms: (i) Algorithm 5 constructed from $\bar{V}_{1/2}$ (the contribution of this paper); (ii) Algorithm 5 constructed from $\bar{V}_{-1/2}$; and (iii) the classical *Krichevsky-Trofimov* (KT) algorithm [OP16] which is an optimistic version of (ii) with similar guarantees. Each algorithm requires one hyperparameter: we set $C=1$ for the first two algorithms, and set the "initial wealth" as $\sqrt{e}$ for KT. Such choices make a fair comparison, as discussed in Appendix D.2.

Since $\text{Regret}_T(u^*)$ depends on both u^* and T, there are multiple ways to visualize our results. In Figure 1a, we fix $u^*=10$ and plot $\text{Regret}_T(u^*)$ as a function of T (lower is better), with more settings of u^* tested in Appendix D.3. For comparison, we also plot the regret upper bound based on $\bar{V}_{1/2}$ (Corollary 13). Consistent



(a) Fix u^* and vary T.

(b) Fix T and vary u^*.

(c) The progression of x_t.

Figure 1: One-dimensional synthetic task with loss $|x_t - u^*|$. Specifically, (b) fixes $T = 500$ and plots $\text{Regret}_T(u^*)$ of KT minus $\text{Regret}_T(u^*)$ of our algorithm ($\bar{V}_{1/2}$) as a function of u^*.

with our theory, (*i*) the upper bound (red dashed) closely captures the actual performance of our algorithm (blue); (*ii*) the two baselines (orange and green) exhibit similar performance, and our algorithm improves both when $u^* = 10$.

In Figure 1b, we fix $T = 500$ and plot the difference between the regret of KT and our algorithm (i.e., $\text{Regret}_T(u^*)|_{KT} - \text{Regret}_T(u^*)|_{ours}$ as a function of u^*, higher is better for us). The obtained curve demonstrates the benefit of our special loss-regret trade-off: while sacrificing the regret at small $|u^*|$, our algorithm significantly improves the baseline when u^* is far-away. Notably, the magnitude of $|u^*|$ represents the quality of initialization: with an oracle guess $\tilde{u}$, one can shift the origin to $\tilde{u}$, and the effective distance to u^* becomes $|\tilde{u} - u^*|$. Figure 1b shows that in order to beat our algorithm, the baseline has to guess u^* beforehand with error at most 1, which is obviously very hard. Therefore, our algorithm prevails in most situations.

To strengthen the intuition, let us fix $u^* = 100$ and take a closer look at the progression of predictions x_t (Figure 1c). Similar to both baselines, our algorithm approaches u^* with exponentially growing speed at the beginning, which is a key benefit of parameter-free algorithms over gradient descent [OT17, Section 5]. However, after overshooting, the prediction of our algorithm exhibits a much smaller "dip". This is very intuitive, as our algorithm allows higher $\text{Regret}_T(0)$. In other words, compared to the baselines, our algorithm has a weaker belief that the initialization is correct; instead, it believes more in the incoming information. Such a property leads to advantages when the initialization is indeed far from the optimum.

5.2 High-dimensional linear regression

Finally, we consider a high-dimensional regression task with real data. We use the YearPredictionMSD dataset [BMEWL11] available from the UCI Machine Learning Repository [DG17], and the context of this dataset is to predict the release year of a song from its audio features. The data-preprocessing steps are introduced in Appendix D.4. After that, we use a linear model with absolute loss $l_t(x) = |\langle z_t, x\rangle - y_t|$, where $z_t \in \mathbb{R}^{90}$ and $y_t \in \mathbb{R}_+$ are the t-th sampled feature vector and target. This can be converted into a 90-dimensional unconstrained OLO problem: the adversary picks $g_t = z_t$ if $\langle z_t, x_t\rangle \geq y_t$, while $g_t = -z_t$ otherwise. Same as before, we consider three algorithms with $C = 1$: (*i*) Algorithm 4 constructed from $\bar{V}_{1/2}$; (*ii*) Algorithm 4 constructed from $\bar{V}_{-1/2}$; and (*iii*) KT.

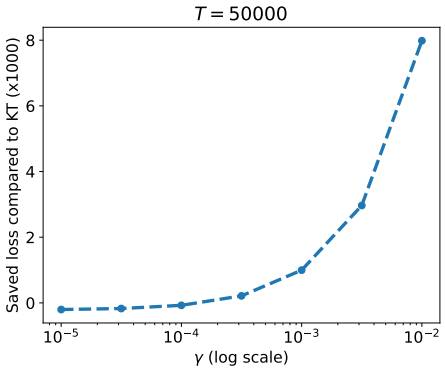


Figure 2: High-dimensional regression. Plot shows the total loss of KT minus the total loss of our algorithm.

To study how these algorithms adapt to the distance to the optimal comparator, we use a parameter γ to scale the target y_t. That is, we assign $y_t \leftarrow \gamma y_t$, and γ is varied across different settings. Due to the linearity of our regression model, the optimal comparator is effectively scaled by γ. With a small γ, the comparators are brought closer to our initialization 0. Note that such a scaling

does not work if we use a nonlinear regression model. In those general cases, one may only care about the unscaled setting ($\gamma = 1$).

We present our results in two ways, (i) fix γ and vary T; (ii) fix T and vary γ. For conciseness, we defer (i) to Appendix D.4, and (ii) is presented in Figure 2. Specifically, since (z_t, y_t) is sampled from the dataset, in each setting (of γ) we run each algorithm 5 times and use the average cumulative OCO loss $\sum_{t=1}^{T} l_t(x_t)$ as the "TotalLoss" of this algorithm. Figure 2 shows the difference between KT and our algorithm ($\text{TotalLoss}|_{KT} - \text{TotalLoss}|_{ours}$) as a function of γ. We can observe a similar behavior as in Figure 1b: our algorithm outperforms the baseline when the optimal comparator is far-away from the initial prediction.

6 Conclusion

We propose a framework that generates unconstrained OLO potentials by solving a PDE. To demonstrate the power of this framework, we use it to design a novel unconstrained OLO algorithm with the optimal leading constant in its regret bound. Our algorithm is the first parameter-free one with such optimality. Moreover, it achieves a loss-regret trade-off largely unexplored in existing works, which leads to practical advantages when a good initialization is not available.

Our result may inspire future works in multiple directions. First, we use a PDE to achieve a special trade-off in online learning. In a similar spirit, one may apply this technique to other multi-objective learning tasks. Second, our regret bound is data-independent, and the Lipschitz constant needs to be known a priori. Borrowing recent ideas [CB16, CB17, MK20], it might be possible to make our algorithm *adaptive* and *scale-free*, thus further improving its practical performance.

Acknowledgement

We thank Francesco Orabona for valuable discussion.

References

[AHR08] Jacob Abernethy, Elad E Hazan, and Alexander Rakhlin. Competing in the dark: An efficient algorithm for bandit linear optimization. In *21st Annual Conference on Learning Theory, COLT 2008*, pages 263–273, 2008.

[AP13] Alexandr Andoni and Rina Panigrahy. A differential equations approach to optimizing regret trade-offs. *arXiv preprint arXiv:1305.1359*, 2013.

[AWH13] George B Arfken, Hans J Weber, and Frank E Harris. *Mathematical Methods for Physicists: A Comprehensive Guide*. Academic Press, 2013.

[Ber12] Dimitri Bertsekas. *Dynamic programming and optimal control: Volume I*, volume 1. Athena scientific, 2012.

[BEZ20a] Erhan Bayraktar, Ibrahim Ekren, and Xin Zhang. Finite-time 4-expert prediction problem. *Communications in Partial Differential Equations*, 45(7):714–757, 2020.

[BEZ20b] Erhan Bayraktar, Ibrahim Ekren, and Yili Zhang. On the asymptotic optimality of the comb strategy for prediction with expert advice. *The Annals of Applied Probability*, 30(6):2517–2546, 2020.

[BEZ21] Erhan Bayraktar, Ibrahim Ekren, and Xin Zhang. Prediction against a limited adversary. *Journal of Machine Learning Research*, 22(72):1–33, 2021.

[BMEWL11] Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. The million song dataset. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*, 2011.

[BPZ20] Erhan Bayraktar, H Vincent Poor, and Xin Zhang. Malicious experts versus the multiplicative weights algorithm in online prediction. *IEEE Transactions on Information Theory*, 67(1):559–565, 2020.

[CB16] Ashok Cutkosky and Kwabena Boahen. Online convex optimization with unconstrained domains and losses. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 748–756, 2016.

[CB17] Ashok Cutkosky and Kwabena Boahen. Online learning without prior information. In *Conference on Learning Theory*, pages 643–677. PMLR, 2017.

[CBL03] Nicolo Cesa-Bianchi and Gábor Lugosi. Potential-based algorithms in on-line prediction and game theory. *Machine Learning*, 51(3):239–261, 2003.

[CBL06] Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

[CLO20] Keyi Chen, John Langford, and Francesco Orabona. Better parameter-free stochastic optimization with ode updates for coin-betting. *arXiv preprint arXiv:2006.07507*, 2020.

[CO18] Ashok Cutkosky and Francesco Orabona. Black-box reductions for parameter-free online learning in banach spaces. In *Conference On Learning Theory*, pages 1493–1529. PMLR, 2018.

[Cov66] Thomas M Cover. Behavior of sequential predictors of binary sequences. Technical report, STANFORD UNIV CALIF STANFORD ELECTRONICS LABS, 1966.

[Cut19] Ashok Cutkosky. Combining online learning guarantees. In *Conference on Learning Theory*, pages 895–913. PMLR, 2019.

[Cut20] Ashok Cutkosky. Parameter-free, dynamic, and strongly-adaptive online learning. In *International Conference on Machine Learning*, pages 2250–2259. PMLR, 2020.

[DC20] Nadejda Drenska and Jeff Calder. Online prediction with history-dependent experts: the general case. *arXiv preprint arXiv:2008.00052*, 2020.

[DG17] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[DK20a] Nadejda Drenska and Robert V Kohn. A pde approach to the prediction of a binary sequence with advice from two history-dependent experts. *arXiv preprint arXiv:2007.12732*, 2020.

[DK20b] Nadejda Drenska and Robert V Kohn. Prediction with expert advice: A pde perspective. *Journal of Nonlinear Science*, 30(1):137–173, 2020.

[DM19] Amit Daniely and Yishay Mansour. Competitive ratio vs regret minimization: achieving the best of both worlds. In *Algorithmic Learning Theory*, pages 333–368. PMLR, 2019.

[Due10] Lutz Duembgen. Bounding standard gaussian tail probabilities. *arXiv preprint arXiv:1012.2063*, 2010.

[FRS18] Dylan J Foster, Alexander Rakhlin, and Karthik Sridharan. Online learning: Sufficient statistics and the burkholder method. In *Conference On Learning Theory*, pages 3028–3064. PMLR, 2018.

[Haz19] Elad Hazan. Introduction to online convex optimization. *arXiv preprint arXiv:1909.05207*, 2019.

[HLPR20] Nicholas JA Harvey, Christopher Liaw, Edwin A Perkins, and Sikander Randhawa. Optimal anytime regret for two experts. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1404–1415. IEEE, 2020.

[JO19] Kwang-Sung Jun and Francesco Orabona. Parameter-free locally differentially private stochastic subgradient descent. *arXiv preprint arXiv:1911.09564*, 2019.

[Kj56] JL Kelly jr. A new interpretation of information rate. *the bell system technical journal*, 1956.

[KKW20a] Vladimir A Kobzar, Robert V Kohn, and Zhilei Wang. New potential-based bounds for prediction with expert advice. In *Conference on Learning Theory*, pages 2370–2405. PMLR, 2020.

[KKW20b] Vladimir A Kobzar, Robert V Kohn, and Zhilei Wang. New potential-based bounds for the geometric-stopping version of prediction with expert advice. In *Mathematical and Scientific Machine Learning*, pages 537–554. PMLR, 2020.

[Kle13] Achim Klenke. *Probability theory: a comprehensive course*. Springer Science & Business Media, 2013.

[KP10] Michael Kapralov and Rina Panigrahy. Prediction strategies without loss. *arXiv preprint arXiv:1008.3672*, 2010.

[LW94] Nick Littlestone and Manfred K Warmuth. The weighted majority algorithm. *Information and computation*, 108(2):212–261, 1994.

[MA13] Brendan McMahan and Jacob Abernethy. Minimax optimal algorithms for unconstrained linear optimization. *Advances in Neural Information Processing Systems*, 26:2724–2732, 2013.

[MK20] Zakaria Mhammedi and Wouter M Koolen. Lipschitz and comparator-norm adaptivity in online learning. In *Conference on Learning Theory*, pages 2858–2887. PMLR, 2020.

