Unconstrained Dynamic Regret via Sparse Coding

Zhiyu Zhang
Boston University
zhiyuz@bu.edu

Ashok Cutkosky
Boston University
ashok@cutkosky.com

Ioannis Ch. Paschalidis
Boston University
yannisp@bu.edu

Abstract

Motivated by time series forecasting, we study Online Linear Optimization (OLO) under the coupling of two problem structures: the domain is unbounded, and the performance of an algorithm is measured by its dynamic regret. Handling either of them requires the regret bound to depend on certain *complexity measure* of the comparator sequence – specifically, the *comparator norm* in unconstrained OLO, and the *path length* in dynamic regret. In contrast to a recent work [JC22] that adapts to the combination of these two complexity measures, we propose an alternative complexity measure by recasting the problem into sparse coding. Adaptivity can be achieved by a simple modular framework, which naturally exploits more intricate prior knowledge of the environment. Along the way, we also present a new gradient adaptive algorithm for static unconstrained OLO, designed using novel continuous time machinery. This could be of independent interest.

1 Introduction

Time series forecasting is a fundamental problem in science and engineering. To design forecasting strategies, a classical procedure is to model the time series based on batched data, either statistically or empirically, and then deploy such models online. The effectiveness of this procedure critically relies on certain stationarity of the environment, thus may fail under distribution shifts. The present work addresses this issue from the perspective of online learning – we design an online fine-tuning framework such that given *any* oracle forecaster, the fine-tuned predictions are equipped with robustness guarantees that do not rely on statistical assumptions at all.

Concretely, we study the following variant of *Online Convex Optimization* (OCO). In the t-th round,

1. We query an oracle forecaster $\mathcal{A}$ for its prediction $w_t \in \mathbb{R}^d$, determine a fine-tuning adjustment $x_t \in \mathbb{R}^d$, and then predict their sum $x_t + w_t$.

2. The environment reveals a true value $z_t \in \mathbb{R}^d$ and a convex G-Lipschitz[1] loss function $l_t : \mathbb{R}^d \to \mathbb{R}$, minimized at z_t. We suffer the loss $l_t(x_t + w_t)$.

Our goal is to achieve low regret against *any* alternative sequence of predictions $y_1, \ldots, y_T \in \mathbb{R}^d$ selected in hindsight, where T is the time horizon. The $y_{1:T}$ sequence does not have to be the true time series $z_{1:T}$, which will be clear shortly.

Since for any subgradient $g_t \in \partial l_t(x_t + w_t)$ we have $l_t(x_t + w_t) - l_t(y_t) \leq \langle g_t, x_t + w_t - y_t \rangle$, for the rest of the paper we will assume only observing g_t (instead of l_t and z_t), and define the regret in the formulation of *Online Linear Optimization* (OLO) [Haz16, Ora19]. Let $u_t = y_t - w_t$ be the "ideal" fine-tuning adjustment had we known the comparing sequence $y_{1:T}$ beforehand. Then,

$$\mathrm{Regret}_T(u_{1:T}) := \sum_{t=1}^{T} \langle g_t, x_t - u_t \rangle = \sum_{t=1}^{T} \langle g_t, x_t + w_t - y_t \rangle . \tag{1}$$

As an example, if $y_{1:T} = z_{1:T}$, then bounding the regret subsumes bounding the forecasting error $\sum_{t=1}^{T} l_t(x_t + w_t)$.

Through the lens of OLO, we call Eq.(1) the unconstrained dynamic regret. Compared to the most standard setting of OLO, the challenge here is due to the coupling of two problem structures.

[1] With respect to $\|\cdot\|_2$.

- The domain $\mathbb{R}^d$ is unbounded.
- The comparator sequence $u_{1:T}$ is time-varying.

Even under only one of these conditions, it appears that the environment is given too much power: no matter how we predict, there always exist some $g_{1:T}$ and $u_{1:T}$ sequences inducing large regret. Circumventing this issue relies on *comparator adaptivity* – instead of only depending on the time horizon T, the regret bound also depends on certain *complexity measures* of $u_{1:T}$. For the time series application, this allows us to write[2]

$$\sum_{t=1}^{T} l_t(x_t + w_t) \le \inf_{y_{1:T}} \left[\sum_{t=1}^{T} l_t(y_t) + \mathrm{Regret}_T(y_{1:T} - w_{1:T}) \right], \tag{2}$$

where the minimizing argument $y_{1:T}$ trades off its cumulative loss and the complexity of $u_{1:T} = y_{1:T} - w_{1:T}$. Choosing the complexity measure introduces inductive bias into the associated algorithm, which is ubiquitous in high dimensional statistics and machine learning. The rationale is that no algorithm works well universally for all problem instances, therefore a meaningful goal is to find the suitable inductive bias for the considered application, and design the corresponding optimal algorithm for that.

Specifically for our setting, prior works mostly studied the two problem structures separately, as reviewed in Section 1.2. For static regret ($u_t = u$) on unconstrained domains, the standard complexity measure is the *comparator norm* $\|u\|$ [MO14, OP16, CO18], whereas for dynamic regret on bounded domains, one typically considers the *path length* $\sum_{t=1}^{T-1} \|u_t - u_{t+1}\|$ [Zin03, ZLZ18]. A recent work [JC22] studied unconstrained dynamic regret by combining these two complexity measures, resulting in the regret bound (simplified)

$$\mathrm{Regret}_T(u_{1:T}) = \tilde{O}\left(G\sqrt{\left(\sum_{t=1}^{T} \|u_t\|_2\right)\left(\sum_{t=1}^{T-1} \|u_{t+1} - u_t\|_2\right)} \right).$$

However, by closely examining its inductive bias, such a bound may not be the most natural one for "non-converging" environments. In order to achieve low regret (hence low cumulative loss via Eq.(2)), it is implicitly assumed that the residual sequence $u_{1:T} = y_{1:T} - w_{1:T}$ is small and almost constant. The latter poses a somewhat stringent requirement on the oracle forecaster $\mathcal{A}$. For example, if $u_{1:T}$ is periodic, as often encountered in time series with seasonality, then the regret bound is in general linear in T. Moreover, the regret bound is achieved by a heavily customized mirror descent algorithm, which deviates from classical frameworks and relies on rather sophisticated algebra.

In this paper, we will take a conceptually different *sparse coding* approach. The obtained regret bound adapts to a new complexity measure of $u_{1:T}$, which naturally exploits more intricate prior knowledge of the environment.

1.1 Contribution

The contributions of this paper are twofold.

- Our first contribution is a simple framework that achieves a new type of unconstrained dynamic regret bounds (Section 2). In a broad sense, it is based on two ideas.
    1. We consider the sequence space $\mathbb{R}^{dT}$ that contains $x_{1:T}$, $g_{1:T}$ and $u_{1:T}$, rather than the default domain $\mathbb{R}^d$ that contains per-round quantities. This is a fundamental view for the batch analysis of sequential data, such as in signal processing [Mal08, VKG14] and time series modeling [BD16, SS17], but (in our opinion) under-explored in the online learning literature.[3] Static online learning can be considered as a special case.
    2. We use advances in static unconstrained OLO to aggregate dynamic base algorithms. In contrast to expert-based model selection approaches, this enables learning *linear combinations* of the base algorithms, rather than their *convex combinations*.

[2] Such bounds are called *oracle inequalities* in statistical learning.

[3] Possibly due to the emphasis on static regret by the community: the sequence $u_{1:T}$ collapses into a time-invariant u.

Combining the two ideas converts our problem into *Online Linear Regression* (OLR). If the comparator $u_{1:T}$ can be linearly represented by a certain collection of feature vectors (i.e., a dictionary) in $\mathbb{R}^{dT}$, then our regret bound adapts to (i) the *energy* of $u_{1:T}$; and (ii) the *sparsity* of its representation, without knowing either conditions beforehand. This brings two advantages.

1. Our approach is built upon close connections to signal processing, thus can benefit from prior works there. For example, a major research topic[4] in signal processing is finding the appropriate (typically redundant) dictionary for specific applications, such that the considered signal admits a sparse representation. We allow taking such a dictionary as prior knowledge and adapting to its quality.
2. Instead of requiring heavy customization like [JC22], many static unconstrained OLO algorithm, given the dictionary, can be used as a black box to solve OLR. Therefore, our approach automatically inherits a wide range of favorable properties from the static regret setting, such as Lipschitz constant adaptivity [Cut19a], scale-freeness [MK20] and generalized loss-regret tradeoffs [ZCP22a].

Overall, the proposed approach is not a replacement, but a complement to [JC22]. They represent different inductive bias, thus should be selected based on the specific application at hand. Nonetheless, simply adding them can always theoretically guarantee the best of both worlds.

- Our second contribution is a new static unconstrained OLO algorithm, which can be used as a subroutine of the sparse coding framework (Section 3).

  To explain what it does, let us consider again the time series application. Intuitively, given an oracle forecaster $\mathcal{A}$, we have to determine how much we trust it. This is essentially a tradeoff: if we want low regret on comparators $y_{1:T}$ that are close to $w_{1:T}$, we have to sacrifice the regret with respect to far-away comparators, and vice versa. In the setting of static regret, our prior work [ZCP22a] proposed a continuous-time-inspired algorithm with the optimal tradeoff,[5] but the bound is not simultaneously adaptive to the gradient variance. Such gradient adaptivity has been a hallmark of practical algorithms, as popularized by AdaGrad [DHS11].

  In this paper, we propose an algorithm that closes this gap. The key technique is a new discretization argument that quantifies the deviation of the discrete time algorithm from its ideal, continuous time counterpart. Plugging it into the sparse coding framework, we obtain a dynamic regret bound that adapts to not only the *sparsity of the comparator* (on the transform domain), but also the *sparsity of the observed gradients* (on the time domain).

1.2 Related work

Our paper addresses the connection between unconstrained online learning and dynamic regret. Although they both embody the idea of comparator adaptivity, unified studies have been scarce.

Unconstrained OLO To obtain static regret bounds in OLO, *Online Gradient Descent* (OGD) [Zin03] is often the default approach. With learning rate η, it guarantees $O(\eta^{-1}\left\|u\right\|_2^2+\eta T)$ regret with respect to any static comparator $u \in \mathbb{R}^d$. Without the prior knowledge of $\left\|u\right\|_2$, it is impossible to tune η optimally. To address this issue, a series of works (also called *parameter-free online learning*) [SM12, MO14, OP16, CO18, FRS18, MK20, ZCP22a] developed vastly different strategies to achieve the oracle optimal rate $O(\left\|u\right\|\sqrt{T})$ up to logarithmic factors. Most recent works are based on a dual space analysis and an elegant loss-regret duality [MO14], with the model selection approach from [FKMS17, CLW21, JC22] being a notable exception.

In these regret bounds, the complexity of u is measured by the *comparator norm* $\left\|u\right\|$, or more generally, $\left\|u-w\right\|$ given a prior w. L_1 and L_2 norm bounds were presented in [SM12], while general Banach norm bounds were developed by [FRS18, CO18]. Historically, the L_1 norm has renowned connections to sparsity, as suggested by LASSO [Tib96], compressed sensing [CRT06], and several works in online learning [KW95, SM12, Ger13, vdH19]. However, we are not aware of any prior use of such regret bounds in characterizing the structural simplicity of nonstationary environments.

[4] As the title of [Mal08] suggests. Often framed as *representation learning*.
[5] Defined as achieving $O(\sqrt{T})$ regret without the doubling trick, c.f., Section 3.1.

Our second contribution is dedicated to static unconstrained OLO itself, thus requires a more detailed review of existing results. This is deferred to Section 3.1 for cleaner exposition.

Dynamic regret Although the field of online learning primarily focused on the static regret, comparing against dynamic sequences has been studied by several lines of works. The closest topic to ours is the *universal dynamic regret*, where the regret bound adapts to the complexity of the comparator $u_{1:T}$ on a bounded domain with diameter D. Typically, the complexity measure is the path length $P_{T,p} = \sum_{t=1}^{T-1} \|u_t - u_{t+1}\|_p$ [HW01] or its generalization, e.g., norm squared [KMBAY15]. The optimal bound for OLO is $O(G\sqrt{DTP_{T,2}})$ [Zin03, HW15, JRSS15, ZLZ18]. With curved losses, the accelerated rate $\tilde{O}(T^{1/3}P_{T,1}^{2/3})$ is achievable [BW21, BW22].

As expected, one cannot go beyond linear dynamic regret in the worst case. The hope is that for "converging" environments where reasonable comparators have short path lengths, the overall regret bound can be sublinear in T. Except [JC22, LZZZ22], a shared limitation is the requirement of a bounded domain. A practical solution is to estimate the range of the problem offline, but since the diameter D is used to select the hyperparameter, wrong estimates will deteriorate the empirical performance of the algorithm.

Besides the universal dynamic regret, there are other notions of dynamic regret that do not induce oracle inequalities like Eq.(2), e.g., (i) the *restricted dynamic regret* [YZJY16, ZYY+17, BW19, BW20, BZW21], which depends on the complexity of certain offline optimal comparator; and (ii) regret bounds that depend on the functional variation $\sum_{t=1}^{T-1} \max_x |l_t(x) - l_{t+1}(x)|$ [BGZ15, CWW19]. They are both incompatible with OLO on unbounded domains.

Notably, we emphasize the difference between our work and a dynamic model approach from [HW15, ZLZ18]. On a bounded domain $\mathcal{X}$, their algorithms can take N dynamic models $\Phi_{t,n} : \mathcal{X} \to \mathcal{X}$, $n \in [1:N]$ as input. The regret bound has a similar form as path length bounds [Zin03], but replaces the path length with the error of the best dynamic model on the comparator, i.e., $\min_n \sum_{t=1}^{T-1} \|u_{t+1} - \Phi_{t,n}(u_t)\|$. Our dictionary also represents certain dynamic prior knowledge, but a key difference is that instead of using the best dictionary element to model the comparator, we use the best *linear combination* of the dictionary. This allows handling unconstrained domain through subspace modeling.

Online regression Our framework builds on online regression, which, in its nonparametric form, has been connected to the path length characterization of dynamic regret [RS14, GG15]. Prior works are mostly restricted to the square loss, and efficient computation can be a challenge [BW21].

For the special case of *Online Linear Regression* (OLR) with square loss, the celebrated VAW forecaster [AW01, Vov01] guarantees $O(N \log T)$ regret against any unbounded coefficient vector $\hat{u} \in \mathbb{R}^N$, where N is the dimension of the feature space. Such a fast rate becomes vacuous when $N > T$ [GY14], therefore [Ger13] proposed a *sparsity regret bound* $\tilde{O}(\|\hat{u}\|_0)$ and an accompanying inefficient algorithm as its high dimensional generalization. Efficient computation was addressed by [GW18], but the obtained result only applies to bounded $\hat{u}$. In some sense, such sparsity regret bounds are the square loss analogue of the L_1-norm parameter-free bounds in OLO. They are also closely related to *sparsity oracle inequalities* in statistics, as reviewed by [Ger13].

Parametric time series models Besides the dynamic regret approach to time series forecasting, significant research effort has been devoted to parametric strategies with stronger inductive bias, such as the ARMA model, state space models, and more recent deep learning models. Online learning has been applied to such models as well [AHMS13, AHZ15, AM16, KM16, HLS+18], leading to forecasting guarantees under mild statistical assumptions. Taking the *autoregressive* (AR) model for example, we will show that learning it can be converted to an instance of the sparse coding framework.

Other sparsity topics in OL Finally, we review other sparsity-related topics in online learning, which do not fit into the scope of this paper. [LLZ09, Xia09, DSSST10, SST11] considered using online learning to solve batch L_1 regularized problems. The goal is to achieve sparse predictions instead of sparsity adaptive regret bounds. [Kal14, FKK16, KKLP17] studied *online sparse regression*, where only a subset of features are available in each round. The challenge is to handle bandit feedback in OLR.

1.3 Notation

For two integers $a \le b$, $[a:b]$ is the set of all integers c such that $a \le c \le b$. Treating all vectors as column vectors, $\mathrm{span}(A)$ denotes the column space of a matrix A. For a function $\Phi : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, assuming differentiability, let $\partial_1\Phi$ and $\partial_2\Phi$ be its first order partial derivatives with respect to the two arguments. Similarly, $\partial_{11}\Phi$, $\partial_{12}\Phi$ and $\partial_{22}\Phi$ denote second order partial derivatives. f^* is the Fenchel conjugate of a function f. log represents natural logarithm when the base is omitted, and $\log_+(\cdot) := 0 \vee \log(\cdot)$. $\mathrm{KL}(\cdot||\cdot)$ is the KL divergence. $\Pi_X(x)$ is the Euclidean projection from x to a closed convex set X.

We define the *imaginary error function* as $\mathrm{erfi}(x) = \int_0^x \exp(u^2)du$. Note that it is scaled by $\sqrt{\pi}/2$ from the usual definition, thus can also be queried from standard software packages like SciPy. Let erfi^{-1} be its inverse function. Specialized notations for the sparse coding framework are detailed in Section 2.1.

