Universal Online Convex Optimization with 1 Projection per Round

Wenhao Yang[1,2], **Yibo Wang**[1,2], **Peng Zhao**[1,2], **Lijun Zhang**[1,2,*]
[1]National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
[2]School of Artificial Intelligence, Nanjing University, Nanjing, China
{yangwh, wangyb, zhaop, zhanglj}@lamda.nju.edu.cn

Abstract

To address the uncertainty in function types, recent progress in online convex optimization (OCO) has spurred the development of universal algorithms that simultaneously attain minimax rates for multiple types of convex functions. However, for a T-round online problem, state-of-the-art methods typically conduct $O(\log T)$ projections onto the domain in each round, a process potentially time-consuming with complicated feasible sets. In this paper, inspired by the black-box reduction of Cutkosky and Orabona [2018], we employ a surrogate loss defined over simpler domains to develop universal OCO algorithms that only require 1 projection. Embracing the framework of prediction with expert advice, we maintain a set of experts for each type of functions and aggregate their predictions via a meta-algorithm. The crux of our approach lies in a uniquely designed expert-loss for strongly convex functions, stemming from an innovative decomposition of the regret into the meta-regret and the expert-regret. Our analysis sheds new light on the surrogate loss, facilitating a rigorous examination of the discrepancy between the regret of the original loss and that of the surrogate loss, and carefully controlling meta-regret under the strong convexity condition. With only 1 projection per round, we establish optimal regret bounds for general convex, exponentially concave, and strongly convex functions simultaneously. Furthermore, we enhance the expert-loss to exploit the smoothness property, and demonstrate that our algorithm can attain small-loss regret for multiple types of convex and smooth functions.

1 Introduction

Online convex optimization (OCO) stands as a pivotal online learning framework for modeling many real-world problems [Hazan, 2016]. OCO is commonly formulated as a repeated game between the learner and the environment with the following protocol. In each round $t \in [T]$, the learner chooses a decision $\mathbf{x}_t$ from a convex domain $\mathcal{X} \subseteq \mathbb{R}^d$; after submitting this decision, the learner suffers a loss $f_t(\mathbf{x}_t)$, where $f_t \colon \mathcal{X} \mapsto \mathbb{R}$ is a convex function selected by the environment. The goal of the learner is to minimize the cumulative loss over T rounds, i.e., $\sum_{t=1}^{T} f_t(\mathbf{x}_t)$, and the standard performance measure is the *regret* [Cesa-Bianchi and Lugosi, 2006]:

$$\textsc{Reg}_T = \sum_{t=1}^{T} f_t(\mathbf{x}_t) - \min_{\mathbf{x}\in\mathcal{X}} \sum_{t=1}^{T} f_t(\mathbf{x}), \tag{1}$$

which quantifies the difference between the cumulative loss of the online learner and that of the best decision chosen in hindsight.

[*]Lijun Zhang is the corresponding author.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

Table 1: A summary of our universal algorithms and previous studies over T rounds d-dimensional functions, where L_T denotes the small-loss quantity. Abbreviations: cvx → convex, exp-concave → exponentially concave, str-cvx → strongly convex, # PROJ → number of projections per round.

Assumption	**Method**	**Regret Bounds**			**# PROJ**
		cvx	exp-concave	str-cvx	
	van Erven and Koolen [2016]	$O(\sqrt{T})$	$O(d \log T)$	$O(d \log T)$	$O(\log T)$
	Mhammedi et al. [2019]	$O(\sqrt{T})$	$O(d \log T)$	$O(d \log T)$	1
	Wang et al. [2019]	$O(\sqrt{T})$	$O(d \log T)$	$O(\log T)$	$O(\log T)$
	Zhang et al. [2022]	$O(\sqrt{T})$	$O(d \log T)$	$O(\log T)$	$O(\log T)$
	Theorem 1 of this work	$O(\sqrt{T})$	$O(d \log T)$	$O(\log T)$	1
$f_t(\cdot)$ is smooth	Wang et al. [2020b]	$O(\sqrt{L_T})$	$O(d \log L_T)$	$O(\log L_T)$	$O(\log T)$
	Zhang et al. [2022]	$O(\sqrt{L_T})$	$O(d \log L_T)$	$O(\log L_T)$	$O(\log T)$
	Theorem 2 of this work	$O(\sqrt{L_T})$	$O(d \log L_T)$	$O(\log L_T)$	1

Although there are plenty of algorithms to minimize the regret of convex functions, including general convex, exponentially concave (abbr. exp-concave) and strongly convex functions [Zinkevich, 2003, Shalev-Shwartz et al., 2007, Hazan et al., 2007], most of them can only handle one specific function type, and need to estimate the moduli of strong convexity and exp-concavity. The demand for prior knowledge motivates the development of *universal* algorithms for OCO, which aim to attain minimax optimal regret guarantees for multiple types of convex functions simultaneously [Bartlett et al., 2008, van Erven and Koolen, 2016, Wang et al., 2019, Mhammedi et al., 2019, Zhang et al., 2022]. State-of-the-art methods typically adopt a two-layer structure following the prediction with expert advice (PEA) framework [Cesa-Bianchi and Lugosi, 2006]. Specifically, they maintain $O(\log T)$ expert-algorithms with different configurations to handle the uncertainty of functions and deploy a meta-algorithm to track the best one. While this two-layer framework has demonstrated effectiveness in endowing algorithms with universality, it raises concerns regarding the computational efficiency. Since each expert-algorithm needs to execute one projection onto the feasible domain $\mathcal{X}$ per round, standard universal algorithms perform $O(\log T)$ projections in each round, which can be time-consuming in practical scenarios particularly when projecting onto complicated domains.

In the literature, there indeed exists an effort to reduce the number of projections required by universal algorithms tailored for *exp-concave functions* [Mhammedi et al., 2019]. This is achieved by applying the black-box reduction of Cutkosky and Orabona [2018], which reduces an OCO problem on the original (but can be complicated) feasible domain to a more manageable one on a simpler domain, such as an Euclidean ball. Deploying an existing universal algorithm [van Erven and Koolen, 2016] on the reduced problem enables us to attain optimal regret for exp-concave functions, crucially, with only *one* single projection per round and no prior knowledge of exp-concavity required. However, this black-box approach *cannot* be extended to strongly convex functions (see Section 3.1 for technical discussions). Therefore, it is still unclear on how to reduce the number of projections of universal algorithms to 1, and at the same time ensure optimal regret for strongly convex functions (as well as general convex and exp-concave functions).

In this paper, we affirmatively solve the above question by introducing an efficient universal OCO algorithm. Our solution employs the black-box reduction Cutkosky [2020] to cast the original problem on the constrained domain $\mathcal{X}$ to an alternative one in terms of the surrogate loss on a simpler domain $\mathcal{Y} \supseteq \mathcal{X}$. Specifically, we construct multiple experts updated in domain $\mathcal{Y}$, each optimizing a expert-loss specialized for a distinct function type. Then, we combine their predictions by a meta-algorithm, and perform the *only projection* onto the feasible domain $\mathcal{X}$. The meta-algorithm chooses the linearized surrogate loss to measure the performance of experts, and is required to yield a second-order regret [Zhang et al., 2022]. The key novelty of our algorithm lies in the uniquely designed *expert-loss for strongly convex functions*, which is motivated by an innovative decomposition of the regret into the meta-regret and the expert-regret. To effectively deal with strongly convex functions, we *explore the domain-converting surrogate loss in depth and illuminate its refined properties*. Our new insights tighten the regret gap in terms of original loss and surrogate loss, and further exploit strong convexity to compensate the meta-regret, thus achieving the optimal regret for strongly convex functions. Section 3.2 provides a formal description of our key ideas. With only 1 projection per round, our algorithm attains $O(\sqrt{T})$, $O(\frac{d}{\alpha}\log T)$, and $O(\frac{1}{\lambda}\log T)$ regret for general convex, α-exp-concave, and λ-strongly convex functions, respectively.

We further establish *small-loss regret* for universal OCO with *smooth* functions. The small-loss quantity $L_T = \min_{\mathbf{x}\in\mathcal{X}}\sum_{t=1}^{T} f_t(\mathbf{x})$ is defined as the cumulative loss of the best decision chosen from the domain $\mathcal{X}$, which is at most $O(T)$ under standard OCO assumptions and meanwhile can be much smaller in benign environments. To achieve small-loss regret bounds, we design an enhanced expert-loss for smooth and strongly convex functions and integrate it into our two-layer framework, which finally leads to a universal OCO algorithm achieving $O(\sqrt{L_T})$, $O(\frac{d}{\alpha}\log L_T)$, and $O(\frac{1}{\lambda}\log L_T)$ small-loss regret for three types of convex functions, respectively. Notably, all those bounds are *optimal* and the algorithm only requires *one* projection per iteration. We summarize our results and compare with previous studies of universal algorithms in Table 1.

Organization. The rest of the paper is organized as follows. Section 2 presents the preliminaries and reviews several mostly related works. Section 3 illuminates the technical challenges and describes our key ideas. Section 4 provides the overall algorithms and regret analysis. We finally conclude the paper in Section 5. All the proofs and omitted details are deferred to appendices.

2 Preliminaries and related works

In this section, we first present preliminaries for OCO, and then review several most related works to our paper, including universal algorithms and projection-efficient algorithms.

2.1 Preliminaries

We introduce two typical assumptions of online convex optimization [Hazan, 2016].

Assumption 1 (bounded domain) *The feasible domain $\mathcal{X} \subseteq \mathbb{R}^d$ contains the origin $\mathbf{0}$, and the diameter is bounded by D, i.e., $\|\mathbf{x}-\mathbf{y}\| \le D$ holds for any $\mathbf{x},\mathbf{y}\in\mathcal{X}$.*

Assumption 2 (bounded gradient norms) *The norm of the gradients of all online functions over the domain $\mathcal{X}$ is bounded by G, i.e., $\|\nabla f_t(\mathbf{x})\| \le G$ holds for all $\mathbf{x}\in\mathcal{X}$ and $t\in[T]$.*

Throughout the paper we use $\|\cdot\|$ for ℓ_2-norm in default. Owing to Assumption 1, we can always construct an Euclidean ball $\mathcal{Y} = \{\mathbf{x} \mid \|\mathbf{x}\| \le D\}$ containing the original feasible domain $\mathcal{X}$.

Next, we state definitions of strong convexity and exp-concavity [Hazan, 2016], and introduce an important property of exp-concave functions [Hazan et al., 2007, Lemma 3].

Definition 1 (strongly convex functions) *A function $f : \mathcal{X} \mapsto \mathbb{R}$ is called λ-strongly convex, if the condition $f(\mathbf{y}) \ge f(\mathbf{x}) + \langle\nabla f(\mathbf{x}), \mathbf{y}-\mathbf{x}\rangle + \frac{\lambda}{2}\|\mathbf{y}-\mathbf{x}\|^2$ holds for all $\mathbf{x},\mathbf{y}\in\mathcal{X}$.*

Definition 2 (exponentially-concave functions) *A function $f : \mathcal{X} \mapsto \mathbb{R}$ is called α-exponentially-concave, if the function $\exp(-\alpha f(\cdot))$ is concave over the feasible domain $\mathcal{X}$.*

Lemma 1 *For an α-exp-concave function $f : \mathcal{X} \mapsto \mathbb{R}$, if the feasible domain $\mathcal{X}$ has a diameter D and $\|\nabla f(\mathbf{x})\| \le G$ holds for $\forall\mathbf{x}\in\mathcal{X}$, then we have*

$$f(\mathbf{y}) \ge f(\mathbf{x}) + \langle\nabla f(\mathbf{x}), \mathbf{y}-\mathbf{x}\rangle + \frac{\beta}{2}\langle\nabla f(\mathbf{x}), \mathbf{y}-\mathbf{x}\rangle^2, \tag{2}$$

for all $\mathbf{x},\mathbf{y}\in\mathcal{X}$, where $\beta = \frac{1}{2}\min\{\frac{1}{4GD}, \alpha\}$.

There are many efforts devoted to minimizing regret, including general convex, α-exp-concave, and λ-strongly convex functions. For general convex functions, online gradient descent (OGD) with step size $\eta_t = O(1/\sqrt{t})$, attains an $O(\sqrt{T})$ regret [Zinkevich, 2003]. For α-exp-concave functions, online Newton step (ONS) is equipped with an $O(\frac{d}{\alpha}\log T)$ regret [Hazan et al., 2007]. For λ-strongly convex functions, OGD with step size $\eta_t = O(1/[\lambda t])$, achieves an $O(\frac{1}{\lambda}\log T)$ regret [Shalev-Shwartz et al., 2007]. These regret bounds are proved to be minimax optimal [Ordentlich and Cover, 1998, Abernethy et al., 2008]. Furthermore, tighter bounds are attainable when the loss functions enjoy additional properties, such as smoothness [Shalev-Shwartz, 2007, Luo and Schapire, 2015, Srebro et al., 2010, Orabona et al., 2012, Chiang et al., 2012, Yang et al., 2014, Mohri and Yang, 2016, Zhang et al., 2019, Zhao et al., 2020, 2024, Chen et al., 2024] and sparsity of gradients [Duchi et al., 2010, Tieleman and Hinton, 2012, Mukkamala and Hein, 2017, Kingma and Ba, 2015, Reddi et al., 2018, Loshchilov and Hutter, 2019, Wang et al., 2020a]. We discuss *small-loss* regret below.

For general convex and smooth functions, Srebro et al. [2010] prove that OGD with constant step size attains an $O(\sqrt{L})$ regret bound, where L is the upper bound of L_T. The limitation of their method is that it requires to know L beforehand. To address this limitation, Zhang et al. [2019] propose scale-free online gradient descent (SOGD), which is a special case of scale-free mirror descent algorithm [Orabona and Pál, 2018], and establish an $O(\sqrt{L_T})$ small-loss regret bound without the prior knowledge of L_T. For α-exp-concave and smooth functions, ONS attains an $O(\frac{d}{\alpha}\log L_T)$ small-loss regret bound [Orabona et al., 2012]. For λ-strongly convex and smooth functions, a variant of OGD, namely S^{2}OGD, is introduced to achieve an $O(\frac{1}{\lambda}\log L_T)$ small-loss regret bound [Wang et al., 2020b]. Such bounds reduce to the minimax optimal bounds in the worst case, but could be much tighter when the comparator has a small loss, i.e., L_T is small.

2.2 Universal algorithms

Most existing online algorithms can only handle one type of convex function and need to know the moduli of strong convexity and exp-concavity beforehand. Universal online learning aims to remove such requirements of domain knowledge. The first universal OCO algorithm is adaptive online gradient descent (AOGD) [Bartlett et al., 2008], which achieves $O(\sqrt{T})$ and $O(\log T)$ regret bounds for general convex and strongly convex functions, respectively. However, the algorithm still needs to know the modulus of strong convexity and does not support exp-concave functions.

An important milestone is the multiple eta gradient (MetaGrad) algorithm [van Erven and Koolen, 2016], which adapt to general convex and exp-concave functions without knowing the modulus of exp-concavity. MetaGrad constructs multiple expert-algorithms with various learning rates and combines their predictions by a meta-algorithm called Tilted Exponentially Weighted Average (TEWA). To avoid prior knowledge, each expert minimizes the expert-loss parameterized by a learning rate η,

$$\ell_{t,\eta}^{\exp}(\mathbf{x}) = -\eta\langle\nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}\rangle + \eta^2\langle\nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}\rangle^2. \tag{3}$$

MetaGrad maintains $O(\log T)$ experts to minimize (3), and attains $O(\sqrt{T\log\log T})$ and $O(\frac{d}{\alpha}\log T)$ regret for general convex and α-exp-concave functions, respectively. To further support strongly convex functions, Wang et al. [2019] propose a new type of expert-losses defined as

$$\ell_{t,\eta}^{\mathrm{sc}}(\mathbf{x}) = -\eta\langle\nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}\rangle + \eta^2 G^2\|\mathbf{x}_t - \mathbf{x}\|^2 \tag{4}$$

where G is the gradient norm upper bound, and introduce an expert-loss for general convex functions

$$\ell_{t,\eta}^{\mathrm{cvx}}(\mathbf{x}) = -\eta\langle\nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}\rangle + \eta^2 G^2 D^2 \tag{5}$$

where D is the upper bound of the diameter of $\mathcal{X}$. Their algorithm, named as Maler, obtains $O(\sqrt{T})$, $O(\frac{1}{\lambda}\log T)$ and $O(\frac{d}{\alpha}\log T)$ regret for general convex, λ-strongly convex functions, and α-exp-concave functions, respectively. Later, Wang et al. [2020b] extend Maler by replacing G^2 in (4) and (5) with $\|\nabla f_t(\mathbf{x}_t)\|^2$, thereby enabling their algorithm to deliver small-loss regret bounds. Under the smoothness condition, their algorithm achieves $O(\sqrt{L_T})$, $O(\frac{1}{\lambda}\log L_T)$ and $O(\frac{d}{\alpha}\log L_T)$ regret for general convex, λ-strongly convex, and α-exp-concave functions, respectively.

MetaGrad and its variants require the carefully designed expert-losses. Zhang et al. [2022] propose a different universal strategy that avoids the construction of losses. The basic idea is to let each expert handle original functions and deploy a meta-algorithm over *linearized loss*. Importantly, the meta-algorithm is required to yield a second-order regret [Gaillard et al., 2014] to exploit strong convexity and exp-concavity. By incorporating existing online algorithms as experts, their approach inherits the regret of any expert designed for strongly convex functions and exp-concave functions, and also obtains minimax optimal regret (and small-loss regret) for general convex functions.