[MO14] H Brendan McMahan and Francesco Orabona. Unconstrained online linear learning in hilbert spaces: Minimax algorithms and normal approximations. In *Conference on Learning Theory*, pages 1020–1039. PMLR, 2014.

[OP16] Francesco Orabona and Dávid Pál. Coin betting and parameter-free online learning. *arXiv preprint arXiv:1602.04128*, 2016.

[Ora13] Francesco Orabona. Dimension-free exponentiated gradient. In *NIPS*, pages 1806–1814, 2013.

[Ora14] Francesco Orabona. Simultaneous model selection and optimization through parameter-free stochastic learning. In *Advances in Neural Information Processing Systems*, pages 1116–1124, 2014.

[Ora19] Francesco Orabona. A modern introduction to online learning. *arXiv preprint arXiv:1912.13213*, 2019.

[OT17] Francesco Orabona and Tatiana Tommasi. Training deep networks without learning rates through coin betting. *Advances in Neural Information Processing Systems*, 30:2160–2170, 2017.

[Roc15] Ralph Tyrell Rockafellar. *Convex analysis*. Princeton university press, 2015.

[Rok17] Dmitry B Rokhlin. Pde approach to the problem of online prediction with expert advice: a construction of potential-based strategies. *arXiv preprint arXiv:1705.01091*, 2017.

[RSS12] Sasha Rakhlin, Ohad Shamir, and Karthik Sridharan. Relax and randomize: From value to algorithms. *Advances in Neural Information Processing Systems*, 25:2141–2149, 2012.

[She11] Irina Shevtsova. On the absolute constants in the berry-esseen type inequalities for identically distributed summands. *arXiv preprint arXiv:1111.6554*, 2011.

[SM12] Matthew Streeter and H Brendan McMahan. No-regret algorithms for unconstrained online convex optimization. *arXiv preprint arXiv:1211.2260*, 2012.

[SS+11] Shai Shalev-Shwartz et al. Online learning and online convex optimization. *Foundations and trends in Machine Learning*, 4(2):107–194, 2011.

[vdH19] Dirk van der Hoeven. User-specified local differential privacy in unconstrained adaptive online learning. In *NeurIPS*, pages 14080–14089, 2019.

[ZCP21] Zhiyu Zhang, Ashok Cutkosky, and Ioannis Ch Paschalidis. Adversarial tracking control via strongly adaptive online learning with memory. *arXiv preprint arXiv:2102.01623*, 2021.

[Zhu14] Kangping Zhu. *Two problems in applications of PDE*. PhD thesis, New York University, 2014.

[Zin03] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th international conference on machine learning (icml-03)*, pages 928–936, 2003.

Appendix

Organization Appendix A presents details on the derivation of our PDE (4). Appendix B solves the one-dimensional PDE and verifies the quality of the induced coin-betting policies. Appendix C converts our theoretical results on coin-betting to unconstrained OLO. Appendix D presents details on our experiments.

A Detail on the derivation of PDE

In this section we present two aspects of our PDE derivation omitted in the main paper. First, we prove Lemma 2.2 which shows that any value function can naturally induce a pair of "dual" player-adversary policies. Next, we discuss our choice of scaling factors for the scaled game (Definition 2.2).

A.1 Proof of Lemma 2.2

Lemma 2.2. *Given any value function V satisfying Definition 2.1,*

1. *We can construct a player policy $\boldsymbol{p}^*$ such that for all $\boldsymbol{a}$ and $T \in \mathbb{N}_+$,*

$$\mathit{Wealth}_T \geq V\left(T, \sum_{t=1}^{T} c_t\right).$$

   In addition, for all t, the player's bet $p_t^*(c_{1:t-1})$ depends on the past coin outcomes only through their sum $\sum_{i=1}^{t-1} c_i$.

2. *We can construct an adversary policy $\boldsymbol{a}^*$ such that for all $\boldsymbol{p}$ and $T \in \mathbb{N}_+$,*

$$\mathit{Wealth}_T \leq V\left(T, \sum_{t=1}^{T} c_t\right).$$

Proof of Lemma 2.2. We only prove the first part by induction. The proof of the second part is similar, therefore omitted. Starting from $t = 0$ and $S = 0$ in the backward recursion,

$$V(t, S) = \min_{x \in \mathcal{X}} \max_{c \in \mathcal{C}} \left[V(t+1, S+c) - \langle c, x\rangle\right].$$

Let x_1 be the outer minimizing argument. Then, for all adversary policy a_1 such that $c_1 = a_1(x_1)$, we have $V(1, c_1) = V(1, c_1) - V(0, 0) \leq \langle c_1, x_1\rangle$.

Consider the following induction hypothesis: there exists $T \in \mathbb{N}_+$, initial bet x_1 and functions $p_2^*, \ldots, p_T^*$ such that for all $\boldsymbol{a}$,

$$\sum_{t=1}^{T} \langle c_t, x_t\rangle \geq V\left(T, \sum_{t=1}^{T} c_t\right).$$

Plugging $(t, S) = (T, \sum_{t=1}^{T} c_t)$ into the backward recursion,

$$V\left(T, \sum_{t=1}^{T} c_t\right) = \min_{x_{T+1} \in \mathcal{X}} \max_{c_{T+1} \in \mathcal{C}} \left[V\left(T+1, \sum_{t=1}^{T+1} c_t\right) - \langle c_{T+1}, x_{T+1}\rangle\right].$$

Given the value function V, there exists x_{T+1} only depending on T and $\sum_{t=1}^{T} c_t$ such that for all c_{T+1},

$$V\left(T, \sum_{t=1}^{T} c_t\right) \geq V\left(T+1, \sum_{t=1}^{T+1} c_t\right) - \langle c_{T+1}, x_{T+1}\rangle.$$

Define the policy p_{T+1}^* in this way, we have

$$\sum_{t=1}^{T+1} \langle c_t, x_t\rangle \geq V\left(T+1, \sum_{t=1}^{T+1} c_t\right).$$

□

A.2 The choice of scaling factors

We now discuss our choice of scaling factors for the scaled game (Definition 2.2). To begin with, let us review the wealth lower bounds for the existing coin-betting setting [MA13, OP16] with budget constraints. For simplicity, assume $d = 1$. Inspired by the celebrated Kelly bettor [Kj56], McMahan and Abernethy [MA13] made an interesting observation: if starting from an initial wealth C and knowing the bias of future coins ($\sum_{t=1}^{T} c_t/T$), the player could bet a fixed fraction $\beta = \sum_{t=1}^{T} c_t/T$ of his wealth in each round and guarantees a wealth lower bound

$$\text{Wealth}_T \geq C \exp\left(\frac{(\sum_{t=1}^{T} c_t)^2}{2T}\right). \tag{13}$$

Of course, this strategy is not implementable in reality. However, using a time-dependent betting fraction β_t that reflects the bias *observed online*, the player can actually implement a strategy [OP16] with

$$\text{Wealth}_T \geq \frac{C}{\sqrt{T}} \exp\left(\frac{(\sum_{t=1}^{T} c_t)^2}{2T}\right),$$

which matches (13) in the important exponential factor. Under the presence of budget constraints, such an exponential factor is optimal. Extended to our unconstrained coin-betting setting (which is a strict generalization of the existing one), this exponential factor characterizes the best result when the player can only tolerate a fixed amount of total loss. This is intuitively similar to the concept of *Pareto optimality*: the optimal policy for the player depends on how risk-tolerant it is.

Back to the design of scaling factors for the ε-scaled game, our guideline is simple: the baseline strategy [MA13] discussed above should guarantee the *same* wealth bound in the scaled game and the original game. In this way, the PDE derived in the scaling limit could recover the specific Pareto optimal result (13). Concretely, let $f(\varepsilon)$ and $g(\varepsilon)$ be the scaling factors on the unit time and the coin space respectively. Without loss of generality, let $g(\varepsilon) = \varepsilon$; we now justify our choice $f(\varepsilon) = \varepsilon^2$.

Consider an extreme adversary whose decisions are always 1. In the original game, the baseline strategy [MA13] guarantees $\text{Wealth}_T \geq C \exp(T/2)$ due to (13). In the scaled game, the adversary decisions are scaled by ε, and the total number of decision rounds is $[f(\varepsilon)]^{-1}$ times as many. Therefore, the baseline strategy [MA13] guarantees

$$\text{Wealth}_T \geq C \exp\left(\frac{(\varepsilon T \cdot [f(\varepsilon)]^{-1})^2}{2T \cdot [f(\varepsilon)]^{-1}}\right) = C \exp\left(\frac{\varepsilon^2 T \cdot [f(\varepsilon)]^{-1}}{2}\right).$$

If $f(\varepsilon) = \varepsilon^2$, then the wealth bounds for the scaled game and the original game are equal.

B Detail on the PDE-based betting policy

In this section we present detailed analysis of the one-dimensional coin-betting game (Section 3). By solving the backward heat equation (5), we obtain three specific limiting value functions (i.e., potential functions) $\bar{V}_1$, $\bar{V}_{-1/2}$ and $\bar{V}_{1/2}$. The performance of their induced coin betting policies (Algorithm 3) will be characterized next.

In particular, $\bar{V}_1$ is a special case where Lemma 2.2 can be directly applied. For the general case (e.g., $\bar{V}_{-1/2}$ and $\bar{V}_{1/2}$), we need to use a different analysis introduced in Appendix B.2 to B.5. Finally, Appendix B.6 shows the optimality of Algorithm 3.

B.1 Special case: Policy induced by $\bar{V}_1$

It is easy to verify that $\bar{V}_1(t, S) = C(S^2 - t)$ satisfies Definition 2.1. Therefore, the performance guarantee of the associated player and adversary policies (Algorithm 2 and 3) can be stated as a corollary of Lemma 2.2.

Corollary 6. *For all $T \in \mathbb{N}_+$,*

1. *Against any adversary policy $\boldsymbol{a}$, Algorithm 3 constructed from $\bar{V}_1$ guarantees*

$$\textit{Wealth}_T \geq C \left[\left(\sum_{t=1}^{T} c_t\right)^2 - T\right].$$

2. *Against any player policy* $\boldsymbol{p}$*, Algorithm 2 constructed from* $\bar{V}_1$ *guarantees*

$$\mathit{Wealth}_T \le C\left[\left(\sum_{t=1}^{T} c_t\right)^2 - T\right].$$

B.2 General case: Discrete Itô formula

In general, the solution of the backward heat equation (5) is only an approximation of a value function (for the discrete-time coin-betting game), therefore Lemma 2.1 cannot be directly applied. Instead, we pursue a perturbed analysis using the *Discrete Itô formula*. Harvey et al. used this technique in the two-expert LEA problem. Here we modify it to a general form applicable in coin-betting.

Lemma 3.1 (Lemma D.3 and D.4 of [HLPR20], adapted)**.** *Consider applying Algorithm 3 against any adversary coin-betting policy* $\boldsymbol{a}$*. For all* $t \in \mathbb{N}$*,*

$$\bar{V}\left(t+1, \sum_{i=1}^{t+1} c_i\right) - \bar{V}\left(t, \sum_{i=1}^{t} c_i\right) \le c_{t+1}x_{t+1} + \underbrace{\left[\bar{\nabla}_t \bar{V}\left(t+1, \sum_{i=1}^{t} c_i\right) + \frac{1}{2}\bar{\nabla}_{SS}\bar{V}\left(t+1, \sum_{i=1}^{t} c_i\right)\right]}_{\Diamond}. \tag{12}$$

Moreover, equality is achieved when $c_{t+1} \in \{-1, 1\}$*.*

Proof of Lemma 3.1. Starting from the LHS of (12),

$$\begin{aligned}
\mathit{LHS} = \bar{V}\left(t+1, \sum_{i=1}^{t+1} c_i\right) &- \frac{1}{2}\left[\bar{V}\left(t+1, \sum_{i=1}^{t} c_i + 1\right) + \bar{V}\left(t+1, \sum_{i=1}^{t} c_i - 1\right)\right] \\
&+ \frac{1}{2}\left[\bar{V}\left(t+1, \sum_{i=1}^{t} c_i + 1\right) + \bar{V}\left(t+1, \sum_{i=1}^{t} c_i - 1\right)\right] - \bar{V}\left(t, \sum_{i=1}^{t} c_i\right) \\
= \bar{V}\left(t+1, \sum_{i=1}^{t+1} c_i\right) &- \frac{1}{2}\left[\bar{V}\left(t+1, \sum_{i=1}^{t} c_i + 1\right) + \bar{V}\left(t+1, \sum_{i=1}^{t} c_i - 1\right)\right] \\
&+ \bar{\nabla}_t \bar{V}\left(t+1, \sum_{i=1}^{t} c_i\right) + \frac{1}{2}\bar{\nabla}_{SS}\bar{V}\left(t+1, \sum_{i=1}^{t} c_i\right).
\end{aligned}$$