2 Sparsity adaptive dynamic regret

In this section we present our sparse coding framework for unconstrained dynamic regret. The basic setting is described in Section 2.1. Focusing on the sequence space $\mathbb{R}^{dT}$, Section 2.2 presents our main result. Section 2.3 discusses our framework from a generalized primal-dual perspective.

2.1 Setting

We start by formally introducing our setup. For sequences $x_{1:T}$, $g_{1:T}$ and $u_{1:T}$, we will flatten everything and treat them as dT dimensional vectors, concatenating per-round quantities in $\mathbb{R}^d$. They are called *signals*.

Our framework requires online access to a *dictionary* matrix $\mathcal{H} \in \mathbb{R}^{dT \times N}$, whose columns are N nonzero *feature vectors*. We write $\mathcal{H}$ in an equivalent block form as $[h_{t,n}]_{1 \le t \le T, 1 \le n \le N}$, where each block $h_{t,n} \in \mathbb{R}^{d \times 1}$. The accompanied linear transform $u = \mathcal{H}\hat{u}$ relates a signal $u \in \mathbb{R}^{dT}$ to a coefficient vector $\hat{u} \in \mathbb{R}^N$. Adopting the convention in signal processing, we will call $\mathbb{R}^{dT}$ the *time domain*, and $\mathbb{R}^N$ the *transform domain*. In general, symbols without hat refer to time domain quantities, while their transform domain counterparts are denoted with hat.

With such notations, we consider the following interaction protocol, which could be termed *multivariate OLR with linear loss*. In the t-th round, our algorithm observes a d-by-N feature matrix $\mathcal{H}_t := [h_{t,n}]_{1 \le n \le N}$, makes a prediction $x_t \in \mathbb{R}^d$, receives a loss gradient $g_t \in \mathbb{R}^d$ satisfying $\|g_t\|_2 \le G$, and then suffers the loss $\langle g_t, x_t \rangle$. The performance metric is the dynamic regret defined in Eq.(1), where the comparator $u_{1:T}$ is unconstrained in $\mathbb{R}^{dT}$.

2.2 Main result

Overall, our strategy is to apply a static unconstrained OLO algorithm on the direction of each feature vector, and then aggregate their predictions. Concretely, let us start with a single feature vector.

Size 1 dictionary Consider an index $n \in [1:N]$, which is associated to the feature $h_{1:T,n} := [h_{1,n}, \ldots, h_{T,n}] \in \mathbb{R}^{dT}$. We suppress the index n and write it as $h_{1:T} = [h_1, \ldots, h_T]$. For any comparator $u_{1:T} \in \mathrm{span}(h_{1:T})$, there exists $\hat{u} \in \mathbb{R}$ such that $u_{1:T} = h_{1:T}\hat{u}$. The cumulative loss of $u_{1:T}$ can be rewritten as

$$\langle g_{1:T}, u_{1:T} \rangle = \langle g_{1:T}, h_{1:T} \rangle \hat{u} = \sum_{t=1}^{T} \langle g_t, h_t \rangle \hat{u},$$

which is the loss of the coefficient $\hat{u}$ on surrogate losses $\langle g_t, h_t \rangle$. To compete with $u_{1:T} \in \mathrm{span}(h_{1:T})$, it suffices to run a 1D static regret algorithm that competes with $\hat{u} \in \mathbb{R}$. Formally, we present this procedure as Algorithm 1.

By further assuming bounded $\|h_t\|_2$, Algorithm 1 could take any static unconstrained OLO algorithm as a black box. However, since the feature h_t is revealed before picking x_t, we can use a better black box that adapts to the scale of h_t, even if $\|h_t\|_2$ is unbounded. This is crucial for our purpose, as it allows the dynamic regret bound to adapt to the *energy* of the comparator, $E(u_{1:T}) := \|u_{1:T}\|_2^2$. As an example, we present such a black box as Algorithm 5 in Appendix A.2, which generalizes a recent result [ZCP22a] to the setting with time-varying but known Lipschitz constants.

Algorithm 1 Sparse coding with size 1 dictionary.

Require: An algorithm $\mathcal{A}$ for static 1D unconstrained OLO, and a nonzero feature vector $h_{1:T} \subset \mathbb{R}^{dT}$ revealed online.

for $t = 1, 2, \ldots, T$ **do**
  Receive $h_t \in \mathbb{R}^d$, and pass $G \|h_t\|_2$ to $\mathcal{A}$ as the Lipschitz constant of its next (t-th) loss.
  Query $\mathcal{A}$ for its t-th output, and assign it to $\hat{x}_t \in \mathbb{R}$.
  Predict $x_t = \hat{x}_t h_t \in \mathbb{R}^d$, and receive a loss gradient $g_t \in \mathbb{R}^d$.
  Compute $\hat{g}_t = \langle g_t, h_t \rangle$, and send it to $\mathcal{A}$ as its t-th surrogate loss gradient.
end for

Although not simultaneously adaptive to the magnitude of g_t, Algorithm 5 enjoys other appealing properties in the static setting, such as the optimal loss-regret tradeoff (reviewed in Section 3.1) and the optimal leading constant. Its analysis goes through a non-gradient-adaptive discretization argument (the *Discrete Itô formula* [HLPR20]), which sets the stage for our improved technique later. Plugging it into Algorithm 1 yields Lemma 2.1 below. Proofs for this subsection are deferred to Appendix A.3.

Lemma 2.1. *Let $\hat{\varepsilon} > 0$ be an arbitrary hyperparameter for Algorithm 5. Applying it as a subroutine, for all $T \in \mathbb{N}_+$ and $u_{1:T} \in \operatorname{span}(h_{1:T})$, Algorithm 1 guarantees*

$$\mathrm{Regret}_T(u_{1:T}) \leq G\varepsilon_T + \sqrt{2} \|u_{1:T}\|_2 G \left[\sqrt{\log\left(1 + \frac{\|u_{1:T}\|_2}{\sqrt{2}\varepsilon_T}\right)} + 1 \right],$$

where $\varepsilon_T = \hat{\varepsilon} \|h_{1:T}\|_2$. The subscript emphasizes that ε_T depends on T.

General dictionary With the single direction learner above, let us turn to the general setting with N features. We run N copies of Algorithm 1 in parallel, aggregate their predictions, and the regret bound sums Lemma 2.1, similar to [Cut19b] in the static setting. An extra twist is that each feature is associated with a different hyperparameter: it introduces a prior on the transform domain, which is essential for the overparameterized regime with $N \gg dT$. In summary, the pseudocode is presented as Algorithm 2, and the regret bound is Theorem 1.

Algorithm 2 Sparse coding with general dictionary.

Require: A dictionary $\mathcal{H} = [h_{t,n}]$, where $h_{t,n} \in \mathbb{R}^d$. Constants $\hat{\varepsilon}_1, \ldots, \hat{\varepsilon}_N > 0$.

For all $n \in [1 : N]$, initialize a copy of Algorithm 1 as $\mathcal{A}_n$. It runs Algorithm 5 as a subroutine, with hyperparameter $\hat{\varepsilon}_n$.
for $t = 1, 2, \ldots, T$ **do**
  Receive $\mathcal{H}_t = [h_{t,n}]_{1 \leq n \leq N}$. For all n, send $h_{t,n}$ to $\mathcal{A}_n$, and query its prediction $w_{t,n}$.
  Predict $x_t = \sum_{n=1}^N w_{t,n}$.
  Receive loss gradient g_t, and send it to $\mathcal{A}_1, \ldots, \mathcal{A}_N$ as loss gradients.
end for

Theorem 1. *For all $T \in \mathbb{N}_+$ and $u_{1:T} \in \mathbb{R}^{dT}$, Algorithm 2 guarantees*

$$\mathrm{Regret}_T(u_{1:T}) \leq 2G\mathcal{E}_T + \sqrt{2} G U_T \left[\sqrt{\log_+ \frac{U_T}{\sqrt{2}\mathcal{E}_T}} + \sqrt{\mathrm{KL}(q \| \pi)} + 2 \right] + G \sum_{t=1}^T \|u_{t,0}\|_2,$$

where

1. *For all $n \in [1 : N]$, $u_{1:T,n}$ is any vector in $\operatorname{span}(h_{1:T,n})$, and $u_{1:T,0} = u_{1:T} - \sum_{n=1}^N u_{1:T,n}$;*
2. *$\mathcal{E}_T = \sum_{n=1}^N \hat{\varepsilon}_n \|h_{1:T,n}\|_2$ and $U_T = \sum_{n=1}^N \|u_{1:T,n}\|_2$;*
3. *π and q are N dimensional probability vectors defined by $\pi_n = \hat{\varepsilon}_n \|h_{1:T,n}\|_2 / \mathcal{E}_T$, and $q_n = \|u_{1:T,n}\|_2 / U_T$.*

To interpret this result, we start from the simplest case. If the size $N = d$, the dictionary $\mathcal{H}_t = I_d$ (the d dimensional identity matrix), and the hyperparameters satisfy $\sum_{n=1}^N \hat{\varepsilon}_n = \varepsilon$, then against any *static* comparator ($u_t = u \in \mathbb{R}^d$), Theorem 1 guarantees

$$\mathrm{Regret}_T(u_{1:T}) \leq 2\varepsilon G\sqrt{T} + \sqrt{2}\left\|u\right\|_1 G\sqrt{T}\left[\sqrt{\log_+ \frac{\|u\|_1}{\sqrt{2}\varepsilon}} + \sqrt{\mathrm{KL}(q||\pi)} + 2\right], \tag{3}$$

where $\pi_n = \hat{\varepsilon}_n/\varepsilon$, and $q = u/\left\|u\right\|_1$. Intuitively, since ε can be tuned arbitrarily low, the first term on the RHS is typically negligible. Within the rest, the KL term gives the bound a Bayesian flavor:[6] we use a prior π to guess the posterior distribution q, i.e., how the "strength" of the comparator is spread across different feature vectors. Simply picking π as the uniform distribution results in $\mathrm{KL}(q||\pi) \leq \log N$, and the bound recovers the standard $\tilde{O}(\left\|u\right\|_1 \sqrt{T})$ bound in static unconstrained OLO [Ora19, Section 9.3].

Next, we enter the dynamic realm. Assume feature vectors are orthogonal, and the comparator $u_{1:T} \in \mathrm{span}(\mathcal{H})$. Within Theorem 1, we are free to set $u_{1:T,0} = 0$, and let $u_{1:T,n}$ be the projection of $u_{1:T}$ onto $\mathrm{span}(h_{1:T,n})$. Due to orthogonality, the projection preserves the energy of the comparator, i.e,

$$E(u_{1:T}) = \sum_{t=1}^T \left\|u_t\right\|_2^2 = \sum_{n=1}^N \left\|u_{1:T,n}\right\|_2^2.$$

By further defining $S_{\mathcal{H}}(u_{1:T}) := (\sum_{n=1}^N \left\|u_{1:T,n}\right\|_2)^2/\sum_{n=1}^N \left\|u_{1:T,n}\right\|_2^2$, we have

$$U_T = \sqrt{S_{\mathcal{H}}(u_{1:T})E(u_{1:T})}.$$

Note that $S_{\mathcal{H}}(u_{1:T})$ is a classical sparsity measure of $\{u_{1:T,n}\}_{1\leq n\leq N}$ [HR09]: if there are only $N_0 \leq N$ nonzero vectors within this collection, then $S_{\mathcal{H}}(u_{1:T}) \leq N_0$ due to the Cauchy-Schwarz inequality. Therefore, through the *complexity measure* U_T, Theorem 1 adapts to (*i*) the energy of $u_{1:T}$; and (*ii*) the sparsity of its representation, without knowing either condition beforehand. With low enough $\mathcal{E}_T$, the bound has the order $\tilde{O}(\sqrt{S_{\mathcal{H}}(u_{1:T})E(u_{1:T})}) \leq \tilde{O}(\sqrt{NT})$: the easier the comparator is (low energy, and sparse on $\mathcal{H}$), the lower the bound becomes.

So far we have only considered the underparameterized regime ($N \leq dT$) where feature vectors can be orthogonal. However, recent trends in signal processing have emphasized overparameterization ($N \gg dT$) as a key to obtain sparser representations. Theorem 1 can be nicely interpreted in this context as well: since it applies to *any* decomposition of $u_{1:T}$, as long as $u_{1:T}$ can be represented by a subset $\tilde{\mathcal{H}}$ of orthogonal features within $\mathcal{H}$, the regret bound adapts to $S_{\tilde{\mathcal{H}}}(u_{1:T})$, i.e., the sparsity of $u_{1:T}$ measured on $\tilde{\mathcal{H}}$. In other words, Theorem 1 adapts to the quality of the optimal (comparator-dependent) sub-dictionary $\tilde{\mathcal{H}}$. Note that:

- Algorithm 2 runs N base algorithms in parallel. For efficient computation with large N, the dictionary itself has to be sparse, which is called the *local property* in signal processing [Mal08]. See Appendix A.4 for a comparison between Fourier and wavelet dictionaries.

- Theorem 1 suffers a large-N penalty through the KL term. In practice, one may pick a good prior π, instead of the uniform distribution, to reduce this root-logarithmic overhead.

Power law phenomenon To further demonstrate the quantitative benefit, let us consider an empirically justified setup. In signal processing, the study of sparsity has been partially motivated by the *power law* [Pri21]: for many real world signals, even with a standard Fourier or wavelet dictionary, the n-th largest transform domain coefficient has magnitude roughly proportional to $n^{-\alpha}$, where $\alpha \in (0.5, 1)$. Suppose $d = 1$, and the comparator $u_{1:T}$ exhibits the power law through an orthogonal transformation of $\mathbb{R}^T$. Then, when T is large,

$$S_{\mathcal{H}}(u_{1:T}) = \frac{(\sum_{n=1}^T n^{-\alpha})^2}{\sum_{n=1}^T n^{-2\alpha}} \approx \frac{2\alpha - 1}{(1-\alpha)^2}(T)^{2-2\alpha} = O\left(T^{2-2\alpha}\right).$$

With $E(u_{1:T}) = O(T)$, we obtain a sublinear $\tilde{O}(T^{1.5-\alpha})$ dynamic regret bound.

[6]Analogous to comparator adaptive bounds in the expert problem [LS15, KVE15, CLW21, NBC$^+$21].

Example Finally, we note that the strength of our framework lies in the incorporation of domain knowledge through the dictionary $\mathcal{H}$. In Appendix A.4, we discuss several concrete examples, including classical Fourier and wavelet dictionaries, the autoregressive dictionary defined by time series, and dictionaries learned by online learning algorithms. As an added bonus, different unconstrained dynamic regret bounds, such as [JC22] and the different instances of Theorem 1, can be combined by simply summing their corresponding predictions (Appendix A.5).

2.3 Primal-dual interpretation

Concluding this section, we discuss our framework from a primal-dual perspective. In static OLO [Ora19], the primal space refers to the domain $\mathbb{R}^d$, while the dual space refers to the space of linear maps on $\mathbb{R}^d$, or intuitively, where we store a *sufficient statistic* of the observed information. The same algorithm can have different but equivalent analysis on the primal space and the dual space, e.g, the *Follow the Regularized Leader* (FTRL) versus the *potential method*. Our framework generalizes the static setting, thus can be understood in a similar way.

Specifically, we consider an analogous primal-dual relation between the time domain $\mathbb{R}^{dT}$ and the transform domain $\mathbb{R}^N$. From this angle, Algorithm 2 runs as follows, c.f., Figure 1.

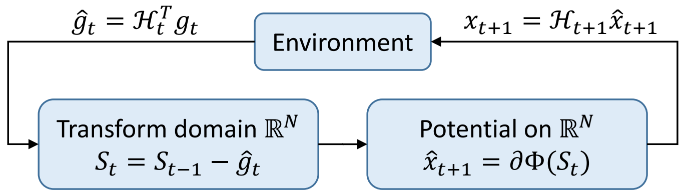

Figure 1: Algorithm 2 as potential method with general sufficient statistic.

- At the end of the t-th round, we multiply $g_t \in \mathbb{R}^d$ by the d-by-N feature matrix $\mathcal{H}_t$. The sufficient statistic in $\mathbb{R}^N$ is updated as $S_t = S_{t-1} - \mathcal{H}_t^T g_t$.
- By evaluating the gradient of a potential function Φ at S_t, we obtain a *transform domain prediction* $\hat{x}_{t+1} \in \mathbb{R}^N$ (analogous to Line 3 of Algorithm 1).
- After the next feature matrix $\mathcal{H}_{t+1}$ is revealed, we define the $t+1$-th prediction as $x_{t+1} = \mathcal{H}_{t+1}\hat{x}_{t+1} \in \mathbb{R}^d$.

Crucially, instead of storing the sum of loss gradients (as in static OLO), we store a N dimensional filtered version of the gradient sequence $g_{1:T}$. Roughly speaking, N captures the complexity of the comparator class, against which our algorithm guarantees sublinear regret.