Although state-of-the-art universal algorithms can adapt to multiple function types, they create $O(\log T)$ experts per round. As a result, they need to perform $O(\log T)$ projections in each round, which can be time-consuming in practical scenarios with complicated domains. To address this limitation, we aim to develop projection-efficient algorithms for universal OCO.

2.3 Projection-efficient algorithms

In the studies of parameter-free online learning, Cutkosky and Orabona [2018] propose a black-box reduction technique from constrained online learning to unconstrained online learning. To avoid regret degeneration, they design the *domain-converting surrogate loss* $\widehat{g}_t : \mathcal{Y} \mapsto \mathbb{R}$ defined as,

$$\widehat{g}_t(\mathbf{y}) = \langle \nabla f_t(\mathbf{x}_t), \mathbf{y} \rangle + \|\nabla f_t(\mathbf{x}_t)\| \cdot S_{\mathcal{X}}(\mathbf{y}) \tag{6}$$

where $S_{\mathcal{X}}(\mathbf{y}) = \|\mathbf{y} - \Pi_{\mathcal{X}}[\mathbf{y}]\|$ is the distance function to the feasible domain $\mathcal{X}$. Then, we can employ an unconstrained online learning algorithm that minimizes (6) to obtain the prediction $\mathbf{y}_t$, and output its prediction on domain $\mathcal{X}$, i.e., $\mathbf{x}_t = \Pi_{\mathcal{X}}[\mathbf{y}_t]$. Cutkosky and Orabona [2018, Theorem 3] have proved that the above surrogate loss satisfies $\|\nabla \widehat{g}_t(\mathbf{y}_t)\| \leq \|\nabla f_t(\mathbf{x}_t)\|$, and

$$\langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x} \rangle \leq 2\big(\widehat{g}_t(\mathbf{y}_t) - \widehat{g}_t(\mathbf{x})\big) \leq 2\langle \nabla \widehat{g}_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{x} \rangle \tag{7}$$

for all $t \in [T]$ and any $\mathbf{x} \in \mathcal{X}$. Based on this fact, we know that the regret of the unconstrained problem directly serves as an upper bound for that of the original problem, hence reducing the original problem to an unconstrained surrogate problem and retaining the order of regret.

Subsequently, Cutkosky [2020] introduces a new surrogate loss $g_t : \mathcal{Y} \mapsto \mathbb{R}$ defined as,

$$g_t(\mathbf{y}) = \langle \nabla f_t(\mathbf{x}_t), \mathbf{y} \rangle - \mathbb{1}_{\{\langle \nabla f_t(\mathbf{x}_t), \mathbf{v}_t \rangle < 0\}} \langle \nabla f_t(\mathbf{x}_t), \mathbf{v}_t \rangle \cdot S_{\mathcal{X}}(\mathbf{y}) \tag{8}$$

where $\mathbf{v}_t = \frac{\mathbf{y}_t - \mathbf{x}_t}{\|\mathbf{y}_t - \mathbf{x}_t\|}$ is the unit vector of the projection direction. As depicted in the following lemma, this surrogate loss avoids the multiplicative constant 2 on the right-hand side of (7).

Lemma 2 (Theorem 2 of Cutkosky [2020]) *The function defined in* (8) *is convex, and it satisfies* $\|\nabla g_t(\mathbf{y}_t)\| \leq \|\nabla f_t(\mathbf{x}_t)\|$. *Furthermore, for all t and all* $\mathbf{x} \in \mathcal{X}$, *we have*

$$\langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x} \rangle \leq g_t(\mathbf{y}_t) - g_t(\mathbf{x}) \leq \langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{x} \rangle. \tag{9}$$

While the black-box reduction is proposed for the constrained-to-unconstrained conversion, it also facilitates the conversion to another constrained problem (i.e., $\mathcal{Y} \neq \mathbb{R}^d$). This enables us to transform OCO problem on a complicated domain into another on simpler domains such that the projection is much easier. Building on this idea, Mhammedi et al. [2019] introduce an efficient implementation of MetaGrad [van Erven and Koolen, 2016], which only conducts 1 projection onto the original domain in each round, and keeps the order of regret bounds. However, as detailed in the following section, the black-box reduction does not adequately extend to strongly convex functions. We also mention that Zhao et al. [2022] recently employ the technique to non-stationary OCO with non-trivial modifications to develop efficient algorithms for minimizing dynamic regret and adaptive regret. However, they focus on the convex functions and do not involve the considerations of exp-concave and strongly convex functions as concerned in our paper.

3 Technical challenge and our key ideas

In this section, we elaborate on the technical challenges and our key ideas.

3.1 Technical challenge

As mentioned, Mhammedi et al. [2019] exploit the black-box reduction scheme of [Cutkosky and Orabona, 2018] to improve the projection efficiency of MetaGrad [van Erven and Koolen, 2016]. We summarize their algorithm in Algorithm 1. In the following, we will demonstrate its effectiveness for exp-concave functions and explain why it fails for strongly convex functions.

Success in exp-concave functions. By applying the black-box reduction as described in Section 2.3, Mhammedi et al. [2019] utilize MetaGrad to minimize the surrogate loss $\widehat{g}_t(\cdot)$ in (6) over an Euclidean ball $\mathcal{Y}$. The projection operations inside MetaGrad are over $\mathcal{Y}$ and thus negligible. Notice that Algorithm 1 demands only 1 projection onto $\mathcal{X}$ in Step 4. According to regret bound of MetaGrad, Algorithm 1 enjoys a second-order bound [Mhammedi et al., 2019, Theorem 10],

$$\sum_{t=1}^{T} \langle \nabla \widehat{g}_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{x} \rangle \leq O\left(\sqrt{d \log T \cdot \sum_{t=1}^{T} \langle \nabla \widehat{g}_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{x} \rangle^2} + d \log T\right). \tag{10}$$

Algorithm 1 Black-box reduction for projection-efficient MetaGrad [Mhammedi et al., 2019]

```
Construct a ball domain $\mathcal{Y} = \{\mathbf{x} \mid \|\mathbf{x}\| \leq D\} \supseteq \mathcal{X}$
for $t = 1$ to $T$ do
    Receive the decision $\mathbf{y}_t \in \mathcal{Y}$ from MetaGrad
    Submit the decision $\mathbf{x}_t = \Pi_{\mathcal{X}}[\mathbf{y}_t]$    ▷ The only step projects onto domain $\mathcal{X}$ per round.
    Suffer the loss $f_t(\mathbf{x}_t)$ and observe the gradient $\nabla f_t(\mathbf{x}_t)$
    Construct the surrogate loss $\widehat{g}_t(\cdot)$ as (6) and send it to MetaGrad
end for
```

The above bound is measured by surrogate loss, thus requiring a further analysis that converts it back to that of the original function. Since $\beta = \frac{1}{2}\min\left\{\frac{1}{4GD}, \alpha\right\}$, the function $x - \beta x^2$ is strictly increasing when $x \in (-\infty, 2GD]$. Therefore, the property of surrogate loss $\widehat{g}_t(\cdot)$ in (7) implies

$$\frac{1}{2}\langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}\rangle - \frac{\beta}{4}\langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}\rangle^2 \leq \langle \nabla \widehat{g}_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{x}\rangle - \beta\langle \nabla \widehat{g}_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{x}\rangle^2. \quad (11)$$

Combining (10) with (11) and applying the AM-GM inequality, we obtain

$$\sum_{t=1}^{T}\langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}\rangle - \frac{\beta}{2}\sum_{t=1}^{T}\langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}\rangle^2 \leq O\left(\frac{d}{\alpha}\log T\right)$$

thus achieving the optimal regret based on Lemma 1.

Failure in strongly convex functions. To handle strongly convex functions, a straightforward way is to use a universal algorithm that supports strongly convex functions, such as Maler [Wang et al., 2019], as the black-box subroutine in Algorithm 1. However, for strongly convex functions, the above analysis cannot be applied, and we are unable to derive a tight regret bound. Specifically, according to the theoretical guarantee of Maler [Wang et al., 2019, Theorem 1], we have

$$\sum_{t=1}^{T}\langle \nabla \widehat{g}_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{x}\rangle \leq O\left(\sqrt{\log T \cdot \sum_{t=1}^{T}\|\mathbf{y}_t - \mathbf{x}\|^2} + \log T\right). \quad (12)$$

From the standard black-box analysis and the definition of strong convexity, we know

$$\sum_{t=1}^{T} f_t(\mathbf{x}_t) - \sum_{t=1}^{T} f_t(\mathbf{x}) \overset{(7)}{\leq} \sum_{t=1}^{T} 2\langle \nabla \widehat{g}_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{x}\rangle - \frac{\lambda}{2}\sum_{t=1}^{T}\|\mathbf{x}_t - \mathbf{x}\|^2. \quad (13)$$

Substituting (12) into (13), we encounter an $\widetilde{O}(\sqrt{\sum_{t=1}^{T}\|\mathbf{y}_t - \mathbf{x}\|^2} - \frac{\lambda}{2}\sum_{t=1}^{T}\|\mathbf{x}_t - \mathbf{x}\|^2)$ term, which is unmanageable since $\|\mathbf{y}_t - \mathbf{x}\| \geq \|\mathbf{x}_t - \mathbf{x}\|$. Here, $\widetilde{O}(\cdot)$ further omits the ploy$(\log T)$ factors.

3.2 Key ideas

To address above challenges, we introduce novel ideas in both algorithm design and regret analysis.

Algorithm design. Our algorithm is still in a two-layer structure. The main contribution lies in a uniquely designed *expert-loss for strongly convex functions*. For simplicity, we consider that the modulus of strong convexity λ is known for a moment, and define

$$\ell_t^{\text{sc}}(\mathbf{y}) = \langle \nabla g_t(\mathbf{y}_t), \mathbf{y}\rangle + \frac{\lambda}{2}\|\mathbf{y} - \mathbf{x}_t\|^2, \quad (14)$$

where $g_t(\cdot)$ is the surrogate loss defined in (8). Let us compare our designed expert-loss (14) with the one when applying existing universal algorithms in a black-box manner. Suppose Maler [Wang et al., 2019] is used, their expert-loss construction (4) indicates that the algorithm over domain $\mathcal{Y}$ essentially optimizes the expert-loss formulated as (up to constant factors).

$$\widehat{\ell}_t^{\text{sc}}(\mathbf{y}) = \langle \nabla g_t(\mathbf{y}_t), \mathbf{y}\rangle + \frac{\lambda}{2}\|\mathbf{y} - \mathbf{y}_t\|^2 \quad (15)$$

An important caveat is that our expert-loss (14) evaluates the performance of the expert (associated with strongly convex functions) based on the distance between its output $\mathbf{y}$ and the *actual* decision $\mathbf{x}_t \in \mathcal{X}$, as opposed to the unprojected intermediate one $\mathbf{y}_t \in \mathcal{Y}$ in (15).

In fact, this design of expert-loss (14) stems from a novel regret decomposition as explained below. First, by strong convexity of f_t and the property of the domain-converting surrogate loss, we have

$$\begin{aligned} \sum_{t=1}^T f_t(\mathbf{x}_t) - \sum_{t=1}^T f_t(\mathbf{x}) &\overset{(9)}{\leq} \sum_{t=1}^T \langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{x}\rangle - \frac{\lambda}{2}\sum_{t=1}^T \|\mathbf{x}_t - \mathbf{x}\|^2 \\ = \sum_{t=1}^T \langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{y}_t^i\rangle &+ \sum_{t=1}^T \langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t^i - \mathbf{x}\rangle - \frac{\lambda}{2}\sum_{t=1}^T \|\mathbf{x}_t - \mathbf{x}\|^2 \end{aligned} \tag{16}$$

where $\mathbf{y}_t^i$ denotes the decision of the i-th expert. The first term of the above bound is the meta-regret in terms of linearized surrogate loss. Then, we reformulate the remaining two terms as follows

$$\begin{aligned} \sum_{t=1}^T \langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t^i - \mathbf{x}\rangle - \frac{\lambda}{2}\sum_{t=1}^T \|\mathbf{x}_t - \mathbf{x}\|^2 &= \sum_{t=1}^T \left(\langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t^i\rangle + \frac{\lambda}{2}\|\mathbf{x}_t - \mathbf{y}_t^i\|^2\right) \\ - \sum_{t=1}^T \left(\langle \nabla g_t(\mathbf{y}_t), \mathbf{x}\rangle + \frac{\lambda}{2}\|\mathbf{x}_t - \mathbf{x}\|^2\right) - \frac{\lambda}{2}\sum_{t=1}^T \|\mathbf{x}_t - \mathbf{y}_t^i\|^2, \end{aligned} \tag{17}$$

where the expert-loss in (14) naturally arises. Combining (16) with (17), we arrive at

$$\sum_{t=1}^T f_t(\mathbf{x}_t) - \sum_{t=1}^T f_t(\mathbf{x}) \leq \underbrace{\sum_{t=1}^T \left(\ell_t^{\mathrm{sc}}(\mathbf{y}_t^i) - \ell_t^{\mathrm{sc}}(\mathbf{x})\right)}_{\texttt{expert-regret}} + \underbrace{\sum_{t=1}^T \langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{y}_t^i\rangle}_{\texttt{meta-regret}} - \frac{\lambda}{2}\sum_{t=1}^T \|\mathbf{x}_t - \mathbf{y}_t^i\|^2. \tag{18}$$

Theoretical analysis. For the expert-regret, since expert-loss (14) is λ-strongly convex and its gradients are bounded (see Lemma 6), we can use OGD to achieve an optimal $O(\frac{1}{\lambda}\log T)$ regret. Following Zhang et al. [2022], we require the meta-algorithm to yield a second-order regret bound

$$\sum_{t=1}^T \langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{y}_t^i\rangle \leq O\left(\sqrt{\sum_{t=1}^T \langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{y}_t^i\rangle^2}\right). \tag{19}$$

Notably, the upper bound of (19) and the negative term in (18) cannot be canceled due to the dismatch between $\mathbf{y}_t - \mathbf{y}_t^i$ and $\mathbf{x}_t - \mathbf{y}_t^i$. To resolve this discrepancy, we demonstrate that the surrogate loss defined in (8) enjoys the following two important improved properties.

Lemma 3 *In addition to enjoying all the properties outlined in Lemma 2, the surrogate loss function $g_t : \mathcal{Y} \mapsto \mathbb{R}$ defined in* (8) *satisfies*

$$\langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}\rangle \leq \langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{x}\rangle - \mathbb{1}_{\{\langle \nabla f_t(\mathbf{x}_t), \mathbf{v}_t\rangle \geq 0\}} \cdot \langle \nabla f_t(\mathbf{x}_t), \mathbf{y}_t - \mathbf{x}_t\rangle, \tag{20}$$

for all t and all $\mathbf{x} \in \mathcal{X}$. Furthermore, we also have

$$\begin{cases} \langle \nabla g_t(\mathbf{y}_t), \mathbf{x}_t - \mathbf{y}_t\rangle = 0, & \textit{when } \langle \nabla f_t(\mathbf{x}_t), \mathbf{v}_t\rangle < 0, \\ \langle \nabla g_t(\mathbf{y}_t), \mathbf{x}_t - \mathbf{y}_t\rangle \leq 0, & \textit{otherwise.} \end{cases} \tag{21}$$

Remark 1 We highlight the improvements of Lemma 3 over Lemma 2. First, we provide a tighter connection between the linearized original function and the surrogate loss in (20). Second, we analyze the difference between the actual decision $\mathbf{x}_t$ and the intermediate decision $\mathbf{y}_t$, along the direction $\nabla g_t(\mathbf{y}_t)$ in (21). As shown later, both of them are crucial for controlling the meta-regret. ◁

Utilizing (20) in Lemma 3, we refine the decomposition in (18) to establish a tighter bound

$$\sum_{t=1}^T f_t(\mathbf{x}_t) - \sum_{t=1}^T f_t(\mathbf{x}) \overset{(16),(17),(20)}{\leq} \mathrm{ER}(T) + \sum_{t=1}^T \langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{y}_t^i\rangle - \frac{\lambda}{2}\sum_{t=1}^T \|\mathbf{x}_t - \mathbf{y}_t^i\|^2 - \Delta_T \tag{22}$$

where $\mathrm{ER}(T)$ is the expert-regret, and $\Delta_T = \sum_{t=1}^T \mathbb{1}_{\{\langle \nabla f_t(\mathbf{x}_t), \mathbf{v}_t \rangle \geq 0\}} \cdot \langle \nabla f_t(\mathbf{x}_t), \mathbf{y}_t - \mathbf{x}_t \rangle \geq 0$ is the negative term introduced in the surrogate loss. Compared to (18), the new upper bound (22) enjoys an additional negative term $-\Delta_T$, which is essential to achieve a favorable regret bound in the analysis.

To utilize the negative quadratic term $-\frac{\lambda}{2} \sum_{t=1}^T \|\mathbf{x}_t - \mathbf{y}_t^i\|^2$ in (22) for compensating the second-order bound in (19), we need to convert $\mathbf{y}_t$ to $\mathbf{x}_t$, a place where (21) comes into play. From (19) and (21), we prove that for any $\gamma \in (0, \frac{G}{2D}]$ it holds that (see Lemma 8 for details):

$$\sum_{t=1}^T \langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{y}_t^i \rangle \leq O\left(\frac{G^2}{2\gamma}\right) + \frac{\gamma}{2G^2} \sum_{t=1}^T \langle \nabla g_t(\mathbf{y}_t), \mathbf{x}_t - \mathbf{y}_t^i \rangle^2 + \Delta_T. \tag{23}$$

Substituting (23) into (22), the additional term Δ_T is automatically *canceled out*, and we have

$$\begin{aligned}\sum_{t=1}^T f_t(\mathbf{x}_t) - \sum_{t=1}^T f_t(\mathbf{x}) &\leq \mathrm{ER}(T) + O\left(\frac{G^2}{2\gamma}\right) + \frac{\gamma}{2G^2} \sum_{t=1}^T \langle \nabla g_t(\mathbf{y}_t), \mathbf{x}_t - \mathbf{y}_t^i \rangle^2 - \frac{\lambda}{2} \sum_{t=1}^T \|\mathbf{x}_t - \mathbf{y}_t^i\|^2 \\ &\leq \mathrm{ER}(T) + O\left(\frac{G^2}{2\gamma}\right) + \left(\frac{\gamma}{2} - \frac{\lambda}{2}\right) \sum_{t=1}^T \|\mathbf{x}_t - \mathbf{y}_t^i\|^2 = O\left(\frac{1}{\lambda} \log T\right)\end{aligned}$$

where the final regret bound is because we set $\gamma = \min\{\frac{G}{2D}, \lambda\}$.