The remaining task is to show

$$\bar{V}\left(t+1, \sum_{i=1}^{t+1} c_i\right) - \frac{1}{2}\left[\bar{V}\left(t+1, \sum_{i=1}^{t} c_i + 1\right) + \bar{V}\left(t+1, \sum_{i=1}^{t} c_i - 1\right)\right] \le c_{t+1}x_{t+1}.$$

Plugging in the player's bet (10), it suffices to show that

$$\bar{V}\left(t+1, \sum_{i=1}^{t+1} c_i\right) \le \frac{1+c_{t+1}}{2}\bar{V}\left(t+1, \sum_{i=1}^{t} c_i + 1\right) + \frac{1-c_{t+1}}{2}\bar{V}\left(t+1, \sum_{i=1}^{t} c_i - 1\right),$$

which follows from the convexity of $\bar{V}$. Equality is achieved when $c_{t+1} \in \{-1, 1\}$. □

B.3 Preliminary: Properties of $\bar{V}_{1/2}$ and $\bar{V}_{-1/2}$

In the Discrete Itô formula, quantifying the perturbation term $\Diamond$ is case-dependent. Before doing so, we present some facts on $\bar{V}_{1/2}$ and $\bar{V}_{-1/2}$ which will be useful later. Let us first consider $\bar{V}_{1/2}$. For clarity, we copy the definition here.

$$\bar{V}_{1/2}(t, S) = C\sqrt{t}\left[2\int_0^{S/\sqrt{2t}}\left(\int_0^u \exp(x^2)dx\right)du - 1\right].$$

Some calculation yields its derivatives. Let ∇_t and ∇_{tt} be the first and second order derivative with respect to t. Let ∇_S, ∇_{SS}, ∇_{SSS}, ∇_{SSSS} be the first to fourth order derivative with respect to S.

$$\nabla_S \bar{V}_{1/2}(t,S) = \sqrt{2}C \int_0^{S/\sqrt{2t}} \exp(x^2)dx. \qquad \nabla_{SSSS} \bar{V}_{1/2}(t,S) = \frac{C}{t^{3/2}} \exp\left(\frac{S^2}{2t}\right)\left(\frac{S^2}{t}+1\right).$$

$$\nabla_{SS} \bar{V}_{1/2}(t,S) = \frac{C}{\sqrt{t}} \exp\left(\frac{S^2}{2t}\right). \qquad \nabla_t \bar{V}_{1/2}(t,S) = -\frac{C}{2\sqrt{t}} \exp\left(\frac{S^2}{2t}\right).$$

$$\nabla_{SSS} \bar{V}_{1/2}(t,S) = \frac{CS}{t^{3/2}} \exp\left(\frac{S^2}{2t}\right). \qquad \nabla_{tt} \bar{V}_{1/2}(t,S) = \frac{C}{4t^{3/2}} \exp\left(\frac{S^2}{2t}\right)\left(\frac{S^2}{t}+1\right).$$

Similarly, for $\bar{V}_{-1/2}$ we have the following.

$$\bar{V}_{-1/2}(t,S) = \frac{C}{\sqrt{t}} \exp\left(\frac{S^2}{2t}\right). \qquad \nabla_{SSSS} \bar{V}_{-1/2}(t,S) = \frac{C}{t^{5/2}} \exp\left(\frac{S^2}{2t}\right)\left(\frac{S^4}{t^2}+\frac{6S^2}{t}+3\right).$$

$$\nabla_S \bar{V}_{-1/2}(t,S) = \frac{CS}{t^{3/2}} \exp\left(\frac{S^2}{2t}\right). \qquad \nabla_t \bar{V}_{-1/2}(t,S) = -\frac{C}{2t^{3/2}} \exp\left(\frac{S^2}{2t}\right)\left(\frac{S^2}{t}+1\right).$$

$$\nabla_{SS} \bar{V}_{-1/2}(t,S) = \frac{C}{t^{3/2}} \exp\left(\frac{S^2}{2t}\right)\left(\frac{S^2}{t}+1\right). \qquad \nabla_{tt} \bar{V}_{-1/2}(t,S) = \frac{C}{4t^{5/2}} \exp\left(\frac{S^2}{2t}\right)\left(\frac{S^4}{t^2}+\frac{6S^2}{t}+3\right).$$

$$\nabla_{SSS} \bar{V}_{-1/2}(t,S) = \frac{C}{t^{3/2}} \exp\left(\frac{S^2}{2t}\right)\left(\frac{S^3}{t^2}+\frac{3S}{t}\right).$$

Let us compare the betting behavior induced by $\bar{V}_{1/2}$ and $\bar{V}_{-1/2}$. The bets in both cases are roughly their derivatives.

Lemma B.1. *For all $t \in \mathbb{N}_+$ and $|S| \le t-1$, $|\nabla_S \bar{V}_{1/2}(t,S)| \ge |\nabla_S \bar{V}_{-1/2}(t,S)|$.*

Proof of Lemma B.1. Due to symmetry, it suffices to consider $0 \le S \le t-1$. Notice that when $S=0$, $\nabla_S \bar{V}_{1/2}(t,0) = \nabla_S \bar{V}_{-1/2}(t,0)$. With $S \le t-1$,

$$\frac{\nabla_{SS} \bar{V}_{-1/2}(t,S)}{\nabla_{SS} \bar{V}_{1/2}(t,S)} = \frac{1}{t}\left(\frac{S^2}{t}+1\right) \le 1 - t^{-1} + t^{-2} \le 1. \qquad \square$$

Moreover, let us specifically compare the derivatives when $|S| = O(\sqrt{t})$: $|\nabla_S \bar{V}_{1/2}(t,S)| = O(1)$ while $|\nabla_S \bar{V}_{-1/2}(t,S)| = O(t^{-1})$. Therefore, the betting behavior induced by $\bar{V}_{1/2}$ cannot be achieved by simply scaling $\bar{V}_{-1/2}$ (i.e., using a different C).

Finally back to $\bar{V}_{1/2}$, the integral definition may not be easy to interpret. We can further lower bound it as the following.

Lemma B.2. *For all (t,S) such that $\sqrt{2t} \le |S| \le t$,*

$$\bar{V}_{1/2}(t,S) \ge C\sqrt{t} \cdot \left[\frac{t}{S^2} \exp\left(\frac{S^2}{2t}\right) - \frac{3}{2}\right].$$

Proof of Lemma B.2. Based on the definition of $\bar{V}_{1/2}$, it suffices to show that for all $z \ge 1$,

$$f(z) := 2\int_0^z \left(\int_0^u \exp(x^2)dx\right) du - \frac{1}{2z^2} \exp\left(z^2\right) \ge -1/2.$$

Notice that $f(1) \ge -1/2$. Taking the derivatives,

$$f'(z) = 2\int_0^z \exp(x^2)dx + \exp(z^2)\left[z^{-3} - z^{-1}\right].$$

$f'(1) \ge 0$, and

$$f''(z) = 3\exp\left(z^2\right)\left(z^{-2} - z^{-4}\right) \ge 0. \qquad \square$$

B.4 Policy induced by $\bar{V}_{1/2}$

In this subsection we characterize the player policy constructed from $\bar{V}_{1/2}$. The first step is to quantify the perturbation error ($\Diamond$ in (12)). After that, the wealth lower bound (Theorem 1) follows from a telescopic sum on the Discrete Itô formula.

Lemma 3.2. *For all $t \in \mathbb{N}_+$ and $S \in [1-t, t-1]$, $\bar{V}_{1/2}$ with any parameter $C > 0$ satisfies*

$$0 \geq \bar{\nabla}_t \bar{V}_{1/2}(t,S) + \bar{\nabla}_{SS} \bar{V}_{1/2}(t,S)/2 \geq \begin{cases} -C, & t = 1, \\ -\frac{C}{8}(t-1)^{-3/2} \exp\left(\frac{S^2}{2(t-1)}\right)\left(\frac{S^2}{t-1}+1\right), & t > 1. \end{cases}$$

Proof of Lemma 3.2. Plugging in the definition of discrete derivative,

$$\bar{\nabla}_t \bar{V}_{1/2}(t,S) + \bar{\nabla}_{SS} \bar{V}_{1/2}(t,S)/2 = \frac{1}{2}\bar{V}_{1/2}(t,S+1) + \frac{1}{2}\bar{V}_{1/2}(t,S-1) - \bar{V}_{1/2}(t-1,S). \tag{14}$$

Let us first consider the upper bound. For clarity, define a function $f : \mathbb{R} \to \mathbb{R}$ as

$$f(z) = 2z \int_0^z \exp(x^2) dx - \exp(z^2).$$

Then, using the definition of $\bar{V}_{1/2}$, it suffices to show that

$$f\left(-\frac{1}{\sqrt{2}}\right) + f\left(\frac{1}{\sqrt{2}}\right) \leq 0,$$

and for all $t > 1$,

$$f\left(\frac{S-1}{\sqrt{2t}}\right) + f\left(\frac{S+1}{\sqrt{2t}}\right) \leq 2\sqrt{1-\frac{1}{t}} f\left(\frac{S}{\sqrt{2(t-1)}}\right).$$

The first inequality can be easily verified by computing the values of $f(1/\sqrt{2})$ and $f(-1/\sqrt{2})$. As for the second inequality, we use an existing result [HLPR20, Lemma C.4]: for all $x \in \mathbb{R}$ and $z \in [0,1)$,

$$f\left(\frac{x-z}{\sqrt{2}}\right) + f\left(\frac{x+z}{\sqrt{2}}\right) \leq 2\sqrt{1-z^2} f\left(\frac{x}{\sqrt{2(1-z^2)}}\right).$$

Taking $x = S/\sqrt{t}$ and $z = 1/\sqrt{t}$ completes the proof.

Next we prove the lower bound. From Taylor's theorem,

$$\bar{V}_{1/2}(t,S+1) = \bar{V}_{1/2}(t,S) + \nabla_S \bar{V}_{1/2}(t,S) + \frac{1}{2}\nabla_{SS}\bar{V}_{1/2}(t,S) + \frac{1}{6}\nabla_{SSS}\bar{V}_{1/2}(t,S) + \frac{1}{24}\nabla_{SSSS}\bar{V}_{1/2}(t,S+a),$$

$$\bar{V}_{1/2}(t,S-1) = \bar{V}_{1/2}(t,S) - \nabla_S \bar{V}_{1/2}(t,S) + \frac{1}{2}\nabla_{SS}\bar{V}_{1/2}(t,S) - \frac{1}{6}\nabla_{SSS}\bar{V}_{1/2}(t,S) + \frac{1}{24}\nabla_{SSSS}\bar{V}_{1/2}(t,S-b),$$

$$\bar{V}_{1/2}(t-1,S) = \bar{V}_{1/2}(t,S) - \nabla_t \bar{V}_{1/2}(t,S) + \frac{1}{2}\nabla_{tt}\bar{V}_{1/2}(t-c,S),$$

where $a, b, c \in [0,1]$. Plugging these into (14) and using the condition $\nabla_t \bar{V}_{1/2} = -\nabla_{SS}\bar{V}_{1/2}/2$ (since $\bar{V}_{1/2}$ is a solution of the backward heat equation), we have

$$\bar{\nabla}_t \bar{V}_{1/2}(t,S) + \bar{\nabla}_{SS}\bar{V}_{1/2}(t,S)/2 = \frac{1}{48}\nabla_{SSSS}\bar{V}_{1/2}(t,S+a) + \frac{1}{48}\nabla_{SSSS}\bar{V}_{1/2}(t,S-b) - \frac{1}{2}\nabla_{tt}\bar{V}_{1/2}(t-c,S).$$

From Appendix B.3, $\nabla_{SSSS}\bar{V}_{1/2}(t,S) \geq 0$ for all (t,S), and

$$\nabla_{tt}\bar{V}_{1/2}(t,S) = \frac{C}{4}t^{-3/2}\exp\left(\frac{S^2}{2t}\right)\left(\frac{S^2}{t}+1\right).$$

Therefore,

$$\begin{aligned}\bar{\nabla}_t \bar{V}_{1/2}(t,S) + \bar{\nabla}_{SS}\bar{V}_{1/2}(t,S)/2 &\geq -\frac{C}{8}\max_{c\in[0,1]}(t-c)^{-3/2}\exp\left(\frac{S^2}{2(t-c)}\right)\left(\frac{S^2}{t-c}+1\right) \\ &= -\frac{C}{8}(t-1)^{-3/2}\exp\left(\frac{S^2}{2(t-1)}\right)\left(\frac{S^2}{t-1}+1\right). \end{aligned}$$

□

Next we prove Theorem 2. It shows that the wealth lower bound (Theorem 1) faithfully characterizes the performance of the player policy constructed from $\bar{V}_{1/2}$.