As for the proof strategy, we have focused on aggregating regret bounds on the time domain. Alternatively, one could use a transform domain analysis to obtain the same result, generalizing the standard workflow in static unconstrained OLO [MO14]. The key is a *loss-regret duality for sequences*.[7]

Lemma 2.2. *If there exists a function $f_T : \mathbb{R}^{dT} \to \mathbb{R}$ such that the prediction sequence $x_{1:T}$ guarantees a loss upper bound $\langle g_{1:T}, x_{1:T}\rangle \leq -f_T(g_{1:T})$, then for all $u_{1:T} \in \mathbb{R}^{dT}$,*

$$\mathrm{Regret}_T(u_{1:T}) \leq f_T^*(-u_{1:T}).$$

In general, the function f_T can be fully nonlinear, so we consider a more structured function class where the nonlinearity acts on a linear sketch of the input, i.e., $f_T(x) = \Phi(\mathcal{H}^T x)$ for some nonlinear potential function $\Phi : \mathbb{R}^N \to \mathbb{R}$. By picking $x_t = \mathcal{H}_t\hat{x}_t$, constructing the loss upper bound is converted into a *coin-betting* problem with decision $\hat{x}_t \in \mathbb{R}^N$, where existing theoretical results are available [Cov66, OP16, ZCP22a].

[7] We present two other versions in Appendix A.6, which are more closely tied to path-length-based dynamic regret bounds.

3 A better static algorithm

As shown in Figure 1, the sparse coding framework consists of two components: (i) choosing a dictionary $\mathcal{H}$ that captures the dynamics of the environment; and (ii) designing a good potential function (or static unconstrained OLO algorithm) with low quantitative regret bound. We now present our second contribution, which addresses the latter. Section 3.1 surveys the background of this topic, while our new algorithm, including its implication for the sparse coding framework, is presented in Section 3.2. For static comparators $u_t = u \in \mathbb{R}^d$, we will write the regret as $\mathrm{Regret}_T(u)$.

3.1 Loss-regret tradeoff

An important topic in static unconstrained OLO is the *loss-regret tradeoff.* Due to a celebrated no free lunch theorem [Cov66], all such algorithms are required to trade off their cumulative loss $\mathrm{Regret}_T(0)$ with their leading regret term, i.e., $\mathrm{Regret}_T(u)$ for large $\|u\|$. Roughly speaking, such a tradeoff represents how much we trust the initialization of the algorithm. Most prior works [MO14, OP16] are natively designed with $O(1)$ loss and $O(\|u\| \sqrt{T \log(\|u\| T)})$ regret, while in principle, the optimal tradeoff corresponds to $O(\sqrt{T})$ loss and $O(\|u\| \sqrt{T \log(\|u\|)})$ regret, which matches the minimax optimal $O(\sqrt{T})$ rate on bounded domains (with respect to T alone). Although different loss-regret tradeoffs are mutually convertible through the doubling trick [SS11], doing so significantly downgrades the empirical performance of the algorithm,[8] thus should (ideally) be avoided in theory as well.

A recent work of ours [ZCP22a] achieved the optimal tradeoff in an "anytime" manner without doubling tricks. However, compared to other frontiers in this field (e.g., [MK20]), the regret bound does not simultaneously adapt to the observed gradient variance V_T. The importance of such gradient adaptivity has been demonstrated in practice [DHS11], but from a technical perspective, it is challenging to add this property to non-gradient-adaptive unconstrained algorithms, as both the algorithm and the analysis need to be modified with considerable sophistication. Existing techniques [CO18, MK20, JC22] are closely tied to the suboptimal loss-regret tradeoff,[9] and their extensions to our objective are unclear.

At the center of the optimal tradeoff [ZCP22a] is a nonstandard erfi potential function, which solves a *Partial Differential Equation* (PDE) that characterizes the *continuous time* (CT) limit of the learning game. In a broader context, the interplay between *discrete time* (DT) online learning and its CT limit has received growing attention [KS10, Zhu14, BEZ20, DK20, HLPR20, KKW20, PLH22, ZCP22b], as the latter is often easier to analyze and gain intuition from. However, a bottleneck here is the discretization of CT-derived algorithms – the standard technique is the Discrete Itô formula [HLPR20], which by construction is not gradient adaptive. Therefore, although gradient adaptivity has been studied in CT before, e.g., [Fre09] and [HLPR20, Appendix B.4], the obtained benefits have not been extended to the DT online learning problem we consider.

In this section, we improve [ZCP22a] by simultaneously achieving gradient adaptivity and the optimal loss-regret tradeoff, without doubling tricks. The key technique is a new discretization argument that further induces gradient adaptivity, which could be of separate interest.

3.2 Main result

Concretely, we consider OLO with domain $\mathbb{R}^d$ and a known Lipschitz constant G. Compared to the setting of Section 2.1, we now focus on the static regret $\mathrm{Regret}_T(u)$. Our approach requires two potential functions defined as follows, where the parameters satisfy $\varepsilon > 0$, $\alpha > 0$ and $z > k > 0$.

$$\phi(x, y) = \varepsilon\sqrt{\alpha x}\left(2\int_0^{\frac{y}{\sqrt{4\alpha x}}} \mathrm{erfi}(u)du - 1\right),$$

$$\Phi(V, S) = \phi(V + z + kS, S). \tag{4}$$

ϕ is the "basic" potential function from [ZCP22a], which solves the *Backward Heat Equation* (BHE) $\partial_1\phi + \alpha\partial_{22}\phi = 0$. Note that ϕ can be evaluated efficiently using Lemma A.1. Φ is the potential function we actually apply, which is constructed from ϕ with a change of variable.

[8] And incurs a multiplicative constant in the bound.

[9] With $O(\sqrt{V_T \log V_T})$ dependence on V_T alone, worse than the optimal $O(\sqrt{V_T})$ rate achieved by adaptive OGD on bounded domains.

Overall, our algorithm has a hierarchical structure. The key component is the 1D base algorithm (Algorithm 3), where for clarity, all the algorithmic quantities are denoted with tilde. We enforce the requirement $-\sum_{i=1}^{t}\tilde{l}_i \geq -1$ to make sure $\tilde{S}_t \geq -1$, thus the gradient computation in Line 3 is well-defined. Then, the meta-algorithm (Algorithm 4) applies two standard techniques [CO18, Cut20] on top of the base algorithm: the first reduces the domain of the base algorithm from $\mathbb{R}$ to $\mathbb{R}_+$, while the second extends it from $\mathbb{R}_+$ to $\mathbb{R}^d$.

Algorithm 3 1D base algorithm

Require: The potential function Φ defined in Eq.(4). Constants $\varepsilon > 0$, $\alpha > 0$ and $z > k > 0$. Surrogate loss gradients $\tilde{l}_{1:T}$ satisfying $\tilde{l}_t \in [-1, 1]$ and $-\sum_{i=1}^{t}\tilde{l}_i \geq -1$ for all t.

1: Initialize $\tilde{V}_0 = 0$, $\tilde{S}_0 = 0$.
2: **for** $t = 1, 2, \ldots, T$ **do**
3:     Predict $\tilde{z}_t = \partial_2\Phi(\tilde{V}_{t-1}, \tilde{S}_{t-1})$.
4:     Receive the surrogate loss gradient $\tilde{l}_t$.
5:     Let $\tilde{V}_t = \tilde{V}_{t-1} + \tilde{l}_t^2$, and $\tilde{S}_t = \tilde{S}_{t-1} - \tilde{l}_t$.
6: **end for**

Algorithm 4 Meta algorithm on $\mathbb{R}^d$.

Define $\mathcal{A}_{1d}$ as a copy of Algorithm 3. Define $\mathcal{A}_{\mathsf{B}}$ as OGD on the d-dimensional unit L_2 norm ball, with adaptive learning rate $\eta_t = \sqrt{2/\sum_{i=1}^{t}\|g_i\|_2^2}$. Initialization of $\mathcal{A}_{\mathsf{B}}$ is arbitrary.

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

    Query $\mathcal{A}_{1d}$ for its prediction $\tilde{z}_t \in \mathbb{R}$. Let $z_t = \Pi_{\mathbb{R}_+}(z_t)$.

    Query $\mathcal{A}_{\mathsf{B}}$ for its prediction $w_t \in \mathbb{R}^d$.

    Predict $x_t = z_t w_t$, receive the loss gradient $g_t \in \mathbb{R}^d$.

    Send $g_t G^{-1}$ as the surrogate loss to $\mathcal{A}_{\mathsf{B}}$.

    Define $l_t = \langle g_t G^{-1}, w_t \rangle$, and

$$\tilde{l}_t = \begin{cases} l_t, & l_t\tilde{z}_t \geq l_t z_t, \\ 0, & \text{else.} \end{cases}$$

    Send $\tilde{l}_t$ as the surrogate loss to $\mathcal{A}_{1d}$.

end for

Before analyzing its performance, Proposition 3 in Appendix B.3 shows that the surrogate loss $\tilde{l}_t$ defined in the meta-algorithm indeed satisfies $-\sum_{i=1}^{t}\tilde{l}_i \geq -1$, therefore the entire hierarchical procedure is well-posed. Then, with the *gradient variance* defined as $V_T = \sum_{t=1}^{T}\|g_t\|_2^2$, we present the regret bound as Theorem 2.

Theorem 2. *With $\varepsilon > 0$, $\alpha = 1$, $k = 2$ and $z = 16$, Algorithm 4 guarantees for all $T \in \mathbb{N}_+$ and $u \in \mathbb{R}^d$,*

$$\mathrm{Regret}_T(u) \leq \varepsilon\sqrt{V_T + 2G\bar{S}} + \|u\|_2\left(\bar{S} + 2\sqrt{2V_T}\right),$$

where

$$\bar{S} = 8G\left(1 + \sqrt{\log(2\|u\|_2\varepsilon^{-1} + 1)}\right)^2 + 2\sqrt{V_T + 16G^2}\left(1 + \sqrt{\log(2\|u\|_2\varepsilon^{-1} + 1)}\right).$$

Let us make this bound a bit more interpretable. Using asymptotic orders, we can simplify it into (see Appendix B.3 for the derivation)

$$\mathrm{Regret}_T(u) \leq \varepsilon\left(\sqrt{V_T} + 6G\right) + \|u\|_2 O\left(\sqrt{V_T\log(\|u\|_2\varepsilon^{-1})} \vee G\log(\|u\|_2\varepsilon^{-1})\right),$$

which is simultaneously valid in two regimes: (i) $\|u\|_2 \gg \varepsilon$ and $V_T \gg G^2$; and (ii) $u = 0$, i.e., $\mathrm{Regret}_T(0) \leq \varepsilon\left(\sqrt{V_T} + 6G\right)$. Note that the logarithmic residual term (outside the root) is standard in gradient adaptive unconstrained OLO. Therefore, with a $O(\sqrt{V_T})$ maximum loss bound, Algorithm 4 guarantees a $O(\|u\|_2\sqrt{V_T}\log\|u\|_2)$

regret bound[10] – this matches the minimax optimal rate $O(\sqrt{V_T})$ on bounded domains, achieved by adaptive OGD [DHS11]. Compared to prior works, we improve [ZCP22a] by achieving second order gradient adaptivity, and [CO18, MK20, JC22] by a better asymptotic rate on V_T.

Sketch of the analysis We now sketch our analysis of the base algorithm (Algorithm 3), including the key idea of discretization. At one point we consider two-case cases. Alternate expressions for the second case are provided in red. Overall, the analysis has a similar procedure as typical potential methods: we first upper-bound the cumulative loss $\sum_{t=1}^T \tilde{l}_t \tilde{z}_t$, and then obtain the regret bound through a *loss-regret duality* [MO14]. The loss upper bound follows from a telescopic sum on the *one-step bound*:

$$\tilde{l}_t \tilde{z}_t = \tilde{l}_t \partial_2 \Phi(\tilde{V}_{t-1}, \tilde{S}_{t-1}) \leq \Phi(\tilde{V}_{t-1}, \tilde{S}_{t-1}) - \Phi(\tilde{V}_t, \tilde{S}_t).$$

Proving it is the main technical challenge of our analysis (Lemma B.3), as in most prior works.

To this end, we aim to show that for all $V \geq 0$, $S \geq -1$ and $c \in [-1, 1]$ satisfying $S + c \geq -1$,

$$f_{V,S}(c) := \Phi(V + c^2, S + c) - \Phi(V, S) - c\partial_2 \Phi(V, S) \leq 0.$$

It is clear that $f_{V,S}(0) = f'_{V,S}(0) = 0$. Therefore, to prove $f_{V,S}(c) \leq 0$, a sufficient condition is the concavity of $f_{V,S}$ on the considered input domain. By calculating the Hessian and using $|c| \leq 1$,

$$f''_{V,S}(c) \leq 2\partial_1 \Phi(V + c^2, S + c) + 4\partial_{11}\Phi(V + c^2, S + c) + 4\left|\partial_{12}\Phi(V + c^2, S + c)\right| + \partial_{22}\Phi(V + c^2, S + c).$$

Furthermore, due to Eq.(4), the derivatives of Φ are concisely related to the derivatives of ϕ: if $\partial_{12}\Phi(V + c^2, S + c) \leq_{(\geq)} 0$, then

$$f''_{V,S}(c) \leq 2\partial_1 \phi + \underbrace{[k -_{(+)} 2]^2 \partial_{11}\phi + 2[k -_{(+)} 2]\partial_{12}\phi}_{:=\Delta} + \partial_{22}\phi, \tag{5}$$

where the derivatives on the RHS are evaluated at the input pair $(V + c^2 + z + k(S + c), S + c)$.

Now, the key observation is that the RHS of Eq.(5) has a striking similarity to the Backward Heat Equation $\partial_1\phi + \alpha\partial_{22}\phi = 0$, which the basic potential function ϕ satisfies. This motivates us to view Δ as the *discretization error*. Ideally, if $\Delta \leq 0$, then $f''_{V,S}(c) \leq 0$ by simply picking $\alpha = 1/2$. The reality is only slightly more complicated:

- We pick $k = 2$ to eliminate the harder case within the two.
- As for the other case, it only occurs when $S + c$ is at most constant-away from 0. Picking a large enough constant offset z, we have $\Delta \leq \partial_{22}\phi$, therefore $f''_{V,S}(c) \leq 0$ follows from $\alpha = 1$.

In summary, through a change of variable, we show how to utilize the CT property (i.e., the BHE) of potential functions in the verification of DT algorithms. The discretization error is more finely characterized compared to the Discrete Itô formula (surveyed in Appendix A.2), which results in additional gradient adaptivity. Moreover, since the BHE succinctly captures a family of adaptive potentials [ZCP22a], the argument above could be applicable to other loss-regret tradeoffs as well. Such generality and simplicity may provide benefits over existing techniques without CT connections [CO18, MK20, JC22].

Application to dynamic regret Finally, we apply this static algorithm to bound the dynamic regret, through our sparse coding framework. Slightly different from Section 2, we impose an additional assumption, $\|h_{t,n}\|_2 \leq 1$. As shown in Lemma B.8, with asymptotic simplification, the dynamic bound against any $u_{1:T} \in \mathrm{span}(\mathcal{H})$ becomes

$$\mathrm{Regret}_T(u_{1:T}) \leq \sum_{n=1}^N \hat{\varepsilon}_n \left(\sqrt{\sum_{t=1}^T \langle g_t, h_{t,n}\rangle^2} + 6G\right) + \tilde{O}\left(\sum_{n=1}^N \sqrt{\sum_{t=1}^T \langle g_t, u_{t,n}\rangle^2}\right),$$

where the first sum is a cumulative loss term tuned to be small, and $\{u_{t,n}\}$ is an arbitrary decomposition of the comparator satisfying $\sum_{n=1}^N u_{t,n} = u_{1:T}$. Compared to Theorem 1 which guarantees a similar form

$$\mathrm{Regret}_T(u_{1:T}) \leq 2G\sum_{n=1}^N \hat{\varepsilon}_n \sqrt{\sum_{t=1}^T \|h_{t,n}\|_2^2} + \tilde{O}\left(G\sum_{n=1}^N \sqrt{\sum_{t=1}^T \|u_{t,n}\|_2^2}\right),$$

[10] Loosely assimilating the residual term for clarity.

our improved approach further achieves gradient adaptivity. In other words, the obtained algorithm adapts to not only the sparsity of the comparator (on the transform domain), but also the sparsity of the observed gradients (on the time domain).

4 Conclusion

In this paper, we presented two complementary results for unconstrained OLO.

- Through a sparse coding framework, one can convert static unconstrained OLO algorithms to the dynamic setting, and the regret bound adapts to both the energy and the sparsity of the comparator sequence. This is closely connected to representation learning, thus may lead to deeper integration of the two research areas.
- We propose an algorithm that simultaneously achieves the gradient variance adaptivity and the optimal loss-regret tradeoff. The key technique is a new discretization argument, which could facilitate the continuous time analysis of online learning in general.