Remark 2 Section 2.3 describes two kinds of surrogate loss, as specified in (6) and (8). Indeed, they *both* are suitable for parameter-free online learning [Cutkosky, 2020] and non-stationary online learning [Zhao et al., 2022]. However, it is essential to adopt the new surrogate loss in our purpose: as established in Lemma 3, both negative terms and the mild difference between $\mathbf{x}_t$ and $\mathbf{y}_t$ are exploited in our regret analysis. By contrast, the old surrogate loss (6) lacks these advanced properties. ◁

4 Efficient algorithm for universal online convex optimization

In this section, we present our efficient algorithms for universal OCO. To reduce the cost of projections, we deploy multiple experts on a ball $\mathcal{Y} = \{\mathbf{x} \mid \|\mathbf{x}\| \leq D\}$ enclosing domain $\mathcal{X}$. After combining their decisions, we project the solution in $\mathcal{Y}$ onto $\mathcal{X}$, which is the only projection onto $\mathcal{X}$ per round.

4.1 Efficient algorithm for minimax universal regret

To handle unknown parameters of strong convexity and exp-concavity, we construct two finite sets, i.e., $\mathcal{P}_{\text{sc}}$ and $\mathcal{P}_{\text{exp}}$, to approximate their values [Zhang et al., 2022]. Taking λ-strongly convex functions as an example, we assume the unknown modulus λ is bounded by $\lambda \in [1/T, 1]$[2], and set $\mathcal{P}_{\text{sc}} = \{1/T, 2/T, \cdots, 2^N/T\}$, where $N = \lceil \log_2 T \rceil$. In this way, for any $\lambda \in [1/T, 1]$, there exists a $\widehat{\lambda} \in \mathcal{P}_{\text{sc}}$ such that $\widehat{\lambda} \leq \lambda \leq 2\widehat{\lambda}$. Moreover, we design three types of expert-losses. For general convex functions, we construct the expert-loss as

$$\ell_t^{\text{cvx}}(\mathbf{y}) = \langle \nabla g_t(\mathbf{y}_t), \mathbf{y} - \mathbf{y}_t \rangle, \tag{24}$$

where $g_t(\mathbf{y})$ is defined in (8). Since $\ell_t^{\text{cvx}}(\cdot)$ is convex, we use OGD as the expert-algorithm to minimize it. To handle exp-concave functions, we construct the expert-loss for each $\widehat{\alpha} \in \mathcal{P}_{\text{exp}}$ as

$$\ell_{t,\widehat{\alpha}}^{\text{exp}}(\mathbf{y}) = \langle \nabla g_t(\mathbf{y}_t), \mathbf{y} - \mathbf{y}_t \rangle + \frac{\widehat{\beta}}{2} \langle \nabla g_t(\mathbf{y}_t), \mathbf{y} - \mathbf{y}_t \rangle^2, \tag{25}$$

where $\widehat{\beta} = \frac{1}{2} \min\{\frac{1}{4GD}, \widehat{\alpha}\}$. It is easy to verify that $\ell_{t,\widehat{\alpha}}^{\text{exp}}(\cdot)$ is $\frac{\widehat{\beta}}{4}$-exp-concave, so we use ONS as the expert-algorithm. For strongly convex functions, we construct the expert-loss for each $\widehat{\lambda} \in \mathcal{P}_{\text{sc}}$ as

$$\ell_{t,\widehat{\lambda}}^{\text{sc}}(\mathbf{y}) = \langle \nabla g_t(\mathbf{y}_t), \mathbf{y} - \mathbf{y}_t \rangle + \frac{\widehat{\lambda}}{2} \|\mathbf{y} - \mathbf{x}_t\|^2. \tag{26}$$

Since $\ell_{t,\widehat{\lambda}}^{\text{sc}}(\cdot)$ is $\widehat{\lambda}$-strongly convex, we use OGD with step size $\eta_t = 1/[\widehat{\lambda} t]$ as the expert-algorithm. Finally, we deploy a meta-algorithm to track the best expert on the fly. Following Zhang et al.

Algorithm 2 Efficient Algorithm for Universal OCO

```
1: **Input:** The modulus set $\mathcal{P}_{\text{sc}}$ and $\mathcal{P}_{\text{exp}}$, the expert set $\mathcal{A} = \emptyset$, the number of experts $k = 0$
2: $k \leftarrow k + 1$, create an expert $E^1$ by running OGD with loss (24) over $\mathcal{Y}$
3: **for all** $\widehat{\alpha} \in \mathcal{P}_{\text{exp}}$ **do**
4:     $k \leftarrow k + 1$, create an expert $E^k$ by running ONS with loss (25) and parameter $\widehat{\alpha}$ over $\mathcal{Y}$
5: **end for**
6: **for all** $\widehat{\lambda} \in \mathcal{P}_{\text{sc}}$ **do**
7:     $k \leftarrow k + 1$, create an expert $E^k$ by running OGD with loss (26) and parameter $\widehat{\lambda}$ over $\mathcal{Y}$
8: **end for**
9: Add all the experts to the set: $\mathcal{A} = \{E^1, E^2, \cdots, E^k\}$
10: **for** $t = 1$ **to** $T$ **do**
11:     Compute the weight $p_t^i$ of each expert $E^i$ by (27)
12:     Receive the decision $\mathbf{y}_t^i$ from each expert $E^i$ in $\mathcal{A}$
13:     Aggregate all the decisions by $\mathbf{y}_t = \sum_{i=1}^{|\mathcal{A}|} p_t^i \mathbf{y}_t^i$
14:     Submit the decision $\mathbf{x}_t = \Pi_{\mathcal{X}}[\mathbf{y}_t]$          ▷ The only step projects onto domain $\mathcal{X}$ per round.
15:     Suffer the loss $f_t(\mathbf{x}_t)$ and observe the gradient $\nabla f_t(\mathbf{x}_t)$
16:     Construct the expert-loss $\ell_t^{\text{cvx}}(\cdot)$, $\ell_t^{\text{sc}}(\cdot)$ or $\ell_t^{\text{exp}}(\cdot)$ and sent it to corresponding expert in $\mathcal{A}$
17: **end for**
```

[2022], we use the linearized surrogate loss to measure the performance of the experts, and choose Adapt-ML-Prod [Gaillard et al., 2014] as the meta-algorithm to yield a second-order bound.

Our efficient algorithm for universal OCO is summarized in Algorithm 2. From Steps 2 to 9, it creates a set of experts by running multiple online algorithms over the ball $\mathcal{Y}$, each specialized for a distinct function type. Then, it maintains a set $\mathcal{A}$ consisting of all experts, and the i-th expert is denoted by E^i. In the t-th round, it computes the weight p_t^i of each expert E^i in Step 11 according to Adapt-ML-Prod. After receiving all the predictions from the experts in Step 12, it aggregates them based on their weights to attain $\mathbf{y}_t$ in Step 13. Next, it conducts the *only* projection onto the original domain $\mathcal{X}$ to obtain the actual decision $\mathbf{x}_t$ in Step 14. In Step 15, it evaluates the gradient $\nabla f_t(\mathbf{x}_t)$ to construct the expert-losses in (24), (25), and (26). In Step 16, it sends the corresponding expert-loss to each expert so that it can make predictions for the next round.

Finally, we elucidate how our algorithm determines the weight of the i-th expert E^i. We measure the performance of expert E^i by the linearized surrogate loss, i.e., $l_t^i = \langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t^i - \mathbf{y}_t \rangle$. According to Lemma 2, we have $|l_t^i| \leq \|\nabla g_t(\mathbf{y}_t)\| \|\mathbf{y}_t^i - \mathbf{y}_t\| \leq 2GD$. Since Adapt-ML-Prod requires the loss to fall within the range of $[0, 1]$, we normalize l_t^i to construct the meta-loss as $\ell_t^i = (\langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t^i - \mathbf{y}_t \rangle)/(4GD) + \frac{1}{2} \in [0, 1]$. The loss of the meta-algorithm in the t-th round is $\ell_t = \sum_{i=1}^{|\mathcal{A}|} p_t^i \ell_t^i$, which is a constant $\frac{1}{2}$ due to its construction and Step 13. For each expert E^i, its weight is updated by:

$$p_t^i = \frac{\eta_{t-1}^i w_{t-1}^i}{\sum_{j=1}^{|\mathcal{A}|} \eta_{t-1}^j w_{t-1}^j}, \quad w_{t-1}^i = \left(w_{t-2}^i \left(1 + \eta_{t-2}^i (\ell_{t-1} - \ell_{t-1}^i) \right) \right)^{\frac{\eta_{t-1}^i}{\eta_{t-2}^i}} \tag{27}$$

where $\eta_{t-1}^i = \min \left\{ \frac{1}{2}, \sqrt{(\ln |\mathcal{A}|)/(1 + \sum_{s=1}^{t-1} (\ell_s - \ell_s^i)^2)} \right\}$. In the first round, we set $w_0^i = 1/|\mathcal{A}|$.

Remark 3 While the surrogate loss in (8) involves the projection operation, our proposed meta-loss and expert-losses only access $g_t(\mathbf{y})$ through $\nabla g_t(\mathbf{y}_t)$, which is given by Cutkosky [2020],

$$\nabla g_t(\mathbf{y}_t) = \nabla f_t(\mathbf{x}_t) - \mathbb{1}_{\{\langle \nabla f_t(\mathbf{x}_t), \mathbf{v}_t \rangle < 0\}} \langle \nabla f_t(\mathbf{x}_t), \mathbf{v}_t \rangle \cdot \mathbf{v}_t$$

where $\mathbf{v}_t = \frac{\mathbf{y}_t - \mathbf{x}_t}{\|\mathbf{y}_t - \mathbf{x}_t\|}$. According to its formulation, the gradient can be directly computed from $\mathbf{x}_t$ and $\mathbf{y}_t$, which means no additional projections are needed. Therefore, in each round, our algorithm requires only 1 projection onto domain $\mathcal{X}$. ◁

Due to page limit, we provide the expert-algorithms, as well as all the proofs, in Appendix B. The theoretical guarantee of Algorithm 2 is given below.

[2] One can verify the degenerated situations where the unknown modulus falls outside the range, which will not be a concern. Formal justifications are provided in Appendix D.

Theorem 1 *Under Assumptions 1 and 2, Algorithm 2 attains $O(\sqrt{T})$, $O(\frac{d}{\alpha}\log T)$ and $O(\frac{1}{\lambda}\log T)$ regret for general convex functions, α-exp-concave functions with $\alpha \in [1/T, 1]$, and λ-strongly convex functions with $\lambda \in [1/T, 1]$, respectively.*

Remark 4 Similar to previous studies [Wang et al., 2019, Zhang et al., 2022], our universal algorithm also achieves the minimax optimal regret, but only requires 1 projection. $\triangleleft$

4.2 Efficient algorithm for small-loss universal regret

Furthermore, we consider the small-loss regret for smooth and non-negative online functions. To this end, an additional assumption is required [Srebro et al., 2010].

Assumption 3 *All the online functions are non-negative, and H-smooth over $\mathcal{X}$.*

To exploit the smoothness, we enhance the expert-loss for strongly convex functions in (26) as

$$\widehat{\ell}^{\text{sc}}_{t,\widehat{\lambda}}(\mathbf{y}) = \langle \nabla g_t(\mathbf{y}_t), \mathbf{y} - \mathbf{y}_t \rangle + \frac{\widehat{\lambda}}{2G^2}\|\nabla g_t(\mathbf{y}_t)\|^2 \|\mathbf{y} - \mathbf{x}_t\|^2. \tag{28}$$

Since $\widehat{\ell}^{\text{sc}}_{t,\widehat{\lambda}}(\cdot)$ is strongly convex and smooth, we use S^{2}OGD [Wang et al., 2020b] as the expert-algorithm. For general convex and exp-concave functions, we reuse (24) and (25) as the expert-losses, and employ ONS [Orabona et al., 2012] and SOGD [Zhang et al., 2019] as the expert-algorithms. The meta-algorithm remains unchanged. In this way, we obtain the following regret guarantee.

Theorem 2 *Under Assumptions 1, 2 and 3, the improved version of Algorithm 2 attains $O(\sqrt{L_T})$, $O(\frac{d}{\alpha}\log L_T)$ and $O(\frac{1}{\lambda}\log L_T)$ regret for general convex functions, α-exp-concave functions with $\alpha \in [1/T, 1]$, and λ-strongly convex functions with $\lambda \in [1/T, 1]$, respectively, where the small-loss quantity $L_T = \min_{\mathbf{x}\in\mathcal{X}} \sum_{t=1}^{T} f_t(\mathbf{x})$ is the cumulative loss of the best decision from the domain $\mathcal{X}$.*

Remark 5 With only 1 projection in each round, our universal algorithm is able to deliver *optimal* small-loss regret bounds for multiple types of convex functions simultaneously. In contrast, Wang et al. [2020b] and Zhang et al. [2022] take $O(\log T)$ projections to achieve the small-loss regret. $\triangleleft$

5 Conclusion and future work

In this paper, we propose a projection-efficient universal algorithm that achieves minimax optimal regret for three types of convex functions with only 1 projection per round. Furthermore, we enhance our algorithm to exploit the smoothness property and demonstrate that it attains small-loss regret for convex and smooth functions. To demonstrate the effectiveness of our proposed method, we also conduct empirical experiments, and the results are presented in Appendix E.

There are several directions for future research. First, one potentially unfavorable characteristic of our work is the requirements of domain and gradient boundedness. Motivated by the recent developments in parameter-free online learning for unbounded domains and gradients [Orabona, 2014, Orabona and Pál, 2016, Cutkosky and Boahen, 2016, 2017, Foster et al., 2017, Luo et al., 2022, Jacobsen and Cutkosky, 2022, 2023], we will investigate whether our algorithms can further avoid these prior knowledge in the future. Second, in addition to the small-loss bound, another important type of problem-dependent guarantee is the *gradient-variation regret bound* [Zhao et al., 2020, 2024], which has been actively studied recently due to its profound relationship to games and stochastic optimization. In the literature, recent studies [Yan et al., 2023, 2024, Xie et al., 2024, Wang et al., 2024a] achieve almost-optimal gradient-variation regret in universal online learning, but also suffer high projection complexity. Therefore, it remains challenging and important to develop a projection-efficient universal algorithm with optimal gradient-variation regret guarantees. Third, to deal with changing environments, adaptive regret has been proposed to minimize the regret over every interval in various setting of online learning [Hazan and Seshadhri, 2007, Daniely et al., 2015, Wan et al., 2021a, Wang et al., 2024b]. Existing universal algorithms [Zhang et al., 2021, Yang et al., 2024] typically conduct $O(\log^2 T)$ projections per round. In the future, we will investigate whether whether we can reduce the projection complexity of universal algorithms for adaptive regret.

Acknowledgments

This work was partially supported by National Science and Technology Major Project (2022ZD0114801), NSFC (62361146852, 62122037), and the Collaborative Innovation Center of Novel Software Technology and Industrialization.

References

J. Abernethy, P. L. Bartlett, A. Rakhlin, and A. Tewari. Optimal strategies and minimax lower bounds for online convex games. In *Proceedings of the 21st Annual Conference on Learning Theory (COLT)*, pages 415–423, 2008.

P. L. Bartlett, E. Hazan, and A. Rakhlin. Adaptive online gradient descent. In *Advances in Neural Information Processing Systems 20 (NIPS)*, pages 65–72, 2008.

N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27:1–27:27, 2011.

S. Chen, Y.-J. Z. W.-W. Tu, P. Zhao, and L. Zhang. Optimistic online mirror descent for bridging stochastic and adversarial online convex optimization. *Journal of Machine Learning Research*, pages 1 – 62, 2024.

C.-K. Chiang, T. Yang, C.-J. Lee, M. Mahdavi, C.-J. Lu, R. Jin, and S. Zhu. Online optimization with gradual variations. In *Proceedings of the 25th Annual Conference on Learning Theory (COLT)*, pages 1–20, 2012.

A. Cutkosky. Parameter-free, dynamic, and strongly-adaptive online learning. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pages 2250–2259, 2020.

A. Cutkosky and K. Boahen. Online learning without prior information. In *Proceedings of the 30th Annual Conference on Learning Theory (COLT)*, pages 643–677, 2017.

A. Cutkosky and K. A. Boahen. Online convex optimization with unconstrained domains and losses. In *Advances in Neural Information Processing Systems 29 (NIPS)*, pages 748–756, 2016.

A. Cutkosky and F. Orabona. Black-box reductions for parameter-free online learning in Banach spaces. In *Proceedings of the 31st Conference On Learning Theory (COLT)*, pages 1493–1529, 2018.

A. Daniely, A. Gonen, and S. Shalev-Shwartz. Strongly adaptive online learning. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pages 1405–1411, 2015.

J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. In *Proceedings of the 23rd Annual Conference on Learning Theory (COLT)*, pages 257–269, 2010.