Theorem 2. *For all $T \in \mathbb{N}_+$ and $S \in [-T, T]$, we can construct $c_1 \in \mathcal{C}$ and $c_2, \ldots, c_T \in \{-1, 1\}$ such that*

1. *$\sum_{t=1}^{T} c_t = S$;*
2. *If the player applies Algorithm 3 constructed from $\bar{V}_{1/2}$ (with parameter C) and the adversary plays the aforementioned coin sequence $c_{1:T}$, then*

$$\mathit{Wealth}_T \le \bar{V}_{1/2}(T, S) + \frac{3C}{8} \exp\left(\frac{S^2}{2T}\right)\left(\frac{S^2}{T} + 1\right) + 2C.$$

Proof of Theorem 2. We first construct the coin sequence. For all $S \in [-T, T]$, there exists an integer $\tilde{S}$ such that $|\tilde{S}| \le T$, $(|\tilde{S}|+1) \bmod 2 = T \bmod 2$ and $|S - \tilde{S}| \le 1$. We define the coins using three phases.

1. $c_1 = S - \tilde{S}$;
2. For all $1 < t \le T - |\tilde{S}|$, let $c_t = \mathrm{sign}(c_1) \cdot (-1)^{t-1}$;
3. If $\tilde{S} \ne 0$, then for all t such that $T - |\tilde{S}| < t \le T$, let $c_t = \tilde{S}/|\tilde{S}|$.

Based on this coin sequence, there are three immediate observations:

1. The sum of coins from the second phase is 0, and the sum of coins from the third phase is $\tilde{S}$; therefore, $\sum_{t=1}^{T} c_t = S$.
2. If $\tau \le T - |\tilde{S}|$ then $|\sum_{t=1}^{\tau} c_t| \le 1$.
3. If $T - |\tilde{S}| < \tau \le T$ then $|\sum_{t=1}^{\tau} c_t| = |S| - T + \tau$.

Next, we derive the wealth upper bound induced by such a coin sequence and the player policy (Algorithm 3). Starting from the first round, $x_1 = 0$, therefore $\mathrm{Wealth}_1 = 0$. $\mathrm{Wealth}_1 = \bar{V}_{1/2}(1, c_1) - \bar{V}_{1/2}(1, c_1) \le \bar{V}_{1/2}(1, c_1) - \bar{V}_{1/2}(1, 0) = \bar{V}_{1/2}(1, c_1) + C$. Considering the rest of the rounds, there are two cases: (i) $|S| \le \sqrt{T}$; (ii) $|S| > \sqrt{T}$.

Case (i) In this case we first show that for all integer τ in $[1:T]$, $|\sum_{t=1}^{\tau} c_t| \le \sqrt{\tau}$. Due to the second observation above, this condition holds for all $\tau \le T - |\tilde{S}|$, and we only need to focus on $T - |\tilde{S}| < \tau \le T$ (the third phase) where $|\sum_{t=1}^{\tau} c_t| = |S| - T + \tau \le \sqrt{T} - T + \tau$; since $T - \sqrt{T} \ge \tau - \sqrt{\tau}$, we further have $|\sum_{t=1}^{\tau} c_t| \le \sqrt{\tau}$. Based on this result, telescoping Lemma 3.1 (notice that equality is achieved) and using Lemma 3.2, we have

$$\begin{aligned}
\mathrm{Wealth}_T &\le \bar{V}_{1/2}\left(T, \sum_{t=1}^{T} c_t\right) + C + \frac{C}{8} \sum_{t=1}^{T-1} t^{-3/2} \exp\left(\frac{(\sum_{i=1}^{t} c_i)^2}{2t}\right)\left(\frac{(\sum_{i=1}^{t} c_i)^2}{t} + 1\right) \\
&\le \bar{V}_{1/2}\left(T, \sum_{t=1}^{T} c_t\right) + C + \frac{\sqrt{e}C}{4} \sum_{t=1}^{T-1} t^{-3/2} \le \bar{V}\left(T, \sum_{t=1}^{T} c_t\right) + \left(\frac{3\sqrt{e}}{4} + 1\right) C.
\end{aligned}$$

Case (ii) In this case we show that for all integer τ in $[1:T]$, $|\sum_{t=1}^{\tau} c_t|/\sqrt{\tau} \le |\sum_{t=1}^{T} c_t|/\sqrt{T}$. Similar to Case (i), we consider $\tau \le T - |\tilde{S}|$ and $T - |\tilde{S}| < \tau \le T$ separately. When $\tau \le T - |\tilde{S}|$, we have $|\sum_{t=1}^{\tau} c_t|/\sqrt{\tau} \le 1 \le |S|/\sqrt{T} = |\sum_{t=1}^{T} c_t|/\sqrt{T}$. On the other hand, when $T - |\tilde{S}| < \tau \le T$ it suffices to show that

$$\frac{|S| - T + \tau}{\sqrt{\tau}} \le \frac{|S|}{\sqrt{T}}.$$

The LHS monotonically increases with respect to τ, and when $\tau = T$ the inequality holds with equality. In summary, the required condition $|\sum_{t=1}^{\tau} c_t|/\sqrt{\tau} \le |\sum_{t=1}^{T} c_t|/\sqrt{T}$ holds for all $\tau \in [1:T]$.

Based on this result, telescoping Lemma 3.1 and using Lemma 3.2, we have

$$\begin{aligned}\text{Wealth}_T &\le \bar{V}_{1/2}\left(T, \sum_{t=1}^{T} c_t\right) + C + \frac{C}{8}\exp\left(\frac{(\sum_{i=1}^{T} c_i)^2}{2T}\right)\left(\frac{(\sum_{i=1}^{T} c_i)^2}{T} + 1\right)\sum_{t=1}^{T-1} t^{-3/2} \\ &\le \bar{V}_{1/2}\left(T, \sum_{t=1}^{T} c_t\right) + C + \frac{3C}{8}\exp\left(\frac{(\sum_{i=1}^{T} c_i)^2}{2T}\right)\left(\frac{(\sum_{i=1}^{T} c_i)^2}{T} + 1\right).\end{aligned}$$

Combining Case (i) and Case (ii) completes the proof. □

Theorem 2 has a special form: it fixes both the player policy (Algorithm 3) and the adversary policy (Algorithm 2), and then bounds the wealth induced by both of them. Results of this form are seldom studied in conventional online learning settings. The reason is that, the performance metric for those settings is usually the uniform regret (a real number), therefore the gap between policy-independent upper and lower bounds is relatively easy to describe. In contrast, we care about the trade-offs on our performance metric, so our upper and lower bounds are both expressed as functions; the characterization of their gap is much richer. We present our player-policy-independent wealth upper bound as Theorem 3. It is related, but incomparable to Theorem 2 stated above.

B.5 Policy induced by $\bar{V}_{-1/2}$

Analogous to the previous subsection, we now characterize the performance of the player policy induced by $\bar{V}_{-1/2}$. The first step is to quantify the perturbation error $\Diamond$.

Lemma B.3. *For all $t \in \mathbb{N}_+$ and $S \in [1-t, t-1]$, $\bar{V}_{-1/2}$ with any parameter $C > 0$ satisfies the following conditions.*

1. *If $t = 1$, then*
$$\bar{\nabla}_t \bar{V}_{-1/2}(t, S) + \bar{\nabla}_{SS} \bar{V}_{-1/2}(t, S)/2 = C\sqrt{e}.$$
2. *If $t > 1$, then*
$$-\frac{C}{8}(t-1)^{-5/2}\exp\left(\frac{S^2}{2(t-1)}\right)\left(\frac{S^4}{(t-1)^2} + \frac{6S^2}{t-1} + 3\right) \le \bar{\nabla}_t \bar{V}_{-1/2}(t, S) + \bar{\nabla}_{SS} \bar{V}_{-1/2}(t, S)/2 \le 0.$$

Proof of Lemma B.3. The case of $t = 1$ can be easily verified. We will prove the second case next. Plugging in the definition, we have

$$\begin{aligned}\bar{\nabla}_t \bar{V}_{-1/2}(t, S) + \bar{\nabla}_{SS} \bar{V}_{-1/2}(t, S)/2 &= \frac{1}{2}\bar{V}_{-1/2}(t, S+1) + \frac{1}{2}\bar{V}_{-1/2}(t, S-1) - \bar{V}_{-1/2}(t-1, S) \\ &= \frac{C}{2\sqrt{t}}\exp\left(\frac{(S-1)^2}{2t}\right) + \frac{C}{2\sqrt{t}}\exp\left(\frac{(S+1)^2}{2t}\right) - \frac{C}{\sqrt{t-1}}\exp\left(\frac{S^2}{2(t-1)}\right).\end{aligned} \tag{15}$$

First, let us consider the upper bound. Since $\exp(-t^{-1}) \ge 1 - t^{-1}$, we have

$$\exp\left(\frac{1}{2t}\right) \le \sqrt{\frac{t}{t-1}}.$$

Therefore,

$$\begin{aligned}\exp\left(\frac{(S-1)^2}{2t}\right) + \exp\left(\frac{(S+1)^2}{2t}\right) &= \exp\left(\frac{S^2+1}{2t}\right)\left[\exp\left(-\frac{S}{t}\right) + \exp\left(\frac{S}{t}\right)\right] \\ &\le \sqrt{\frac{t}{t-1}}\exp\left(\frac{S^2}{2t}\right)\left[\exp\left(-\frac{S}{t}\right) + \exp\left(\frac{S}{t}\right)\right] \\ &\le 2\sqrt{\frac{t}{t-1}}\exp\left(\frac{S^2}{2t}\right)\exp\left(\frac{S^2}{2t^2}\right),\end{aligned}$$

where the last inequality is due to the classical result $\cosh(x) \le \exp(x^2/2)$. Back to (15),

$$\bar{\nabla}_t \bar{V}_{-1/2}(t,S) + \bar{\nabla}_{SS} \bar{V}_{-1/2}(t,S)/2 \le C\sqrt{\frac{1}{t-1}} \left[\exp\left(\frac{S^2}{2t}\right) \exp\left(\frac{S^2}{2t^2}\right) - \exp\left(\frac{S^2}{2(t-1)}\right) \right],$$

and it is straightforward to verify that RHS ≤ 0.

Next, we consider the lower bound. Similar to the proof of Lemma 3.2, using the derivatives from Appendix B.3,

$$\begin{aligned} \bar{\nabla}_t \bar{V}_{-1/2}(t,S) + \bar{\nabla}_{SS} \bar{V}_{-1/2}(t,S)/2 &\ge -\frac{1}{2} \nabla_{tt} \bar{V}_{-1/2}(t-1,S) \\ &= -\frac{C}{8}(t-1)^{-5/2} \exp\left(\frac{S^2}{2(t-1)}\right) \left(\frac{S^4}{(t-1)^2} + \frac{6S^2}{t-1} + 3\right). \end{aligned}$$

□

Similar to the wealth lower bound induced by $\bar{V}_{1/2}$ (Theorem 1), we can plug the above lemma into the Discrete Itô formula (Lemma 3.1) and obtain the following theorem via a telescopic sum. The proof is omitted. Essentially, a wealth lower bound of this form recovers the result from [OP16]. However, our analysis is based on a general framework without budget constraints, therefore does not involve any *betting fractions*.

Theorem 7. *For all $T \in \mathbb{N}_+$, Algorithm 3 constructed from $\bar{V}_{-1/2}$ guarantees a wealth lower bound*

$$\mathit{Wealth}_T \ge \bar{V}_{-1/2}\left(T, \sum_{t=1}^{T} c_t\right) - C\sqrt{e},$$

against any adversary policy $\boldsymbol{a}$.

In addition, analogous to Theorem 2, we can also state a wealth upper bound based on $\bar{V}_{-1/2}$. The proof uses a similar strategy, therefore is omitted.

Theorem 8. *For all $T \in \mathbb{N}_+$ and $S \in [-T, T]$, we can construct $c_1 \in \mathcal{C}$ and $c_2, \dots, c_T \in \{-1, 1\}$ such that*

1. *$\sum_{t=1}^{T} c_t = S$;*
2. *If the player applies Algorithm 3 constructed from $\bar{V}_{-1/2}$ (with parameter C) and the adversary plays the aforementioned coin sequence $c_{1:T}$, then*

$$\mathit{Wealth}_T \le \bar{V}_{-1/2}(T,S) + \frac{5C}{24} \exp\left(\frac{S^2}{2T}\right) \left(\frac{S^4}{T^2} + \frac{6S^2}{T} + 3\right) + 2C.$$

B.6 Detail on the optimality of Algorithm 3

Finally, in this subsection we prove the player-policy-independent wealth upper bounds (Theorem 3 and its analogy based on $\bar{V}_{-1/2}$). The first step is to prove a sharp lower bound for the tail probability of one-dimensional symmetric random walk, based on a normal approximation. The idea is simple, but we could not find the suitable result in existing literature.