References

[AHMS13] Oren Anava, Elad Hazan, Shie Mannor, and Ohad Shamir. Online learning for time series prediction. In *Conference on learning theory*, pages 172–184. PMLR, 2013.

[AHZ15] Oren Anava, Elad Hazan, and Assaf Zeevi. Online time series prediction with missing data. In *International conference on machine learning*, pages 2191–2199. PMLR, 2015.

[AM16] Oren Anava and Shie Mannor. Heteroscedastic sequences: beyond gaussianity. In *International Conference on Machine Learning*, pages 755–763. PMLR, 2016.

[AW01] Katy S Azoury and Manfred K Warmuth. Relative loss bounds for on-line density estimation with the exponential family of distributions. *Machine Learning*, 43(3):211–246, 2001.

[BD16] Peter J Brockwell and Richard A Davis. *Introduction to Time Series and Forecasting*. Springer, 2016.

[BEZ20] Erhan Bayraktar, Ibrahim Ekren, and Xin Zhang. Finite-time 4-expert prediction problem. *Communications in Partial Differential Equations*, 45(7):714–757, 2020.

[BGZ15] Omar Besbes, Yonatan Gur, and Assaf Zeevi. Non-stationary stochastic optimization. *Operations research*, 63(5):1227–1244, 2015.

[BW19] Dheeraj Baby and Yu-Xiang Wang. Online forecasting of total-variation-bounded sequences. *Advances in Neural Information Processing Systems*, 32, 2019.

[BW20] Dheeraj Baby and Yu-Xiang Wang. Adaptive online estimation of piecewise polynomial trends. *Advances in Neural Information Processing Systems*, 33:20462–20472, 2020.

[BW21] Dheeraj Baby and Yu-Xiang Wang. Optimal dynamic regret in exp-concave online learning. In *Conference on Learning Theory*, pages 359–409. PMLR, 2021.

[BW22] Dheeraj Baby and Yu-Xiang Wang. Optimal dynamic regret in proper online learning with strongly convex losses and beyond. In *International Conference on Artificial Intelligence and Statistics*, pages 1805–1845. PMLR, 2022.

[BZW21] Dheeraj Baby, Xuandong Zhao, and Yu-Xiang Wang. An optimal reduction of tv-denoising to adaptive online learning. In *International Conference on Artificial Intelligence and Statistics*, pages 2899–2907. PMLR, 2021.

[CLW21] Liyu Chen, Haipeng Luo, and Chen-Yu Wei. Impossible tuning made possible: A new expert algorithm and its applications. In *Conference on Learning Theory*, pages 1216–1259. PMLR, 2021.

[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 University, 1966.

[CRT06] Emmanuel J Candès, Justin Romberg, and Terence Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on information theory*, 52(2):489–509, 2006.

[Cut19a] Ashok Cutkosky. Artificial constraints and hints for unbounded online learning. In *Conference on Learning Theory*, pages 874–894. PMLR, 2019.

[Cut19b] 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.

[CWW19] Xi Chen, Yining Wang, and Yu-Xiang Wang. Nonstationary stochastic optimization under l p, q-variation measures. *Operations Research*, 67(6):1752–1765, 2019.

[DHS11] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

[DK20] Nadejda Drenska and Robert V Kohn. Prediction with expert advice: A PDE perspective. *Journal of Nonlinear Science*, 30(1):137–173, 2020.

[DSSST10] John C Duchi, Shai Shalev-Shwartz, Yoram Singer, and Ambuj Tewari. Composite objective mirror descent. In *COLT*, volume 10, pages 14–26. Citeseer, 2010.

[FKK16] Dean Foster, Satyen Kale, and Howard Karloff. Online sparse linear regression. In *Conference on Learning Theory*, pages 960–970. PMLR, 2016.

[FKMS17] Dylan J Foster, Satyen Kale, Mehryar Mohri, and Karthik Sridharan. Parameter-free online learning via model selection. *Advances in Neural Information Processing Systems*, 30, 2017.

[Fre09] Yoav Freund. A method for hedging in continuous time. *arXiv preprint arXiv:0904.3356*, 2009.

[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.

[Ger13] Sébastien Gerchinovitz. Sparsity regret bounds for individual sequences in online linear regression. *The Journal of Machine Learning Research*, 14(1):729–769, 2013.

[GG15] Pierre Gaillard and Sébastien Gerchinovitz. A chaining algorithm for online nonparametric regression. In *Conference on Learning Theory*, pages 764–796. PMLR, 2015.

[GW18] Pierre Gaillard and Olivier Wintenberger. Efficient online algorithms for fast-rate regret bounds under sparsity. *Advances in Neural Information Processing Systems*, 31, 2018.

[GY14] Sébastien Gerchinovitz and Jia Yuan Yu. Adaptive and optimal online linear regression on l1-balls. *Theoretical Computer Science*, 519:4–28, 2014.

[Haz16] Elad Hazan. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

[HLPR20] Nicholas JA Harvey, Christopher Liaw, Edwin Perkins, and Sikander Randhawa. Optimal anytime regret with two experts. *arXiv preprint arXiv:2002.08994v2*, 2020.

[HLS+18] Elad Hazan, Holden Lee, Karan Singh, Cyril Zhang, and Yi Zhang. Spectral filtering for general linear dynamical systems. *Advances in Neural Information Processing Systems*, 31, 2018.

[HR09] Niall Hurley and Scott Rickard. Comparing measures of sparsity. *IEEE Transactions on Information Theory*, 55(10):4723–4741, 2009.

[HW01] Mark Herbster and Manfred K Warmuth. Tracking the best linear predictor. *The Journal of Machine Learning Research*, 1:281–309, 2001.

[HW15] Eric C Hall and Rebecca M Willett. Online convex optimization in dynamic environments. *IEEE Journal of Selected Topics in Signal Processing*, 9(4):647–662, 2015.

[JC22] Andrew Jacobsen and Ashok Cutkosky. Parameter-free mirror descent. In *Conference on Learning Theory*, pages 4160–4211. PMLR, 2022.

[JRSS15] Ali Jadbabaie, Alexander Rakhlin, Shahin Shahrampour, and Karthik Sridharan. Online optimization: Competing with dynamic comparators. In *Artificial Intelligence and Statistics*, pages 398–406. PMLR, 2015.

[Kal14] Satyen Kale. Open problem: Efficient online sparse regression. In *Conference on Learning Theory*, pages 1299–1301. PMLR, 2014.

[KKLP17] Satyen Kale, Zohar Karnin, Tengyuan Liang, and Dávid Pál. Adaptive feature selection: Computationally efficient online sparse linear regression under rip. In *International Conference on Machine Learning*, pages 1780–1788. PMLR, 2017.

[KKW20] 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.

[KM16] Vitaly Kuznetsov and Mehryar Mohri. Time series prediction and online learning. In *Conference on Learning Theory*, pages 1190–1213. PMLR, 2016.

[KMBAY15] Wouter M Koolen, Alan Malek, Peter L Bartlett, and Yasin Abbasi Yadkori. Minimax time series prediction. *Advances in Neural Information Processing Systems*, 28, 2015.

[KS10] Robert V Kohn and Sylvia Serfaty. A deterministic-control-based approach to fully nonlinear parabolic and elliptic equations. *Communications on pure and applied mathematics*, 63(10):1298–1350, 2010.

[KVE15] Wouter M Koolen and Tim Van Erven. Second-order quantile methods for experts and combinatorial games. In *Conference on Learning Theory*, pages 1155–1175. PMLR, 2015.

[KW95] Jyrki Kivinen and Manfred K Warmuth. Additive versus exponentiated gradient updates for linear prediction. In *Proceedings of the twenty-seventh annual ACM symposium on Theory of computing*, pages 209–218, 1995.

[LLZ09] John Langford, Lihong Li, and Tong Zhang. Sparse online learning via truncated gradient. *Journal of Machine Learning Research*, 10(3), 2009.

[LS15] Haipeng Luo and Robert E Schapire. Achieving all with no parameters: Adanormalhedge. In *Conference on Learning Theory*, pages 1286–1304. PMLR, 2015.

[LZZZ22] Haipeng Luo, Mengxiao Zhang, Peng Zhao, and Zhi-Hua Zhou. Corralling a larger band of bandits: A case study on switching regret for linear bandits. In *Conference on Learning Theory*, pages 3635–3684. PMLR, 2022.

[Mal08] Stephane Mallat. *A Wavelet Tour of Signal Processing: The Sparse Way*. Academic Press, 2008.

[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.

[NBC+21] Jeffrey Negrea, Blair Bilodeau, Nicolò Campolongo, Francesco Orabona, and Dan Roy. Minimax optimal quantile and semi-adversarial regret via root-logarithmic regularizers. *Advances in Neural Information Processing Systems*, 34, 2021.

[OP16] Francesco Orabona and Dávid Pál. Coin betting and parameter-free online learning. *Advances in Neural Information Processing Systems*, 29, 2016.

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

[PLH22] Victor Sanches Portella, Christopher Liaw, and Nicholas JA Harvey. Continuous prediction with experts' advice. *arXiv preprint arXiv:2206.00236*, 2022.

[Pri21] Eric Price. Sparse recovery. In Tim Roughgarden, editor, *Beyond the Worst-Case Analysis of Algorithms*, page 140–164. Cambridge University Press, 2021.

[RS14] Alexander Rakhlin and Karthik Sridharan. Online non-parametric regression. In *Conference on Learning Theory*, pages 1232–1264. PMLR, 2014.

[SM12] Matthew Streeter and Brendan Mcmahan. No-regret algorithms for unconstrained online convex optimization. *Advances in Neural Information Processing Systems*, 25, 2012.

[SS11] Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and trends in Machine Learning*, 4(2):107–194, 2011.

[SS17] Robert H Shumway and David S Stoffer. *Time Series Analysis and Its Applications: With R Examples*. Springer, 2017.

[SST11] Shai Shalev-Shwartz and Ambuj Tewari. Stochastic methods for l1-regularized loss minimization. *Journal of Machine Learning Research*, 12:1865–1892, 2011.

[Tib96] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[vdH19] Dirk van der Hoeven. User-specified local differential privacy in unconstrained adaptive online learning. *Advances in Neural Information Processing Systems*, 32, 2019.

[VKG14] Martin Vetterli, Jelena Kovačević, and Vivek K Goyal. *Foundations of signal processing*. Cambridge University Press, 2014.

[Vov01] Volodya Vovk. Competitive on-line statistics. *International Statistical Review*, 69(2):213–248, 2001.

[Xia09] Lin Xiao. Dual averaging method for regularized stochastic learning and online optimization. *Advances in Neural Information Processing Systems*, 22, 2009.

[YZJY16] Tianbao Yang, Lijun Zhang, Rong Jin, and Jinfeng Yi. Tracking slowly moving clairvoyant: Optimal dynamic regret of online learning with true and noisy gradient. In *International Conference on Machine Learning*, pages 449–457. PMLR, 2016.

[ZCP22a] Zhiyu Zhang, Ashok Cutkosky, and Ioannis Paschalidis. PDE-based optimal strategy for unconstrained online learning. In *International Conference on Machine Learning*, pages 26085–26115. PMLR, 2022.

[ZCP22b] Zhiyu Zhang, Ashok Cutkosky, and Ioannis Ch Paschalidis. Optimal parameter-free online learning with switching cost. *arXiv preprint arXiv:2205.06846*, 2022.

[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*, pages 928–936, 2003.

[ZLZ18] Lijun Zhang, Shiyin Lu, and Zhi-Hua Zhou. Adaptive online learning in dynamic environments. *Advances in neural information processing systems*, 31, 2018.

[ZYY+17] Lijun Zhang, Tianbao Yang, Jinfeng Yi, Rong Jin, and Zhi-Hua Zhou. Improved dynamic regret for non-degenerate functions. *Advances in Neural Information Processing Systems*, 30, 2017.

Appendix

Organization Appendix A presents details on our sparse coding framework (Section 2). Appendix B presents details on the improved static unconstrained OLO algorithm (Section 3).

A Detail on sparse coding

The sparse coding framework requires running a static OLO algorithm as a subroutine. We present some basic facts on the erfi function in Appendix A.1, and a non-gradient-adaptive static OLO subroutine in Appendix A.2. Appendix A.3 contains the proof of our sparsity adaptive regret bound. Appendix A.4 discusses several concrete choices of the dictionary. Appendix A.5 shows how to combine algorithms with different unconstrained dynamic regret guarantees. Appendix A.6 complements the primal-dual interpretation of our framework from Section 2.3.

A.1 Fact on the erfi function

First of all, we will use the following facts on the erfi function. Note that in this paper, we scale it from its usual definition, c.f., Section 1.3.

Lemma A.1. *For all $x \in \mathbb{R}$,*

$$\int_0^x \mathrm{erfi}(u)du = x\mathrm{erfi}(x) - \frac{1}{2}\exp(x^2) + \frac{1}{2}.$$

The proof follows from a simple integration by parts, therefore omitted.

Lemma A.2. *For all $x \geq 1$, $\mathrm{erfi}(x) \geq \exp(x^2)/2x$.*

Proof of Lemma A.2. Let $f(x) = \mathrm{erfi}(x) - \exp(x^2)/2x$. $f(1) = \mathrm{erfi}(1) - e/2 > 0$. For all $x \geq 1$,

$$f'(x) = \frac{1}{2x^2}\exp(x^2) > 0.$$ □

Lemma A.3 (From Theorem 4 of [ZCP22a])**.** *For all $x \geq 0$, $\mathrm{erfi}^{-1}(x) \leq 1 + \sqrt{\log(x+1)}$.*

A.2 Unconstrained OL with varying Lipschitzness

We present a non-gradient-adaptive, static 1D unconstrained OLO algorithm as Algorithm 5. It is designed to exploit time-varying, but known Lipschitz constants on the loss functions. The regret bound is Lemma A.4.

Algorithm 5 1D Static unconstrained OLO with time-varying Lipschitzness.

Require: A hyperparameter $\hat{\varepsilon} > 0$. A sequence of Lipschitz constants $G_{1:T}$ such that each loss gradient $\hat{g}_t \in \mathbb{R}$ satisfies $|\hat{g}_t| \leq G_t$.

Initialize $V_0 = S_0 = 0$. Define a potential function as

$$\Phi(V,S) = \hat{\varepsilon}\sqrt{V}\left(2\int_0^{\frac{S}{\sqrt{2V}}} \mathrm{erfi}(x)dx - 1\right). \tag{6}$$

Note that $\int \mathrm{erfi}(x)dx$ can be evaluated using Lemma A.1.

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

  Receive the t-th Lipschitz constant G_t, and let $V_t = V_{t-1} + G_t^2$.

  If $G_t = 0$, predict $\hat{x}_t = 0$. Otherwise, predict

$$\hat{x}_t = \frac{1}{2G_t}\left[\Phi\left(V_t, S_{t-1} + G_t\right) - \Phi\left(V_t, S_{t-1} - G_t\right)\right].$$

  Observe the loss gradient $\hat{g}_t \in \mathbb{R}$, and let $S_t = S_{t-1} - \hat{g}_t$.

end for

Lemma A.4. *For all $T \in \mathbb{N}_+$ and $\hat{u} \in \mathbb{R}$, Algorithm 5 guarantees*

$$\sum_{t=1}^{T} \langle \hat{g}_t, \hat{x}_t - \hat{u} \rangle \le \hat{\varepsilon} \sqrt{\sum_{t=1}^{T} G_t^2} + \sqrt{2}\,|\hat{u}| \sqrt{\sum_{t=1}^{T} G_t^2} \left[\sqrt{\log\left(1 + \frac{|\hat{u}|}{\sqrt{2}\hat{\varepsilon}}\right)} + 1 \right].$$

The proof of Lemma A.4 generalizes the argument of [HLPR20, ZCP22a] by allowing arbitrary time-varying gap parameters. It also demonstrates the existing technique (the Discrete Itô formula) for discretizing continuous-time-derived algorithms. This is in contrast to our improved technique in Section 3.

First, consider a function $\Phi : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$. In light of standard partial derivatives $\partial_1 \Phi$, $\partial_2 \Phi$ and $\partial_{22} \Phi$, given a gap parameter $\delta > 0$, we define *discrete derivatives* (denoted with bars) as

$$\bar{\partial}_1^\delta \Phi(V, S) = \frac{1}{\delta^2} \left[\Phi(V, S) - \Phi(V - \delta^2, S) \right],$$

$$\bar{\partial}_2^\delta \Phi(V, S) = \frac{1}{2\delta} \left[\Phi(V, S + \delta) - \Phi(V, S - \delta) \right],$$

$$\bar{\partial}_{22}^\delta \Phi(V, S) = \frac{1}{\delta^2} \left[\Phi(V, S + \delta) + \Phi(V, S - \delta) - 2\Phi(V, S) \right].$$

If $\delta = 0$, define $\bar{\partial}_1^\delta \Phi(V, S) = \bar{\partial}_2^\delta \Phi(V, S) = \bar{\partial}_{22}^\delta \Phi(V, S) = 0$.

