# A Universal Proximal Oracle for Optimization, Sampling, and Online Learning

Jiaming Liang * Yongxin Chen †

June 30, 2023

**Abstract**

We consider sampling problems with possibly non-smooth potentials (negative log-densities). In particular, we study two specific settings of sampling where the convex potential is either semi-smooth or in composite form as the sum of a smooth component and a semi-smooth component. To overcome the challenges caused by the non-smoothness, we propose a Markov chain Monte Carlo algorithm that resembles proximal methods in optimization for these sampling tasks. The key component of our method is a sampling scheme for a quadratically regularized target potential. This scheme relies on rejection sampling with a carefully designed Gaussian proposal whose center is an approximate minimizer of the regularized potential. We develop a novel technique (a modified Gaussian integral) to bound the complexity of this rejection sampling scheme in spite of the non-smoothness in the potentials. We then combine this scheme with the alternating sampling framework (ASF), which uses Gibbs sampling on an augmented distribution, to accomplish the two settings of sampling tasks we consider. Furthermore, by combining the complexity bound of the rejection sampling we develop and the remarkable convergence properties of ASF discovered recently, we are able to establish several non-asymptotic complexity bounds for our algorithm, in terms of the total number of queries of subgradient of the target potential. Our algorithm achieves state-of-the-art complexity bounds compared with all existing methods in the same settings.

**Key words.** High-dimensional sampling, non-smooth potential, semi-smooth potential, composite potential, complexity analysis, alternating sampling framework, rejection sampling, proximal bundle method, restricted Gaussian oracle

## 1 Introduction

Read Lin Xiao's "Dual Averaging Methods for Regularized Stochastic Learning and Online Optimization" and John Duchi's "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization" when revising this paper and discussing the application in online learning

Drawing samples from a given (often unnormalized) probability density plays a crucial role in many scientific and engineering problems that face uncertainty (either physically or algorithmically). Sampling algorithms are widely used in many areas such as statistical inference/estimation, operations research, physics, biology, and machine learning, etc [3, 14, 16, 19, 22, 23, 24, 48]. For instance, in Bayesian inference, one draws samples from the posterior distribution to infer its mean, covariance, or other important statistics. Sampling is also heavily used in molecular dynamics to discover new molecular structures.

*Department of Computer Science, Yale University, New Haven, CT 06511. (email: jiaming.liang@yale.edu).

†School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA, 30332. (email: yongchen@gatech.edu).

Over the past decades, many algorithms have been developed for sampling [1, 8, 16, 25, 26, 34, 46]. A widely used framework for sampling from complex distributions is the Markov chain Monte Carlo (MCMC) algorithm [7, 9, 10, 13, 14]. In MCMC, a Markov chain is constructed so that its invariant distribution is the given target distribution we want to sample from. Several widely used MCMC methods include Langevin Monte Carlo (LMC) [12, 21, 40, 43], Metropolis-adjusted Langevin algorithm (MALA) [4, 42, 43], and Hamiltonian Monte Carlo (HMC) [37]. All these three algorithms use gradient information of the potential (negative log-density) to construct the Markov chain. Recently, many theoretical results (see [8, 12, 13, 15, 25, 27, 41, 43] and references therein) have been established to understand the complexities of these MCMC algorithms. A typical assumption in most of these theoretical results is that the target potential is smooth [12, 25, 51], i.e., its gradient is Lipschitz continuous.

In this work we consider sampling problems where the target potential is not smooth. Many applications of sampling involve potentials that lack smoothness. For instance, in Bayesian inference, the prior is naturally non-smooth when its support is constrained. Many problems in deep learning are also non-smooth, not only due to non-smooth activation functions like ReLU used in the neural networks, but also due to intrinsic scaling symmetries. Nevertheless, the study of such sampling problems without smoothness is nascent, compared with that for smooth potentials.

This work is along the recent line of researches that lie in the interface of sampling and optimization [13, 45]. Indeed, sampling is closely related to optimization. On the one hand, optimization can be viewed as the limit of sampling when the temperature parameter, or equivalently the randomness in the problem, goes to 0. On the other hand, sampling can be viewed as an optimization over the manifold of probability distributions [51, 53]. The popular gradient-based MCMC methods such as LMC, MALA, and HMC resemble the gradient-based algorithms in optimization and can be viewed as the sampling counterparts of them. However, in sharp contrast to optimization where a plethora of algorithms, e.g., subgradient method, proximal algorithm, bundle method have been developed for non-smooth optimization [29, 30, 32, 33, 35, 44, 52], the sampling problem without smoothness remains largely unexplored, compared with its smooth counterpart.

The goal of this paper is to develop an efficient algorithm to draw samples from potentials that lack smoothness. We consider two specific settings where the convex potential is either semi-smooth (i.e., the (sub)gradient of the potential is Hölder-continuous with exponent $\alpha \in [0, 1]$) or composite with a semi-smooth component. This is a non-trivial extension of our previous work [31] where the target potential is assumed to be convex and Lipschitz continuous. The core of our algorithm is a scheme to sample from a quadratically regularized version of the target potential. Our scheme is based on rejection sampling with a carefully designed Gaussian proposal whose center is an approximate minimizer of the regularized potential. We develop a novel technique to bound the complexity of our sampling scheme by estimating a modified Gaussian integral. Moreover, we establish an iteration-complexity bound for the proximal bundle optimization algorithm we use to compute an approximate minimizer of the regularized potential so that total complexity of our sampling scheme is properly bounded.