Lemma B.4. *For all $T \in \mathbb{N}_+$, let $z_1, \dots, z_T$ be i.i.d. Rademacher random variables. Then for any $k > 0$,*

$$\mathbb{P}\left[\left|\sum_{t=1}^{T} c_t\right| \ge k\right] \ge \sqrt{\frac{2}{\pi}} \frac{k\sqrt{T}}{k^2 + T} \exp\left(-\frac{k^2}{2T}\right) - \frac{1}{\sqrt{T}}.$$

Proof of Lemma B.4. Due to Central Limit Theorem, the random variable $(\sum_{t=1}^{T} c_t)/\sqrt{T}$ converges in distribution to standard normal $N(0,1)$. Concretely, the nonasymptotic convergence rate can be characterized via the Berry-Esseen Theorem [She11]: Let $F_T(x)$ be the CDF of $(\sum_{t=1}^{T} c_t)/\sqrt{T}$ and $\Phi(x)$ be the standard normal CDF, then,

$$\sup_{x \in \mathbb{R}} |F_T(x) - \Phi(x)| \le \frac{1}{2\sqrt{T}}.$$

For the tail probability of standard normal distribution, there is a standard lower bound (e.g., [Due10]) which can be verified via a derivative argument: For all $x > 0$,

$$1 - \Phi(x) \geq \frac{1}{\sqrt{2\pi}} \frac{1}{x + x^{-1}} \exp\left(-\frac{x^2}{2}\right).$$

Therefore,

$$\mathbb{P}\left[\left|\sum_{t=1}^{T} c_t\right| \geq k\right] = 2 \cdot \left[1 - F_T(k/\sqrt{T})\right] \geq 2 \cdot \left[1 - \Phi(k/\sqrt{T}) - \frac{1}{2\sqrt{T}}\right] \geq \sqrt{\frac{2}{\pi}} \frac{k\sqrt{T}}{k^2 + T} \exp\left(-\frac{k^2}{2T}\right) - \frac{1}{\sqrt{T}}. \quad \square$$

Compared to similar tail lower bounds from existing works on unconstrained OLO [SM12, Ora13], Lemma B.4 has the tight exponent (1/2) in the exponential function. This allows us to justify the optimality of our PDE-based coin-betting policy (Algorithm 3 constructed from $\bar{V}_{1/2}$), and eventually the converted unconstrained OLO algorithm.

Theorem 3. *For all $\lambda \geq \exp[(\sqrt{2}+1)/2]$, $T \geq 8\pi\lambda^2 \log \lambda$, and any player policy $\boldsymbol{p}$ that guarantees $\text{Wealth}_T \geq -C\sqrt{T}$ (e.g., Algorithm 3 constructed from $\bar{V}_{1/2}$), there exists an adversary policy $\boldsymbol{a}$ such that the following statement holds. In the coin-betting game induced by the policy pair $(\boldsymbol{p}, \boldsymbol{a})$,*

1. $|\sum_{t=1}^{T} c_t| \geq \sqrt{2T \log \lambda}$;
2. $\text{Wealth}_T \leq 2\sqrt{2\pi}\lambda\sqrt{\log \lambda} \cdot C\sqrt{T}$.

Proof of Theorem 3. Let us first generalize the unconstrained coin-betting game to allow random adversary on the coin space $\{-1, 1\}$. That is, based on past player bets $x_1, \ldots, x_t$, the adversary decides a distribution on $\{-1, 1\}$ and samples c_t from this distribution.

Now, consider the setting where the player applies any policy $\boldsymbol{p}$ that guarantees $\text{Wealth}_T \geq -C\sqrt{T}$, and the adversary picks coin outcomes according to a Rademacher distribution: regardless of $x_1, \ldots, x_t$, the coin c_t equals -1 and 1 with probability 1/2 respectively. Then for all $T \in \mathbb{N}_+$, let $k = \sqrt{2T \log \lambda}$.

$$\begin{aligned}
0 &= \mathbb{E}\left[\sum_{t=1}^{T} c_t x_t\right] \\
&= \mathbb{E}\left[\sum_{t=1}^{T} c_t x_t \middle| \left|\sum_{t=1}^{T} c_t\right| \geq k\right] \mathbb{P}\left[\left|\sum_{t=1}^{T} c_t\right| \geq k\right] + \mathbb{E}\left[\sum_{t=1}^{T} c_t x_t \middle| \left|\sum_{t=1}^{T} c_t\right| < k\right] \mathbb{P}\left[\left|\sum_{t=1}^{T} c_t\right| < k\right] \\
&\geq \mathbb{E}\left[\sum_{t=1}^{T} c_t x_t \middle| \left|\sum_{t=1}^{T} c_t\right| \geq k\right] \mathbb{P}\left[\left|\sum_{t=1}^{T} c_t\right| \geq k\right] - C\sqrt{T}.
\end{aligned}$$

Applying Lemma B.4, using $\lambda \geq \exp[(\sqrt{2}+1)/2]$ and $T \geq 8\pi\lambda^2 \log \lambda$,

$$\begin{aligned}
\mathbb{P}\left[\left|\sum_{t=1}^{T} c_t\right| \geq k\right] &\geq \sqrt{\frac{2}{\pi}} \frac{\sqrt{2\log\lambda}}{1 + 2\log\lambda} \lambda^{-1} - \frac{1}{\sqrt{T}} \\
&\geq \frac{1}{\sqrt{2\pi\log\lambda}} \lambda^{-1} - \frac{1}{\sqrt{T}} \geq \frac{1}{2\sqrt{2\pi\log\lambda}} \lambda^{-1}.
\end{aligned}$$

$$\mathbb{E}\left[\sum_{t=1}^{T} c_t x_t \middle| \left|\sum_{t=1}^{T} c_t\right| \geq k\right] \leq \frac{C\sqrt{T}}{\mathbb{P}\left[\left|\sum_{t=1}^{T} c_t\right| \geq k\right]} \leq 2\sqrt{2\pi}\lambda\sqrt{\log\lambda} \cdot C\sqrt{T}.$$

Therefore, for any player policy $\boldsymbol{p}$ there exists an adversary policy $\boldsymbol{a}$ which induces $|\sum_{t=1}^{T} c_t| \geq \sqrt{2T \log \lambda}$ and $\text{Wealth}_T \leq 2\sqrt{2\pi}\lambda\sqrt{\log\lambda} \cdot C\sqrt{T}$. $\square$

A similar result can be stated with respect to $\bar{V}_{-1/2}$, using a different "barrier" k that depends on T. This introduces a specific technical issue: when we use Lemma B.4, the normal approximation error $(1/\sqrt{T})$ is comparable in magnitude to the Gaussian tail bound (the first term in Lemma B.4) which we care about. Therefore, the following theorem has a slightly weaker form than Theorem 3.

Theorem 9. *For all $\lambda \geq \exp[(\sqrt{2}+1)/2]$, there exists $T_0 \in \mathbb{N}_+$ (depending on λ) such that for all $T \geq T_0$ and any player policy $\boldsymbol{p}$ which guarantees $\mathrm{Wealth}_T \geq -C\sqrt{e}$ (e.g., Algorithm 3 constructed from $\bar{V}_{-1/2}$), there exists an adversary policy $\boldsymbol{a}$ with the following property. In the coin-betting game induced by the policy pair $(\boldsymbol{p}, \boldsymbol{a})$,*

1. $\left|\sum_{t=1}^T c_t\right| \geq \sqrt{2T\log(\lambda\sqrt{T}/\log T)}$;
2. $\mathrm{Wealth}_T \leq 2\sqrt{2\pi e}\lambda(\log T)^{-1}\sqrt{\log(\lambda\sqrt{T}/\log T)} \cdot C\sqrt{T}$.

Proof of Theorem 9. We follow a similar analysis as Theorem 3 but use a different barrier. Let us only consider $T > 1$ and let $k = \sqrt{2T\log(\lambda\sqrt{T}/\log T)}$. Using the Rademacher random adversary,

$$\mathbb{E}\left[\sum_{t=1}^T c_t x_t \middle| \left|\sum_{t=1}^T c_t\right| \geq k\right] \mathbb{P}\left[\left|\sum_{t=1}^T c_t\right| \geq k\right] \leq C.$$

Using $\lambda \geq \exp[(\sqrt{2}+1)/2]$, we have $1 \leq 2(\sqrt{2}-1)\log(\lambda) \leq 2(\sqrt{2}-1)\log(\lambda\sqrt{T}/\log T)$. Therefore,

$$\begin{aligned}\mathbb{P}\left[\left|\sum_{t=1}^T c_t\right| \geq k\right] &\geq \sqrt{\frac{2}{\pi}}\frac{\sqrt{2\log(\lambda\sqrt{T}/\log T)}}{1+2\log(\lambda\sqrt{T}/\log T)}\frac{\log T}{\lambda\sqrt{T}} - \frac{1}{\sqrt{T}} \\ &\geq \frac{\log T}{\lambda\sqrt{2\pi\log(\lambda\sqrt{T}/\log T)}}T^{-1/2} - T^{-1/2}.\end{aligned}$$

Since the first term decays slower (with respect to T) than the second term $T^{-1/2}$, there exists T_0 depending on λ such that for all $T \geq T_0$,

$$\mathbb{P}\left[\left|\sum_{t=1}^T c_t\right| \geq k\right] \geq \frac{\log T}{2\lambda\sqrt{2\pi\log(\lambda\sqrt{T}/\log T)}}T^{-1/2},$$

$$\mathbb{E}\left[\sum_{t=1}^T c_t x_t \middle| \left|\sum_{t=1}^T c_t\right| \geq k\right] \leq \frac{C\sqrt{e}}{\mathbb{P}\left[\left|\sum_{t=1}^T c_t\right| \geq k\right]} \leq 2\sqrt{2\pi e}\lambda(\log T)^{-1}\sqrt{\log(\lambda\sqrt{T}/\log T)} \cdot C\sqrt{T}. \qquad \square$$

C Detail on unconstrained OLO

In this section we present detailed analysis on unconstrained OLO. First, using the conversion from coin-betting to OLO (Algorithm 1), our coin-betting policy (Algorithm 3) can be directly converted into a one-dimensional unconstrained OLO algorithm. For clarity, we restate its pseudo-code as Algorithm 5.

Algorithm 5 PDE-based one-dimensional unconstrained OLO algorithm.

Require: A one-dimensional limiting value function $\bar{V}$ which satisfies (5).

1: **for** $t = 1, 2, \ldots$ **do**

2: Predict

$$x_t = \frac{1}{2}\left[\bar{V}\left(t, -\sum_{i=1}^{t-1} g_i + 1\right) - \bar{V}\left(t, -\sum_{i=1}^{t-1} g_i - 1\right)\right].$$

3: Observe the loss gradient g_t and store it.

4: **end for**

For general d-dimensional problems, we rely on a classical reduction [CO18] to the one-dimensional problem. Its pseudo-code is Algorithm 6, and the associated performance guarantee is Lemma C.1 whose proof follows from [CO18, Theorem 2] and the standard regret bound of OGD (e.g., [Ora19, Section 4.2.1]). Our final product is Algorithm 4 presented in the main paper.

Lemma C.1 (Theorem 2 of [CO18], adapted)**.** *For all $T \in \mathbb{N}_+$, if $\mathcal{A}_{1d}$ guarantees regret bound $\mathrm{Regret}_T(u) \leq R_T(u)$ for all $u \in \mathbb{R}$, then Algorithm 6 guarantees $\mathrm{Regret}_T(u) \leq R_T(\|u\|) + \|u\|\sqrt{2T}$ for all $u \in \mathbb{R}^d$.*

Algorithm 6 Reducing unconstrained OLO from $\mathbb{R}^d$ to $\mathbb{R}$.

Require: A one-dimensional unconstrained OLO algorithm $\mathcal{A}_{1d}$.

Define $\mathcal{A}_B$ as the standard Online Gradient Descent (OGD) on B^d with learning rate $\eta_t = 1/\sqrt{t}$, initialized at the origin.

for $t = 1, 2, \dots$ **do**

  Obtain predictions $y_t \in \mathbb{R}$ from $\mathcal{A}_{1d}$ and $z_t \in \mathbb{R}^d$ from $\mathcal{A}_B$.