The Discrete Itô formula [HLPR20, Lemma 3.13 and 3.14] has been shown useful in connecting discrete time online learning algorithms with their continuous time counterparts. We generalize it as follows.

Lemma A.5 (Discrete Itô formula with general gap)**.** *Consider any function $\Phi : \mathbb{R}_{\ge 0} \times \mathbb{R} \to \mathbb{R}$, convex in its second argument. For all $V \ge 0$, $S, c \in \mathbb{R}$ and $\delta \ge 0$ satisfying $|c| \le \delta$, we have*

$$\Phi(V + \delta^2, S + c) - \Phi(V, S) \le c \bar{\partial}_2^\delta \Phi(V + \delta^2, S) + \delta^2 \left[\bar{\partial}_1^\delta \Phi(V + \delta^2, S) + \frac{1}{2} \bar{\partial}_{22}^\delta \Phi(V + \delta^2, S) \right]. \tag{7}$$

Proof of Lemma A.5. The case of $\delta = 0$ trivially holds. As for $\delta > 0$, applying the discrete derivatives,

$$\begin{aligned} \text{LHS} &= \Phi(V + \delta^2, S + c) - \frac{1}{2} \left[\Phi(V + \delta^2, S + \delta) + \Phi(V + \delta^2, S - \delta) \right] \\ &\qquad + \frac{1}{2} \left[\Phi(V + \delta^2, S + \delta) + \Phi(V + \delta^2, S - \delta) \right] - \Phi(V, S) \\ &= \Phi(V + \delta^2, S + c) - \frac{1}{2} \left[\Phi(V + \delta^2, S + \delta) + \Phi(V + \delta^2, S - \delta) \right] \\ &\qquad + \delta^2 \left[\bar{\partial}_1^\delta \Phi(V + \delta^2, S) + \frac{1}{2} \bar{\partial}_{22}^\delta \Phi(V + \delta^2, S) \right]. \end{aligned}$$

Comparing it with our objective, it remains to show

$$\begin{aligned} \Phi(V + \delta^2, S + c) - \frac{1}{2} \left[\Phi(V + \delta^2, S + \delta) + \Phi(V + \delta^2, S - \delta) \right] &\le c \bar{\partial}_2^\delta \Phi(V + \delta^2, S) \\ &= \frac{c}{2\delta} \left[\Phi(V + \delta^2, S + \delta) - \Phi(V + \delta^2, S - \delta) \right]. \end{aligned}$$

Regrouping the terms, it suffices to show

$$\Phi(V + \delta^2, S + c) \le \frac{\delta + c}{2\delta} \Phi(V + \delta^2, S + \delta) + \frac{\delta - c}{2\delta} \Phi(V + \delta^2, S - \delta),$$

which follows from the convexity of Φ. $\square$

To apply the Discrete Itô formula to Algorithm 5, we need to fix a minor issue: the function $\Phi(V, S)$ from Eq.(6) is not well-defined for $V = 0$. Without loss of generality, we will impose $\Phi(0, S) = 0$ for all S. Notice that Φ from Eq.(6) is convex in S, and the prediction in Algorithm 5 is precisely the discrete derivative, i.e., $\hat{x}_t = \bar{\partial}_2^{G_t}\Phi(V_t, S_{t-1})$. Plugging in $V \leftarrow V_{t-1}$, $S \leftarrow S_{t-1}$, $\delta \leftarrow G_t$ and $c \leftarrow -\hat{g}_t$ into Lemma A.5, we have

$$\Phi(V_t, S_t) - \Phi(V_{t-1}, S_{t-1}) \leq -\hat{g}_t\hat{x}_t + G_t^2\left[\bar{\partial}_1^{G_t}\Phi(V_t, S_{t-1}) + \frac{1}{2}\bar{\partial}_{22}^{G_t}\Phi(V_t, S_{t-1})\right].$$

The second term on the RHS can be seen as a perturbation on an otherwise clean recursive inequality. The form of this perturbation term also closely resembles the *Backward Heat Equation* (BHE) $\partial_1\Phi + \frac{1}{2}\partial_{22}\Phi = 0$, which, as shown in [ZCP22a], is satisfied by existing potential functions in unconstrained OLO, including Eq.(6). This explains why the Discrete Itô formula is useful: to convert continuous-time-derived algorithms to discrete time, it suffices to characterize the discretization error on the BHE. As long as the discretization error (the perturbation term above) is upper-bounded, we can still control the cumulative loss of the algorithm by a telescopic sum, i.e.,

$$\sum_{t=1}^{T}\hat{g}_t\hat{x}_t \leq \Phi(0,0) - \Phi(V_T, S_T) + \sum_{t=1}^{T} G_t^2\left[\bar{\partial}_1^{G_t}\Phi(V_t, S_{t-1}) + \frac{1}{2}\bar{\partial}_{22}^{G_t}\Phi(V_t, S_{t-1})\right]. \tag{8}$$

Specifically for Algorithm 5, we bound the perturbation term as follows. It uses a key result from [HLPR20].

Lemma A.6. *For all $t \in \mathbb{N}_+$, Algorithm 5 guarantees*

$$G_t^2\left[\bar{\partial}_1^{G_t}\Phi(V_t, S_{t-1}) + \frac{1}{2}\bar{\partial}_{22}^{G_t}\Phi(V_t, S_{t-1})\right] \leq 0.$$

Proof of Lemma A.6. Let us define $f(x) = 2x\mathrm{erfi}(x) - \exp(x^2)$. Due to the definition of Φ in Eq.(6) and the simplification of $\int \mathrm{erfi}(x)dx$ in Lemma A.1, for all $V > 0$ and $S \in \mathbb{R}$,

$$\Phi(V, S) = \hat{\varepsilon}\sqrt{V} f\left(\frac{S}{\sqrt{2V}}\right).$$

When $V = 0$, we have defined $\Phi(0, S) = 0$.

Now, consider the quantities in Algorithm 5. To proceed, there are two cases: (*i*) $V_{t-1} > 0$; (*ii*) $V_{t-1} = 0$. If $V_{t-1} > 0$, plugging in the discrete derivatives,

$$\begin{aligned}
&G_t^2\left[\bar{\partial}_1^{G_t}\Phi(V_t, S_{t-1}) + \frac{1}{2}\bar{\partial}_{22}^{G_t}\Phi(V_t, S_{t-1})\right]\\
=\ &\frac{1}{2}\Phi(V_t, S_{t-1} + G_t) + \frac{1}{2}\Phi(V_t, S_{t-1} - G_t) - \Phi(V_{t-1}, S_{t-1})\\
=\ &\frac{1}{2}\hat{\varepsilon}\sqrt{V_t}\left[f\left(\frac{S_{t-1} + G_t}{\sqrt{2V_t}}\right) + f\left(\frac{S_{t-1} - G_t}{\sqrt{2V_t}}\right) - 2\sqrt{\frac{V_{t-1}}{V_t}} f\left(\frac{S_{t-1}}{\sqrt{2V_{t-1}}}\right)\right].
\end{aligned}$$

Due to [HLPR20, Lemma 3.10], 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 $x = S_{t-1}/\sqrt{V_t}$ and $z = G_t/\sqrt{V_t}$ proves the first case.

As for the second case ($V_{t-1} = 0$), note that $S_{t-1} = 0$ and $V_t = G_t^2$. Then,

$$G_t^2\left[\bar{\partial}_1^{G_t}\Phi(V_t, S_{t-1}) + \frac{1}{2}\bar{\partial}_{22}^{G_t}\Phi(V_t, S_{t-1})\right] = \frac{1}{2}\Phi(G_t^2, G_t) + \frac{1}{2}\Phi(G_t^2, -G_t).$$

If $G_t = 0$, then it holds trivially that RHS $= 0$. Otherwise,

$$\mathrm{RHS} = \frac{\hat{\varepsilon}}{2}G_t\left[f\left(\frac{1}{\sqrt{2}}\right) + f\left(-\frac{1}{\sqrt{2}}\right)\right] \leq 0,$$

due to straightforward evaluation of f. This completes the proof of the second case. □

With Lemma A.5 and A.6, it becomes fairly standard to prove the guarantee of Algorithm 5, i.e., Lemma A.4. See, for example, [Ora19, Chapter 9] for the overall proof strategy.

Proof of Lemma A.4. Plugging Lemma A.6 into Eq.(8), we obtain a cumulative loss bound

$$\sum_{t=1}^{T} \hat{g}_t \hat{x}_t \leq -\Phi(V_T, S_T).$$

Due to a standard loss-regret duality [Ora19, Theorem 9.6], the regret can be bounded by

$$\sum_{t=1}^{T} \langle \hat{g}_t, \hat{x}_t - \hat{u} \rangle \leq \Phi^*_{V_T}(\hat{u}),$$

where $\Phi^*_{V_T}$ denotes the Fenchel conjugate of the function $\Phi_{V_T}(\cdot) := \Phi(V_T, \cdot)$. Finally, due to the proof of [ZCP22a, Theorem 4],

$$\Phi^*_{V_T}(\hat{u}) \leq \hat{\varepsilon}\sqrt{V_T} + |\hat{u}| \sqrt{2V_T} \left[\sqrt{\log\left(1 + \frac{|\hat{u}|}{\sqrt{2}\hat{\varepsilon}}\right)} + 1 \right]. \qquad \square$$

A.3 Proof of main results

This subsection presents the omitted proofs for Section 2.2.

Lemma 2.1. *Let $\hat{\varepsilon} > 0$ be an arbitrary hyperparameter for Algorithm 5. Applying it as a subroutine, for all $T \in \mathbb{N}_+$ and $u_{1:T} \in \operatorname{span}(h_{1:T})$, Algorithm 1 guarantees*

$$\mathrm{Regret}_T(u_{1:T}) \leq G\varepsilon_T + \sqrt{2} \left\| u_{1:T} \right\|_2 G \left[\sqrt{\log\left(1 + \frac{\left\| u_{1:T} \right\|_2}{\sqrt{2}\varepsilon_T}\right)} + 1 \right],$$

where $\varepsilon_T = \hat{\varepsilon} \left\| h_{1:T} \right\|_2$. The subscript emphasizes that ε_T depends on T.

Proof of Lemma 2.1. We start by rewriting the dynamic regret as

$$\mathrm{Regret}_T(u_{1:T}) = \sum_{t=1}^{T} \langle g_t, x_t - u_t \rangle = \sum_{t=1}^{T} \langle g_t, h_t \hat{x}_t - h_t \hat{u}_t \rangle = \sum_{t=1}^{T} \langle \hat{g}_t, \hat{x}_t - \hat{u}_t \rangle.$$

The static regret on the RHS can be bounded from Lemma A.4, with the Lipschitz constant $G_t = G \left\| h_t \right\|_2$. Concretely,

$$\begin{aligned}
\mathrm{Regret}_T(u_{1:T}) &\leq G\hat{\varepsilon}\sqrt{\sum_{t=1}^{T} \left\| h_t \right\|_2^2} + \sqrt{2} G \left| \hat{u} \right| \sqrt{\sum_{t=1}^{T} \left\| h_t \right\|_2^2} \left[\sqrt{\log\left(1 + \frac{|\hat{u}|}{\sqrt{2}\hat{\varepsilon}}\right)} + 1 \right] \\
&= G\hat{\varepsilon} \left\| h_{1:T} \right\|_2 + \sqrt{2} G \left| \hat{u} \right| \left\| h_{1:T} \right\|_2 \left[\sqrt{\log\left(1 + \frac{|\hat{u}| \left\| h_{1:T} \right\|_2}{\sqrt{2}\hat{\varepsilon} \left\| h_{1:T} \right\|_2}\right)} + 1 \right] \\
&= G\varepsilon_T + \sqrt{2} \left\| u_{1:T} \right\|_2 G \left[\sqrt{\log\left(1 + \frac{\left\| u_{1:T} \right\|_2}{\sqrt{2}\varepsilon_T}\right)} + 1 \right]. \qquad \square
\end{aligned}$$

Theorem 1. *For all $T \in \mathbb{N}_+$ and $u_{1:T} \in \mathbb{R}^{dT}$, Algorithm 2 guarantees*

$$\mathrm{Regret}_T(u_{1:T}) \leq 2G\mathcal{E}_T + \sqrt{2} G U_T \left[\sqrt{\log_+ \frac{U_T}{\sqrt{2}\mathcal{E}_T}} + \sqrt{\mathrm{KL}(q||\pi)} + 2 \right] + G \sum_{t=1}^{T} \left\| u_{t,0} \right\|_2,$$

where

1. *For all $n \in [1:N]$, $u_{1:T,n}$ is any vector in* $\mathrm{span}(h_{1:T,n})$, *and* $u_{1:T,0} = u_{1:T} - \sum_{n=1}^{N} u_{1:T,n}$*;*
2. $\mathcal{E}_T = \sum_{n=1}^{N} \hat{\varepsilon}_n \left\|h_{1:T,n}\right\|_2$ *and* $U_T = \sum_{n=1}^{N} \left\|u_{1:T,n}\right\|_2$*;*
3. π *and* q *are* N *dimensional probability vectors defined by* $\pi_n = \hat{\varepsilon}_n \left\|h_{1:T,n}\right\|_2 / \mathcal{E}_T$*, and* $q_n = \left\|u_{1:T,n}\right\|_2 / U_T$*.*

Proof of Theorem 1. To begin with, we apply a dynamic analogue of [Cut19b] to sum the regret bound of single direction learners. Any comparator $u_{1:T}$ can be decomposed into the directions of feature vectors plus an unconstrained residual. Therefore, for all decomposition $u_{1:T} = \sum_{n=0}^{N} u_{1:T,n}$ such that $u_{1:T,n} \in \mathrm{span}(h_{1:T,n})$ for all $n \in [1:T]$, we have

$$\mathrm{Regret}_T(u_{1:T}) = \langle g_{1:T}, x_{1:T} - u_{1:T} \rangle = \langle -g_{1:T}, u_{1:T,0} \rangle + \sum_{n=1}^{N} \langle g_{1:T}, w_{1:T,n} - u_{1:T,n} \rangle .$$

On each direction we apply Lemma 2.1. Moreover, $\langle -g_{1:T}, u_{1:T,0} \rangle \leq G \sum_{t=1}^{T} \left\|u_{t,0}\right\|_2$. It leads to

$$\begin{aligned}
\mathrm{Regret}_T(u_{1:T}) &\leq G \sum_{n=1}^{N} \left\{ \hat{\varepsilon}_n \left\|h_{1:T,n}\right\|_2 + \sqrt{2} \left\|u_{1:T,n}\right\|_2 \left[\sqrt{\log\left(1 + \frac{\left\|u_{1:T,n}\right\|_2}{\sqrt{2}\hat{\varepsilon}_n \left\|h_{1:T,n}\right\|_2}\right)} + 1 \right] \right\} + G \sum_{t=1}^{T} \left\|u_{t,0}\right\|_2 \\
&= G\mathcal{E}_T + \sqrt{2} G U_T + \sqrt{2} G U_T \sum_{n=1}^{N} q_n \sqrt{\log\left(1 + \frac{q_n U_T}{\sqrt{2}\pi_n \mathcal{E}_T}\right)} + G \sum_{t=1}^{T} \left\|u_{t,0}\right\|_2 .
\end{aligned}$$

Now consider the third term. Without loss of generality, assume $q_n > 0$ for all n, and $U_T > 0$. Applying $\log(1+x) \leq \log x + x^{-1}$,

$$\begin{aligned}
\sum_{n=1}^{N} q_n \sqrt{\log\left(1 + \frac{q_n U_T}{\sqrt{2}\pi_n \mathcal{E}_T}\right)} &\leq \sum_{n=1}^{N} q_n \sqrt{\frac{\sqrt{2}\pi_n \mathcal{E}_T}{q_n U_T} + \log \frac{U_T}{\sqrt{2}\mathcal{E}_T} + \log \frac{q_n}{\pi_n}} \\
&= \sum_{n=1}^{N} \sqrt{q_n} \sqrt{\frac{\sqrt{2}\pi_n \mathcal{E}_T}{U_T} + q_n \log \frac{U_T}{\sqrt{2}\mathcal{E}_T} + q_n \log \frac{q_n}{\pi_n}} \\
&\leq \sqrt{\frac{\sqrt{2}\mathcal{E}_T}{U_T} + \log \frac{U_T}{\sqrt{2}\mathcal{E}_T} + \mathrm{KL}(q||\pi)} && \text{(Cauchy-Schwarz)} \\
&\leq \sqrt{\frac{\sqrt{2}\mathcal{E}_T}{U_T}} + \sqrt{\log_+ \frac{U_T}{\sqrt{2}\mathcal{E}_T}} + \sqrt{\mathrm{KL}(q||\pi)}.
\end{aligned}$$

Finally, note that $\sqrt{U_T \mathcal{E}_T} \leq (U_T + \mathcal{E}_T)/2$. Combining everything completes the proof. ☐

A.4 Example

The idea of the sparse coding framework is closely related to signal processing and representation learning, where a fundamental objective is to find a dictionary that sparsely represents the signal structure. Through a few examples, we show that it ties several distinct applications together.