To sample from the original target distribution, we utilize the alternative sampling framework (ASF) [25]. The latter is an application of Gibbs sampling over a specially designed distribution that augments the target one. The ASF has shown to exhibit remarkable convergence properties under mild assumptions [6]. In particular, smoothness of the target potential is not required to ensure global convergence. To use ASF in practice, one needs to realize, in each iteration, a step known as the restricted Gaussian oracle (RGO) [25]. However, except for some very special cases e.g., the target potential is decomposable along each dimension, it was not clear how to implement RGO efficiently. It turns out that the sampling scheme we developed to sample from a quadratically regularized potential of the target potential is exactly an RGO for the target distribution. Thus,

by combining ASF and the sampling scheme we developed, we establish an efficient algorithm to sample from any convex semi-smooth potential and any composite potential with a semi-smooth component. Moreover, the complexity of the resulting algorithm can be bounded by combining that of our sampling scheme and that of the ASF. We summarize our contributions as follows.

i) We develop an efficient scheme to sample from a quadratically regularized potential that lacks smoothness and establish novel techniques to bound its complexity.

ii) We combine our sampling scheme and the ASF to form a general sampling algorithm for convex semi-smooth potentials as well as composite potentials with semi-smooth components. The complexity of our algorithm is better than all existing results under the same assumptions.

iii) Though these are not the focus of this work, we establish complexity bounds of the proximal bundle subroutine for convex semi-smooth functions and composite functions with semi-smooth components. These results can be readily used to bound the iteration-complexity of the proximal bundle algorithm in optimization for these functions.

**Related Works:** Several new algorithms and theoretical results in sampling with semi-smooth and composite potentials have been established over the last few years. We begin with the literature on sampling from semi-smooth potentials or non-smooth potentials. In [28], the author developed the projected LMC algorithm and analyzed its complexity for non-smooth potentials. In [13], the authors presented an optimization approach to analyze the complexity of sampling and established a complexity result for sampling with non-smooth potentials. In [5], the authors proposed an LMC type algorithm for sampling from semi-smooth potentials based on Gaussian smoothing. In [11, 17], the authors analyzed LMC under functional inequalities and semi-smoothness (except the non-smooth case) and established corresponding complexity results.

Next we consider sampling algorithms for composite potentials of the form $\exp(-f_1-f_2)$, where $f_1$ is convex and smooth, and $f_2$ is convex and non-smooth/semi-smooth. In [36], the authors developed an algorithm that needs an oracle to sample from the target potential regularized by a large isotropic quadratic term and to compute the corresponding partition function, akin to the RGO used in ASF [47]. In [14], the authors introduced an algorithm by running LMC on the Moreau envelope of the potential. In [13], the authors proposed an algorithm embedding the proximal map of $f_2$ into LMC and analyzed the convergence of the average of distributions over iterates. In [18], the authors improved the results in [13] for the cases where $f_2$ is an isotropic quadratic term. The paper [45] provided a primal-dual interpretation of the algorithm proposed in [13], and established a slightly improved complexity result when the smooth part $f_1$ is also strongly convex. In [2], the authors also examined the problem from an optimization perspective and established a complexity bound in the cases where $f_1$ is strongly convex. Another approach for sampling from the composite density $\exp(-f_1 - f_2)$ is to apply LMC on the Gaussian smoothing of the potentials. In [5], the authors proposed this algorithm for sampling from composite densities. Following this paper, [38] further developed algorithms based on generalized Gaussian smoothing and obtained improved results when $f_1$ is strongly convex.

We compare these existing complexity bounds with our results in Tables 1 and 2. We highlight only the dependence of the complexity on the dimension $d$, the accuracy $\varepsilon$, the level of smoothness $\alpha$ and the semi-smoothness coefficient $L_\alpha$, the smoothness coefficient $L_1$ of $f_1$, and the strong convexity coefficient $\lambda$ of $f_1$ if it is positive. To this end, we make the following simplifications. The initial distance (either in KL/Rényi or in $W_2^2$) to the target distribution is set to be $\tilde{\mathcal{O}}(d)$. The forth order moment $\mathcal{M}_4 = \mathcal{O}(d^2)$. We also omit the dependence of the Poincaré coefficient if there is any. We make the following remarks regarding these complexity bounds. Complexity results

established in [13] is for the average of distributions over iterates, while other results presented in Tables 1 and 2 are for the distributions over the last iterates. The non-smooth case (i.e., $\alpha = 0$) is not covered in complexity results of [11] and [17] in Table 1, and the result of [38] in Table 2, as these bounds blow up as $\alpha \to 0$. Papers [2, 5, 13, 14, 45] rely on the composite form $f_1 + f_2$ of the composite potentials and the proximal map of $f_2$. In contrast, our algorithm does not depend the decomposition of the potential and does not necessarily require the proximal map of $f_2$.

| Source | Complexity | Assumption | Metric |
|---|---|---|---|
| [5] | $\tilde{\mathcal{O}}\left(\frac{L_\alpha^{6/(1+\alpha)} d^{8-3\alpha}}{\varepsilon^{(10+4\alpha)/(1+\alpha)}}\right)$ | semi-smooth | TV |
| [13] | $\mathcal{O}\left(\frac{L_0^2 d}{\varepsilon^2}\right)$ | non-smooth | KL |
| [28] | $\tilde{\mathcal{O}}\left(\frac{L_0^2 d^2}{\varepsilon^2}\right)$ | non-smooth | $W_1^2$ |
| [11] | $\tilde{\mathcal{O}}\left(\frac{L_\alpha^{2/\alpha} d^{2+1/\alpha}}{\varepsilon^{1/\alpha}}\right)$ | semi-smooth | Rényi |
| [17] | $\tilde{\mathcal{O}}\left(\frac{L_\alpha^{2/\alpha} d^{2+3/\alpha}}{\varepsilon^{1/\alpha}}\right)$ | semi-smooth | KL |
| this paper (Thm. **??**) | $\tilde{\mathcal{O}}\left(\frac{L_\alpha^{2/(1+\alpha)} d^2}{\varepsilon}\right)$ | semi-smooth | TV |
| this paper (Thm. 5.7) | $\tilde{\mathcal{O}}\left(L_\alpha^{2/(1+\alpha)} d^2\right)$ | semi-smooth | Rényi |

Table 1: Complexity bounds for sampling from semi-smooth/non-smooth potentials.