  Predict $x_t = y_t z_t \in \mathbb{R}^d$, observe $g_t \in \mathbb{R}^d$.

  Return $\langle g_t, z_t \rangle$ and g_t as the t-th loss gradient to $\mathcal{A}_r$ and $\mathcal{A}_B$, respectively.

end for

C.1 OLO algorithm induced by $\bar{V}_{1/2}$

Next, we consider Algorithm 4 and prove the regret upper bound induced by $\bar{V}_{1/2}$.

Theorem 4. *For all $T \in \mathbb{N}_+$ and $u \in \mathbb{R}^d$, against any adversary, Algorithm 4 constructed from $\bar{V}_{1/2}$ guarantees*

$$\mathit{Regret}_T(u) \leq C\sqrt{T} + \|u\| \sqrt{2T} \left[\sqrt{\log\left(1 + \frac{\|u\|}{\sqrt{2}C}\right)} + 2 \right].$$

Proof of Theorem 4. The proof follows from the combination of Lemma 2.1, Theorem 1 and Lemma C.1. Specifically, let us first guarantee the performance of the y_t sequence. For clarity, given any T, define a one-dimensional function f_T as $f_T(S) = \bar{V}_{1/2}(T, S)$. Combining Lemma 2.1 and Theorem 1, for any $T \in \mathbb{N}_+$ and $w \in \mathbb{R}$ we have

$$\sum_{t=1}^{T} \langle g_t, z_t \rangle y_t - \sum_{t=1}^{T} \langle g_t, z_t \rangle w \leq f_T^*(w).$$

Then, due to Lemma C.1, for all $T \in \mathbb{N}_+$ and $u \in \mathbb{R}^d$ Algorithm 4 guarantees

$$\mathrm{Regret}_T(u) \leq f_T^*(\|u\|) + \|u\| \sqrt{2T}.$$

The remaining task is to bound the Fenchel conjugate f_T^*. For all $w \in \mathbb{R}$,

$$f_T^*(w) = \sup_{S \in \mathbb{R}} Sw - f_T(S).$$

Let S^* be the maximizing argument. Without loss of generality (due to symmetry), assume $w \geq 0$ and therefore $S^* \geq 0$. We have

$$w = \nabla f_T(S^*) = \sqrt{2}C \int_0^{S^*/\sqrt{2T}} \exp(z^2) dz.$$

For any $x \geq 0$, consider the function $f(x) = \int_0^x \exp(z^2) dz$. It is lower bounded by $g(x) = \exp(x^2 - x) - 1$, as $f(0) = g(0)$, and

$$f'(x) = \exp(x^2) \geq \exp(x^2 - x)(2x - 1) = g'(x),$$

due to the standard inequality $\exp(x) \geq 2x - 1$. Therefore,

$$\frac{w}{\sqrt{2}C} = \int_0^{S^*/\sqrt{2T}} \exp(z^2) dz \geq \exp\left[\left(\frac{S^*}{\sqrt{2T}} - \frac{1}{2} \right)^2 - \frac{1}{4} \right] - 1,$$

$$S^* \leq \sqrt{2T} \left[\sqrt{\frac{1}{4} + \log\left(1 + \frac{w}{\sqrt{2}C}\right)} + \frac{1}{2} \right].$$

Now consider $f_T^*(w)$. Since $f_T(S^*) \geq -C\sqrt{T}$ and $\sqrt{x + (1/4)} \leq \sqrt{x} + (1/2)$,

$$f_T^*(w) = S^* w - f_T(S^*) \leq S^* w + C\sqrt{T} \leq C\sqrt{T} + w\sqrt{2T} \left[\sqrt{\log\left(1 + \frac{w}{\sqrt{2}C}\right)} + 1 \right].$$

Combining everything completes the proof. □

Converting Theorem 3 to unconstrained OLO, we also have a regret lower bound with respect to all algorithms (satisfying a condition).

Theorem 10. *For all $\eta \in (0,1)$, $U \geq 12\eta^{-1}C$, $T \geq 2\eta^2 U^2 C^{-2} \log(\eta U C^{-1})$ and any unconstrained OLO algorithm $\mathcal{A}$ that guarantees $Regret_T(0) \leq C\sqrt{T}$ (e.g., Algorithm 4 constructed from $\bar{V}_{1/2}$), there exists an adversary and a comparator $u \in \mathbb{R}^d$ such that $\|u\| = U$ and*

$$Regret_T(u) \geq (1-\eta)\|u\| \sqrt{2T \log \frac{\eta \|u\|}{2\sqrt{\pi}C}}.$$

Proof of Theorem 10. We start by proving the regret lower bound for one-dimensional unconstrained OLO. Extension to the general d-dimensional problem will be considered later.

For the one-dimensional problem, we first invoke a particular version of Theorem 3 on unconstrained coin-betting. Specifically, for any constants $\eta \in (0,1)$ and $u \in \mathbb{R}/\{0\}$ we define λ in Theorem 3 as

$$\lambda = \frac{\eta |u|}{2\sqrt{\pi}C}.$$

For convenience of notation we also define

$$T_0 = \frac{2\eta^2 |u|^2}{C^2} \log\left(\frac{\eta |u|}{2\sqrt{\pi}C}\right).$$

Then, Theorem 3 yields the following result: For all $\eta \in (0,1)$, $|u| \geq 2\sqrt{\pi}\exp[(\sqrt{2}+1)/2]\eta^{-1}C$, $T \geq T_0$ and any coin-betting player policy $\boldsymbol{p}$ that guarantees $\text{Wealth}_T \geq -C\sqrt{T}$, there exists a coin-betting adversary policy $\boldsymbol{a}$ such that in the game induced by $(\boldsymbol{p}, \boldsymbol{a})$,

1. $|\sum_{t=1}^T c_t| \geq \sqrt{2T \log \lambda}$;
2. $\text{Wealth}_T \leq \eta |u| \sqrt{2T \log \lambda}$.

Using Algorithm 1, we can equivalently convert OLO to coin-betting by letting $c_t = -g_t$. Then, the above result immediately translates to the following statement on one-dimensional unconstrained OLO: For all $\eta \in (0,1)$, $|u| \geq 2\sqrt{\pi}\exp[(\sqrt{2}+1)/2]\eta^{-1}C$, $T \geq T_0$ and any unconstrained OLO algorithm $\mathcal{A}$ that guarantees the cumulative loss bound $\sum_{t=1}^T g_t x_t \leq C\sqrt{T}$, there exists an OLO adversary such that in the induced game,

1. $|\sum_{t=1}^T g_t| \geq \sqrt{2T \log \lambda}$;
2. $-\sum_{t=1}^T g_t x_t \leq \eta |u| \sqrt{2T \log \lambda}$.

Let us consider the regret of $\mathcal{A}$ in this setting with respect to comparators u and $-u$. Using the above result,

$$\begin{aligned}
\max\{\text{Regret}_T(u), \text{Regret}_T(-u)\} &= \sum_{t=1}^T g_t x_t + \max\left\{-\sum_{t=1}^T g_t u, \sum_{t=1}^T g_t u\right\} \\
&= \sum_{t=1}^T g_t x_t + \left|\sum_{t=1}^T g_t\right| |u| \\
&\geq (1-\eta)|u| \sqrt{2T \log \lambda} \\
&= (1-\eta)|u| \sqrt{2T \log \frac{\eta |u|}{2\sqrt{\pi}C}}.
\end{aligned}$$

Thus we have proved the desirable result when $d = 1$.

Extending this result to d-dimension follows from a standard technique: consider adversaries whose loss vectors g_t are only nonzero in one coordinate. Let $g_t = [g_{t,1}, \ldots, g_{t,d}]$, and assume $g_{t,2} = \ldots = g_{t,d} = 0$. Then, for any player who plays against this adversary and competes against $u = [u_1, 0, \ldots, 0]$,

$$\text{Regret}_T(u) = \sum_{t=1}^T \langle g_t, x_t \rangle - \sum_{t=1}^T \langle g_t, u \rangle = \sum_{t=1}^T g_{t,1} x_{t,1} - \sum_{t=1}^T g_{t,1} u_1,$$

$\|u\| = |u_1|$, and the cumulative loss satisfies $\sum_{t=1}^T \langle g_t, x_t\rangle = \sum_{t=1}^T g_{t,1}x_{t,1}$. Therefore, any d-dimensional algorithm that guarantees $\mathrm{Regret}_T(0) \le C\sqrt{T}$ is translated into a one-dimensional algorithm with the same guarantee, and our one-dimensional regret lower bound can be applied. □

Finally, for a clear comparison of the upper and lower bounds, we have the following theorem presented in the main paper.

Theorem 5. *Define $\mathcal{A}_{1/2}$ as Algorithm 4 constructed from $\bar{V}_{1/2}$, then Theorem 4 leads to*

$$\limsup_{U\to\infty}\limsup_{T\to\infty}\sup_{\|u\|=U,adv}\frac{\mathit{Regret}_T^{\mathcal{A}_{1/2},adv}(u)}{\|u\|\sqrt{T\log\|u\|}} \le \sqrt{2}.$$

Conversely, for all C and any unconstrained OLO algorithm $\mathcal{A}$ (e.g., $\mathcal{A}_{1/2}$) that guarantees $\mathit{Regret}_T^{\mathcal{A},adv}(0) \le C\sqrt{T}$ for all adv and T, we have

$$\liminf_{U\to\infty}\liminf_{T\to\infty}\sup_{\|u\|=U,adv}\frac{\mathit{Regret}_T^{\mathcal{A},adv}(u)}{\|u\|\sqrt{T\log\|u\|}} \ge \sqrt{2}.$$

Proof of Theorem 5. Let us first consider the upper bound. Plugging in Theorem 4,

$$\begin{aligned}
&\limsup_{U\to\infty}\limsup_{T\to\infty}\sup_{\|u\|=U,adv}\frac{\mathrm{Regret}_T^{\mathcal{A}_{1/2},adv}(u)}{\|u\|\sqrt{T\log\|u\|}}\\
\le\ &\limsup_{U\to\infty}\limsup_{T\to\infty}\sup_{\|u\|=U,adv}\left(\frac{C+2\sqrt{2}\|u\|}{\|u\|\sqrt{\log\|u\|}}+\sqrt{2\log\left(1+\frac{\|u\|}{\sqrt{2}C}\right)\log^{-1}\|u\|}\right)\\
\le\ &\lim_{U\to\infty}\frac{C+2\sqrt{2}U}{U\sqrt{\log U}}+\lim_{U\to\infty}\sqrt{2\log\left(1+\frac{U}{\sqrt{2}C}\right)\log^{-1}U}=\sqrt{2}
\end{aligned}$$

As for the lower bound, we use Theorem 10. We first fix any C and any $\mathcal{A}$ satisfying the condition in the theorem to be proved. For all $\eta\in(0,1)$, with $U \ge 12\eta^{-1}C$ and $T \ge 2\eta^2U^2C^{-2}\log(\eta UC^{-1})$,

$$\begin{aligned}
\sup_{\|u\|=U,adv}\frac{\mathrm{Regret}_T^{\mathcal{A},adv}(u)}{\|u\|\sqrt{T\log\|u\|}} &\ge (1-\eta)\sqrt{2\log\frac{\eta U}{2\sqrt{\pi}C}\log^{-1}U}\\
&= (1-\eta)\sqrt{2\left(1+\frac{\log\eta}{\log U}-\frac{\log(2\sqrt{\pi}C)}{\log U}\right)}.
\end{aligned}$$

Taking lim inf on both sides, for all $\eta\in(0,1)$,

$$\liminf_{U\to\infty}\liminf_{T\to\infty}\sup_{\|u\|=U,adv}\frac{\mathrm{Regret}_T^{\mathcal{A},adv}(u)}{\|u\|\sqrt{T\log\|u\|}} \ge \sqrt{2}(1-\eta).$$

Rewriting this statement, we have: for all $\varepsilon\ge 0$ and $\eta\in(0,1)$, there exists U_0 depending on ε and η such that for all $U\ge U_0$,

$$\liminf_{T\to\infty}\sup_{\|u\|=U,adv}\frac{\mathrm{Regret}_T^{\mathcal{A},adv}(u)}{\|u\|\sqrt{T\log\|u\|}} \ge \sqrt{2}-\sqrt{2}\eta-\varepsilon.$$

Finally, using the definition of lim inf completes the proof. □

C.2 OLO algorithm induced by $\bar{V}_{-1/2}$

Similar to the previous subsection, we can also convert our results on $\bar{V}_{-1/2}$ (Appendix B.5) to the OLO setting. Since $\bar{V}_{1/2}$ recovers the existing coin-betting potentials, the converted regret upper bound recovers the classical bound (2). See also [OP16, Corollary 5].