Fourier dictionary Many prediction tasks exhibit natural periodicity, such as the daily temperature, the seasonal sale of a product, and the load on a power grid. Here, trigonometric feature vectors are a reasonable choice. Taking $d = 1$ for example, with a known base frequency ω and an order $K \in \mathbb{N}$, one can define a size $2K$ dictionary from (for all $k \in [1:K]$)

$$\begin{aligned}
h_{t,2k-1} &= \cos(k\omega t), \\
h_{t,2k} &= \sin(k\omega t).
\end{aligned}$$

It is also optional to add an all-one feature to track the constant offset of $u_{1:T}$.

Alternatively, if T is fixed, we may set $N = T$ and define $\mathcal{H}$ as the *Discrete Fourier Transform* (DFT) matrix. Since we only consider real inputs, the complex DFT dictionary can be simplified into the real form above with $\omega = 2\pi/T$, which is intuitively suitable for tasks with unknown periodicity.

Wavelet dictionary Wavelets are powerful tools to handle multi-scale signal structures, and specifically in our framework, "shifting" environments. With $d = 1$ and $N = T$, we consider the simplest *Haar wavelet*, where the dictionary $\mathcal{H}$ is set as the transpose of the (un-normalized) *Haar matrix*. The precise definition is standard, but out of our scope. However, the idea can be clearly illustrated in the special case with $T = 8$:

$$\mathcal{H} = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & -1 & 0 & 0 & 0 \\ 1 & 1 & -1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & -1 & 0 & 0 & -1 & 0 & 0 \\ 1 & -1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & -1 & 0 & 1 & 0 & 0 & -1 & 0 \\ 1 & -1 & 0 & -1 & 0 & 0 & 0 & 1 \\ 1 & -1 & 0 & -1 & 0 & 0 & 0 & -1 \end{bmatrix}.$$

It is an orthogonal basis of $\mathbb{R}^{dT}$. Projecting a signal onto features on the left is equivalent to downsampling, while the removed local details are captured by features on the right. Compared to using the dense DFT matrix, such *local property* simplifies the computation in our framework, as the base algorithm $\mathcal{A}_n$ in Algorithm 2 trivially outputs $w_{t,0} = 0$ when the input feature $h_{t,0} = 0$. Therefore, in each round, the Haar-wavelet-based algorithm only maintains $O(\log T)$ black-box 1D algorithms, as opposed to $O(T)$ in the Fourier-based algorithm.

Dictionary from time series Specifically for time series forecasting, we can learn classical parametric strategies, such as the *autoregressive* (AR) model, by choosing $\mathcal{H}$ properly. As shown in [AHMS13], learning it is a fundamental task for learning the more general ARMA models. If a time series $z_{1:T}$ is generated by a (noiseless) AR(k) model, then with parameters $\alpha_{1:k}$, it satisfies $z_t = \sum_{i=1}^{k} \alpha_i z_{t-i}$.

We consider the time series setup from the beginning of the paper, with $d = 1$ and $w_{1:T} = 0$. Setting $N = p$ and $h_{t,n} = z_{t-n}$, Theorem 1 translates to a forecasting regret bound against any prediction sequence $y_{1:T}$ generated by an AR(k) model. In particular, the bound adapts to the magnitude and sparsity of the comparator model parameter, which induces an oracle inequality similar to Eq.(2). This improves the non-adaptive approach from [AHMS13].

Learned dictionary Since $\mathcal{H}$ is only queried online, we may generate $\mathcal{H}$ itself using an online learning algorithm. If the base learner guarantees a regret bound against certain normalized comparators, then our approach can enhance it by adapting to the actual scale of the comparator, which is unbounded a priori. Concretely, consider $N = 1$, i.e., Algorithm 1. For any (unknown) $\hat{u} \in \mathbb{R}$,

$$\mathrm{Regret}_T(u_{1:T}) = \hat{u} \sum_{t=1}^{T} \left\langle g_t, h_t - \frac{u_t}{\hat{u}} \right\rangle + \sum_{t=1}^{T} \langle g_t, h_t \rangle (\hat{x}_t - \hat{u}).$$

The first sum can be bounded by the guarantee of $h_{1:T}$ – this is the ideal adaptive bound we aim for. In this regard, the second sum is the overhead of such adaptivity, similar to the objective in Lemma 2.1. Specific applications of this technique can be found in [CO18] and [JC22, Appendix I], while here we show that in general, it can be viewed as an instance of the sparse coding framework.

A.5 Model selection by summation

For unconstrained dynamic regret, an appealing property is that different regret bounds can be simply aggregated by summation. This is essentially the idea of Theorem 1 itself. From this angle, our sparse coding framework and the path length bound from [JC22] are mutually complementary.

Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be two algorithms, each guaranteeing an unconstrained dynamic regret bound $f_i(u_{1:T})$ for all $u_{1:T} \in \mathbb{R}^{dT}$, $i = 1$ or 2. Consider a master algorithm $\mathcal{A}$ that simply predicts the sum of their predictions. Then, for *any* decomposition of the comparator $u_{1:T} = u^{(1)}_{1:T} + u^{(2)}_{1:T}$,

$$\mathrm{Regret}_T(u_{1:T}) \leq f_1(u^{(1)}_{1:T}) + f_2(u^{(2)}_{1:T}).$$

For example, take $\mathcal{A}_1$ as the sparse coding algorithm, and $\mathcal{A}_2$ as the algorithm from [JC22]. Then, within Theorem 1, we can replace the trivial characterization of $u_{t,0}$ by a path length bound on $u_{t,0}$. The sacrifice is only a slightly larger cumulative loss term, i.e., $2G\mathcal{E}_T$ in Theorem 1. The result can also be plugged into the oracle inequality Eq.(2).

A.6 Detail on the primal-dual interpretation

Supplementing the primal-dual discussion in Section 2.3, we present several versions of the loss-regret duality on the sequence space $\mathbb{R}^{dT}$. The first one generalizes a classical argument in static unconstrained OLO [MO14, Theorem 1]. The other two are to our knowledge new, and are more closely tied to the path length characterization of the comparator. Define $s_t = \sum_{i=1}^t g_i$, i.e., the sum of past gradients.

Lemma 2.2. *If there exists a function $f_T : \mathbb{R}^{dT} \to \mathbb{R}$ such that the prediction sequence $x_{1:T}$ guarantees a loss upper bound $\langle g_{1:T}, x_{1:T}\rangle \leq -f_T(g_{1:T})$, then for all $u_{1:T} \in \mathbb{R}^{dT}$,*

$$\mathrm{Regret}_T(u_{1:T}) \leq f_T^*(-u_{1:T}).$$

Proof of Lemma 2.2. This follows from a standard Fenchel duality argument.

$$\mathrm{Regret}_T(u_{1:T}) = \langle g_{1:T}, x_{1:T} - u_{1:T}\rangle \leq \langle g_{1:T}, -u_{1:T}\rangle - f_T(g_{1:T}) \leq \sup_{x\in\mathbb{R}^{dT}} \langle x, -u_{1:T}\rangle - f_T(x) = f_T^*(-u_{1:T}). \qquad \square$$

Lemma A.7. *Recall that we defined $s_t = \sum_{i=1}^t g_i$. If there exists $f_T : \mathbb{R}^{dT} \to \mathbb{R}$ such that $x_{1:T}$ guarantees $\langle g_{1:T}, x_{1:T}\rangle \leq -f_T(s_{1:T})$, then for all $u_{1:T} \in \mathbb{R}^{dT}$,*

$$\mathrm{Regret}_T(u_{1:T}) \leq f_T^*(u_2 - u_1, u_3 - u_2, \ldots, u_T - u_{T-1}, -u_T).$$

Proof of Lemma A.7. We start by rewriting the comparator loss.

$$\begin{aligned}\langle g_{1:T}, u_{1:T}\rangle &= \langle s_T, u_T\rangle + \sum_{t=1}^T \langle g_t, u_t - u_T\rangle = \langle s_T, u_T\rangle + \sum_{t=1}^T\sum_{i=t}^{T-1} \langle g_t, u_i - u_{i+1}\rangle \\ &= \langle s_T, u_T\rangle + \sum_{i=1}^{T-1}\sum_{t=1}^{i} \langle g_t, u_i - u_{i+1}\rangle = \langle s_T, u_T\rangle + \sum_{t=1}^{T-1} \langle s_t, u_t - u_{t+1}\rangle.\end{aligned}$$

Given the loss upper bound,

$$\begin{aligned}\mathrm{Regret}_T(u_{1:T}) &\leq \sum_{t=1}^{T-1} \langle s_t, u_{t+1} - u_t\rangle + \langle s_T, -u_T\rangle - f_T(s_{1:T}) \\ &\leq \sup_{x_{1:T}\in\mathbb{R}^{dT}} \left(\sum_{t=1}^{T-1} \langle x_t, u_{t+1} - u_t\rangle + \langle x_T, -u_T\rangle - f_T(x_{1:T})\right) \\ &= f_T^*(u_2 - u_1, u_3 - u_2, \ldots, u_T - u_{T-1}, -u_T). \qquad \square\end{aligned}$$

By reversing the index, we have the following lemma.

Lemma A.8. *If there exists $f_T : \mathbb{R}^{dT} \to \mathbb{R}$ such that $x_{1:T}$ guarantees $\langle g_{1:T}, x_{1:T}\rangle \leq -f_T(s_T, s_T - s_1, \ldots, s_T - s_{T-1})$, then for all $u_{1:T} \in \mathbb{R}^{dT}$,*

$$\mathrm{Regret}_T(u_{1:T}) \leq f_T^*(-u_1, u_1 - u_2, u_2 - u_3, \ldots, u_{T-1} - u_T).$$

Proof of Lemma A.8. Similar to the proof above,

$$\begin{aligned}\langle g_{1:T}, u_{1:T}\rangle &= \langle s_T, u_1\rangle + \sum_{t=1}^T \langle g_t, u_t - u_1\rangle = \langle s_T, u_1\rangle + \sum_{t=1}^T\sum_{i=1}^{t-1} \langle g_t, u_{i+1} - u_i\rangle \\ &= \langle s_T, u_1\rangle + \sum_{i=1}^{T-1}\sum_{t=i+1}^{T} \langle g_t, u_{i+1} - u_i\rangle = \langle s_T, u_1\rangle + \sum_{t=1}^{T-1} \langle s_T - s_t, u_{t+1} - u_t\rangle.\end{aligned}$$

$$
\begin{aligned}
\mathrm{Regret}_T(u_{1:T}) &\le \langle s_T, -u_1\rangle + \sum_{t=1}^{T-1} \langle s_T - s_t, u_t - u_{t+1}\rangle - f_T(s_T, s_T - s_1, \dots, s_T - s_{T-1}) \\
&\le \sup_{x_{1:T}\in\mathbb{R}^{dT}} \left(\langle x_1, -u_1\rangle + \sum_{t=2}^{T} \langle x_t, u_{t-1} - u_t\rangle - f_T(x_{1:T}) \right) \\
&= f_T^*(-u_1, u_1 - u_2, u_2 - u_3, \dots, u_{T-1} - u_T). \qquad \square
\end{aligned}
$$

B Detail on the improved static algorithm

This section presents the second contribution of the paper, an improved static unconstrained OLO algorithm. Appendix B.1 contains the derivatives of our potential functions, which will be useful in the proof. Appendix B.2 analyzes the base algorithm (Algorithm 3), with its regret bound presented as Lemma B.6. Appendix B.3 presents the analysis of the meta algorithm, resulting in Theorem 2, the main theorem of this section. Appendix B.4 discusses the application of this static algorithm to the dynamic regret problem, through the sparse coding framework.

B.1 Facts of the potential function

For the two potential functions defined in Section 3.2, we compute their derivatives as follows. This will be useful later on.

$$
\partial_1\phi(x,y) = -\frac{\varepsilon\sqrt{\alpha}}{2\sqrt{x}} \exp\left(\frac{y^2}{4\alpha x}\right),
$$

$$
\partial_2\phi(x,y) = \varepsilon \mathrm{erfi}\left(\frac{y}{\sqrt{4\alpha x}}\right),
$$

$$
\partial_{11}\phi(x,y) = \frac{\varepsilon\sqrt{\alpha}}{4x^{3/2}} \left(\frac{y^2}{2\alpha x} + 1\right) \exp\left(\frac{y^2}{4\alpha x}\right),
$$

$$
\partial_{12}\phi(x,y) = -\frac{\varepsilon y}{4\sqrt{\alpha} x^{3/2}} \exp\left(\frac{y^2}{4\alpha x}\right),
$$

$$
\partial_{22}\phi(x,y) = \frac{\varepsilon}{2\sqrt{\alpha x}} \exp\left(\frac{y^2}{4\alpha x}\right).
$$

Due to the change of variable, the derivatives of Φ can be concisely represented as

$$
\partial_1\Phi(V,S) = \partial_1\phi(V+z+kS, S),
$$

$$
\begin{aligned}
\partial_2\Phi(V,S) &= k\partial_1\phi(V+z+kS,S) + \partial_2\phi(V+z+kS,S), \\
\partial_{11}\Phi(V,S) &= \partial_{11}\phi(V+z+kS,S), \\
\partial_{12}\Phi(V,S) &= k\partial_{11}\phi(V+z+kS,S) + \partial_{12}\phi(V+z+kS,S), \\
\partial_{22}\Phi(V,S) &= k^2\partial_{11}\phi(V+z+kS,S) + 2k\partial_{12}\phi(V+z+kS,S) + \partial_{22}\phi(V+z+kS,S).
\end{aligned}
$$

B.2 Analysis of the base algorithm

The key component of our approach is the base algorithm (Algorithm 3). Within its analysis, the most crucial part is the characterization of the one step change of the potential (Lemma B.3). This subsection is outlined as follows. We first present two simple lemmas on the property of our potential function Φ. Then, we prove the key lemma (Lemma B.3), which leads to a cumulative loss upper bound (Lemma B.4). As in the standard analysis of potential methods, converting the loss upper bound to the regret bound relies on computing the Fenchel conjugate of Φ – this is the focus of Lemma B.5. Finally, Lemma B.6 combines everything into the regret bound of the base algorithm.

To begin with, we first show that $\Phi(V,S)$ is convex in S, just like more standard potential functions.

Lemma B.1. *If $\varepsilon > 0$, $\alpha > 0$ and $z > k > 0$, the potential function $\Phi(V, S)$ satisfies $\partial_{22}\Phi(V, S) \geq 0$ for all $V \geq 0$ and $S \geq -1$.*

Proof of Lemma B.1. Define the shorthands $x = V + z + kS$ and $y = S$. For all $V \geq 0$ and $S \geq -1$, we have $x > 0$, therefore

$$\begin{aligned}\partial_{22}\Phi(V,S) &= k^2\partial_{11}\phi(x,y) + 2k\partial_{12}\phi(x,y) + \partial_{22}\phi(x,y)\\ &= \frac{\varepsilon\sqrt{\alpha}}{4x^{3/2}}\exp\left(\frac{y^2}{4\alpha x}\right)\left(\frac{k^2y^2}{2\alpha x} + k^2 - \frac{2ky}{\alpha} + \frac{2x}{\alpha}\right)\\ &= \frac{\varepsilon\sqrt{\alpha}}{4x^{3/2}}\exp\left(\frac{y^2}{4\alpha x}\right)\left(\frac{k^2y^2}{2\alpha x} + k^2 + \frac{2(V+z)}{\alpha}\right)\\ &\geq 0.\end{aligned}$$ □

Next, we show that the base algorithm makes strictly negative prediction $\tilde{z}_t$ when S_{t-1} is negative. This will be exploited by the meta-algorithm to ensure that the surrogate losses received by the base algorithm satisfy the $S_t \geq -1$ constraint.

Lemma B.2. *If $\varepsilon > 0$, $\alpha > 0$ and $z > k > 0$, the potential function $\Phi(V, S)$ satisfies $\partial_2\Phi(V, S) < 0$ for all $V \geq 0$ and $-1 \leq S \leq 0$.*

Proof of Lemma B.2. Let us check $\partial_2\Phi(V, 0) < 0$. Indeed,

$$\partial_2\Phi(V,0) = k\partial_1\phi(V+z,0) + \partial_2\phi(V+z,0) = -\frac{\varepsilon k\sqrt{\alpha}}{2\sqrt{V+z}} < 0.$$

Moreover, $\partial_{22}\Phi(V, S) \geq 0$ due to Lemma B.1. Therefore, we have $\partial_2\Phi(V, S) < 0$ for all $V \geq 0$ and $-1 \leq S \leq 0$. □

The key lemma in our analysis is the following, which says a suitable combination of parameters yields a one-step loss bound on the potential function Φ, as long as the second argument of Φ is always larger than -1. Such a lemma is typically the central component in the classical potential analysis.