| Source | Complexity | Assumption | Metric |
|---|---|---|---|
| [5] | $\tilde{\mathcal{O}}\left(\frac{L_\alpha^{6/(1+\alpha)} d^{5-3\alpha}}{\lambda^3 \varepsilon^{(7+\alpha)/(1+\alpha)}}\right)$ | smooth+semi-smooth | TV |
| [14] | $\tilde{\mathcal{O}}\left(\frac{L_0^2 d^5}{\varepsilon^2}\right)$ | smooth+non-smooth | TV |
| [13] | $\mathcal{O}\left(\frac{L_1 d^2 + L_0^2 d}{\varepsilon^2}\right)$ | smooth+non-smooth | KL |
| [45] | $\tilde{\mathcal{O}}\left(\frac{L_0^2 + L_1 d}{\lambda^2 \varepsilon}\right)$ | smooth+non-smooth | $W_2^2$ |
| [38] | $\tilde{\mathcal{O}}\left(\frac{(L_\alpha \vee L_1)^{2/\alpha} d^{1/\alpha}}{\lambda^{1+1/\alpha} \varepsilon^{1/\alpha}}\right)$ | smooth+semi-smooth | KL |
| [2] | $\tilde{\mathcal{O}}\left(\frac{L_0^2 d}{\lambda \varepsilon^4}\right)$ | smooth+non-smooth | $W_2$ |
| this paper (Thm. 5.10) | $\tilde{\mathcal{O}}\left(\frac{(L_\alpha^{2/(\alpha+1)} \vee L_1) d^2}{\varepsilon}\right)$ | smooth+semi-smooth | TV |
| this paper (Thm. 5.10) | $\tilde{\mathcal{O}}\left((L_\alpha^{2/(\alpha+1)} \vee L_1) d^2\right)$ | smooth+semi-smooth | Rényi |
| this paper (Thm. **??**) | $\tilde{\mathcal{O}}\left(\frac{(L_\alpha^{2/(\alpha+1)} \vee L_1) d}{\lambda}\right)$ | smooth+semi-smooth | Rényi |

Table 2: Complexity bounds for sampling from composite potentials.

## 2 Background, Proximal operator and motivating examples

motivation, significance, challenges

We are interested in sampling problems associated with convex potentials that are not necessarily smooth. In particular, we consider two specific scenarios of sampling tasks with target distribution

$$\nu \propto \exp(-f(x)) \tag{1}$$

in $\mathbb{R}^d$. In the first setting, the potential $f$ is assumed to be convex and semi-smooth, i.e.,

$$\|f'(u) - f'(v)\| \leq L_\alpha \|u - v\|^\alpha, \quad \forall u, v \in \mathbb{R}^d \tag{2}$$

for some $\alpha \in [0, 1]$ and coefficient $L_\alpha > 0$, where $f'$ denotes a subgradient of $f$. Clearly, when $\alpha = 0$, (2) reduces to a Lipschitz continuous condition, and when $\alpha = 1$, it reduces to a smoothness condition. In the second setting, the potential is assumed to be composite as $f(x) = f_1(x) + f_2(x)$ with $f_1$ being convex and smooth and $f_2$ being convex and semi-smooth.

Most existing gradient-based sampling algorithms are not applicable to these problems due to the lack of smoothness. In this work, we develop **a proximal algorithm for sampling from semi-smooth potentials and composite potentials**. Our method is based on the alternating sampling framework (ASF) introduced in [25], which is a generic framework for sampling from a distribution $\pi^X(x) \propto \exp(-g(x))$. Starting from a point $x_0 \in \mathbb{R}^d$, the alternating sampling framework with stepsize $\eta > 0$ repeats the two steps as in Algorithm 1.

**Algorithm 1** Alternating Sampling Framework [25]

1. Sample $y_k \sim \pi^{Y|X}(y \mid x_k) \propto \exp[-\frac{1}{2\eta}\|x_k - y\|^2]$
2. Sample $x_{k+1} \sim \pi^{X|Y}(x \mid y_k) \propto \exp[-g(x) - \frac{1}{2\eta}\|x - y_k\|^2]$

Apparently, the ASF is a special case of Gibbs sampling [20] of the joint distribution

$$\pi(x, y) \propto \exp(-f(x) - \frac{1}{2\eta}\|x - y\|^2). \tag{3}$$

In Algorithm 1, sampling $y_k$ given $x_k$ in step 1 can be easily done since $\pi^{Y|X}(y \mid x_k) = \mathcal{N}(x_k, \eta I)$ is a simple Gaussian distribution. Sampling $x_{k+1}$ given $y_k$ in step 2 is however a nontrivial task; it corresponds to the so-called restricted Gaussian oracle for $g$ introduced in [25], which is defined as follows.

**Definition 2.1.** *Given a point $y \in \mathbb{R}^d$ and stepsize $\eta > 0$, the restricted Gaussian oracle (RGO) for $f : \mathbb{R}^d \to \mathbb{R}$ is a sampling oracle that returns a random sample from a distribution proportional to $\exp(-f(\cdot) - \|\cdot - y\|^2/(2\eta))$.*

The RGO is an analogy of the proximal map

$$\mathrm{Prox}_{\eta g}(y) := \mathrm{argmin}_x[g(x) + \frac{1}{2\eta}\|x - y\|^2]$$

in optimization, which is widely used in proximal algorithms for optimization [39]. To use the ASF in practice, one needs to efficiently implement the RGO. Some examples of $g$ that admits a computationally efficient RGO have been presented in [36, 47]. These instances of $g$ have simple structures such as coordinate-separable regularizers, $\ell_1$-norm, and group Lasso. For general $g$, especially non-smooth ones considered in this work, it was not clear how to realize the RGO efficiently.

# 3 Proximal operator algorithms and complexities

## 3.1 A proximal bundle method subroutine for semi-smooth optimization

We consider the optimization problem

$$f_y^\eta(x^*) = \min\left\{f_y^\eta(x) = f(x) + \frac{1}{2\eta}\|x - y\|^2 : x \in \mathbb{R}^d\right\}, \tag{4}$$

and we aim at obtaining a $\delta$-solution, i.e., a point $\bar{x}$ such that

$$f_y^\eta(\bar{x}) - f_y^\eta(x^*) \le \delta. \tag{5}$$

To achieve this goal, we borrow ideas from the proximal bundle method [32, 33], which is an efficient algorithm for solving convex non-smooth optimization problems. The proximal bundle method solves a non-smooth optimization via sequentially solving a sequence of sub-problems of the form (4) approximately. We adopt this subroutine in the proximal bundle method to obtain a $\delta$-solution to (4). This is summarized in Algorithm 2.