D. J. Foster, S. Kale, M. Mohri, and K. Sridharan. Parameter-free online learning via model selection. In *Advances in Neural Information Processing Systems 30 (NIPS)*, pages 6020–6030, 2017.

P. Gaillard, G. Stoltz, and T. van Erven. A second-order bound with excess losses. In *Proceedings of the 27th Conference on Learning Theory (COLT)*, pages 176–196, 2014.

D. Garber and B. Kretzu. Projection-free online exp-concave optimization. In *Proceedings of Thirty Sixth Conference on Learning Theory (COLT)*, pages 1259–1284, 2023.

E. Hazan. Introduction to Online Convex Optimization. *Foundations and Trends in Optimization*, 2 (3-4):157–325, 2016.

E. Hazan and S. Kale. Projection-free online learning. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, pages 521–528, 2012.

E. Hazan and E. Minasyan. Faster projection-free online learning. In *Proceedings of Thirty Third Conference on Learning Theory (COLT)*, pages 1877–1893, 2020.

E. Hazan and C. Seshadhri. Adaptive algorithms for online decision problems. *Electronic Colloquium on Computational Complexity*, 88, 2007.

E. Hazan, A. Agarwal, and S. Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.

A. Jacobsen and A. Cutkosky. Parameter-free mirror descent. In *Proceedings of 35th Conference on Learning Theory (COLT)*, pages 4160–4211, 2022.

A. Jacobsen and A. Cutkosky. Unconstrained online learning with unbounded losses. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, pages 14590–14630, 2023.

D. P. Kingma and J. L. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

H. Luo and R. E. Schapire. Achieving all with no parameters: AdaNormalHedge. In *Proceedings of the 28th Conference on Learning Theory (COLT)*, pages 1286–1304, 2015.

H. Luo, M. Zhang, P. Zhao, and Z.-H. Zhou. Corralling a larger band of bandits: A case study on switching regret for linear bandits. In *Proceedings of the 35th Conference on Learning Theory (COLT)*, pages 3635–3684, 2022.

Z. Mhammedi, W. M. Koolen, and T. Van Erven. Lipschitz adaptivity with multiple learning rates in online learning. In *Proceedings of the 32nd Conference on Learning Theory (COLT)*, pages 2490–2511, 2019.

M. Mohri and S. Yang. Accelerating online convex optimization via adaptive prediction. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 848–856, 2016.

M. C. Mukkamala and M. Hein. Variants of RMSProp and Adagrad with logarithmic regret bounds. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 2545–2553, 2017.

F. Orabona. Simultaneous model selection and optimization through parameter-free stochastic learning. In *Advances in Neural Information Processing Systems 27 (NIPS)*, pages 1116–1124, 2014.

F. Orabona and D. Pál. Coin betting and parameter-free online learning. In *Advances in Neural Information Processing Systems 29 (NIPS)*, pages 577–585, 2016.

F. Orabona and D. Pál. Scale-free online learning. *Theoretical Computer Science*, 716:50–69, 2018.

F. Orabona, N. Cesa-Bianchi, and C. Gentile. Beyond logarithmic bounds in online learning. In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 823–831, 2012.

E. Ordentlich and T. M. Cover. The cost of achieving the best portfolio in hindsight. *Mathematics of Operations Research*, 23(4):960–982, 1998.

D. Prokhorov. IJCNN 2001 neural network competition. Technical report, Ford Research Laboratory, 2001.

S. J. Reddi, S. Kale, and S. Kumar. On the convergence of Adam and beyond. In *International Conference on Learning Representations (ICLR)*, 2018.

S. Shalev-Shwartz. *Online Learning: Theory, Algorithms, and Applications*. PhD thesis, The Hebrew University of Jerusalem, 2007.

S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: primal estimated sub-gradient solver for SVM. In *Proceedings of the 24th International Conference on Machine Learning (ICML)*, pages 807–814, 2007.

S. Shalev-Shwartz, Y. Singer, N. Srebro, and A. Cotter. Pegasos: primal estimated sub-gradient solver for svm. *Mathematical Programming*, 127(1):3–30, 2011.

N. Srebro, K. Sridharan, and A. Tewari. Smoothness, low-noise and fast rates. In *Advances in Neural Information Processing Systems 23 (NIPS)*, pages 2199–2207, 2010.

T. Tieleman and G. Hinton. Lecture 6.5-RMSProp: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, pages 26–31, 2012.

T. van Erven and W. M. Koolen. MetaGrad: Multiple learning rates in online learning. In *Advances in Neural Information Processing Systems 29 (NIPS)*, pages 3666–3674, 2016.

Y. Wan and L. Zhang. Projection-free online learning over strongly convex sets. *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 10076–10084, 2021.

Y. Wan, W.-W. Tu, and L. Zhang. Strongly adaptive online learning over partial intervals. *Science China Information Sciences*, 2021a.

Y. Wan, B. Xue, and L. Zhang. Projection-free online learning in dynamic environments. *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 10067–10075, 2021b.

Y. Wan, W.-W. Tu, and L. Zhang. Online frank-wolfe with arbitrary delays. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 19703–19715, 2022.

G. Wang, S. Lu, and L. Zhang. Adaptivity and optimality: A universal algorithm for online convex optimization. In *Proceedings of the 35th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 659–668, 2019.

G. Wang, S. Lu, Q. Cheng, W.-W. Tu, and L. Zhang. SAdam: A variant of Adam for strongly convex functions. In *International Conference on Learning Representations (ICLR)*, 2020a.

G. Wang, S. Lu, Y. Hu, and L. Zhang. Adapting to smoothness: A more universal algorithm for online convex optimization. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI)*, pages 6162–6169, 2020b.

Y. Wang, Y. Wan, S. Zhang, and L. Zhang. Distributed projection-free online learning for smooth and convex losses. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence (AAAI)*, pages 10226–10234, 2023.

Y. Wang, S. Chen, W. Jiang, W. Yang, Y. Wan, and L. Zhang. Online composite optimization between stochastic and adversarial environments. In *Advances in Neural Information Processing Systems 37 (NeurIPS)*, 2024a.

Y. Wang, W. Yang, W. Jiang, S. Lu, B. Wang, H. Tang, Y. Wan, and L. Zhang. Non-stationary projection-free online learning with dynamic and adaptive regret guarantees. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence (AAAI)*, pages 15671–15679, 2024b.

Y.-F. Xie, P. Zhao, and Z.-H. Zhou. Gradient-variation online learning under generalized smoothness. In *Advances in Neural Information Processing Systems 37 (NeurIPS)*, 2024.

Y.-H. Yan, P. Zhao, and Z.-H. Zhou. Universal online learning with gradient variations: A multi-layer online ensemble approach. In *Advances in Neural Information Processing Systems 36 (NeurIPS)*, pages 37682–37715, 2023.

Y.-H. Yan, P. Zhao, and Z.-H. Zhou. A simple and optimal approach for universal online learning with gradient variations. In *Advances in Neural Information Processing Systems 37 (NeurIPS)*, 2024.

T. Yang, M. Mahdavi, R. Jin, and S. Zhu. Regret bounded by gradual variation for online convex optimization. *Machine Learning*, 95:183–223, 2014.

W. Yang, W. Jiang, Y. Wang, P. Yang, Y. Hu, and L. Zhang. Small-loss adaptive regret for online convex optimization. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, pages 56156–56195, 2024.

L. Zhang, T.-Y. Liu, and Z.-H. Zhou. Adaptive regret of convex and smooth functions. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pages 7414–7423, 2019.

L. Zhang, G. Wang, W.-W. Tu, W. Jiang, and Z.-H. Zhou. Dual adaptivity: A universal algorithm for minimizing the adaptive regret of convex functions. In *Advances in Neural Information Processing Systems 34 (NeurIPS)*, pages 24968–24980, 2021.

L. Zhang, G. Wang, J. Yi, and T. Yang. A simple yet universal strategy for online convex optimization. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, pages 26605–26623, 2022.

P. Zhao, Y.-J. Zhang, L. Zhang, and Z.-H. Zhou. Dynamic regret of convex and smooth functions. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, pages 12510–12520, 2020.

P. Zhao, Y.-F. Xie, L. Zhang, and Z.-H. Zhou. Efficient methods for non-stationary online learning. In *Advances in Neural Information Processing Systems 35 (NeurIPS)*, pages 11573–11585, 2022.

P. Zhao, Y.-J. Zhang, L. Zhang, and Z.-H. Zhou. Adaptivity and non-stationarity: Problem-dependent dynamic regret for online convex optimization. *Journal of Machine Learning Research*, 25(98):1 – 52, 2024.

M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning (ICML)*, pages 928–936, 2003.

A Algorithms for experts

In this section, we provide the detailed procedures of the expert-algorithms in our efficient algorithm.

A.1 Online gradient descent for convex functions

We use OGD [Zinkevich, 2003] to minimize $\ell_t^{\text{cvx}}(\cdot)$ in (24). The procedure of the expert-algorithm for general convex functions is summarized in Algorithm 3.

Algorithm 3 Expert E^i: OGD for Convex Functions

```
1: Let $\mathbf{y}_1^i$ be any point in $\mathcal{Y}$
2: for $t = 1$ to $T$ do
3:     Submit $\mathbf{y}_t^i$ to the meta-algorithm
4:     Update
           $$\widehat{\mathbf{y}}_{t+1}^i = \mathbf{y}_t^i - \frac{1}{\sqrt{t}} \nabla g_t(\mathbf{y}_t)$$
5:     Conduct a projection onto $\mathcal{Y}$
           $$\mathbf{y}_{t+1}^i = \begin{cases} \widehat{\mathbf{y}}_{t+1}^i, & \text{if } \|\widehat{\mathbf{y}}_{t+1}^i\| \leq D, \\ \widehat{\mathbf{y}}_{t+1}^i \cdot \frac{D}{\|\widehat{\mathbf{y}}_{t+1}^i\|}, & \text{otherwise .} \end{cases}$$
6: end for
```

A.2 Online newton step for exp-concave (and smooth) functions

Lemma 4 *Under Assumptions 1 and 2, $\ell_{t,\widehat{\alpha}}^{\exp}(\cdot)$ in (25) is $\frac{\widehat{\beta}}{4}$-exp-concave, and $\|\nabla \ell_{t,\widehat{\alpha}}^{\exp}(\mathbf{y})\|^2 \leq 2G^2$.*

Thus, we use ONS to minimize $\ell_{t,\widehat{\alpha}}^{\exp}(\cdot)$. Different from OGD, the projection of ONS onto $\mathcal{Y}$ cannot be achieved through a simple rescaling like Step 5 in Algorithm 3. Here, we employ an efficient implementation of ONS [Mhammedi et al., 2019] that enhances the efficiency of its projection onto $\mathcal{Y}$. The procedure is summarized in Algorithm 4.

Algorithm 4 Expert E^i: ONS for Exp-concave (and Smooth) Functions

```
1: Let $\mathbf{y}_1^i$ be any point in $\mathcal{Y}$ and $\Sigma_1 = \frac{1}{\widehat{\beta}^2 D^2} \mathbf{I}_d$
2: for $t = 1$ to $T$ do
3:     Submit $\mathbf{y}_t^i$ to the meta-algorithm
4:     Update
           $$\Sigma_{t+1} = \Sigma_t + \nabla \ell_{t,\widehat{\alpha}}^{\exp}(\mathbf{y}_t^i) \nabla \ell_{t,\widehat{\alpha}}^{\exp}(\mathbf{y}_t^i)^\top, \quad \widehat{\mathbf{y}}_{t+1}^i = \mathbf{y}_t^i - \frac{1}{\widehat{\beta}} \Sigma_{t+1}^{-1} \nabla \ell_{t,\widehat{\alpha}}^{\exp}(\mathbf{y}_t^i)$$
       where
           $$\nabla \ell_{t,\widehat{\alpha}}^{\exp}(\mathbf{y}_t^i) = \nabla g_t(\mathbf{y}_t) + \widehat{\beta} \nabla g_t(\mathbf{y}_t) \nabla g_t(\mathbf{y}_t)^\top (\mathbf{y}_t^i - \mathbf{y}_t)$$
5:     Conduct a projection onto $\mathcal{Y}$ in (29)
6: end for
```

Lemma 5 *Let $\mathbf{\Lambda}_{t+1} := \operatorname{diag}((\lambda_t^k)_{k \in [d]})$ and $\mathbf{Q}_{t+1}$ are the matrices of eigenvalues and eigenvectors of $(\Sigma_{t+1} - \frac{1}{\widehat{\beta}^2 D^2} \mathbf{I}_d)$, respectively. Then, the projection onto the ball $\mathcal{Y}$ in Step 5 can be formulated as*

$$\mathbf{y}_{t+1}^i = \begin{cases} \widehat{\mathbf{y}}_{t+1}^i, & \textit{if } \|\widehat{\mathbf{y}}_{t+1}^i\| \leq D, \\ \mathbf{Q}_{t+1}^\top \left(x_{t+1}^i \mathbf{I} + \Lambda_{t+1} \right)^{-1} \mathbf{Q}_{t+1} \Sigma_{t+1} \widehat{\mathbf{y}}_{t+1}^i, & \textit{otherwise .} \end{cases} \tag{29}$$

where x_{t+1}^i is the unique solution of $\rho(x) := \sum_{k=1}^d \frac{\langle \mathbf{e}_k, \mathbf{Q}_{t+1} \Sigma_{t+1} \widehat{\mathbf{y}}_{t+1}^i \rangle^2}{(x + \lambda_t^k)^2} = D^2$.

A.3 Online gradient descent for strongly convex functions

We establish the following lemma for function $\ell_t^{\mathrm{sc}}(\cdot)$ in (14).

Lemma 6 *Under Assumptions 1 and 2, the loss function $\ell_t^{\mathrm{sc}}(\cdot)$ in* (14) *is λ-strongly convex, and $\|\nabla \ell_t^{\mathrm{sc}}(\mathbf{y})\|^2 \leq (G+2D)^2$.*

Since $\ell_{t,\widehat{\lambda}}^{\mathrm{sc}}(\cdot)$ in (26) shares the same formulation as $\ell_t^{\mathrm{sc}}(\cdot)$, $\ell_{t,\widehat{\lambda}}^{\mathrm{sc}}(\cdot)$ also benefits from the aforementioned properties, with the distinction being the substitution of λ for $\widehat{\lambda}$. Therefore, we use a variant of OGD [Shalev-Shwartz et al., 2007] to minimize $\ell_{t,\widehat{\lambda}}^{\mathrm{sc}}(\cdot)$. The procedure is summarized in Algorithm 5.

Algorithm 5 Expert E^i: OGD for Strongly Convex Functions

```
Let $\mathbf{y}_1^i$ be any point in $\mathcal{Y}$
for $t = 1$ to $T$ do
    Submit $\mathbf{y}_t^i$ to the meta-algorithm
    Update
        $\widehat{\mathbf{y}}_{t+1}^i = \mathbf{y}_t^i - \frac{1}{\widehat{\lambda} t} \nabla \ell_{t,\widehat{\lambda}}^{\mathrm{sc}}(\mathbf{y}_t^i)$
    where
        $\nabla \ell_{t,\widehat{\lambda}}^{\mathrm{sc}}(\mathbf{y}_t^i) = \nabla g_t(\mathbf{y}_t) + \widehat{\lambda}(\mathbf{y}_t^i - \mathbf{x}_t)$
    Conduct a projection onto $\mathcal{Y}$
        $\mathbf{y}_{t+1}^i = \begin{cases} \widehat{\mathbf{y}}_{t+1}^i, & \text{if } \|\widehat{\mathbf{y}}_{t+1}^i\| \leq D, \\ \widehat{\mathbf{y}}_{t+1}^i \cdot \frac{D}{\|\widehat{\mathbf{y}}_{t+1}^i\|}, & \text{otherwise .} \end{cases}$
end for
```

A.4 Scale-free online gradient descent for convex and smooth functions

To exploit smoothness, we use scale-free online gradient descent (SOGD) [Zhang et al., 2019] to minimize $\ell_t^{\mathrm{cvx}}(\cdot)$ in (24). The procedure is summarized in Algorithm 6.

Algorithm 6 Expert E^i: Scale-free OGD for Convex and Smooth Functions

```
Let $\mathbf{y}_1^i$ be any point in $\mathcal{Y}$
for $t = 1$ to $T$ do
    Submit $\mathbf{y}_t^i$ to the meta-algorithm
    Update
        $\widehat{\mathbf{y}}_{t+1}^i = \mathbf{y}_t^i - \eta_t \nabla g_t(\mathbf{y}_t)$
    where
        $\eta_t = \frac{\alpha}{\sqrt{\delta + \sum_{s=1}^t \|\nabla g_s(\mathbf{y}_s)\|^2}}, \quad \alpha, \delta > 0$
    Conduct a projection onto $\mathcal{Y}$
        $\mathbf{y}_{t+1}^i = \begin{cases} \widehat{\mathbf{y}}_{t+1}^i, & \text{if } \|\widehat{\mathbf{y}}_{t+1}^i\| \leq D, \\ \widehat{\mathbf{y}}_{t+1}^i \cdot \frac{D}{\|\widehat{\mathbf{y}}_{t+1}^i\|}, & \text{otherwise .} \end{cases}$
end for
```

A.5 Smooth and strongly convex online gradient descent

Recall that to exploit smoothness, we enhance the expert-loss for strongly convex functions as follows

$$\widehat{\ell}_{t,\widehat{\lambda}}^{\mathrm{sc}}(\mathbf{y}) = \langle \nabla g_t(\mathbf{y}_t), \mathbf{y} - \mathbf{y}_t \rangle + \frac{\widehat{\lambda}}{2G^2} \|\nabla g_t(\mathbf{y}_t)\|^2 \|\mathbf{y} - \mathbf{x}_t\|^2.$$

The above expert-loss enjoys the following property.