Theorem 11. *For all $T \in \mathbb{N}_+$ and $u \in \mathbb{R}^d$, against any adversary, Algorithm 4 constructed from $\bar{V}_{-1/2}$ guarantees*

$$\mathit{Regret}_T(u) \le C\sqrt{e} + \|u\| \sqrt{2T} \left[\sqrt{\log\left(1 + \frac{\|u\| T}{C}\right)} + 1 \right].$$

Proof of Theorem 11. Following the proof of Theorem 4, the only difference here is to upper bound the Fenchel conjugate of $f_T(S) = \bar{V}_{-1/2}(T, S)$. We use the existing result [OP16, Lemma 18]: for any function $f(x) = \beta \exp(x^2/(2\alpha))$ with $\alpha, \beta > 0$,

$$f^*(y) \le |y| \sqrt{\alpha \log\left(1 + \frac{\alpha y^2}{\beta^2}\right)} - \beta.$$

Therefore,

$$f_T^*(\|u\|) \le C\sqrt{e} + \|u\| \sqrt{T \log\left(1 + \frac{\|u\|^2 T^2}{C^2}\right)} \le C\sqrt{e} + \|u\| \sqrt{2T \log\left(1 + \frac{\|u\| T}{C}\right)}.$$

The rest of the proof is similar to the proof of Theorem 4. □

Next we present the regret lower bound induced by $\bar{V}_{-1/2}$, parallel to Theorem 10.

Theorem 12. *For all $\eta \in (0, 1)$ and $U \ge 12\eta^{-1}C$, there exists $T_0 \in \mathbb{N}_+$ (depending on η, U and C) such that the following statement holds. For all $T \ge T_0$ and any unconstrained OLO algorithm $\mathcal{A}$ that guarantees $\mathit{Regret}_T(0) \le C\sqrt{e}$ (e.g., Algorithm 4 constructed from $\bar{V}_{-1/2}$), there exists an adversary and a comparator $u \in \mathbb{R}^d$ such that $\|u\| = U$ and*

$$\mathit{Regret}_T(u) \ge \left[1 - \eta \left(\log T\right)^{-1}\right] \|u\| \sqrt{2T \log \frac{\eta \|u\| \sqrt{T}}{2\sqrt{\pi e} C \log T}}.$$

The proof is similar to Theorem 10 therefore omitted. In particular, we plug a slightly different choice of λ into Theorem 9: $\lambda = \eta |u| / (2\sqrt{\pi e} C)$.

To our knowledge, existing lower bounds for unconstrained OLO ([SM12, Theorem 7], [Ora13, Theorem 2], [Ora19, Theorem 5.12]) all focused on the "budget constraint" $\mathrm{Regret}_T(0) \le$ constant. Such a setting is different from Theorem 10 presented in the main paper, but same as Theorem 12 above. Compared to those results, Theorem 12 improves the leading constant: previously the best known constant (on the leading term $\|u\| \sqrt{T \log(\|u\| \sqrt{T})}$) was $1/\sqrt{\log 2} \approx 1.201$ [Ora13], while we improve it to $\sqrt{2} \approx 1.414$. This is due to the use of a tighter tail lower bound for one-dimensional random walk (Lemma B.4).

Finally let us compare Theorem 12 to Theorem 11. The leading constants in the upper and lower bounds are 2 and $\sqrt{2}$ respectively (on the leading term $\|u\| \sqrt{T \log(\|u\| \sqrt{T})}$). Future works may consider closing this gap.

C.3 Converting Theorem 2 into OLO

In this subsection we convert our player-dependent wealth upper bound (Theorem 2) into an algorithm-dependent regret lower bound for unconstrained OLO. The first step is to fix an unconstrained OLO algorithm for our analysis. The ideal choice would be our high-dimensional algorithm (Algorithm 4) constructed from $\bar{V}_{1/2}$. However, the polar decomposition adopted in Algorithm 4 introduces some technicalities that are non-essential for understanding the nature of this problem. Therefore, we consider the one-dimensional algorithm (Algorithm 5), where the polar decomposition is not needed.

For Algorithm 5 constructed from $\bar{V}_{1/2}$, we can state the following regret upper bound using the proof of Theorem 4. Since we do not further bound $f_T^*(|u|)$, such a result is tighter than Theorem 4.

Corollary 13. *Denote $f_T(S) = \bar{V}_{1/2}(T, S)$. For all $T \in \mathbb{N}_+$ and $u \in \mathbb{R}$, against any adversary, Algorithm 5 constructed from $\bar{V}_{1/2}$ guarantees*

$$\mathit{Regret}_T(u) \le f_T^*(|u|).$$

The Fenchel conjugate can be slightly simplified: if we define z through $|u| = \sqrt{2}C\int_0^z \exp(x^2)dx$, then $f_T^*(|u|) = C\sqrt{T}\exp(z^2)$. Although the order of $|u|$ is not as clear as in Theorem 4, we can numerically evaluate this bound as in our experiments.

Converting Theorem 2 to OLO, we have

Theorem 14. *Denote $f_T(S) = \bar{V}_{1/2}(T, S)$. For all $T \in \mathbb{N}_+$ and $|u| \leq (3/8)C(T+3)\exp(T/2)$, we can construct a finite sequence of loss gradients $g_1, \dots, g_T \in [-1, 1]$ such that Algorithm 5 constructed from $\bar{V}_{1/2}$ has the regret lower bound*

$$\mathit{Regret}_T(u) \geq f_T^*(|u|) - O(|u|\log|u|),$$

against the aforementioned loss gradients. $O(\cdot)$ subsumes absolute constants.

Proof of Theorem 14. For convenience, let us define the function

$$h_T(S) = \bar{V}_{1/2}(T, S) + \frac{3C}{8}\exp\left(\frac{S^2}{2T}\right)\left(\frac{S^2}{T} + 1\right) + 2C.$$

Directly applying Theorem 2 yields the following result. For all $T \in \mathbb{N}_+$ and $S \in [-T, T]$, there exists $g_1, \dots, g_T \in [-1, 1]$ such that (i) $-\sum_{t=1}^T g_t = S$; and (ii) Algorithm 5 constructed from $\bar{V}_{1/2}$ satisfies $\sum_{t=1}^T g_t x_t \geq -h_T(S)$ against loss gradients $g_{1:T}$.

Define a variable u^* as

$$u^* = h_T'(S) = \sqrt{2}C\int_0^{S/\sqrt{2T}} \exp(x^2)dx + \frac{3CS}{8T}\exp\left(\frac{S^2}{2T}\right)\left(\frac{S^2}{T} + 3\right).$$

Since S is arbitrary within the interval $[-T, T]$, u^* can take any value within $[-U, U]$, where $U = (3/8)C(T+3)\exp(T/2)$. Due to a standard result from convex analysis [Roc15, Theorem 23.5], $h_T(S) + h_T^*(u^*) = Su^*$. Therefore,

$$\mathrm{Regret}_T(u^*) = \sum_{t=1}^T g_t x_t - \sum_{t=1}^T g_t u^* \geq -h_T(S) + Su^* = h_T^*(u^*).$$

The remaining task is to lower bound $h_T^*(\cdot)$.

Without loss of generality, assume $u \geq 0$. Let us define a variable $\tilde{S}$ through the equation

$$u = \sqrt{2}C\int_0^{\tilde{S}/\sqrt{2T}} \exp(z^2)dz.$$

Then, using the proof of Theorem 4,

$$h_T^*(u) = \sup_{S\in\mathbb{R}} Su - h_T(S) \geq \tilde{S}u - h_T(\tilde{S}) = f_T^*(u) - \frac{3C}{8}\exp\left(\frac{\tilde{S}^2}{2T}\right)\left(\frac{\tilde{S}^2}{T} + 1\right) - 2C,$$

and

$$\tilde{S} \leq \sqrt{2T}\left[\sqrt{\log\left(1 + \frac{u}{\sqrt{2}C}\right)} + 1\right].$$

Combining the above completes the proof. □

Comparing Corollary 13 to Theorem 14, the leading terms in the player-dependent bounds are exactly the same. The gap between the upper and lower bounds does not depend on time. That is, we have a good estimate of the worst case performance of Algorithm 5.

D Detail on experiments

We now present details on our experiments. First, we introduce the KT algorithm [OP16] as our baseline. It is perhaps the most well-known parameter-free algorithm for unconstrained OLO. Essentially, it is an optimistic version of Algorithm 5 induced by the existing potential $\bar{V}_{-1/2}$. Next, we discuss the choice of hyperparameters in our experiments. In the last two subsections, we present empirical results omitted from the main paper.

D.1 Baseline: Krichevsky-Trofimov algorithm

We first consider the one-dimensional version of the KT algorithm, whose pseudo-code is presented as Algorithm 7. Theoretically it guarantees a similar bound as Theorem 11, with only minor differences on the non-leading constants.

Lemma D.1 (Corollary 5 of [OP16]). *For all $T \in \mathbb{N}_+$ and $u \in \mathbb{R}$, against any adversary, Algorithm 7 guarantees*

$$\mathit{Regret}_T(u) \le \varepsilon + |u| \sqrt{T \log\left(1 + \frac{24\,|u|^2\, T^2}{\varepsilon^2}\right)}.$$

Algorithm 7 The Krichevsky-Trofimov algorithm.

Require: Initial wealth $\varepsilon > 0$.
1: **for** $t = 1, 2, \dots$ **do**
2: Predict $x_t = (-\sum_{i=1}^{t-1} g_i/t) \cdot (\varepsilon - \sum_{i=1}^{t-1} g_i x_i)$
3: Observe g_t and store it.
4: **end for**

The one-dimensional KT algorithm can be naturally extended to higher dimensions. Specifically, we wrap it using Algorithm 6 (the reduction from [CO18]), just like how Algorithm 4 extends Algorithm 5 to higher dimensions.

D.2 Choice of hyperparameters

We first discuss the choice of hyperparameter C in the two versions of Algorithm 5. Note that since both versions are parameter-free algorithms, the hyperparameter C does not affect their performance as critically as the learning rate in OGD: for any C, the regret upper bound has the same asymptotic order (but with different constants). Specifically we choose $C = 1$ in both versions. One reason is that this is the most natural choice when no information is available beforehand. More importantly, at the beginning of the optimization process, $C = 1$ induces the same asymptotic exponential growth rate for the predictions of the two versions. (As we discussed in Section 5, such an exponential growth is the key for the success of parameter-free algorithms.)

Concretely, the predictions of the both versions are roughly the gradients of the potentials, which are $\nabla_S \bar{V}_{-1/2}(t, S) = CSt^{-3/2} \exp[S^2/(2t)]$ for $\bar{V}_{-1/2}$ and $\nabla_S \bar{V}_{1/2}(t, S) = \sqrt{2}C \int_0^{S/\sqrt{2t}} \exp(x^2) dx$ for $\bar{V}_{1/2}$. At the beginning, all the gradient feedback are one-sided, therefore $|S| = t$. Applying $S = t$ and taking the derivative with respect to t, the growth rate of predictions based on $\bar{V}_{-1/2}$ is

$$\nabla_t \left[\nabla_S \bar{V}_{-1/2}(t, S)|_{S=t}\right] = \frac{C}{2\sqrt{t}} \left(1 - \frac{1}{t}\right) \exp\left(\frac{t}{2}\right).$$

For $\bar{V}_{1/2}$ we have

$$\nabla_t \left[\nabla_S \bar{V}_{1/2}(t, S)|_{S=t}\right] = \frac{C}{2\sqrt{t}} \exp\left(\frac{t}{2}\right).$$

The leading terms would match if the hyperparameters of the two versions are the same.

As for the initial wealth ε in KT, by comparing Theorem 11 and Lemma D.1 we can see that $\varepsilon = \sqrt{e}C$ is the most reasonable choice. It matches the maximum allowable $\mathrm{Regret}_T(0)$ in the KT algorithm and the version of Algorithm 5 based on $\bar{V}_{-1/2}$.

D.3 Omitted results on the 1d task

We first present more cases of u^* to support Figure 1a. Figure 3 shows that for $u^* \ge 1$, our algorithm consistently beats the baselines. Note that the vertical scale in each subfigure is different. Using a unified scale, Figure 1b in the main paper plots the gap between the green line and the blue line at $T = 500$. (The two baselines are similar, therefore the orange line is not considered in Figure 1b.)