Lemma B.3 (Key lemma: one step potential bound)**.** *With $\alpha = 1$, $k = 2$, $z = 16$ and an arbitrary $\varepsilon > 0$, the potential function $\Phi(V, S)$ satisfies*

$$\Phi(V + c^2, S + c) - \Phi(V, S) - c\partial_2\Phi(V, S) \leq 0,$$

for all $V \geq 0$, $S \geq -1$ and $c \in [-1, 1] \cap [-1 - S, \infty)$.

Proof of Lemma B.3. Let us view our objective as a function of c,

$$f_{V,S}(c) := \Phi(V + c^2, S + c) - \Phi(V, S) - c\partial_2\Phi(V, S).$$

Taking the derivatives,

$$f'_{V,S}(c) = 2c\partial_1\Phi(V + c^2, S + c) + \partial_2\Phi(V + c^2, S + c) - \partial_2\Phi(V, S),$$

$$\begin{aligned}f''_{V,S}(c) &= 2\partial_1\Phi(V+c^2,S+c) + 4c^2\partial_{11}\Phi(V+c^2,S+c) + 4c\partial_{12}\Phi(V+c^2,S+c) + \partial_{22}\Phi(V+c^2,S+c)\\ &\leq 2\partial_1\Phi(V+c^2,S+c) + 4\partial_{11}\Phi(V+c^2,S+c) + 4\left|\partial_{12}\Phi(V+c^2,S+c)\right| + \partial_{22}\Phi(V+c^2,S+c). \quad (9)\end{aligned}$$

Note that $f_{V,S}(0) = f'_{V,S}(0) = 0$. Therefore, to prove $f_{V,S}(c) \leq 0$, it suffices to show $f''_{V,S}(c) \leq 0$ for all considered values of V, S and c. The RHS of Eq.(9) has a striking similarity to the Backward Heat Equation – in fact, after a change of variable, the resulting expressions, namely Eq.(10) and Eq.(11) below, will resemble a BHE on ϕ ($\partial_1\phi + \alpha\partial_{22}\phi = 0$) plus a perturbation. The main goal of this proof is to control such perturbations by properly choosing α, k and z.

Concretely, due to the absolute value in Eq.(9), we will analyze two cases. Technically, the first case is harder, therefore we pick k to simplify its analysis. The second case requires $S + c$ to be around zero – this is an "edge" case and relatively easier to handle.

Case 1: $\partial_{12}\Phi(V+c^2, S+c) \leq 0$**.** Substituting the derivatives of Φ by the derivatives of ϕ, we have

$$f''_{V,S}(c) \leq 2\partial_1\phi + (k-2)^2\partial_{11}\phi + 2(k-2)\partial_{12}\phi + \partial_{22}\phi\Big|_{(V+c^2+z+k(S+c),S+c)}. \tag{10}$$

The RHS means we evaluate all the derivative functions at $(V+c^2+z+k(S+c), S+c)$. Plugging in our specific choice of k and α,

$$\begin{aligned} f''_{V,S}(c) &\leq 2\partial_1\phi + \partial_{22}\phi\Big|_{(V+c^2+z+2(S+c),S+c)} && (k=2) \\ &\leq 2\partial_1\phi + 2\partial_{22}\phi\Big|_{(V+c^2+z+2(S+c),S+c)} && (\partial_{22}\phi \geq 0) \\ &= 0. && (\phi \text{ satisfies the BHE with } \alpha = 1.) \end{aligned}$$

Case 2: $\partial_{12}\Phi(V+c^2, S+c) \geq 0$**.** Similar to the first case,

$$f''_{V,S}(c) \leq 2\partial_1\phi + (k+2)^2\partial_{11}\phi + 2(k+2)\partial_{12}\phi + \partial_{22}\phi\Big|_{(V+c^2+z+k(S+c),S+c)}. \tag{11}$$

Consider the k-dependent "perturbation" terms in Eq.(11), i.e., $(k+2)^2\partial_{11}\phi + 2(k+2)\partial_{12}\phi$. Our goal is to upper bound it by $\partial_{22}\phi$, such that an upper bound of $f''_{V,S}(c)$ follows from the BHE. Plugging in the derivatives of ϕ from Appendix B.1, for all inputs (x, y),

$$(k+2)^2\partial_{11}\phi + 2(k+2)\partial_{12}\phi - \partial_{22}\phi\Big|_{(x,y)} = \frac{\varepsilon}{4\sqrt{\alpha}x^{3/2}}\exp\left(\frac{y^2}{4\alpha x}\right)\left[(k+2)^2\left(\frac{y^2}{2x}+\alpha\right) - 2(k+2)y - 2x\right].$$

We aim to show the bracket on the RHS is negative at $x = V + c^2 + z + k(S+c)$ and $y = S + c$. Also plugging in our choice of $\alpha = 1$ and $k = 2$, this amounts to showing

$$\lozenge := \frac{2(S+c)^2}{V+c^2+z+2S+2c} + 4 - 3(S+c) - \frac{1}{2}(V+c^2+z) \leq 0.$$

The idea is that we can pick a large enough z to make it hold. Concretely,

- If $S + c > 0$, then
$$\lozenge \leq \frac{2(S+c)^2}{2S+2c} + 4 - 3(S+c) - \frac{1}{2}z \leq 4 - \frac{1}{2}z,$$
and it suffices to pick $z \geq 8$.
- If $S + c \leq 0$, then since $c \in [-1-S, \infty)$, we have $S + c \geq -1$. As long as $z > 2$,
$$\lozenge \leq \frac{2}{z-2} + 7 - \frac{1}{2}z.$$
It suffices to pick $z \geq 16$.

In summary, $z = 16$ ensures $\lozenge \leq 0$. Due to the BHE on ϕ,

$$f''_{V,S}(c) \leq 2\partial_1\phi + 2\partial_{22}\phi\Big|_{(V+c^2+z+2(S+c),S+c)} = 0.$$

Combining the two cases completes the proof. □

Based on Lemma B.3, we immediately obtain a cumulative loss bound of the base algorithm. The proof is a straightforward telescopic sum, therefore omitted.

Lemma B.4 (Cumulative loss bound)**.** *With $\alpha = 1$, $k = 2$ and $z = 16$, Algorithm 3 guarantees for all $T \in \mathbb{N}_+$,*

$$\sum_{t=1}^{T} \tilde{l}_t\tilde{z}_t \leq \Phi(0,0) - \Phi(\tilde{V}_T, \tilde{S}_T).$$

As for the regret bound, similar to the standard duality argument [Ora19, Chapter 9], we need the Fenchel conjugate of the potential function Φ. With any $V \geq 0$, define

$$\Phi_V^*(u) := \sup_{S \in [-1,\infty)} uS - \Phi(V,S)$$

as the conjugate of $\Phi(V,S)$ with respect to S. Slightly different from the standard definition where the supremum is over $\mathbb{R}$, here the supremum is over $[-1,\infty)$, since the surrogate losses in the base algorithm satisfy $\tilde{S}_t \geq -1$ for all t. To proceed, we will only consider the dual variable satisfying $u \geq 0$.

Lemma B.5 (Conjugate)**.** *With $\varepsilon > 0$, $\alpha > 0$ and $z > k > 0$, for all $u \geq 0$,*

$$\begin{aligned}\Phi_V^*(u) &:= \sup_{S \in [-1,\infty)} uS - \Phi(V,S)\\ &\leq u\bar{S} + \varepsilon\sqrt{\alpha(V+z+k\bar{S})},\end{aligned}$$

where

$$\bar{S} = 4\alpha k\left(1+\sqrt{\log(2u\varepsilon^{-1}+1)}\right)^2 + \sqrt{4\alpha(V+z)}\left(1+\sqrt{\log(2u\varepsilon^{-1}+1)}\right).$$

Proof of Lemma B.5. We first show that the supremum over S in the Fenchel conjugate is attainable by some $S^* \in [0,\infty)$. To this end, define a function $f(S) = uS - \Phi(V,S)$. f is continuous, with $f'(S) = u - \partial_2\Phi(V,S)$. Moreover, due to Lemma B.1, f is concave on $[-1,\infty)$. The existence of S^* then follows from analyzing the boundary.

- For all $S \in [-1,0]$, we have $f'(S) \geq 0$. The reason is $u \geq 0$, and $\partial_2\Phi(V,S) \leq 0$ due to Lemma B.2.
- For sufficiently large S, we aim to show $f'(S) < 0$. Let us begin by writing down $\partial_2\Phi(V,S)$, from Appendix B.1.

$$\partial_2\Phi(V,S) = \varepsilon\mathrm{erfi}\left(\frac{S}{\sqrt{4\alpha(V+z+kS)}}\right) - \frac{\varepsilon k\sqrt{\alpha}}{2\sqrt{V+z+kS}}\exp\left(\frac{S^2}{4\alpha(V+z+kS)}\right).$$

  Now consider large S that satisfies $S \geq \sqrt{4\alpha(V+z+kS)}$. Due to an estimate of the erfi function (Lemma A.2),

$$\mathrm{erfi}\left(\frac{S}{\sqrt{4\alpha(V+z+kS)}}\right) \geq \frac{\sqrt{\alpha(V+z+kS)}}{S}\exp\left(\frac{S^2}{4\alpha(V+z+kS)}\right).$$

  Moreover,

$$\frac{\sqrt{\alpha(V+z+kS)}}{S} - \frac{k\sqrt{\alpha}}{\sqrt{V+z+kS}} = \frac{\sqrt{\alpha}(V+z)}{S\sqrt{V+z+kS}} \geq 0.$$

  Therefore,

$$\begin{aligned}\partial_2\Phi(V,S) &= \left[\frac{\varepsilon}{2}\mathrm{erfi}\left(\frac{S}{\sqrt{4\alpha(V+z+kS)}}\right) - \frac{\varepsilon k\sqrt{\alpha}}{2\sqrt{V+z+kS}}\exp\left(\frac{S^2}{4\alpha(V+z+kS)}\right)\right]\\ &\qquad + \frac{\varepsilon}{2}\mathrm{erfi}\left(\frac{S}{\sqrt{4\alpha(V+z+kS)}}\right)\\ &\geq \frac{\varepsilon}{2}\mathrm{erfi}\left(\frac{S}{\sqrt{4\alpha(V+z+kS)}}\right). \end{aligned} \tag{12}$$

  For sufficiently large S, we have RHS $> u$, hence $f'(S) < 0$.

Summarizing the above, we have shown that there exists $S^* \in [0,\infty)$ such that

$$\Phi_V^*(u) := \sup_{S \in [-1,\infty)} uS - \Phi(V,S) = uS^* - \Phi(V,S^*).$$

Moreover, S^* should satisfy the first order condition $f'(S^*) = 0$, i.e., $u = \partial_2\Phi(V,S^*)$. Our goal next is to upper bound S^* by a function of u. Again, we analyze two cases.

Case 1. If S^* satisfies $S^* < \sqrt{4\alpha(V+z+kS^*)}$, then by regrouping the terms, we have $(S^*)^2 - 4\alpha k S^* - 4\alpha(V+z) < 0$. Solving this quadratic inequality,

$$\begin{aligned}
S^* &\le \frac{1}{2}\left[4\alpha k + \sqrt{(4\alpha k)^2 + 16\alpha(V+z)}\right] \\
&= 2\alpha k + \sqrt{4\alpha^2 k^2 + 4\alpha(V+z)} \\
&\le 4\alpha k + \sqrt{4\alpha(V+z)}.
\end{aligned}$$

Case 2. If S^* satisfies $S^* \ge \sqrt{4\alpha(V+z+kS^*)}$, then same as the earlier analysis in the present proof, Eq.(12), we have

$$u \ge \frac{\varepsilon}{2}\mathrm{erfi}\left(\frac{S^*}{\sqrt{4\alpha(V+z+kS^*)}}\right).$$

For conciseness, define the notation $p = \mathrm{erfi}^{-1}(2u\varepsilon^{-1})$. Then, $(S^*)^2 - 4\alpha k p^2 S^* - 4\alpha p^2(V+z) < 0$. Solving the quadratic inequality,

$$\begin{aligned}
S^* &\le \frac{1}{2}\left[4\alpha k p^2 + \sqrt{(4\alpha k p^2)^2 + 16\alpha p^2(V+z)}\right] \\
&\le 2\alpha k p^2 + \sqrt{4\alpha^2 k^2 p^4 + 4\alpha p^2(V+z)} \\
&\le 4\alpha k p^2 + \sqrt{4\alpha(V+z)}p.
\end{aligned}$$

Now we can combine the above two cases. Specifically, $p \le 1 + \sqrt{\log(2u\varepsilon^{-1}+1)}$ due to Lemma A.3. Therefore,

$$S^* \le 4\alpha k\left(1 + \sqrt{\log(2u\varepsilon^{-1}+1)}\right)^2 + \sqrt{4\alpha(V+z)}\left(1 + \sqrt{\log(2u\varepsilon^{-1}+1)}\right).$$

Define the RHS as $\bar{S}$. Then, from the definition of the Fenchel conjugate,

$$\begin{aligned}
\Phi_V^*(u) &= uS^* - \Phi(V, S^*) \\
&= uS^* - \varepsilon\sqrt{\alpha(V+z+kS^*)}\left[2\int_0^{\frac{S^*}{\sqrt{4\alpha(V+z+kS^*)}}} \mathrm{erfi}(u)du - 1\right] \\
&\le u\bar{S} + \varepsilon\sqrt{\alpha(V+z+k\bar{S})}.
\end{aligned}$$

Plugging in $\bar{S}$ completes the proof. □

Finally, we assemble the cumulative loss bound (Lemma B.4) and the conjugate of the potential (Lemma B.5) into the regret bound of the base algorithm.

Lemma B.6 (Regret of the base algorithm)**.** *With $\varepsilon > 0$, $\alpha = 1$, $k = 2$ and $z = 16$, Algorithm 3 guarantees for all $T \in \mathbb{N}_+$ and $\tilde{u} \ge 0$,*

$$\sum_{t=1}^{T} \tilde{l}_t(\tilde{z}_t - \tilde{u}) \le \varepsilon\sqrt{\tilde{V}_T + 2\bar{S}} + \tilde{u}\bar{S},$$

where

$$\bar{S} = 8\left(1 + \sqrt{\log(2\tilde{u}\varepsilon^{-1}+1)}\right)^2 + 2\sqrt{\tilde{V}_T + 16}\left(1 + \sqrt{\log(2\tilde{u}\varepsilon^{-1}+1)}\right).$$

Proof of Lemma B.6. Due to the standard loss-regret duality [Ora19, Theorem 9.6], starting from the cumulative loss bound (Lemma B.4), the regret can be bounded by

$$\begin{aligned}
\sum_{t=1}^{T}\tilde{l}_t(\tilde{z}_t-\tilde{u}) &\leq S_T\tilde{u}+\Phi(0,0)-\Phi(\tilde{V}_T,\tilde{S}_T)\\
&\leq \Phi(0,0)+\sup_{S\in[-1,\infty)}\left[S\tilde{u}-\Phi(\tilde{V}_T,S)\right]\\
&= \Phi(0,0)+\Phi^*_{\tilde{V}_T}(\tilde{u})\\
&\leq -4\varepsilon+\varepsilon\sqrt{\tilde{V}_T+16+2\bar{S}}+\tilde{u}\bar{S}\\
&\leq \varepsilon\sqrt{\tilde{V}_T+2\bar{S}}+\tilde{u}\bar{S}.
\end{aligned}$$

Plugging in $\bar{S}$ from Lemma B.5 completes the proof. □

B.3 Proof of the main result

This subsection presents the theoretical guarantees of the meta algorithm (Algorithm 4). We first show that when combined with the base algorithm (Algorithm 3), the whole procedure is well-posed, in the sense that the surrogate loss $\tilde{l}_t$ satisfies $-\sum_{i=1}^{t}\tilde{l}_i\geq -1$ for all t.

Proposition 3 (Well-posedness)**.** *The surrogate loss $\tilde{l}_t$ defined in Algorithm 4 satisfies $-\sum_{i=1}^{t}\tilde{l}_i\geq -1$ for all t.*

Proof of Proposition 3. First, notice that $|\tilde{l}_t|\leq|l_t|=\left|\left\langle g_tG^{-1},w_t\right\rangle\right|\leq 1$.

Next, we prove by induction. Consider $-\sum_{i=1}^{t-1}\tilde{l}_i$, which is defined as $\tilde{S}_{t-1}$ in the base algorithm (Algorithm 3). Suppose $\tilde{S}_{t-1}\geq -1$, which trivially holds at $t=1$. Let us analyze two cases.