Lemma 7 *Under Assumptions 1 and 2, $\widehat{\ell}^{\text{sc}}_{t,\widehat{\lambda}}(\cdot)$ in* (28) *is $\frac{\widehat{\lambda}}{G^2}\|\nabla g_t(\mathbf{y}_t)\|^2$-strongly convex, and $\|\widehat{\ell}^{\text{sc}}_{t,\widehat{\lambda}}(\mathbf{y})\|^2 \leq \left(1+\frac{2D}{G}\right)^2\|\nabla g_t(\mathbf{y}_t)\|^2$.*

Due to the modulus of strong convexity is not fixed, we choose Smooth and Strongly Convex OGD (S^{2}OGD) as the expert-algorithm [Wang et al., 2020b] to minimize $\widehat{\ell}^{\text{sc}}_{t,\widehat{\lambda}}(\cdot)$. The procedure is summarized in Algorithm 7.

Algorithm 7 Expert E^i: Smooth and Strongly Convex OGD

```
Let $\mathbf{y}_1^i$ be any point in $\mathcal{Y}$
for $t = 1$ to $T$ do
    Submit $\mathbf{y}_t^i$ to the meta-algorithm
    Update
        $$\widehat{\mathbf{y}}_{t+1}^i = \mathbf{y}_t^i - \eta_t \nabla g_t(\mathbf{y}_t)$$
    where
        $$\eta_t = \frac{\alpha}{\delta + \sum_{s=1}^t \|\nabla \widehat{\ell}^{\text{sc}}_{s,\widehat{\lambda}}(\mathbf{y}_s^i)\|^2}, \quad \alpha, \delta > 0$$
    Conduct a projection onto $\mathcal{Y}$
        $$\mathbf{y}_{t+1}^i = \begin{cases} \widehat{\mathbf{y}}_{t+1}^i, & \text{if } \|\widehat{\mathbf{y}}_{t+1}^i\| \leq D, \\ \widehat{\mathbf{y}}_{t+1}^i \cdot \frac{D}{\|\widehat{\mathbf{y}}_{t+1}^i\|}, & \text{otherwise .} \end{cases}$$
end for
```

B Proofs

In this section, we provide the proofs of the theorems presented in the main paper (Theorem 1 and Theorem 2), as well as proofs of two important lemmas (Lemma 3 and Lemma 8).

B.1 Proof of Theorem 1

We present the exact bounds of the theoretical guarantee provided in Theorem 1. When functions are general convex, we have

$$\begin{aligned}\sum_{t=1}^T f_t(\mathbf{x}_t) - \sum_{t=1}^T f_t(\mathbf{x}) &\leq 4\Gamma GD\left(2 + \frac{1}{\sqrt{\ln|\mathcal{A}|}}\right) + \left(\frac{2\Gamma GD}{\sqrt{\ln|\mathcal{A}|}} + 2D^2 + G^2\right)\sqrt{T} - \frac{G^2}{2} \\ &= O(\sqrt{T})\end{aligned}$$

where $|\mathcal{A}| = 1 + 2\lceil \log_2 T\rceil$ and

$$\Gamma = 3\ln|\mathcal{A}| + \ln\left(1 + \frac{|\mathcal{A}|}{2e}(1 + \ln(T+1))\right) = O(\log\log T). \tag{30}$$

When functions are α-exp-concave, we have

$$\begin{aligned}\sum_{t=1}^T f_t(\mathbf{x}_t) - \sum_{t=1}^T f_t(\mathbf{x}) &\leq 4\Gamma GD\left(2 + \frac{1}{\sqrt{\ln|\mathcal{A}|}}\right) + \frac{\Gamma^2}{\beta\ln|\mathcal{A}|} + 5\left(\frac{8}{\beta} + 2\sqrt{2}GD\right)d\log T \\ &= O\left(\frac{d}{\alpha}\log T\right).\end{aligned}$$

When functions are λ-strongly convex, we have

$$\sum_{t=1}^{T} f_t(\mathbf{x}_t) - \sum_{t=1}^{T} f_t(\mathbf{x}) \leq 4\Gamma GD \left(2 + \frac{1}{\sqrt{\ln|\mathcal{A}|}}\right) + \frac{\Gamma^2 G^2}{\min\{\frac{G}{D}, \lambda\} \ln|\mathcal{A}|} + \frac{(G+D)^2}{\lambda} \log T$$
$$= O\left(\frac{1}{\lambda} \log T\right).$$

B.1.1 Analysis for general convex functions

We introduce the following decomposition for general convex functions,

$$\begin{aligned}\sum_{t=1}^{T} f_t(\mathbf{x}_t) - \sum_{t=1}^{T} f_t(\mathbf{x}) \leq \sum_{t=1}^{T} \langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x} \rangle &\overset{(9)}{\leq} \sum_{t=1}^{T} \langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{x} \rangle \\ &= \sum_{t=1}^{T} \langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{y}_t^i \rangle + \sum_{t=1}^{T} \langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t^i - \mathbf{x} \rangle \\ &\overset{(24)}{=} \underbrace{\sum_{t=1}^{T} \langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{y}_t^i \rangle}_{\texttt{meta-regret}} + \underbrace{\sum_{t=1}^{T} \left(\ell_t^{\text{cvx}}(\mathbf{y}_t^i) - \ell_t^{\text{cvx}}(\mathbf{x})\right)}_{\texttt{expert-regret}}.\end{aligned} \tag{31}$$

First, we start with the expert-regret. Since we are employing OGD to minimize $\ell_t^{\text{cvx}}(\cdot)$, using standard OGD analysis [Zinkevich, 2003, Theorem 1] can obtain the following upper bound

$$\sum_{t=1}^{T} \ell_t^{\text{cvx}}(\mathbf{y}_t^i) - \sum_{t=1}^{T} \ell_t^{\text{cvx}}(\mathbf{x}) \leq (2D^2 + G^2)\sqrt{T} - \frac{G^2}{2}, \tag{32}$$

for any expert $\mathbf{y}_t^i \in \mathcal{Y}$ and any $\mathbf{x} \in \mathcal{X}$.

Next, we move to bound the meta-regret. According to (48), we have

$$\begin{aligned}\sum_{t=1}^{T} \langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{y}_t^i \rangle &\leq 8\Gamma GD + \frac{\Gamma}{\sqrt{\ln|\mathcal{A}|}} \sqrt{16G^2D^2 + \sum_{t=1}^{T} \langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{y}_t^i \rangle^2} \\ &\leq 4\Gamma GD \left(2 + \frac{1}{\sqrt{\ln|\mathcal{A}|}}\right) + \frac{\Gamma}{\sqrt{\ln|\mathcal{A}|}} \sqrt{\sum_{t=1}^{T} \langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{y}_t^i \rangle^2} \\ &\leq 4\Gamma GD \left(2 + \frac{1}{\sqrt{\ln|\mathcal{A}|}}\right) + \frac{\Gamma}{\sqrt{\ln|\mathcal{A}|}} \sqrt{\sum_{t=1}^{T} \|\nabla g_t(\mathbf{y}_t)\|^2 \|\mathbf{y}_t - \mathbf{y}_t^i\|^2} \\ &\leq 4\Gamma GD \left(2 + \frac{1}{\sqrt{\ln|\mathcal{A}|}}\right) + \frac{2\Gamma GD}{\sqrt{\ln|\mathcal{A}|}} \sqrt{T},\end{aligned} \tag{33}$$

for all expert $E^i \in \mathcal{A}$, where Γ is defined in (30) and the last set is due to

$$\|\nabla g_t(\mathbf{y}_t)\| \leq \|\nabla f_t(\mathbf{x}_t)\| \leq G. \tag{34}$$

Finally, substituting (32) and (33) into (31), we have

$$\sum_{t=1}^{T} f_t(\mathbf{x}_t) - \sum_{t=1}^{T} f_t(\mathbf{x}) \leq 4\Gamma GD \left(2 + \frac{1}{\sqrt{\ln|\mathcal{A}|}}\right) + \left(\frac{2\Gamma GD}{\sqrt{\ln|\mathcal{A}|}} + 2D^2 + G^2\right) \sqrt{T} - \frac{G^2}{2}.$$

B.1.2 Analysis for exp-concave functions

For α-exp-concave functions, there exits $\widehat{\alpha}^* \in \mathcal{P}_{\text{exp}}$ that $\widehat{\alpha}^* \leq \alpha \leq 2\widehat{\alpha}^*$, where $\widehat{\alpha}^*$ is the modulus of the i-th expert E^i. This inequality also indicates

$$\widehat{\beta}^* \leq \beta \leq 2\widehat{\beta}^*, \quad \widehat{\beta}^* = \frac{1}{2} \min\{\frac{1}{4GD}, \widehat{\alpha}^*\}. \tag{35}$$

Since $x - \frac{\widehat{\beta}^*}{2}x^2$ is strictly increasing where $\widehat{\beta}^* = \frac{1}{2}\min\{\frac{1}{4GD}, \widehat{\alpha}^*\}$ when $x \in (-\infty, 2GD]$, (9) implies that

$$\langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}\rangle - \frac{\widehat{\beta}^*}{2}\langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}\rangle^2 \leq \langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{x}\rangle - \frac{\widehat{\beta}^*}{2}\langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{x}\rangle^2. \tag{36}$$

Then, we introduce the following decomposition for α-exp-concave functions,

$$\begin{aligned}
&\sum_{t=1}^{T} f_t(\mathbf{x}_t) - \sum_{t=1}^{T} f_t(\mathbf{x}) \leq \sum_{t=1}^{T}\langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}\rangle - \frac{\beta}{2}\sum_{t=1}^{T}\langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}\rangle^2 \\
\overset{(35)}{\leq} &\sum_{t=1}^{T}\langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}\rangle - \frac{\widehat{\beta}^*}{2}\sum_{t=1}^{T}\langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}\rangle^2 \\
\overset{(36)}{\leq} &\sum_{t=1}^{T}\langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{x}\rangle - \frac{\widehat{\beta}^*}{2}\sum_{t=1}^{T}\langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{x}\rangle^2 \\
= &\sum_{t=1}^{T}\langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{y}_t^i\rangle + \sum_{t=1}^{T}\langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t^i - \mathbf{x}\rangle - \frac{\widehat{\beta}^*}{2}\sum_{t=1}^{T}\langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{x}\rangle^2 \\
\overset{(25)}{=} &\underbrace{\sum_{t=1}^{T}\langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{y}_t^i\rangle}_{\texttt{meta-regret}} + \underbrace{\sum_{t=1}^{T}\left(\ell_{t,\widehat{\alpha}^*}^{\exp}(\mathbf{y}_t^i) - \ell_{t,\widehat{\alpha}^*}^{\exp}(\mathbf{x})\right)}_{\texttt{expert-regret}} - \frac{\widehat{\beta}^*}{2}\sum_{t=1}^{T}\langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{y}_t^i\rangle^2.
\end{aligned} \tag{37}$$

For the expert-regret, we can use the analysis of ONS [Hazan et al., 2007, Theorem 2] to obtain

$$\sum_{t=1}^{T}\ell_{t,\widehat{\alpha}^*}^{\exp}(\mathbf{y}_t^i) - \sum_{t=1}^{T}\ell_{t,\widehat{\alpha}^*}^{\exp}(\mathbf{x}) \leq 5\left(\frac{4}{\widehat{\beta}^*} + 2\sqrt{2}GD\right) d\log T \tag{38}$$

for any expert $\mathbf{y}_t^i \in \mathcal{Y}$ and any $\mathbf{x} \in \mathcal{X}$, where $\widehat{\beta}^*$ is defined in (35). Next, we move to bound the meta-regret. According to (48), we have

$$\begin{aligned}
\sum_{t=1}^{T}\langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{y}_t^i\rangle &\leq 8\Gamma GD + \frac{\Gamma}{\sqrt{\ln|\mathcal{A}|}}\sqrt{16G^2D^2 + \sum_{t=1}^{T}\langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{y}_t^i\rangle^2} \\
&\leq 4\Gamma GD\left(2 + \frac{1}{\sqrt{\ln|\mathcal{A}|}}\right) + \frac{\Gamma}{\sqrt{\ln|\mathcal{A}|}}\sqrt{\sum_{t=1}^{T}\langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{y}_t^i\rangle^2} \\
&\leq 4\Gamma GD\left(2 + \frac{1}{\sqrt{\ln|\mathcal{A}|}}\right) + \frac{\Gamma^2}{2\widehat{\beta}^*\ln|\mathcal{A}|} + \frac{\widehat{\beta}^*}{2}\langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{y}_t^i\rangle^2
\end{aligned} \tag{39}$$

for all expert $E^i \in \mathcal{A}$, where Γ is defined in (30) and the last step is due to $\sqrt{ab} \leq \frac{a}{2} + \frac{b}{2}$. Substituting (38) and (39) into (37), we have

$$\sum_{t=1}^{T} f_t(\mathbf{x}_t) - \sum_{t=1}^{T} f_t(\mathbf{x}) \leq 4\Gamma GD\left(2 + \frac{1}{\sqrt{\ln|\mathcal{A}|}}\right) + \frac{\Gamma^2}{2\widehat{\beta}^*\ln|\mathcal{A}|} + 5\left(\frac{4}{\widehat{\beta}^*} + 2\sqrt{2}GD\right) d\log T.$$

Finally, we use (35) to simplify the above bound.

B.1.3 Analysis for strongly convex functions

For λ-strongly convex functions, there exits $\widehat{\lambda}^* \in \mathcal{P}_{\text{sc}}$ that $\widehat{\lambda}^* \le \lambda \le 2\widehat{\lambda}^*$, where $\widehat{\lambda}^*$ is the modulus of the i-th expert E^i. Then, we introduce the following decomposition for λ-strongly convex functions

$$\begin{aligned}
&\sum_{t=1}^T f_t(\mathbf{x}_t) - \sum_{t=1}^T f_t(\mathbf{x}) \le \sum_{t=1}^T \langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}\rangle - \frac{\lambda}{2}\sum_{t=1}^T \|\mathbf{x}_t - \mathbf{x}\|^2 \\
\le{}& \sum_{t=1}^T \langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}\rangle - \frac{\widehat{\lambda}^*}{2}\sum_{t=1}^T \|\mathbf{x}_t - \mathbf{x}\|^2 \\
\overset{(20)}{\le}{}& \sum_{t=1}^T \langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{x}\rangle - \Delta_T - \frac{\widehat{\lambda}^*}{2}\sum_{t=1}^T \|\mathbf{x}_t - \mathbf{x}\|^2 \\
\overset{(26)}{=}{}& \underbrace{\sum_{t=1}^T \langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{y}_t^i\rangle}_{\texttt{meta-regret}} + \underbrace{\sum_{t=1}^T \left(\ell^{\text{sc}}_{t,\widehat{\lambda}^*}(\mathbf{y}_t^i) - \ell^{\text{sc}}_{t,\widehat{\lambda}^*}(\mathbf{x})\right)}_{\texttt{expert-regret}} - \frac{\widehat{\lambda}^*}{2}\sum_{t=1}^T \|\mathbf{y}_t^i - \mathbf{x}_t\|^2 - \Delta_T
\end{aligned} \tag{40}$$

where $\Delta_T = \sum_{t=1}^T \mathbb{1}_{\{\langle \nabla f_t(\mathbf{x}_t), \mathbf{v}_t\rangle \ge 0\}} \cdot \langle \nabla f_t(\mathbf{x}_t), \mathbf{y}_t - \mathbf{x}_t\rangle$. To bound the meta-regret, we derive the following theoretical guarantee.

Lemma 8 *Under Assumptions 1 and 2, the meta-regret of Algorithm 2 satisfies*

$$\begin{aligned}
&\sum_{t=1}^T \langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{y}_t^i\rangle \le 8\Gamma GD + \frac{\Gamma}{\sqrt{\ln|\mathcal{A}|}}\sqrt{16G^2D^2 + \sum_{t=1}^T \langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{y}_t^i\rangle^2} \\
\le{}& 4\Gamma GD\left(2 + \frac{1}{\sqrt{\ln|\mathcal{A}|}}\right) + \frac{\Gamma^2 G^2}{2\gamma\ln|\mathcal{A}|} + \frac{\gamma}{2G^2}\sum_{t=1}^T \langle \nabla g_t(\mathbf{y}_t), \mathbf{x}_t - \mathbf{y}_t^i\rangle^2 + \Delta_T
\end{aligned}$$

for any $\gamma \in (0, \frac{G}{2D}]$*, where* $\Delta_T = \sum_{t=1}^T \mathbb{1}_{\{\langle \nabla f_t(\mathbf{x}_t), \mathbf{v}_t\rangle \ge 0\}} \cdot \langle \nabla f_t(\mathbf{x}_t), \mathbf{y}_t - \mathbf{x}_t\rangle$ *and* Γ *is in* (30).