We remark that though Algorithm 2 is widely used in the proximal bundle method and is not new, the complexity analysis (Theorem 3.5) of it associated with a semi-smooth function $f$ is novel.

To our best knowledge, Theorem 3.5 is the first iteration-complexity result for the optimization of the form (4) for semi-smooth functions. This result can be readily used for complexity analysis of the proximal bundle method for semi-smooth functions (This is not the focus of this paper and will not be explored here).

**Algorithm 2** Proximal Bundle Method Subroutine

1. Let $y \in \mathbb{R}^d$, $\eta > 0$, and $\delta > 0$ be given, and set $x_0 = \tilde{x}_0 = y$, and $j = 1$
2. Update $f_j(x) = \max\{f(x_i) + \langle f'(x_i), x - x_i\rangle : 0 \le i \le j-1\}$
3. Compute
$$x_j = \operatorname*{argmin}_{u\in\mathbb{R}^d} \left\{ f_j^\eta(x) := f_j(x) + \frac{1}{2\eta}\|x-y\|^2 \right\}, \tag{6}$$
$$\tilde{x}_j \in \text{Argmin}\left\{ f_y^\eta(x) : x \in \{x_j, \tilde{x}_{j-1}\}\right\} \tag{7}$$
4. If $f_y^\eta(\tilde{x}_j) - f_j^\eta(x_j) \le \delta$, then **stop** and **return** $J = j, x_J, \tilde{x}_J$; else, go to step 5
5. Set $j \leftarrow j+1$ and go to step 2.

The basic idea of Algorithm 2 is to approximate the non-smooth part of the objective function $f_y^\eta$ with piece-wise affine functions constructed by a collection of cutting-planes and solve the resulting simplified problem. As the approximation becomes more and more accurate, the solutions to the approximated problems converge to that of (4). We make several remarks regarding Algorithm 2. First, $f_j$ is the standard cutting-plane model and $\{f_j\}$ is a sequence of increasing functions underneath $f$. Second, (50) can be reformulated into convex quadratic programming with $j$ affine constraints.

We next show Algorithm 2 can compute a $\delta$-solution to (4) within a reasonable number of iterations. To this end, we present a technical lemma for Algorithm 2. This lemma is also useful in the complexity analysis in Section **??** for sampling.

**Lemma 3.1.** *Assume $f$ is convex and $L_\alpha$-semi-smooth. Let $J, x_J, \tilde{x}_J$ be the outputs of Algorithm 2, then the following statements hold:*

*a) $f_j(x) \le f_{j+1}(x)$ and $f_j(x) \le f(x)$ for $\forall x \in \mathbb{R}^d$ and $\forall j$;*

*b) $f_j^\eta(x_j) + \|x - x_j\|^2/(2\eta) \le f_j^\eta(x)$ for $\forall x \in \mathbb{R}^d$ and $\forall j$;*

*c) $f_y^\eta(\tilde{x}_J) - f_J^\eta(x_J) \le \delta$;*

*d) $f(\tilde{x}_J) - f(u) \le \delta + \frac{1}{2\eta}\|u-y\|^2 - \frac{1}{2\eta}\|x_J - u\|^2$;*

*e) $-\frac{1}{\eta}(x^* - y) \in \partial f(x^*)$ where $\partial f$ denotes the subdifferential of $f$.*

**Proof**: a) The first inequality follows from the definition of $f_j$ in step 2 of ALgorithm 2. The second inequality directly follows from the definition of $f_j$ and the convexity of $f$.

b) This statement follows from (50) and the fact that $f_j^\eta$ is $(1/\eta)$-strongly convex.

c) This statement immediately follows from step 4 of Algorithm 2.

d)

$$\begin{aligned} f(\tilde{x}_J) - f(x) + \frac{1}{2\eta}\|x_J - x\|^2 &\le f(\tilde{y}_J) - f_J(x) + \frac{1}{2\eta}\|x_J - x\|^2 \\ &\overset{\text{(b)}}{\le} f(\tilde{x}_J) - f_J^\eta(x_J) + \frac{1}{2\eta}\|x-y\|^2 \overset{\text{(c)}}{\le} \delta + \frac{1}{2\eta}\|x-y\|^2 \end{aligned}$$

e) This statement directly follows from the optimality condition of (4). ∎

Clearly, when Algorithm 2 terminates, the output $\tilde{x}_J$ is a $\delta$-solution to (4). To see this, note that, following Lemma 3.1(c), (50), and the fact $f_J^\eta \le f_y^\eta$,

$$f_y^\eta(\tilde{x}_J) \le \delta + f_J^\eta(x_J) \le \delta + f_J^\eta(x^*) \le \delta + f_y^\eta(x^*).$$

In the remaining part of this subsection, we establish the iteration-complexity of Algorithm 2 for solving the proximal semi-smooth optimization problem (4). The following lemma provides basic recursive formulas and is the starting point of the analysis of Algorithm 2.

**Lemma 3.2.** *Define*

$$t_j := f_y^\eta(\tilde{x}_j) - f_j^\eta(x_j). \tag{8}$$

*Then, for every $j \ge 1$, the following statements hold:*

*a)* $t_{j+1} + \frac{1}{2\eta}\|x_{j+1} - x_j\|^2 \le t_j$;

*b)* $t_j \le \frac{L_\alpha}{\alpha+1}\|x_j - x_{j-1}\|^{\alpha+1}$;

*c)* $t_{j+1} + \frac{1}{2\eta}\left(\frac{\alpha+1}{L_\alpha} t_{j+1}\right)^{\frac{2}{\alpha+1}} \le t_j$.