- If $\tilde{S}_{t-1}\geq 0$, then $-\sum_{i=1}^{t}\tilde{l}_i=\tilde{S}_{t-1}-\tilde{l}_t\geq\tilde{S}_{t-1}-|\tilde{l}_t|\geq -1$.
- If $-1\leq\tilde{S}_{t-1}<0$, then due to Lemma B.2, the prediction $\tilde{z}_t$ of the base algorithm satisfies $\tilde{z}_t<0$. The meta algorithm projects it to $z_t=0$. Then, due to our definition of $\tilde{l}_t$ in the meta algorithm,

$$\tilde{l}_t=\begin{cases} l_t, & l_t\leq 0,\\ 0, & \text{else},\end{cases}$$

which is non-positive. Therefore, $-\sum_{i=1}^{t}\tilde{l}_i=\tilde{S}_{t-1}-\tilde{l}_t\geq\tilde{S}_{t-1}\geq -1$.

An induction completes the proof. □

Next we present the main result, the static regret bound of Algorithm 4. Here we define the gradient variance $V_T=\sum_{t=1}^{T}\|g_t\|_2^2$.

Theorem 2. *With $\varepsilon>0$, $\alpha=1$, $k=2$ and $z=16$, Algorithm 4 guarantees for all $T\in\mathbb{N}_+$ and $u\in\mathbb{R}^d$,*

$$\mathrm{Regret}_T(u)\leq\varepsilon\sqrt{V_T+2G\bar{S}}+\|u\|_2\left(\bar{S}+2\sqrt{2V_T}\right),$$

where

$$\bar{S}=8G\left(1+\sqrt{\log(2\|u\|_2\varepsilon^{-1}+1)}\right)^2+2\sqrt{V_T+16G^2}\left(1+\sqrt{\log(2\|u\|_2\varepsilon^{-1}+1)}\right).$$

Proof of Theorem 2. Since the meta algorithm simply applies two existing black-box reductions [CO18, Cut20], the proof is straightforward given Lemma B.6. First, due to a polar decomposition theorem [CO18, Theorem 2], the regret can be decomposed into the regret of $\mathcal{A}_B$ with respect to $u/\|u\|_2$, plus the regret of z_t with respect to $\|u\|_2$. Then, the latter is upper-bounded by the regret of $\tilde{z}_t$ – this is because our definition of z_t and $\tilde{l}_t$ follows the procedure of [Cut20, Theorem 2], where a convex constraint can be added to an unconstrained algorithm without changing its regret bound. In summary, we have

$$\begin{aligned}\mathrm{Regret}_T(u) &\le G\sum_{t=1}^{T} l_t(z_t - \|u\|_2) + \|u\|_2 \sum_{t=1}^{T} \langle g_t, w_t - u/\|u\|_2\rangle \\ &\le G\sum_{t=1}^{T} \tilde{l}_t(\tilde{z}_t - \|u\|_2) + \|u\|_2 \sum_{t=1}^{T} \langle g_t, w_t - u/\|u\|_2\rangle .\end{aligned}$$

The two regret terms on the RHS represent the regret bound of $\mathcal{A}_{1d}$ and $\mathcal{A}_{\mathsf{B}}$, respectively.

Now, the first term follows from Lemma B.6, where $\tilde{V}_T = \sum_{t=1}^{T} \tilde{l}_t^2 \le \sum_{t=1}^{T} l_t^2 = \sum_{t=1}^{T} \langle g_t G^{-1}, w_t\rangle^2 \le V_T/G^2$. As for the regret of $\mathcal{A}_{\mathsf{B}}$, due to [Ora19, Theorem 4.14],

$$\sum_{t=1}^{T} \langle g_t, w_t - u/\|u\|_2\rangle \le 2\sqrt{2V_T}.$$

Combining these two components completes the proof. □

Finally, let us use asymptotic orders to make this bound a bit more interpretable. Consider the regime of large $\|u\|_2$ and V_T, i.e., $\|u\|_2 \gg \varepsilon$ and $V_T \gg G^2$. We preserve the dependence of ε, as it is an arbitrary hyperparameter. In contrast, α, z and k are absolute constants, therefore subsumed by the big-Oh.

Using $\log(1+x) \le x$, we can crudely bound $\bar{S}$ by

$$\begin{aligned}\bar{S} &\le 8G\left(1 + \sqrt{2\|u\|_2 \varepsilon^{-1}}\right)^2 + 2\sqrt{V_T + 16G^2}\left(1 + \sqrt{2\|u\|_2 \varepsilon^{-1}}\right) \\ &= 8G + 2\sqrt{V_T + 16G^2} + o\left(\|u\|_2 \varepsilon^{-1}\sqrt{V_T}\right).\end{aligned}$$

Plugging this crude bound of $\bar{S}$ into the first term of the regret bound, we have

$$\begin{aligned}\mathrm{Regret}_T(u) &\le \varepsilon\sqrt{V_T + 16G^2 + 4G\sqrt{V_T + 16G^2}} + \varepsilon\sqrt{o\left(G\|u\|_2 \varepsilon^{-1}\sqrt{V_T}\right)} + \|u\|_2\left(\bar{S} + 2\sqrt{2V_T}\right) \\ &\le \varepsilon(\sqrt{V_T + 16G^2} + 2G) + o\left(\sqrt{G}\|u\|_2 V_T^{1/4}\right) + \|u\|_2\left(\bar{S} + 2\sqrt{2V_T}\right) \\ &\le \varepsilon(\sqrt{V_T} + 6G) + o\left(\sqrt{G}\|u\|_2 V_T^{1/4}\right) + \|u\|_2\left(\bar{S} + 2\sqrt{2V_T}\right).\end{aligned}$$

Next, notice that $\bar{S} = O\left(\sqrt{V_T \log(\|u\|_2 \varepsilon^{-1})} \vee G\log(\|u\|_2 \varepsilon^{-1})\right)$. Using it to replace the remaining $\bar{S}$ above,

$$\mathrm{Regret}_T(u) \le \varepsilon\left(\sqrt{V_T} + 6G\right) + o\left(\sqrt{G}\|u\|_2 V_T^{1/4}\right) + \|u\|_2 O\left(\sqrt{V_T \log(\|u\|_2 \varepsilon^{-1})} \vee G\log(\|u\|_2 \varepsilon^{-1})\right).$$

The second term can be assimilated into the third term. The result becomes

$$\mathrm{Regret}_T(u) \le \varepsilon\left(\sqrt{V_T} + 6G\right) + \|u\|_2 O\left(\sqrt{V_T \log(\|u\|_2 \varepsilon^{-1})} \vee G\log(\|u\|_2 \varepsilon^{-1})\right), \quad (13)$$

where $O(\cdot)$ subsumes absolute constants.

Note that this bound is not only valid for large $\|u\|_2$, but also valid when $u = 0$ (this can be directly verified from Theorem 2). Therefore, we can use it to characterize the loss-regret tradeoff. It is the same loss-regret tradeoff as [ZCP22a], but with time T replaced by the gradient variance V_T.

Towards the optimal leading constant Without considering gradient adaptivity, [ZCP22a] showed that the optimal leading term in the regret bound (including the multiplicative constant) is $\|u\|_2 G\sqrt{2T\log(\|u\|_2 \varepsilon^{-1})}$, c.f., Eq.(3). In Theorem 2, if we ignore the logarithmic residue $\log(\|u\|_2 \varepsilon^{-1})$ outside the square root,[11] then the

[11] Which does not depend on V_T.

leading term is $2\|u\|_2\sqrt{V_T\log(\|u\|\varepsilon^{-1})}$. In the worst case with $V_T=G^2T$, the constant of the latter has a $\sqrt{2}$ gap with respect to the lower bound. This is essentially due to our analysis, where α is picked as 1 instead of $1/2$ to handle the second case in the proof sketch (Section 3.2). One can use a smaller α (corresponding to smaller leading constant in the regret) in exchange for a larger z (the additive constant on V_T). However, achieving the lower bound $\sqrt{2}$ without blowing up the additive term remains to be studied in future works.

B.4 Application to dynamic regret

Given the improved static algorithm, we now apply its 1D version to the sparse coding framework. For clarity, we will adopt the asymptotic regret bound, Eq.(13), and loosely assimilate the residual term. Since it is applied on the transform domain, we will denote transform domain quantities with hat, according to our convention in Section 2.1. Then, analogous to Lemma A.4 applied in the main paper, our improved static algorithm, given an arbitrary hyperparameter $\hat{\varepsilon}>0$, guarantees for all $T\in\mathbb{N}_+$ and $\hat{u}\in\mathbb{R}$,

$$\sum_{t=1}^T\langle\hat{g}_t,\hat{x}_t-\hat{u}\rangle\leq\hat{\varepsilon}\left(\sqrt{\sum_{t=1}^T\hat{g}_t^2}+6\hat{G}\right)+|\hat{u}|\,O\left[\sqrt{\sum_{t=1}^T\hat{g}_t^2\log(|\hat{u}|\,\varepsilon^{-1})}\right],\tag{14}$$

where $\hat{G}$ is the Lipschitz constant for the surrogate loss $\hat{g}_t$.

In the sparse coding framework, we will assume the dictionary satisfies $\|h_t\|_2\leq 1$, which holds for Fourier and (un-normalized) wavelet dictionaries (Appendix A.4). Then, let us apply the improved static algorithm to a single feature vector (Algorithm 1). Note that instead of setting the surrogate Lipschitz constant as $G\|h_t\|_2$ (Line 2 of Algorithm 1), we now set it as G. The resulting dynamic regret bound is the following, which is analogous to Lemma 2.1.

Lemma B.7. *Let $\hat{\varepsilon}>0$ be an arbitrary hyperparameter for our improved static algorithm (the 1D version of Algorithm 4). Applying it as a subroutine, for all $T\in\mathbb{N}_+$ and $u_{1:T}\in\operatorname{span}(h_{1:T})$, Algorithm 1 guarantees*

$$\mathrm{Regret}_T(u_{1:T})\leq\varepsilon_T+\sqrt{\sum_{t=1}^T\langle g_t,u_t\rangle^2}\,O\left[\log\left(\varepsilon_T^{-1}\sqrt{\sum_{t=1}^T\langle g_t,u_t\rangle^2}\right)\right],$$

where

$$\varepsilon_T=\hat{\varepsilon}\left(\sqrt{\sum_{t=1}^T\langle g_t,h_t\rangle^2}+6G\right).$$

In particular, the big-Oh bound holds in two regimes: *(i)* $\sum_{t=1}^T\langle g_t,u_t\rangle^2\gg\hat{\varepsilon}^2\sum_{t=1}^T\langle g_t,h_t\rangle^2$ and $\sum_{t=1}^T\langle g_t,h_t\rangle^2\gg G^2$; and *(ii)* $u_{1:T}=0$. Compared to Lemma 2.1, the better underlying static algorithm essentially improves $G^2\|h_{1:T}\|_2^2$ in the dynamic regret bound with $\sum_{t=1}^T\langle g_t,h_t\rangle^2$, and $G^2\|u_{1:T}\|_2^2$ with $\sum_{t=1}^T\langle g_t,u_t\rangle^2$.

Proof of Lemma B.7. Similar to the proof of Lemma 2.1, the dynamic regret of Algorithm 1 equals Eq.(14) on the transform domain. In particular, $\hat{u}$ satisfies $u_{1:T}=\hat{u}h_{1:T}$. The surrogate Lipschitz constant $\hat{G}$ in Eq.(14)

equals G, the actual Lipschitz constant for the dynamic regret problem. With $\sum_{t=1}^T \langle g_t, h_t\rangle^2 \gg G^2$,

$$\begin{aligned}
\mathrm{Regret}_T(u_{1:T}) &\leq \hat{\varepsilon}\left(\sqrt{\sum_{t=1}^T \langle g_t, h_t\rangle^2} + 6G\right) + |\hat{u}|\, O\left[\sqrt{\sum_{t=1}^T \langle g_t, h_t\rangle^2 \log(|\hat{u}|\,\hat{\varepsilon}^{-1})}\right]\\
&= \varepsilon_T + |\hat{u}|\sqrt{\sum_{t=1}^T \langle g_t, h_t\rangle^2} O\left[\log\left(\varepsilon_T^{-1}\,|\hat{u}|\sqrt{\sum_{t=1}^T \langle g_t, h_t\rangle^2}\right)\right]\\
&= \varepsilon_T + \sqrt{\sum_{t=1}^T \langle g_t, |\hat{u}|\,|h_t|\rangle^2} O\left[\log\left(\varepsilon_T^{-1}\sqrt{\sum_{t=1}^T \langle g_t, |\hat{u}|\,|h_t|\rangle^2}\right)\right]\\
&= \varepsilon_T + \sqrt{\sum_{t=1}^T \langle g_t, u_t\rangle^2} O\left[\log\left(\varepsilon_T^{-1}\sqrt{\sum_{t=1}^T \langle g_t, u_t\rangle^2}\right)\right]. \qquad \square
\end{aligned}$$

Next, let us consider general size N dictionaries, analogous to Theorem 1. Still, we assume that for all t and n, $\|h_{t,n}\|_2 \leq 1$.

Lemma B.8. *Consider Algorithm 2, with its static subroutine replaced by the 1D version of Algorithm 4. For all $T \in \mathbb{N}_+$ and $u_{1:T} \in \mathbb{R}^{dT}$, it guarantees*

$$\mathrm{Regret}_T(u_{1:T}) \leq \mathcal{E}_T + U_T \cdot O\left(\log \frac{U_T}{\mathcal{E}_T} + \mathrm{KL}(q||\pi)\right) + G\sum_{t=1}^T \|u_{t,0}\|_2\,,$$

where

1. *For all $n \in [1:N]$, $u_{1:T,n}$ is any vector in $\mathrm{span}(h_{1:T,n})$, and $u_{1:T,0} = u_{1:T} - \sum_{n=1}^N u_{1:T,n}$;*
2. *$\mathcal{E}_T$ and U_T are non-negative numbers defined by*

$$\mathcal{E}_T = \sum_{n=1}^N \hat{\varepsilon}_n\left(\sqrt{\sum_{t=1}^T \langle g_t, h_{t,n}\rangle^2} + 6G\right),$$

$$U_T = \sum_{n=1}^N \sqrt{\sum_{t=1}^T \langle g_t, u_{t,n}\rangle^2};$$

3. *π and q are N dimensional probability vectors defined by*

$$\pi_n = \frac{\hat{\varepsilon}_n}{\mathcal{E}_T}\left(\sqrt{\sum_{t=1}^T \langle g_t, h_{t,n}\rangle^2} + 6G\right),$$

$$q_n = \frac{1}{U_T}\sqrt{\sum_{t=1}^T \langle g_t, u_{t,n}\rangle^2}.$$

If π is the uniform distribution, and $u_{1:T} \in \mathrm{span}(\mathcal{H})$, then the bound can be simplified into

$$\mathrm{Regret}_T(u_{1:T}) \leq \mathcal{E}_T + \tilde{O}(U_T).$$

Note that the big-Oh bound is meant for the regime where for all $n \in [1:N]$, either (i) $u_{1:T,n} = 0$; or (ii) $\sum_{t=1}^T \langle g_t, u_{t,n}\rangle^2 \gg \hat{\varepsilon}^2 \sum_{t=1}^T \langle g_t, h_{t,n}\rangle^2$ and $\sum_{t=1}^T \langle g_t, h_{t,n}\rangle^2 \gg G^2$. Compared to Theorem 1, we improve $G\sum_{n=1}^N \sqrt{\sum_{t=1}^T \|u_{t,n}\|_2}$ to $\sum_{n=1}^N \sqrt{\sum_{t=1}^T \langle g_t, u_{t,n}\rangle^2}$, which adapts to the complexity of both the gradient sequence and the comparator.

Proof of Lemma B.8. By summing Lemma B.7,

$$
\begin{aligned}
\mathrm{Regret}_T(u_{1:T}) &\le \sum_{n=1}^{N} \left\{ \hat{\varepsilon}_n \left(\sqrt{\sum_{t=1}^{T} \langle g_t, h_{t,n} \rangle^2} + 6G \right) + \sqrt{\sum_{t=1}^{T} \langle g_t, u_{t,n} \rangle^2} O \left[\log \left(\hat{\varepsilon}_n^{-1} \frac{\sqrt{\sum_{t=1}^{T} \langle g_t, u_{t,n} \rangle^2}}{\sqrt{\sum_{t=1}^{T} \langle g_t, h_{t,n} \rangle^2} + 6G} \right) \right] \right\} \\
&= \mathcal{E}_T + U_T \cdot O \left(\sum_{n=1}^{N} q_n \log \frac{q_n U_T}{\pi_n \mathcal{E}_T} \right) \\
&\le \mathcal{E}_T + U_T \cdot O \left(\log \frac{U_T}{\mathcal{E}_T} + \sum_{n=1}^{N} q_n \log \frac{q_n}{\pi_n} \right) \\
&= \mathcal{E}_T + U_T \cdot O \left(\log \frac{U_T}{\mathcal{E}_T} + \mathrm{KL}(q||\pi) \right). \qquad \square
\end{aligned}
$$