Remark 6 As mentioned in Section 3.2, Lemma 8 is pivotal in delivering optimal regret for strongly convex functions. Specifically, when the meta-algorithm enjoys a second-order bound in terms of the surrogate loss in (8), we can then convert the intermediate decision $\mathbf{y}_t$ in the meta-regret bound to the actual one $\mathbf{x}_t$, at the cost of adding an addition positive term, as presented in the analysis in (23). ◁

Combining Lemma 8 with (40) , we have

$$\begin{aligned}
&\sum_{t=1}^T f_t(\mathbf{x}_t) - \sum_{t=1}^T f_t(\mathbf{x}) \\
\le{}& 4\Gamma GD\left(2 + \frac{1}{\sqrt{\ln|\mathcal{A}|}}\right) + \frac{\Gamma^2 G^2}{2\gamma\ln|\mathcal{A}|} + \frac{\gamma}{2G^2}\sum_{t=1}^T \langle \nabla g_t(\mathbf{y}_t), \mathbf{x}_t - \mathbf{y}_t^i\rangle^2 \\
&+ \mathrm{ER}(T) - \frac{\widehat{\lambda}^*}{2}\sum_{t=1}^T \|\mathbf{y}_t^i - \mathbf{x}_t\|^2 \\
\overset{(34)}{\le}{}& 4\Gamma GD\left(2 + \frac{1}{\sqrt{\ln|\mathcal{A}|}}\right) + \frac{\Gamma^2 G^2}{2\gamma\ln|\mathcal{A}|} + \left(\frac{\gamma}{2} - \frac{\widehat{\lambda}^*}{2}\right)\sum_{t=1}^T \|\mathbf{x}_t - \mathbf{y}_t^i\|^2 + \mathrm{ER}(T) \\
\le{}& 4\Gamma GD\left(2 + \frac{1}{\sqrt{\ln|\mathcal{A}|}}\right) + \frac{\Gamma^2 G^2}{2\gamma\ln|\mathcal{A}|} + \mathrm{ER}(T)
\end{aligned} \tag{41}$$

where $\mathrm{ER}(T) = \sum_{t=1}^T (\ell^{\text{sc}}_{t,\widehat{\lambda}^*}(\mathbf{y}_t^i) - \ell^{\text{sc}}_{t,\widehat{\lambda}^*}(\mathbf{x}))$ and the last step is because we set $\gamma = \min\{\frac{G}{2D}, \widehat{\lambda}^*\}$. Next, we bound the expert-regret [Shalev-Shwartz et al., 2011, Lemma 1]

$$\mathrm{ER}(T) = \sum_{t=1}^T \ell^{\text{sc}}_{t,\widehat{\lambda}^*}(\mathbf{y}_t^i) - \sum_{t=1}^T \ell^{\text{sc}}_{t,\widehat{\lambda}^*}(\mathbf{x}) \le \frac{(G+D)^2}{2\widehat{\lambda}^*}\log T. \tag{42}$$

for any expert $\mathbf{y}_t^i \in \mathcal{Y}$ and any $\mathbf{x} \in \mathcal{X}$. Substituting (42) into (41), we have

$$\sum_{t=1}^{T} f_t(\mathbf{x}_t) - \sum_{t=1}^{T} f_t(\mathbf{x}) \leq 4\Gamma G D \left(2 + \frac{1}{\sqrt{\ln |\mathcal{A}|}}\right) + \frac{\Gamma^2 G^2}{2\gamma \ln |\mathcal{A}|} + \frac{(G+D)^2}{2\widehat{\lambda}^*} \log T.$$

Finally, we use $\widehat{\lambda}^* \leq \lambda \leq 2\widehat{\lambda}^*$ to simplify the above bound.

B.2 Proof of Lemma 3

According to (8), the (sub-)gradients of $g_t(\cdot)$ can be formulated as

$$\nabla g_t(\mathbf{y}) = \begin{cases} \nabla f_t(\mathbf{x}_t), & \text{if } \langle \nabla f_t(\mathbf{x}_t), \mathbf{v}_t \rangle \geq 0, \\ \nabla f_t(\mathbf{x}_t) - \langle \nabla f_t(\mathbf{x}_t), \mathbf{v}_t \rangle \cdot \frac{\mathbf{y} - \Pi_{\mathcal{X}}[\mathbf{y}]}{\|\mathbf{y} - \Pi_{\mathcal{X}}[\mathbf{y}]\|}, & \text{if } \langle \nabla f_t(\mathbf{x}_t), \mathbf{v}_t \rangle < 0. \end{cases} \tag{43}$$

(i) When $\langle \nabla f_t(\mathbf{x}_t), \mathbf{v}_t \rangle \geq 0$. We have $g_t(\mathbf{y}) = \langle \nabla f_t(\mathbf{x}_t), \mathbf{y} \rangle$ and $\nabla g_t(\mathbf{y}) = \nabla f_t(\mathbf{x}_t)$. Thus,

$$\langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x} \rangle = \langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{x} \rangle - \langle \nabla f_t(\mathbf{x}_t), \mathbf{y}_t - \mathbf{x}_t \rangle. \tag{44}$$

By the definition of $\mathbf{v}_t = (\mathbf{y}_t - \mathbf{x}_t)/\|\mathbf{y}_t - \mathbf{x}_t\|$, we have $\langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t \rangle \leq \langle \nabla f_t(\mathbf{x}_t), \mathbf{y}_t \rangle$ and thus

$$\langle \nabla g_t(\mathbf{y}_t), \mathbf{x}_t \rangle \leq \langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t \rangle \tag{45}$$

(ii) When $\langle \nabla f_t(\mathbf{x}_t), \mathbf{v}_t \rangle < 0$. According to Lemma 2, we obtain

$$\langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x} \rangle \leq \langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{x} \rangle. \tag{46}$$

Moreover, we derive the following equation

$$\begin{aligned} \langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{x}_t \rangle &= \langle \nabla f_t(\mathbf{x}_t), \mathbf{y}_t - \mathbf{x}_t \rangle - \langle \nabla f_t(\mathbf{x}_t), \mathbf{v}_t \rangle \cdot \langle \mathbf{v}_t, \mathbf{y}_t - \mathbf{x}_t \rangle \\ = \langle \nabla f_t(\mathbf{x}_t), \mathbf{y}_t - \mathbf{x}_t \rangle &- \langle \nabla f_t(\mathbf{x}_t), \mathbf{y}_t - \mathbf{x}_t \rangle \cdot \frac{1}{\|\mathbf{y}_t - \mathbf{x}_t\|} \left\langle \frac{\mathbf{y}_t - \mathbf{x}_t}{\|\mathbf{y}_t - \mathbf{x}_t\|}, \mathbf{y}_t - \mathbf{x}_t \right\rangle = 0. \end{aligned} \tag{47}$$

Finally, combining (44) and (46) obtains (20), further combining (45) and (47) yields (21).

B.3 Proof of Lemma 8

By the regret guarantee of Adapt-ML-Prod [Gaillard et al., 2014, Corollary 4], we have

$$\sum_{t=1}^{T} \left(\ell_t - \ell_t^i\right) \leq 2\Gamma + \frac{\Gamma}{\sqrt{\ln |\mathcal{A}|}} \sqrt{1 + \sum_{t=1}^{T} \left(\ell_t - \ell_t^i\right)^2}$$

for all expert $E^i \in \mathcal{A}$, where $\Gamma = 3 \ln |\mathcal{A}| + \ln(1 + \frac{|\mathcal{A}|}{2e}(1 + \ln(T+1))) = O(\log \log T)$. By the definition of ℓ_t and ℓ_t^i, we have

$$\begin{aligned} &\sum_{t=1}^{T} \langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{y}_t^i \rangle \leq 8\Gamma G D + \frac{\Gamma}{\sqrt{\ln |\mathcal{A}|}} \sqrt{16 G^2 D^2 + \sum_{t=1}^{T} \langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{y}_t^i \rangle^2} \\ &\leq 4\Gamma G D \left(2 + \frac{1}{\sqrt{\ln |\mathcal{A}|}}\right) + \frac{\Gamma^2 G^2}{2\gamma \ln |\mathcal{A}|} + \frac{\gamma}{2G^2} \sum_{t=1}^{T} \langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{y}_t^i \rangle^2, \end{aligned} \tag{48}$$

for any $\gamma > 0$, where the last step uses AM-GM inequality.

Next, we handle the term $\langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{y}_t^i \rangle^2$. We will consider two cases separately.

(i) When $\langle \nabla f_t(\mathbf{x}_t), \mathbf{v}_t \rangle \geq 0$, we have

$$\langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{y}_t^i \rangle \leq \langle \nabla f_t(\mathbf{x}_t), \mathbf{y}_t - \mathbf{y}_t^i \rangle \leq \|\nabla f_t(\mathbf{x}_t)\| \|\mathbf{y}_t - \mathbf{y}_t^i\| \leq 2GD. \tag{49}$$

As the function $q(x) = x - \frac{\gamma}{2G^2} x^2$ is strictly increasing when $x \in (-\infty, \frac{G^2}{\gamma}]$, (49) implies that

$$\langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{y}_t^i \rangle - \frac{\gamma}{2G^2} \langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{y}_t^i \rangle^2 \leq \langle \nabla f_t(\mathbf{x}_t), \mathbf{y}_t - \mathbf{y}_t^i \rangle - \frac{\gamma}{2G^2} \langle \nabla f_t(\mathbf{x}_t), \mathbf{y}_t - \mathbf{y}_t^i \rangle^2.$$

for any $\gamma \in (0, \frac{G}{2D}]$. By rearranging terms, we obtain

$$\begin{aligned}\frac{\gamma}{2G^2}\langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{y}_t^i\rangle^2 &\overset{(43)}{=} \frac{\gamma}{2G^2}\langle \nabla f_t(\mathbf{x}_t), \mathbf{y}_t - \mathbf{y}_t^i\rangle^2 \\ &\leq \langle \nabla f_t(\mathbf{x}_t), \mathbf{y}_t - \mathbf{x}_t\rangle + \frac{\gamma}{2G^2}\langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{y}_t^i\rangle^2 \\ &\overset{(43)}{=} \langle \nabla f_t(\mathbf{x}_t), \mathbf{y}_t - \mathbf{x}_t\rangle + \frac{\gamma}{2G^2}\langle \nabla g_t(\mathbf{y}_t), \mathbf{x}_t - \mathbf{y}_t^i\rangle^2.\end{aligned} \tag{50}$$

(ii) When $\langle \nabla f_t(\mathbf{x}_t), \mathbf{v}_t\rangle < 0$, (47) directly implies $\langle \nabla g_t(\mathbf{y}_t), \mathbf{x}_t - \mathbf{y}_t^i\rangle = \langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{y}_t^i\rangle$. Thus,

$$\frac{\gamma}{2G^2}\langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{y}_t^i\rangle^2 = \frac{\gamma}{2G^2}\langle \nabla g_t(\mathbf{y}_t), \mathbf{x}_t - \mathbf{y}_t^i\rangle^2. \tag{51}$$

Combining (50) and (51), we have

$$\frac{\gamma}{2G^2}\langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{y}_t^i\rangle^2 \leq \mathbb{1}_{\{\langle \nabla f_t(\mathbf{x}_t), \mathbf{v}_t\rangle \geq 0\}}\langle \nabla f_t(\mathbf{x}_t), \mathbf{y}_t - \mathbf{x}_t\rangle + \frac{\gamma}{2G^2}\langle \nabla g_t(\mathbf{y}_t), \mathbf{x}_t - \mathbf{y}_t^i\rangle^2 \tag{52}$$

for any $\gamma \in (0, \frac{G}{2D}]$. Substituting (52) into (48), we finish the proof.

B.4 Proof of Theorem 2

The analysis is similar to Theorem 1. Also, we present the exact bounds of the theoretical guarantee provided in Theorem 2. When functions are general convex, we have

$$\begin{aligned}&\sum_{t=1}^T f_t(\mathbf{x}_t) - \sum_{t=1}^T f_t(\mathbf{x}) \\ \leq\ & 4\Gamma G D\left(2 + \frac{1}{\sqrt{\ln|\mathcal{A}|}}\right) + \sqrt{2D^2\delta} + 4H\left(\frac{2\Gamma D}{\sqrt{\ln|\mathcal{A}|}} + \sqrt{2}(D+2G)\right)^2 \\ &+ 2\sqrt{H}\left(\frac{2\Gamma D}{\sqrt{\ln|\mathcal{A}|}} + \sqrt{2}(D+2G)\right)\sqrt{L_T + 4\Gamma G D\left(2 + \frac{1}{\sqrt{\ln|\mathcal{A}|}}\right) + \sqrt{2D^2\delta}} \\ =\ & O(\sqrt{L_T}).\end{aligned}$$

where $|\mathcal{A}| = 1 + 2\lceil\log_2 T\rceil$, Γ is defined in (30), and $L_T = \min_{\mathbf{x}\in\mathcal{X}}\sum_{t=1}^T f_t(\mathbf{x})$. When functions are α-exp-concave, we have

$$\begin{aligned}&\sum_{t=1}^T f_t(\mathbf{x}_t) - \sum_{t=1}^T f_t(\mathbf{x}) \\ \leq\ & 4\Gamma G D\left(2 + \frac{1}{\sqrt{\ln|\mathcal{A}|}}\right) + \frac{\Gamma^2}{2\beta\ln|\mathcal{A}|} + \frac{2d}{\beta}\log\left(\frac{\beta^2 D^2 H}{d}\sum_{t=1}^T f_t(\mathbf{x}_t) + 1\right) + \frac{2}{\beta} \\ \leq\ & \widehat{\Gamma} + \frac{2d}{\beta}\log\left(\frac{2\beta^2 D^2 H}{d}\sum_{t=1}^T f_t(\mathbf{x}) + \frac{2\beta^2 D^2 H}{d}\widehat{\Gamma} + 2D^2 H\log(2D^2 H) + 2\right) \\ =\ & O\left(\frac{d}{\alpha}\log L_T\right)\end{aligned}$$

where $\widehat{\Gamma} = 4\Gamma G D\left(2 + \frac{1}{\sqrt{\ln|\mathcal{A}|}}\right) + \frac{\Gamma^2}{2\beta\ln|\mathcal{A}|} + \frac{2}{\beta}$. When functions are λ-strongly convex, we have

$$\begin{aligned}&\sum_{t=1}^T f_t(\mathbf{x}_t) - \sum_{t=1}^T f_t(\mathbf{x}) \\ \leq\ & \widetilde{\Gamma} + \frac{(G+2D)^2}{2\lambda}\log\left(\frac{8H\lambda}{(G+2D)^2}\sum_{t=1}^T f_t(\mathbf{x}) + \frac{8H\lambda}{(G+2D)^2}\widetilde{\Gamma} + 2H\log(2H) + 2\right) \\ =\ & O\left(\frac{1}{\lambda}\log L_T\right)\end{aligned}$$

where $\widetilde{\Gamma} = 4\Gamma G D\left(2 + \frac{1}{\sqrt{\ln|\mathcal{A}|}}\right) + \frac{\Gamma^2 G^2}{2\gamma\ln|\mathcal{A}|} + 1$.

B.4.1 Analysis for general convex functions

We start with the meta-expert regret decomposition as presented in (31),

$$\sum_{t=1}^{T} f_t(\mathbf{x}_t) - \sum_{t=1}^{T} f_t(\mathbf{x}) \le \underbrace{\sum_{t=1}^{T}\langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{y}_t^i\rangle}_{\texttt{meta-regret}} + \underbrace{\sum_{t=1}^{T}\left(\ell_t^{\text{cvx}}(\mathbf{y}_t^i) - \ell_t^{\text{cvx}}(\mathbf{x})\right)}_{\texttt{expert-regret}}. \tag{53}$$

For the meta-regret, we reuse (33) to obtain

$$\begin{aligned}\sum_{t=1}^{T}\langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{y}_t^i\rangle &\le 4\Gamma GD\left(2 + \frac{1}{\sqrt{\ln|\mathcal{A}|}}\right) + \frac{\Gamma}{\sqrt{\ln|\mathcal{A}|}}\sqrt{\sum_{t=1}^{T}\|\nabla g_t(\mathbf{y}_t)\|^2\|\mathbf{y}_t - \mathbf{y}_t^i\|^2} \\ &\le 4\Gamma GD\left(2 + \frac{1}{\sqrt{\ln|\mathcal{A}|}}\right) + \frac{2\Gamma D}{\sqrt{\ln|\mathcal{A}|}}\sqrt{\sum_{t=1}^{T}\|\nabla g_t(\mathbf{y}_t)\|^2},\end{aligned} \tag{54}$$

for all expert $E^i \in \mathcal{A}$, where Γ is defined in (30). For the expert-regret, we can use the analysis of SOGD [Zhang et al., 2019, Theorem 2] to obtain

$$\sum_{t=1}^{T}\ell_t^{\text{cvx}}(\mathbf{y}_t^i) - \sum_{t=1}^{T}\ell_t^{\text{cvx}}(\mathbf{x}) \le \sqrt{2D^2}\sqrt{\delta + \left(1 + \frac{2G}{D}\right)^2 \sum_{t=1}^{T}\|\nabla g_t(\mathbf{y}_t)\|^2}.$$

for any expert $\mathbf{y}_t^i \in \mathcal{Y}$ and any $\mathbf{x} \in \mathcal{X}$. From the above formulation, we have

$$\sum_{t=1}^{T}\ell_t^{\text{cvx}}(\mathbf{y}_t^i) - \sum_{t=1}^{T}\ell_t^{\text{cvx}}(\mathbf{x}) \le \sqrt{2D^2\delta} + \sqrt{2\left(D + 2G\right)^2 \sum_{t=1}^{T}\|\nabla g_t(\mathbf{y}_t)\|^2}. \tag{55}$$

Substituting (54) and (55) into (53), we have

$$\begin{aligned}&\sum_{t=1}^{T} f_t(\mathbf{x}_t) - \sum_{t=1}^{T} f_t(\mathbf{x}) \\ &\overset{(34)}{\le} 4\Gamma GD\left(2 + \frac{1}{\sqrt{\ln|\mathcal{A}|}}\right) + \sqrt{2D^2\delta} + \left(\frac{2\Gamma D}{\sqrt{\ln|\mathcal{A}|}} + \sqrt{2}(D + 2G)\right)\sqrt{\sum_{t=1}^{T}\|\nabla f_t(\mathbf{x}_t)\|^2}.\end{aligned}$$

Next, we introduce the self-bounding property of smooth functions.