**Proof**: a) Using the first inequality in Lemma 3.1(a) and Lemma 3.1(b) with $x = x_{j+1}$, we obtain

$$f_{j+1}^\eta(x_{j+1}) \ge f_j^\eta(x_{j+1}) \ge f_j^\eta(x_j) + \frac{1}{2\eta}\|x_{j+1} - x_j\|^2.$$

It follows from the above inequality, the definition of $\tilde{x}_j$ in (7), and the definition of $t_j$ in (8), that

$$t_{j+1} = f_y^\eta(\tilde{x}_{j+1}) - f_{j+1}^\eta(x_{j+1}) \le f_y^\eta(\tilde{x}_j) - f_j^\eta(x_j) - \frac{1}{2\eta}\|x_{j+1} - x_j\|^2 = t_j - \frac{1}{2\eta}\|x_{j+1} - x_j\|^2.$$

b) It follows from the definition of $t_j$ in (8) and the definition of $\tilde{x}_j$ in (7) that

$$\begin{aligned} t_j &\le f_y^\eta(x_j) - f_j^\eta(x_j) = f(x_j) - f_j(x_j) \\ &\le f(x_j) - f(x_{j-1}) - \langle f'(x_{j-1}), x_j - x_{j-1}\rangle \\ &\le \frac{L_\alpha}{\alpha+1}\|x_j - x_{j-1}\|^{\alpha+1}, \end{aligned}$$

where the second inequality is due to the definition of $f_j$ in the step 2 of Algorithm 2, and the third inequality is due to (60) with $(u, v) = (x_j, x_{j-1})$.

c) This statement directly follows from a) and b). ∎

It is easy to observe from Lemma 3.2 that $\{t_j\}$ is non-increasing. The next proposition gives a bound on $j$ so that $t_j \le \delta$, i.e., the termination criterion in step 4 of Algorithm 2 is satisfied.

**Proposition 3.3.** *Define*

$$\beta := \frac{1}{2\eta}\left(\frac{\alpha+1}{L_\alpha}\right)^{\frac{2}{\alpha+1}} \delta^{\frac{1-\alpha}{\alpha+1}}, \quad j_0 = 1 + \left\lceil \frac{1+\beta}{\beta} \log\left(\frac{t_1}{\delta}\right)\right\rceil. \tag{9}$$

*Then, the following statements hold:*

*a) if $t_j > \delta$, then $(1+\beta)t_j \le t_{j-1}$;*

*b)* $t_j \le \delta$ *for every* $j \ge j_0$. $J \le j_0$

**Proof**: a) Using the definition of $\beta$ in (9), the assumption that $t_j > \delta$, and Lemma 3.2(c), we obtain

$$(1+\beta)t_j = t_j + \frac{1}{2\eta}\left(\frac{\alpha+1}{L_\alpha}\right)^{\frac{2}{\alpha+1}} \delta^{\frac{1-\alpha}{\alpha+1}} t_j \le t_j + \frac{1}{2\eta}\left(\frac{\alpha+1}{L_\alpha} t_j\right)^{\frac{2}{\alpha+1}} \le t_{j-1}.$$

b) Since $\{t_j\}$ is non-increasing, it suffices to prove that $t_{j_0} \le \delta$ We prove this statement by contradiction. Suppose that $t_{j_0} > \delta$, then we have $t_j > \delta$ for $j \le j_0$. Hence, statement (a) holds for $j \le j_0$. Using this conclusion repeatedly and the fact that $\tau \le \exp(\tau - 1)$ with $\tau = 1/(1+\beta)$, we have

$$t_{j_0} \le \frac{1}{(1+\beta)^{j_0-1}} t_1 \le \frac{1}{\exp\left(\frac{\beta}{1+\beta}(j_0-1)\right)} t_1 \le \delta,$$

where the last inequality is due to the definition of $j_0$ in (9). This contradicts with the assumption that $t_{j_0} > \delta$, and hence we prove this statement. ∎

The following result shows that $t_1$ is bounded from above, and hence the bound in Proposition 3.3 is meaningful.

**Lemma 3.4.** *We have*

$$t_1 \le \frac{L_\alpha \eta^{\alpha+1}}{\alpha+1} \|f'(x_0)\|^{\alpha+1}.$$

**Proof**: Following the optimality condition of (50) with $j = 1$, we have $x_0 - x_1 = \eta f'(x_0)$. This identity and Lemma 3.2(b) with $j = 1$ then imply the lemma. ∎

We conclude this subsection by presenting the iteration-complexity bound for Algorithm 2.

**Theorem 3.5.** *Algorithm 2 takes* $\tilde{\mathcal{O}}\left(\eta L_\alpha^{\frac{2}{\alpha+1}} \left(\frac{1}{\delta}\right)^{\frac{1-\alpha}{\alpha+1}} + 1\right)$ *iterations to terminate.*

**Proof**: It follows directly from Proposition 3.3 and Lemma 3.4. ∎

### 3.2 A proximal bundle method subroutine for composite optimization

The main goal of this subsection is to study Algorithm 2 for solving the proximal optimization problem (4) under the assumption that $f$ is convex and $(L_1, L_\alpha)$-smooth-semi-smooth.

**Lemma 3.6.** *Let* $t_j$ *be as in* (8). *For* $\delta > 0$, *define*

$$M = L_1 + \frac{L_\alpha^{\frac{2}{\alpha+1}}}{[(\alpha+1)\delta]^{\frac{1-\alpha}{\alpha+1}}}. \tag{10}$$

*Then, for every* $j \ge 1$, *the following statements hold:*

*a)* $t_{j+1} + \frac{1}{2\eta}\|x_{j+1} - x_j\|^2 \le t_j$;

*b)* $t_j \le \frac{M}{2}\|x_{j+1} - x_j\|^2 + \frac{(1-\alpha)\delta}{2}$;

*c)* $\left(1 + \frac{1}{\eta M}\right)\left(t_{j+1} - \frac{(1-\alpha)\delta}{2}\right) \le t_j - \frac{(1-\alpha)\delta}{2}$.