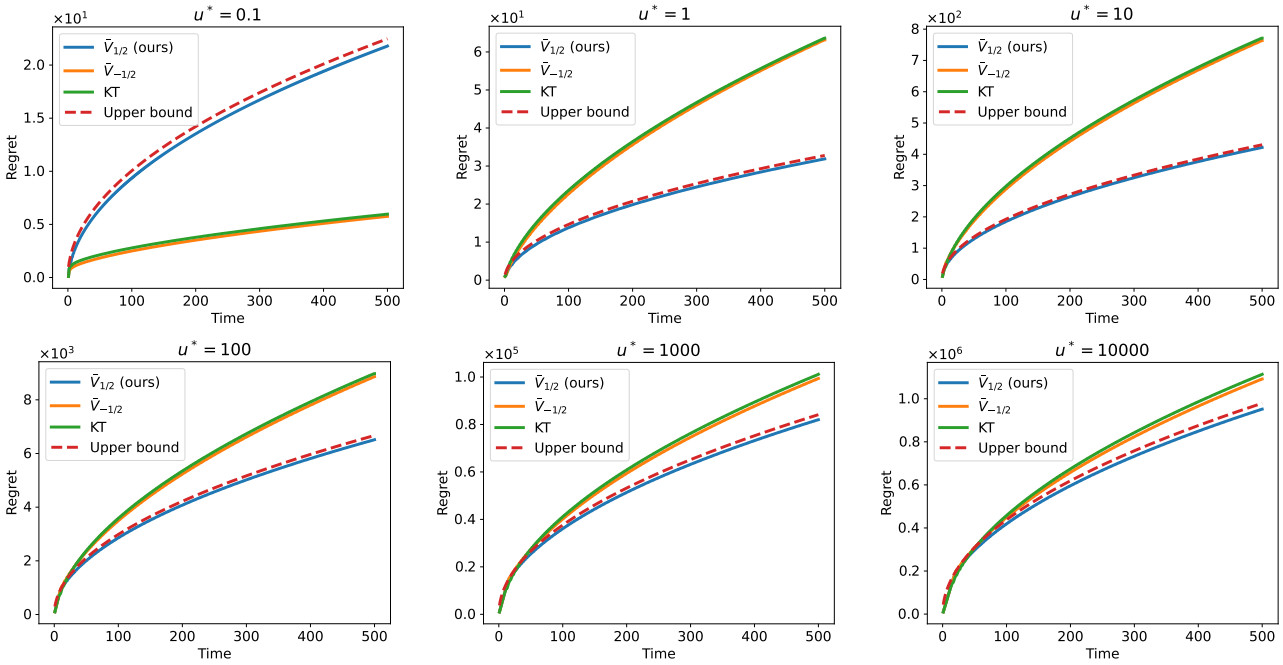


Figure 3: More cases of u^* to support Figure 1a; $C = 1$, $T = 500$.

Next, we investigate the effect of the maximum time horizon T. When closely comparing the regret upper bounds of the two potential-based algorithms (Theorem 4 and 11), one can see that for all fixed C and nonzero u^*, the upper bound based on the new potential $\bar{V}_{1/2}$ is always better if T is long enough ($O(\sqrt{T})$ as opposed to $O(\sqrt{T \log T})$). Then, a reasonable guess is that for some small u^*, the performance of our algorithm may be weaker than the baselines at $T = 500$ (Figure 3), but better than the baselines at larger T. Such a guess is true, as shown in Figure 4. Specifically, we pick $u^* = 0.3$ and vary the maximum T. Initially our algorithm is worse, but as T increases it can still outperform the baselines.

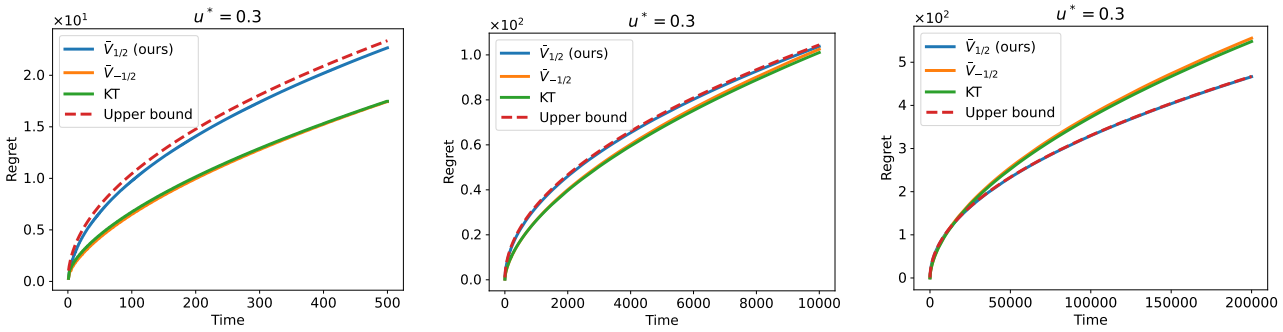


Figure 4: One-dimensional task with $C = 1$ and $u^* = 0.3$. From left to right: $T = 500, 10000, 200000$.

Finally, we investigate the effect of the hyperparameter C on the qualitative comparison of the three algorithms. We change C to 10 and present results parallel to Figure 3 in Figure 5. The initial wealth of KT is scaled accordingly to $10\sqrt{e}$.

Figure 5 exhibits a similar behavior as Figure 3: while sacrificing the regret at small $|u^*|$, our algorithm is better when u^* is far-away. However, we also see that our algorithm exhibits less qualitative improvement over the baselines: in order to beat our algorithm (at $T = 500$), previously (with $C = 1$) the baselines should initialize at $\tilde{u}$ with error $|\tilde{u} - u^*| \leq 1$, but now (with $C = 10$) such an error is allowed to be less than 10. A possible concern is that the advantage of our algorithm becomes harder to justify in this setting. We address

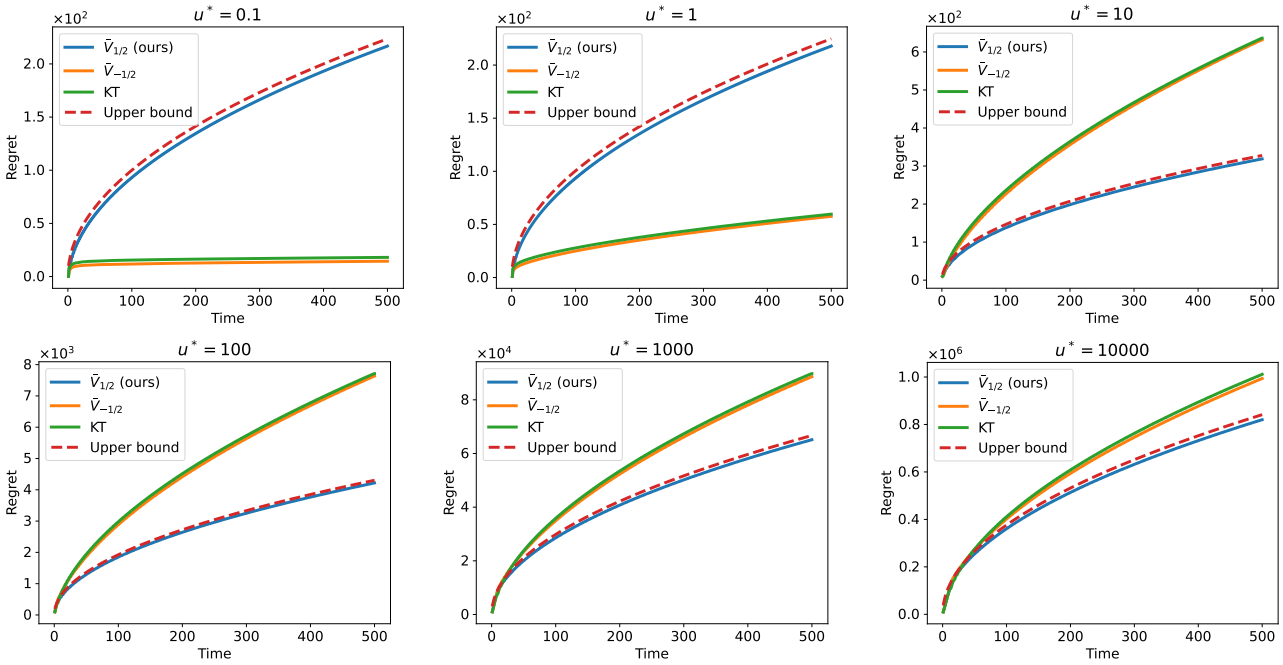


Figure 5: One-dimensional task with $C = 10$.

this concern from three different perspectives.

1. Even with $C = 10$, our algorithm still outperforms the baselines when u^* is *everywhere except on a compact set*. Therefore, our algorithm still works better in more situations (of u^*).

2. Theoretically, for all fixed C and nonzero u^*, our algorithm always guarantees better regret bound than the baselines when T is large enough. Empirically this is validated in Figure 4.

3. The key idea of parameter-free algorithms is to use a simple hyperparameter to replace the laborious tuning of learning rates. In practice (e.g., [OP16, CLO20]), such a hyperparameter is often simply set to 1, and the resulting algorithms already exhibit strong empirical performance. Actually, changing C amounts to a particular loss-regret trade-off; without any prior knowledge, the most *natural* choice is perhaps $C = 1$. Therefore, when our algorithm and the two baselines are in the *most natural configuration*, our algorithm has the best performance unless a very accurate guess of u^* is known a priori (Figure 3).

D.4 Omitted results on high-dimensional regression

We first report our data-preprocessing procedure.

1. Feature normalization. For all the features (columns of the data matrix), we perform min-max scaling to transform their range to $[0, 1]$, which is a common practice.

2. Row scaling. For all the feature vectors (rows of the data matrix), we scale them such that each row has L_2-norm 1. This is due to the Lipschitz requirement in our setting, and the same procedure has been performed in prior works (e.g., [OP16]).

Next, we fix γ and plot the cumulative OCO loss (averaged over 5 random seeds). Such a procedure is similar to Figure 3, and results are shown in Figure 6. Figure 2 in the main paper plots the difference between the green line and the blue line at $T = 50000$. We can draw the same conclusion as in the one-dimensional experiment: our algorithm outperforms the baselines when the optimal comparator is far-away.

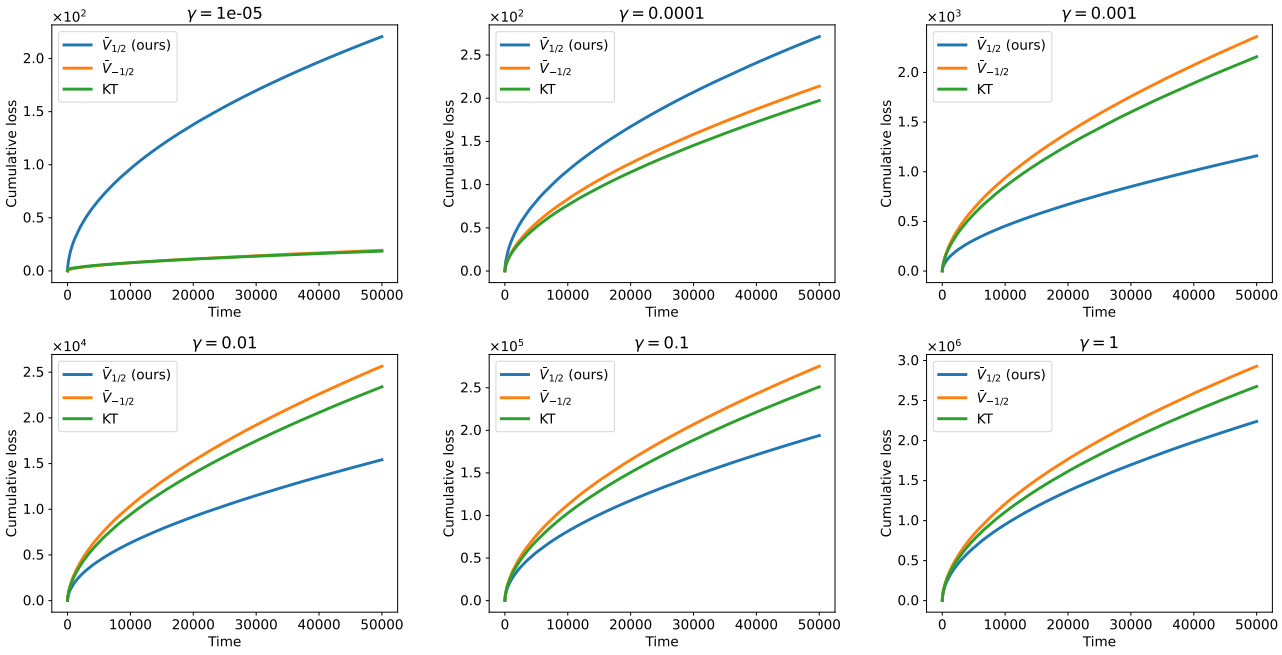


Figure 6: High-dimensional experiment with real data. $C = 1$, $T = 50000$.