Lemma 9 (Lemma 3.1 of Srebro et al. [2010]) *For an H-smooth and nonnegative function, we have $\|\nabla f(\mathbf{x})\| \le \sqrt{4Hf(\mathbf{x})}$.*

Thus, when functions are smooth, we have

$$\begin{aligned}&\sum_{t=1}^{T} f_t(\mathbf{x}_t) - \sum_{t=1}^{T} f_t(\mathbf{x}) \\ &\overset{(34)}{\le} 4\Gamma GD\left(2 + \frac{1}{\sqrt{\ln|\mathcal{A}|}}\right) + \sqrt{2D^2\delta} + \left(\frac{2\Gamma D}{\sqrt{\ln|\mathcal{A}|}} + \sqrt{2}(D + 2G)\right)\sqrt{4H\sum_{t=1}^{T} f_t(\mathbf{x}_t)}.\end{aligned}$$

To simplify the above inequality, we use the following lemma.

Lemma 10 (Lemma 19 of Shalev-Shwartz [2007]) *Let $x, b, c \in \mathbb{R}^+$. Then, we have $x - c \le b\sqrt{x} \Rightarrow x - c \le b^2 + b\sqrt{c}$.*

By utilizing Lemma 10, we finish the proof.

B.4.2 Analysis for exp-concave functions

The analysis is also similar to Theorem 1. We start with (37)

$$\begin{aligned}
&\sum_{t=1}^{T} f_t(\mathbf{x}_t) - \sum_{t=1}^{T} f_t(\mathbf{x}) \\
\leq &\underbrace{\sum_{t=1}^{T} \langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{y}_t^i \rangle}_{\texttt{meta-regret}} + \underbrace{\sum_{t=1}^{T} \left(\ell^{\exp}_{t,\widehat{\alpha}^*}(\mathbf{y}_t^i) - \ell^{\exp}_{t,\widehat{\alpha}^*}(\mathbf{x}) \right)}_{\texttt{expert-regret}} - \frac{\widehat{\beta}^*}{2} \sum_{t=1}^{T} \langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{y}_t^i \rangle^2.
\end{aligned} \tag{56}$$

For the meta-regret, we also use (39) to bound. For the expert-regret, we can use the analysis of ONS under the smoothness condition [Orabona et al., 2012, Theorem 1] to get

$$\sum_{t=1}^{T} \ell^{\exp}_{t,\widehat{\alpha}}(\mathbf{y}_t^i) - \sum_{t=1}^{T} \ell^{\exp}_{t,\widehat{\alpha}}(\mathbf{x}) \leq \frac{2d}{\widehat{\beta}^*} \log \left(\frac{{\widehat{\beta}^*}^2 D^2}{16d} \sum_{t=1}^{T} \|\nabla \ell^{\exp}_{t,\widehat{\alpha}}(\mathbf{y}_t^i)\|^2 + 1 \right) + \frac{2}{\widehat{\beta}^*}.$$

for any expert $\mathbf{y}_t^i \in \mathcal{Y}$ and any $\mathbf{x} \in \mathcal{X}$. Next, we provide an upper bound for $\|\nabla \ell^{\exp}_{t,\widehat{\alpha}}(\mathbf{y}_t^i)\|^2$

$$\begin{aligned}
&\|\nabla \ell^{\exp}_{t,\widehat{\alpha}}(\mathbf{y}_t^i)\|^2 \\
= &\langle \nabla g_t(\mathbf{y}_t) + \widehat{\beta}^* \nabla g_t(\mathbf{y}_t) \nabla g_t(\mathbf{y}_t)^\top (\mathbf{y} - \mathbf{y}_t), \nabla g_t(\mathbf{y}_t) + \widehat{\beta}^* \nabla g_t(\mathbf{y}_t) \nabla g_t(\mathbf{y}_t)^\top (\mathbf{y} - \mathbf{y}_t) \rangle \\
= &\|\nabla g_t(\mathbf{y}_t)\|^2 + 2\widehat{\beta}^* \langle \nabla g_t(\mathbf{y}_t), \mathbf{y} - \mathbf{y}_t \rangle \|\nabla g_t(\mathbf{y}_t)\|^2 + {\widehat{\beta}^*}^2 \|\nabla g_t(\mathbf{y}_t)\|^4 \|\mathbf{y} - \mathbf{y}_t\|^2 \\
\leq &\left(1 + 2{\widehat{\beta}^*}^2 GD \right)^2 \|\nabla g_t(\mathbf{y}_t)\|^2 \leq 4 \|\nabla g_t(\mathbf{y}_t)\|^2.
\end{aligned}$$

Thus, we have

$$\sum_{t=1}^{T} \ell^{\exp}_{t,\widehat{\alpha}}(\mathbf{y}_t^i) - \sum_{t=1}^{T} \ell^{\exp}_{t,\widehat{\alpha}}(\mathbf{x}) \leq \frac{2d}{\widehat{\beta}^*} \log \left(\frac{{\widehat{\beta}^*}^2 D^2}{4d} \sum_{t=1}^{T} \|\nabla g_t(\mathbf{y}_t)\|^2 + 1 \right) + \frac{2}{\widehat{\beta}^*} \tag{57}$$

Substituting (39) and (57) into (56), we have

$$\begin{aligned}
&\sum_{t=1}^{T} f_t(\mathbf{x}_t) - \sum_{t=1}^{T} f_t(\mathbf{x}) \\
\leq &4\Gamma GD \left(2 + \frac{1}{\sqrt{\ln |\mathcal{A}|}} \right) + \frac{\Gamma^2}{2\widehat{\beta}^* \ln |\mathcal{A}|} + \frac{2d}{\widehat{\beta}^*} \log \left(\frac{{\widehat{\beta}^*}^2 D^2}{4d} \sum_{t=1}^{T} \|\nabla g_t(\mathbf{y}_t)\|^2 + 1 \right) + \frac{2}{\widehat{\beta}^*} \qquad (58) \\
\overset{(34)}{\leq} &4\Gamma GD \left(2 + \frac{1}{\sqrt{\ln |\mathcal{A}|}} \right) + \frac{\Gamma^2}{2\widehat{\beta}^* \ln |\mathcal{A}|} + \frac{2d}{\widehat{\beta}^*} \log \left(\frac{{\widehat{\beta}^*}^2 D^2 H}{d} \sum_{t=1}^{T} f_t(\mathbf{x}_t) + 1 \right) + \frac{2}{\widehat{\beta}^*}
\end{aligned}$$

where the last step is due to Lemma 9. Finally, we use the following lemma to simplify the bound.

Lemma 11 (Corollary 5 of Orabona et al. [2012]) *Let $a, b, c, d, x > 0$ satisfy $x - d \leq a \ln(bx + c)$. Then, we have $x - d \leq a \ln(2(ab \ln \frac{2ab}{\mathrm{e}} + db + c))$.*

B.4.3 Analysis for strongly convex functions

Recall that we construct the expert-loss for strongly convex functions as follows

$$\widehat{\ell}^{\mathrm{sc}}_{t,\widehat{\lambda}}(\mathbf{y}) = \langle \nabla g_t(\mathbf{y}_t), \mathbf{y} - \mathbf{y}_t \rangle + \frac{\widehat{\lambda}^*}{2G^2} \|\nabla g_t(\mathbf{y}_t)\|^2 \|\mathbf{y} - \mathbf{x}_t\|^2.$$

Then, we introduce a new decomposition for λ-strongly convex functions

$$
\begin{aligned}
&\sum_{t=1}^{T} f_t(\mathbf{x}_t) - \sum_{t=1}^{T} f_t(\mathbf{x}) \leq \sum_{t=1}^{T} \langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x} \rangle - \frac{\lambda}{2} \sum_{t=1}^{T} \|\mathbf{x}_t - \mathbf{x}\|^2 \\
\leq\ & \sum_{t=1}^{T} \langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x} \rangle - \frac{\widehat{\lambda}^*}{2} \sum_{t=1}^{T} \|\mathbf{x}_t - \mathbf{x}\|^2 \\
\leq\ & \sum_{t=1}^{T} \langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x} \rangle - \frac{\widehat{\lambda}^*}{2G^2} \sum_{t=1}^{T} \|\nabla g_t(\mathbf{y}_t)\|^2 \|\mathbf{x}_t - \mathbf{x}\|^2 \\
\overset{(20)}{\leq}\ & \sum_{t=1}^{T} \langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{x} \rangle - \Delta_T - \frac{\widehat{\lambda}^*}{2G^2} \sum_{t=1}^{T} \|\nabla g_t(\mathbf{y}_t)\|^2 \|\mathbf{x}_t - \mathbf{x}\|^2 \\
\overset{(28)}{=}\ & \underbrace{\sum_{t=1}^{T} \langle \nabla g_t(\mathbf{y}_t), \mathbf{y}_t - \mathbf{y}_t^i \rangle}_{\texttt{meta-regret}} + \underbrace{\sum_{t=1}^{T} \left(\widehat{\ell}_{t,\widehat{\lambda}^*}^{\text{sc}}(\mathbf{y}_t^i) - \widehat{\ell}_{t,\widehat{\lambda}^*}^{\text{sc}}(\mathbf{x}) \right)}_{\texttt{expert-regret}} - \frac{\widehat{\lambda}^*}{2G^2} \sum_{t=1}^{T} \|\nabla g_t(\mathbf{y}_t)\|^2 \|\mathbf{x}_t - \mathbf{y}_t^i\|^2 - \Delta_T
\end{aligned}
\tag{59}
$$

where $\Delta_T = \sum_{t=1}^{T} \mathbb{1}_{\{\langle \nabla f_t(\mathbf{x}_t), \mathbf{v}_t \rangle \geq 0\}} \cdot \langle \nabla f_t(\mathbf{x}_t), \mathbf{y}_t - \mathbf{x}_t \rangle$. To bound the meta-regret, we still incorporate with Lemma 8 to get

$$
\sum_{t=1}^{T} f_t(\mathbf{x}_t) - \sum_{t=1}^{T} f_t(\mathbf{x}) \leq 4\Gamma G D \left(2 + \frac{1}{\sqrt{\ln |\mathcal{A}|}} \right) + \frac{\Gamma^2 G^2}{2\gamma \ln |\mathcal{A}|} + \sum_{t=1}^{T} \left(\widehat{\ell}_{t,\widehat{\lambda}^*}^{\text{sc}}(\mathbf{y}_t^i) - \widehat{\ell}_{t,\widehat{\lambda}^*}^{\text{sc}}(\mathbf{x}) \right).
$$

For the expert-regret, we derive a variant of theoretical guarantee of S^{2}OGD.

Lemma 12 *Under Assumptions 1 and 2, for any expert $\mathbf{y}_t^i \in \mathcal{Y}$ and any $\mathbf{x} \in \mathcal{X}$, we have*

$$
\sum_{t=1}^{T} \widehat{\ell}_{t,\widehat{\lambda}^*}^{\text{sc}}(\mathbf{y}_t^i) - \sum_{t=1}^{T} \widehat{\ell}_{t,\widehat{\lambda}^*}^{\text{sc}}(\mathbf{x}) \leq 1 + \frac{(G+2D)^2}{2\widehat{\lambda}^*} \log \left(\frac{\widehat{\lambda}^*}{(G+2D)^2} \sum_{t=1}^{T} \|\nabla g_t(\mathbf{y}_t)\|^2 + 1 \right)
$$

Combining the above bounds, we have

$$
\begin{aligned}
&\sum_{t=1}^{T} f_t(\mathbf{x}_t) - \sum_{t=1}^{T} f_t(\mathbf{x}) \\
\leq & 4\Gamma G D \left(2 + \frac{1}{\sqrt{\ln |\mathcal{A}|}} \right) + \frac{\Gamma^2 G^2}{2\gamma \ln |\mathcal{A}|} + 1 + \frac{(G+2D)^2}{2\widehat{\lambda}^*} \log \left(\frac{4H\widehat{\lambda}^*}{(G+2D)^2} \sum_{t=1}^{T} f_t(\mathbf{x}) + 1 \right).
\end{aligned}
$$

Finally, we simplify the above bound by utilizing Lemma 11.

C Supporting Lemmas

C.1 Proof of Lemma 4

According to the definition of $\ell_{t,\widehat{\alpha}}^{\exp}(\cdot)$ in (25), we have $\nabla \ell_{t,\widehat{\alpha}}^{\exp}(\mathbf{y}) = \nabla g_t(\mathbf{y}_t) + \widehat{\beta} \nabla g_t(\mathbf{y}_t) \nabla g_t(\mathbf{y}_t)^\top (\mathbf{y} - \mathbf{y}_t)$. Thus, for all $\mathbf{y} \in \mathcal{Y}$, it holds that

$$
\begin{aligned}
\nabla \ell_{t,\widehat{\alpha}}^{\exp}(\mathbf{y}) \nabla \ell_{t,\widehat{\alpha}}^{\exp}(\mathbf{y})^\top &= \nabla g_t(\mathbf{y}_t) \nabla g_t(\mathbf{y}_t)^\top + 2\widehat{\beta} \nabla g_t(\mathbf{y}_t)(\mathbf{y} - \mathbf{y}_t)^\top \nabla g_t(\mathbf{y}_t) \nabla g_t(\mathbf{y}_t)^\top \\
&\quad + \widehat{\beta}^2 \nabla g_t(\mathbf{y}_t) \nabla g_t(\mathbf{y}_t)^\top (\mathbf{y} - \mathbf{y}_t)(\mathbf{y} - \mathbf{y}_t)^\top \nabla g_t(\mathbf{y}_t) \nabla g_t(\mathbf{y}_t)^\top \\
&= \left(1 + \widehat{\beta} \langle \nabla g_t(\mathbf{y}_t), \mathbf{y} - \mathbf{y}_t \rangle \right)^2 \nabla g_t(\mathbf{y}_t) \nabla g_t(\mathbf{y}_t)^\top \\
&\preceq 4 \nabla g_t(\mathbf{y}_t) \nabla g_t(\mathbf{y}_t)^\top = \frac{4}{\widehat{\beta}} \nabla^2 \ell_{t,\widehat{\alpha}}^{\exp}(\mathbf{y})
\end{aligned}
$$

where $\nabla^2 \ell_{t,\widehat{\alpha}}^{\exp}(\mathbf{y})$ denotes the Hessian matrix of $\ell_{t,\widehat{\alpha}}^{\exp}(\mathbf{y})$ and the last inequality is due to a, and the definition of $\widehat{\beta}$. Therefore, $\ell_{t,\widehat{\alpha}}^{\exp}(\cdot)$ is $\frac{\widehat{\beta}}{4}$-exp-concave [Hazan, 2016, Lemma 4.1]. Next, we provide the upper bound of the gradient of $\ell_{t,\widehat{\alpha}}^{\exp}(\cdot)$ as follows

$$\|\nabla \ell_{t,\widehat{\alpha}}^{\exp}(\mathbf{y})\|^2 \overset{(34)}{\leq} (G + 2\widehat{\beta}G^2 D)^2 \leq \frac{25}{16}G^2 \leq 2G^2.$$

This ends the proof.

C.2 Proof of Lemma 6

According to the definition of $\ell_t^{\mathrm{sc}}(\cdot)$ in (14), it holds for any $\mathbf{x}, \mathbf{y} \in \mathcal{X}$ that

$$\ell_t^{\mathrm{sc}}(\mathbf{x}) \geq \ell_t^{\mathrm{sc}}(\mathbf{y}) + \langle \nabla \ell_t^{\mathrm{sc}}(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle + \frac{\lambda}{2}\|\mathbf{x} - \mathbf{y}\|^2.$$

By Definition 1, it can be seen that $\ell_t^{\mathrm{sc}}(\cdot)$ is λ-strongly convex. Next, we provide the upper bound of the gradient of $\ell_t^{\mathrm{sc}}(\cdot)$ as follows

$$\|\nabla \ell_t^{\mathrm{sc}}(\mathbf{y})\|^2 \leq \|\nabla g_t(\mathbf{y}_t) + \lambda(\mathbf{y} - \mathbf{x}_t)\|^2 \overset{(34)}{\leq} (G + 2\lambda D)^2 \leq (G + 2D)^2$$

where the last step is due to our assumption that $\lambda \in [1/T, 1]$.

C.3 Proof of Lemma 7

Similar to analysis of Lemma 6, for any $\mathbf{x}, \mathbf{y} \in \mathcal{X}$, we have

$$\ell_{t,\widehat{\lambda}}^{\mathrm{sc}}(\mathbf{x}) \geq \ell_{t,\widehat{\lambda}}^{\mathrm{sc}}(\mathbf{y}) + \langle \nabla \ell_{t,\widehat{\lambda}}^{\mathrm{sc}}(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle + \frac{\widehat{\lambda}}{2G^2}\|\nabla g_t(\mathbf{y}_t)\|^2 \|\mathbf{x} - \mathbf{y}\|^2$$

By Definition 1, it is established that $\ell_{t,\widehat{\lambda}}^{\mathrm{sc}}(\cdot)$ is $\frac{\widehat{\lambda}}{G^2}\|\nabla g_t(\mathbf{y}_t)\|^2$-strongly convex. Next, we upper bound the gradient of $\ell_{t,\widehat{\lambda}}^{\mathrm{sc}}(\cdot)$ as follows

$$\begin{aligned}
\|\ell_{t,\widehat{\lambda}}^{\mathrm{sc}}(\mathbf{y})\|^2 &\leq \left\langle \nabla g_t(\mathbf{y}_t) + \frac{\widehat{\lambda}}{G^2}\|\nabla g_t(\mathbf{y}_t)\|^2(\mathbf{y} - \mathbf{x}_t), \nabla g_t(\mathbf{y}_t) + \frac{\widehat{\lambda}}{G^2}\|\nabla g_t(\mathbf{y}_t)\|^2(\mathbf{y} - \mathbf{x}_t) \right\rangle \\
&= \|\nabla g_t(\mathbf{y}_t)\|^2 + \frac{2\widehat{\lambda}}{G^2}\|\nabla g_t(\mathbf{y}_t)\|^2 \langle \nabla g_t(\mathbf{y}_t), \mathbf{y} - \mathbf{x}_t \rangle + \frac{\widehat{\lambda}^2}{G^4}\|\nabla g_t(\mathbf{y}_t)\|^4 \|\mathbf{y} - \mathbf{x}_t\|^2 \\
&\overset{(34)}{\leq} \left(1 + \frac{2\widehat{\lambda}D}{G}\right)^2 \|\nabla g_t(\mathbf{y}_t)\|^2 \leq \left(1 + \frac{2D}{G}\right)^2 \|\nabla g_t(\mathbf{y}_t)\|^2
\end{aligned}$$

where the last step is due to our assumption that $\widehat{\lambda} \in [1/T, 1]$.