**Proof**: a) This statement follows from the same argument as in the proof of Lemma 3.2(a).

b) Following a similar argument as in the proof of Lemma 3.2(b) with (60) replaced by (59), we have

$$t_j \leq \frac{L_\alpha}{\alpha+1}\|x_j - x_{j-1}\|^{\alpha+1} + \frac{L_1}{2}\|x_j - x_{j-1}\|^2. \tag{11}$$

Using the Young's inequality $ab \leq a^p/p + b^q/q$ with

$$a = \frac{L_\alpha}{(\alpha+1)\delta^{\frac{1-\alpha}{2}}}\|x_{j+1} - x_j\|^{\alpha+1}, \quad b = \delta^{\frac{1-\alpha}{2}}, \quad p = \frac{2}{\alpha+1}, \quad q = \frac{2}{1-\alpha},$$

we obtain

$$\frac{L_\alpha}{\alpha+1}\|x_{j+1} - x_j\|^{\alpha+1} \leq \frac{L_\alpha^{\frac{2}{\alpha+1}}}{2[(\alpha+1)\delta]^{\frac{1-\alpha}{\alpha+1}}}\|x_{j+1} - x_j\|^2 + \frac{(1-\alpha)\delta}{2}.$$

Combining the above inequality and (11), and using the definition of $M$ in (10), we prove the statement.

c) It immediately follows from (a) and (b) that

$$t_{j+1} + \frac{1}{\eta M}\left(t_{j+1} - \frac{(1-\alpha)\delta}{2}\right) \leq t_{j+1} + \frac{1}{2\eta}\|x_{j+1} - x_j\|^2 \leq t_j,$$

and hence the statement follows. ∎

The following lemma gives an upper bound on $t_1$ similar to Lemma 3.4.

**Lemma 3.7.** *We have*

$$t_1 \leq \frac{L_\alpha \eta^{\alpha+1}}{\alpha+1}\|f'(x_0)\|^{\alpha+1} + \frac{L_1\eta^2}{2}\|f'(x_0)\|^2.$$

**Proof**: This lemma follows from a similar argument as in the proof of Lemma 3.4. ∎

The following proposition is the key result in establishing the iteration-complexity of Algorithm 2.

**Proposition 3.8.** *We have $t_j \leq \delta$, for every $j$ such that*

$$j \geq \left[1 + \eta\left(L_1 + \frac{L_\alpha^{\frac{2}{\alpha+1}}}{[(\alpha+1)\delta]^{\frac{1-\alpha}{\alpha+1}}}\right)\right]\log\left(\frac{2t_1}{\delta}\right). \tag{12}$$

**Proof**: Let

$$\tau = \frac{\eta M}{1+\eta M}, \tag{13}$$

then Lemma 3.6(c) becomes

$$t_{j+1} - \frac{(1-\alpha)\delta}{2} \leq \tau\left(t_j - \frac{(1-\alpha)\delta}{2}\right).$$

Using the above inequality and the fact that $\tau \leq \exp(\tau - 1)$, we have for every $j \geq 1$,

$$t_j - \frac{(1-\alpha)\delta}{2} \leq \tau^{j-1}\left(t_1 - \frac{(1-\alpha)\delta}{2}\right) \leq \tau^{j-1}t_1 \leq \exp\{(\tau-1)(j-1)\}t_1.$$

Hence, it is easy to see that $t_j \leq \delta$ if $j \geq \frac{1}{1-\tau}\log\left(\frac{2t_1}{\delta}\right)$. Using the definition of $\tau$ in (54), we have if $j$ is as in (12), then $t_j \leq \delta$. ∎

**Theorem 3.9.** *Algorithm 2 takes $\tilde{\mathcal{O}}\left(\eta L_1 + \eta L_\alpha^{\frac{2}{\alpha+1}}\left(\frac{1}{\delta}\right)^{\frac{1-\alpha}{\alpha+1}} + 1\right)$ iterations to terminate.*

**Proof**: It follows directly from Proposition 3.8 and Lemma 3.7. ∎

# 4 Adaptive proximal bundle method

The proximal bundle method we studied in Section 3 is parameter-free since it does not require problem-dependent parameters such as $\alpha$ and $L_\alpha$. However, in order to implement the method, we still need to choose a stepsize $\eta$, which dictates the practical performance of many proximal-type algorithms. In general, it is not easy to select a constant stepsize $\eta$ to be a good one in practice.

In this section, we develop an adaptive proximal bundle method that automatically searches for suitable stepsizes. From practical observations, the proximal bundle method works well when the number of inner iterations stays as a constant much larger than 1 (i.e., that of the subgradient method), say 10. Recall from Theorem 3.5 that inner complexity is $\tilde{\mathcal{O}}\left(\eta L_\alpha^{\frac{2}{\alpha+1}}\left(\frac{1}{\delta}\right)^{\frac{1-\alpha}{\alpha+1}}+1\right)$. Since we do not know $\alpha$ and $L_\alpha$, we cannot choose a constant stepsize $\eta$ so that the number of inner iterations is close to a desired number such as 10. Hence, an adaptive stepsize rule is indeed needed.

By carefully examining Proposition 3.3 and Theorem 3.5, we find that the inner complexity is $\tilde{\mathcal{O}}(\beta^{-1}+1)$ where $\beta$ is as in (9). Suppose we want to prescribe the number of inner iterations to be close to $\beta_0^{-1}$ for some $\beta_0 \in (0,1]$, if $\beta_0 \le \beta$, then by Proposition 3.3(a), we have

$$(1+\beta_0)t_j \le t_{j-1}. \tag{14}$$

Hence, it suffices to begin with a relatively large $\eta$, check (14) to determine whether the $\eta$ is small enough (i.e., $\beta$ is large enough), and adjust $\eta$ (if necessary) by progressively halving it.

The following adaptive proximal bundle method is a formal statement based on the above intuition.