C.4 Proof of Lemma 12

The analysis is similar to Wang et al. [2020b]. Let $\widetilde{\mathbf{y}}_{t+1}^i = \mathbf{y}_t^i - \frac{1}{\eta_t}\nabla \ell_{t,\widehat{\alpha}}^{\mathrm{sc}}(\mathbf{y}_t^i)$. According to the definition of (28), we have

$$\begin{aligned}
\ell_{t,k}^{\mathrm{sc}}(\mathbf{y}_t^i) - \ell_{t,k}^{\mathrm{sc}}(\mathbf{x}) &\leq \langle \nabla \ell_{t,k}^{\mathrm{sc}}(\mathbf{y}_t^i), \mathbf{y}_t^i - \mathbf{x} \rangle - \frac{\widehat{\lambda}}{2G^2}\|\nabla g_t(\mathbf{y}_t)\|^2 \|\mathbf{y}_t^i - \mathbf{x}\|^2 \\
&= \eta_t \langle \mathbf{y}_t^i - \widetilde{\mathbf{y}}_{t+1}^i, \mathbf{y}_t^i - \mathbf{x} \rangle - \frac{\widehat{\lambda}}{2G^2}\|\nabla g_t(\mathbf{y}_t)\|^2 \|\mathbf{y}_t^i - \mathbf{x}\|^2.
\end{aligned}$$

For the first term, it can be verified that

$$\begin{aligned}
\langle \mathbf{y}_t^i - \widetilde{\mathbf{y}}_{t+1}^i, \mathbf{y}_t^i - \mathbf{x} \rangle &= \|\mathbf{y}_t^i - \mathbf{x}\|^2 + \langle \mathbf{x} - \widetilde{\mathbf{y}}_{t+1}^i, \mathbf{y}_t^i - \mathbf{x} \rangle \\
&= \|\mathbf{y}_t^i - \mathbf{x}\|^2 - \|\widetilde{\mathbf{y}}_{t+1}^i - \mathbf{x}\|^2 - \langle \mathbf{y}_t^i - \widetilde{\mathbf{y}}_{t+1}^i, \widetilde{\mathbf{y}}_{t+1}^i - \mathbf{x} \rangle \\
&= \|\mathbf{y}_t^i - \mathbf{x}\|^2 - \|\widetilde{\mathbf{y}}_{t+1}^i - \mathbf{x}\|^2 + \|\widetilde{\mathbf{y}}_{t+1}^i - \mathbf{y}_t^i\|^2 + \langle \widetilde{\mathbf{y}}_{t+1}^i - \mathbf{y}_t^i, \mathbf{y}_t^i - \mathbf{x} \rangle
\end{aligned}$$

which implies that

$$\langle \mathbf{y}_t^i - \widetilde{\mathbf{y}}_{t+1}^i, \mathbf{y}_t^i - \mathbf{x} \rangle = \frac{1}{2}\left(\|\mathbf{y}_t^i - \mathbf{x}\|^2 - \|\widetilde{\mathbf{y}}_{t+1}^i - \mathbf{x}\|^2 + \|\widetilde{\mathbf{y}}_{t+1}^i - \mathbf{y}_t^i\|^2 \right).$$

Thus,

$$\begin{aligned} \ell_{t,k}^{\text{sc}}(\mathbf{y}_t^i) - \ell_{t,k}^{\text{sc}}(\mathbf{w}) \leq & \frac{\eta_t}{2}\left(\|\mathbf{y}_t^i - \mathbf{x}\|^2 - \|\widetilde{\mathbf{y}}_{t+1}^i - \mathbf{x}\|^2 \right) \\ & + \frac{1}{2\eta_t}\|\nabla \ell_{t,\widehat{\alpha}}^{\text{sc}}(\mathbf{y}_t^i)\|^2 - \frac{\widehat{\lambda}}{2G^2}\|\nabla g_t(\mathbf{y}_t)\|^2 \|\mathbf{y}_t^i - \mathbf{x}\|^2. \end{aligned}$$

Summing the above bound up over $t = 1$ to T, we attain

$$\begin{aligned} & \sum_{t=1}^{T} \ell_{t,\widehat{\alpha}}^{\text{sc}}(\mathbf{y}_t^i) - \sum_{t=1}^{T} \ell_{t,\widehat{\alpha}}^{\text{sc}}(\mathbf{x}) \\ \leq & \frac{\eta_1}{2}\|\mathbf{y}_1^i - \mathbf{x}\|^2 + \sum_{t=1}^{T}\left(\eta_t - \eta_{t-1} - \frac{\widehat{\lambda}}{G^2}\|\nabla g_t(\mathbf{y}_t)\|^2 \right)\frac{\|\mathbf{y}_t^i - \mathbf{x}\|^2}{2} + \sum_{t=1}^{T}\frac{1}{2\eta_t}\|\nabla \ell_{t,\widehat{\alpha}}^{\text{sc}}(\mathbf{y}_t^i)\|^2 \\ \leq & 1 + \sum_{t=1}^{T}\frac{1}{2\eta_t}\|\nabla \ell_{t,\widehat{\lambda}}^{\text{sc}}(\mathbf{y}_t^i)\|^2 \leq 1 + \frac{(G+2D)^2}{2\widehat{\lambda}}\sum_{t=1}^{T}\frac{\|\nabla g_t(\mathbf{y}_t)\|^2}{(G+2D)^2/\widehat{\lambda} + \sum_{i=1}^{t}\|\nabla g_i(\mathbf{y}_i)\|^2}. \end{aligned}$$

where the last two inequalities is due to $\eta_t = (1 + 2D/G)^2 + \frac{\widehat{\lambda}}{G^2}\sum_{i=1}^{t}\|\nabla g_i(\mathbf{y}_i)\|^2$ which is specifically set for new expert-loss. Further, we will use the following lemma to bound the last term.

Lemma 13 (Lemma 11 of Hazan et al. [2007]) *Let $l_1, \cdots, l_T$ and δ be non-negative real numbers. Then, we have $\sum_{t=1}^{T} \frac{l_t^2}{\sum_{i=1}^{t} l_i^2 + \delta} \leq \log\left(\frac{1}{\delta}\sum_{t=1}^{T} l_t^2 + 1 \right)$.*

This completes the proof of Lemma 12.

C.5 Proof of Lemma 5

The analysis is similar to Mhammedi et al. [2019, Lemma 9]. When $\|\widehat{\mathbf{y}}_{t+1}^i\| \geq D$, then we need to solve the following quadratic problem:

$$\mathbf{y}_{t+1}^i = \operatorname*{arg\,min}_{\mathbf{y} \in \mathcal{Y}} (\widehat{\mathbf{y}}_{t+1}^i - \mathbf{y})^\top \Sigma_{t+1} (\widehat{\mathbf{y}}_{t+1}^i - \mathbf{y}).$$

We use the Lagrangian multiplier to solve the above problem

$$\mathcal{L}(\mathbf{y}, \mu) = (\widehat{\mathbf{y}}_{t+1}^i - \mathbf{y})^\top \Sigma_{t+1} (\widehat{\mathbf{y}}_{t+1}^i - \mathbf{y}) + \mu(\mathbf{y}^\top \mathbf{y} - D^2).$$

We set $\frac{\partial \mathcal{L}}{\partial \mathbf{y}} = \mathbf{0}$ to attain $\Sigma_{t+1}(\mathbf{y} - \widehat{\mathbf{y}}_{t+1}^i) + \mu \mathbf{y} = 0$, which implies

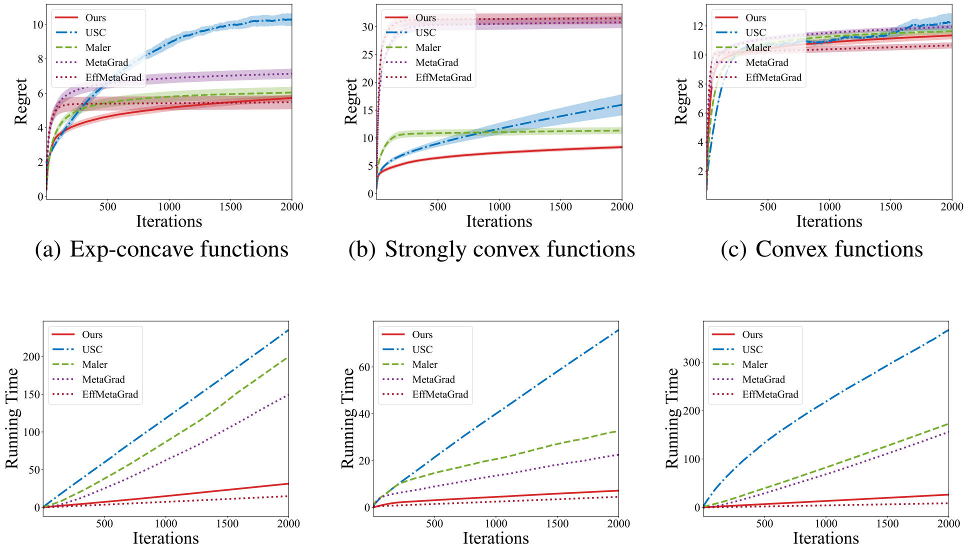
$$\mathbf{y} = (\mu \mathbf{I}_d + \Sigma_{t+1})^{-1} \Sigma_{t+1} \widehat{\mathbf{y}}_{t+1}^i = \mathbf{Q}_{t+1}^\top \left(x\mathbf{I} + \Lambda_{t+1} \right)^{-1} \mathbf{Q}_{t+1} \Sigma_{t+1} \widehat{\mathbf{y}}_{t+1}^i$$

where $x = \mu + 1/(\widehat{\beta}^2 D^2)$. Due to $\mathbf{y}^\top \mathbf{y} = D^2$, x is the solution of the following problem

$$\rho(x) := \sum_{k=1}^{d} \frac{\langle \mathbf{e}_k, \mathbf{Q}_{t+1} \Sigma_{t+1} \widehat{\mathbf{y}}_{t+1}^i \rangle^2}{(x + \lambda_t^k)^2} = D^2.$$

D Clarifications on bounded modulus

In this section, we explain that bounded moduli are generally acceptable in practical scenarios, which is also stated in previous study [Zhang et al., 2022]. Taking λ-strongly convex functions as an example, we assume that $\lambda \in [1/T, 1]$, since other cases that $\lambda < 1/T$ and $\lambda > 1$ can be disregarded. (i) If $\lambda < 1/T$, the regret bound for strongly convex functions becomes $\Omega(T)$, which cannot benefit from strong convexity. Therefore, we should treat these functions as general convex functions. (ii) If $\lambda > 1$, λ-strongly convex functions are also 1-strongly convex according to Definition 1. Thus, we can treat these functions as 1-strongly convex functions.



(a) Exp-concave functions (b) Strongly convex functions (c) Convex functions

(d) Exp-concave functions (e) Strongly convex functions (f) Convex functions

Figure 1: Regret (first row) and running time (second row) of different methods.

E Experiments

In this section, we conduct empirical experiments to validate the effectiveness of our proposed methods, and present the details of experiments.

Settings We conduct experiments on the ijcnn1 dataset from LIBSVM Data [Chang and Lin, 2011, Prokhorov, 2001], where the dimension of features is $d = 22$. We consider the following online classification problem. In each round $t \in [T]$, the online learner chooses a decision $\mathbf{x}_t \in \mathcal{X}$. After submitting the decision, the online learner receives a batch of data samples $\{(x_t^{(i)}, y_t^{(i)})\}_{i=1}^m$ which are sampled from the dataset, where $x_t^{(i)}$ is the feature vector of the i-th sample, and $y_t^{(i)}$ is the corresponding label. The learner can evaluate the model by the online convex loss $f_t(\mathbf{x}_t)$ and update the decision for the next round. In our study, we set $T = 2000$, the domain diameter as $D = 20$, and the gradient norm upper bound as $G = \sqrt{22}$. Following the general setup of Zhao et al. [2022], we set the feasible domain to be an ellipsoid $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^d \mid \mathbf{x}^\top \mathbf{E}\mathbf{x} \leq \lambda_{\min}(\mathbf{E}) \cdot (D/2)^2\}$, where $\mathbf{E}$ is a certain diagonal matrix and $\lambda_{\min}(\mathbf{E})$ denotes its minimum eigenvalue. We remark that the cost of one projection onto $\mathcal{X}$ is generally expensive since it requires solving a convex programming.

In the following, we consider three types of online convex functions to simulate the unknown environment and demonstrate the universality of our method. First, for exp-concave functions, the online learner suffers a logistic loss: $f_t(\mathbf{x}_t) = \frac{1}{m}\sum_{i=1}^m \log\left(1 + \exp(-y_t^{(i)}\mathbf{x}_t^\top x_t^{(i)})\right)$. Second, for strongly convex functions, we choose the regularized hinge loss: $f_t(\mathbf{x}_t) = \frac{1}{m}\sum_{i=1}^m \max\left(0, 1 - y_t^{(i)}\mathbf{x}_t^\top x_t^{(i)}\right) + \frac{\lambda}{2}\|\mathbf{x}_t\|^2$. Third, for general convex functions, the online learner suffers the absolute loss: $f_t(\mathbf{x}_t) = \frac{1}{m}\sum_{i=1}^m \left|\mathbf{x}_t^\top x_t^{(i)} - y_t^{(i)}\right|$. Based on the above experimental settings, we conduct the empirical studies of our method, as well as other universal algorithms in the literature.

Algorithms We compare the performance of our proposed method for minimax universal regret with existing universal algorithms, including MetaGrad [van Erven and Koolen, 2016], Maler [Wang et al., 2019], efficient implementation of MetaGrad [Mhammedi et al., 2019], and USC [Zhang et al., 2022]. All the baselines share the same experimental setting as our method.

Table 2: A summary of state-of-the-art projection-free algorithms for different types of convex functions.

Algorithm	Condition on Loss	Regret Bound
OFW [Hazan and Kale, 2012]	convex	$O(T^{3/4})$
OSPF [Hazan and Minasyan, 2020]	convex and smooth	$O(T^{2/3})$
SC-OFW [Wan and Zhang, 2021]	strongly convex	$O(T^{2/3})$
AFP-ONS [Garber and Kretzu, 2023]	exp-concave and smooth	$O(T^{2/3})$

Results We repeat the experiments for five times and record the results in Figure 1. We conduct the experiments on a machine with a single CPU (Apple M1 pro) and 16GB memory. We record both regret and running time (in seconds) for all methods. As shown in Figure 1, the running time of our method is comparable to that of EffMetaGrad, yet it achieves better results for strongly convex functions. Compared to other algorithms which conduct $O(\log T)$ projections, i.e., USC, Maler, and MetaGrad, the running time of our projection-efficient method is 5 to 20 times faster, and it also attains nearly optimal regret for three types of convex functions. In conclusion, the empirical results demonstrate the effectiveness of our method in achieving optimal regret guarantee and also significant enhancement in computational efficiency.

F Further discussion on projection-free algorithms

In the literature, there exists a class of projection-free algorithms [Hazan and Kale, 2012, Hazan and Minasyan, 2020, Wan and Zhang, 2021, Wan et al., 2021b, 2022, Wang et al., 2023, Garber and Kretzu, 2023]. Therefore, it is natural to ask whether projection-free algorithms such as variants of Online Frank Wolfe could be used instead of OGD and ONS to remove all projections while still being adaptive to the smoothness. Here, we provide some targeted discussions on this matter.

In fact, we can choose projection-free algorithms as the expert-algorithms. However, given the current studies on projection-free algorithms, this approach will lead to a deterioration of the regret bound and can not handle certain cases. As is shown in Table 2, in the literature, there are no suitable projection-free algorithms for exp-concave functions, neither for strongly convex and smooth functions. Moreover, when functions are smooth, existing projection-free algorithms are unable to achieve problem-dependent bounds, such as the small-loss bounds in this work.

Finally, we would like to highlight that although using projection-free algorithms can remove all projections, they may not achieve greater efficiency based on the universal framework. Specifically, most projection-free algorithms, such as OFW and its variants, replace the original projection operation with a linear optimization step. Since the universal framework requires maintaining $O(\log T)$ expert-algorithms, this approach needs to perform $O(\log T)$ linear optimization steps per round, which can be time-consuming when T is large.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our contributions and scope are clearly written in the abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See the future work in Section 5.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: See the assumptions in Section 2.1. The complete proofs can be found in Appendix B.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All the information needed to reproduce the experimental results is provided.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: Due to privacy concerns, we do not include the code.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: See the experiments in Appendix E.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: See the experiments in Appendix E.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: See the experiments in Appendix E.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have read the ethics carefully.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: This work is mainly theoretical.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to

generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This work is mainly theoretical.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite the creators of the dataset used in our experiments properly.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: This work does not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This work does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This work does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.