**Algorithm 3** Adaptive Proximal Bundle Method

1. Let $y_0 \in \mathbb{R}^d$, $\eta_0 > 0$, $\beta_0 \in (0,1]$, and $\varepsilon > 0$ be given, and set $k = 1$
2. Call Algorithm 2 with $(y, \eta, \delta) = (y_{k-1}, \eta_{k-1}, \varepsilon/2)$ and output $(y_k, \tilde{y}_k) = (x_J, \tilde{x}_J)$
3. In the execution of Algorithm 2, if (14) is always true, then set $\eta_k = \eta_{k-1}$; otherwise, set $\eta_k = \eta_{k-1}/2$
4. Set $k \leftarrow k+1$ and go to step 2.

**Lemma 4.1.** *The following statements hold for APBM:*

*a) for every $k \ge 1$ and $u \in \mathbb{R}^d$, we have*

$$2\eta_{k-1}[f(\tilde{y}_k) - f(u)] \le \|y_{k-1} - u\|^2 - \|y_k - u\|^2 + \eta_{k-1}\varepsilon; \tag{15}$$

*b) for every $k \ge 1$, if*

$$\eta_{k-1} \le \frac{1}{2\beta_0}\left(\frac{\alpha+1}{L_\alpha}\right)^{\frac{2}{\alpha+1}}\left(\frac{\varepsilon}{2}\right)^{\frac{1-\alpha}{\alpha+1}}, \tag{16}$$

*then (15) holds with $\eta_k = \eta_{k-1}$;*

*c) $\{\eta_k\}$ is a non-increasing sequence;*

*d) for every $k \ge 0$,*

$$\eta_k \ge \underline{\eta} := \min\left\{\frac{1}{4\beta_0}\left(\frac{\alpha+1}{L_\alpha}\right)^{\frac{2}{\alpha+1}}\left(\frac{\varepsilon}{2}\right)^{\frac{1-\alpha}{\alpha+1}}, \eta_0\right\}. \tag{17}$$

**Proof**: a) This statement directly follows from Lemma 3.1(d) and the fact $(\eta, y, x_J, \tilde{x}_J) = (\eta_{k-1}, y_{k-1}, y_k, \tilde{y}_k)$ (see step 2 of Algorithm 3).

b) Since (16) implies that (14) with

$$\beta = \frac{1}{2\eta_{k-1}} \left(\frac{\alpha+1}{L_\alpha}\right)^{\frac{2}{\alpha+1}} \left(\frac{\varepsilon}{2}\right)^{\frac{1-\alpha}{\alpha+1}}$$

always holds in the execution of Algorithm 2, it follows from step 3 of Algorithm 3 that $\eta_k = \eta_{k-1}$.

c) This statement clearly follows from step 3 of Algorithm 3.

d) This statement immediately follows from b) and step 3 of Algorithm 3. ∎

**Theorem 4.2.** *For some $\tilde{\eta} > 0$ such that*

$$\tilde{\eta} = \mathcal{O}\left(\frac{\|y_0 - x_*\|^2}{\varepsilon}\right), \quad \tilde{\eta} = \Omega\left(\frac{\varepsilon^{\frac{1-\alpha}{\alpha+1}}}{L_\alpha^{\frac{2}{\alpha+1}}}\right), \tag{18}$$

*if $\eta_k \equiv \tilde{\eta}$ for $k \geq k_0$, then the iteration complexity to obtain an $\varepsilon$-solution to (??) is given by*

$$\tilde{\mathcal{O}}\left(\frac{L_\alpha^{\frac{2}{\alpha+1}} \|y_0 - x_*\|^2}{\varepsilon^{\frac{2}{\alpha+1}}} + \eta_0 L_\alpha^{\frac{2}{\alpha+1}} \left(\frac{1}{\varepsilon}\right)^{\frac{1-\alpha}{\alpha+1}} \log\left(\frac{\eta_0}{\underline{\eta}}\right) + 1\right) \tag{19}$$

*where $\underline{\eta}$ is as in* (17).

**Proof**: Summing (15) from $k = 1$ to $n$, we have

$$2\sum_{k=1}^{n} \eta_{k-1} \left(\min_{1\leq k\leq n} f(\tilde{y}_k) - f(u)\right) \leq 2\sum_{k=1}^{n} \eta_{k-1}[f(\tilde{y}_k) - f(u)] \leq \|y_0 - u\|^2 + \varepsilon \sum_{k=1}^{n} \eta_{k-1}.$$

The above inequality and the assumption that $\eta_k \equiv \tilde{\eta}$ for $k \geq k_0$ imply that

$$\min_{1\leq k\leq n} f(\tilde{y}_k) - f_* \leq \frac{\|y_0 - x_*\|^2}{2\sum_{k=1}^{n} \eta_{k-1}} + \frac{\varepsilon}{2} \leq \frac{\|y_0 - x_*\|^2}{2(n-k_0)\tilde{\eta}} + \frac{\varepsilon}{2}.$$

In order to have $\min_{1\leq k\leq n} f(\tilde{y}_k) - f_* \leq \varepsilon$, we need

$$n - k_0 = \mathcal{O}\left(\frac{\|y_0 - x_*\|^2}{\tilde{\eta}\varepsilon} + 1\right). \tag{20}$$

Moreover, it follows from the way $\eta_k$ is updated in step 3 and Lemma 4.1(d) that

$$k_0 = \mathcal{O}\left(\log\left(\frac{\eta_0}{\underline{\eta}}\right) + 1\right). \tag{21}$$

Using Theorem 3.5, we have the iteration complexity of every call to Algorithm 2 is bounded by

$$\tilde{\mathcal{O}}\left(\tilde{\eta} L_\alpha^{\frac{2}{\alpha+1}} \left(\frac{1}{\varepsilon}\right)^{\frac{1-\alpha}{\alpha+1}} + 1\right) \tag{22}$$

for every cycle $k \geq k_0$ and by

$$\tilde{\mathcal{O}}\left(\eta_0 L_\alpha^{\frac{2}{\alpha+1}} \left(\frac{1}{\varepsilon}\right)^{\frac{1-\alpha}{\alpha+1}} + 1\right) \tag{23}$$

for every cycle $k \le k_0 - 1$. Hence, combining (20) and (22) and using (18), we obtain the iteration complexity

$$\tilde{\mathcal{O}}\left(\frac{L_\alpha^{\frac{2}{\alpha+1}}\|y_0 - x_*\|^2}{\varepsilon^{\frac{2}{\alpha+1}}} + 1\right)$$

for cycles $k \ge k_0$, and combining (21) and (23), we obtain the iteration complexity

$$\tilde{\mathcal{O}}\left(\eta_0 L_\alpha^{\frac{2}{\alpha+1}}\left(\frac{1}{\varepsilon}\right)^{\frac{1-\alpha}{\alpha+1}}\log\left(\frac{\eta_0}{\underline{\eta}}\right) + 1\right)$$

for cycles $k \le k_0 - 1$. Finally, the total iteration complexity clearly follows from the above two bounds. ∎

---

universal method is essentially the same as A-CS

the complexity is

$$\mathcal{O}\left(\frac{L_\alpha^{\frac{2}{\alpha+1}}\|y_0 - x_*\|^2}{\varepsilon^{\frac{2}{\alpha+1}}}\right)$$

universal method (A-CS) is different from APBM

universal method is adaptive subgradient method and the nature of subgradient method is that its convergence relies on small enough stepsize, so it enforces $\eta$ to be small

if $f(x) - \ell_f(x; x_j) - \|x - x_j\|^2/(2\lambda) > \bar{\varepsilon}/2$, it sets $\lambda = \lambda/2$

bundle method converges with any constant stepsize $\eta$, it guarantees $t_J \le \varepsilon/2$ by the cutting-plane approach but not by small $\eta$, so it halves $\eta$ when necessary. The goal in the adaptive method is to regulate the inner complexity to a desired number, say 10. It is beyond convergence but a desired convergence.

Even the total complexity has been covered by bundle smooth (see (40)), this paper has two advantages: 1) the complexity for solving the proximal oracle which is the crux for sampling and online learning as well; 2) the adaptive approach for the proximal point framework

# 5 Proximal Sampling algorithm

Assuming the RGO in the ASF can be realized, the ASF exhibits remarkable convergence properties. In [25] it was shown that Algorithm 1 converges linearly when $f$ is strongly convex. This convergence result is recently improved in [6] under various weaker assumptions on the target distribution $\pi^X \propto \exp(-f)$. Below we present several convergence results established in [6] that will be used in this paper, under the assumptions that $\pi^X$ is log-concave, or satisfies the log-Sobolev inequality (LSI) or Poincaré inequality (PI). Recall that a probability distribution $\nu$ satisfies PI with constant $\lambda > 0$ ($\lambda$-PI) if for any smooth bounded function $\psi : \mathbb{R}^d \to \mathbb{R}$,

$$\mathbb{E}_\nu[(\psi - \mathbb{E}_\nu(\psi))^2] \le \frac{1}{\lambda}\mathbb{E}_\nu[\|\nabla\psi\|^2]\,. \tag{24}$$

To this end, for two probability distributions $\rho \ll \nu$, we denote by

$$H_\nu(\rho) := \int \rho \log\frac{\rho}{\nu}, \quad \chi^2_\nu(\rho) := \int \frac{\rho^2}{\nu} - 1, \quad R_{q,\nu}(\rho) := \frac{1}{q-1}\log\int\frac{\rho^q}{\nu^{q-1}}$$

the *KL divergence*, the *Chi-squared divergence*, and the *Rényi divergence*, respectively. Note that $R_{2,\nu} = \log(1+\chi^2_\nu)$, $R_{1,\nu} = H_\nu$, and $R_{q,\nu} \le R_{q',\nu}$ for any $1 \le q \le q' < \infty$. We denote by $W_2$ the Wasserstein-2 distance defined by [49]

$$W_2^2(\nu,\rho) := \min_{\gamma\in\Pi(\nu,\rho)} \int \|x-y\|^2 \mathrm{d}\gamma,$$

where $\Pi(\nu,\rho)$ represents the set of all couplings between $\nu$ and $\rho$.

**Theorem 5.1** ([6, Theorem 2]). *Assume that $\pi^X \propto \exp(-f)$ is log-concave (i.e., $f$ is convex). For any initial distribution $\rho_0^X$, the $k$-th iterate $\rho_k^X$ of Algorithm 1 satisfies*

$$H_{\pi^X}(\rho_k^X) \le \frac{W_2^2(\rho_0^X,\pi^X)}{k\eta}\,.$$

**Theorem 5.2** ([6, Theorem 4]). *Assume $\pi^X \propto \exp(-f)$ satisfies $\lambda$-PI. For any initial distribution $\rho_0^X$, the $k$-th iterate $\rho_k^X$ of Algorithm 1 with step size $\eta > 0$ satisfies*

$$\chi^2_{\pi^X}(\rho_k^X) \le \frac{\chi^2_{\pi^X}(\rho_0^X)}{(1+\lambda\eta)^{2k}}\,. \tag{25}$$

*Furthermore, for all $q \ge 2$,*

$$R_{q,\pi^X}(\rho_k^X) \le \begin{cases} R_{q,\pi^X}(\rho_0^X) - \frac{2k\log(1+\lambda\eta)}{q}\,, & \text{if } k \le \frac{q}{2\log(1+\lambda\eta)}\left(R_{q,\pi^X}(\rho_0^X)-1\right), \\ 1/(1+\lambda\eta)^{2(k-k_0)/q}\,, & \text{if } k \ge k_0 := \left\lceil \frac{q}{2\log(1+\lambda\eta)}\left(R_{q,\pi^X}(\rho_0^X)-1\right)\right\rceil. \end{cases} \tag{26}$$

As discussed earlier, to use ASF in sampling problems, we need to realize the RGO with efficient implementations. In the rest of this section, we develop efficient algorithms for RGO associated with the two scenarios of sampling we are interested in, and then combine them with the ASF to establish a proximal algorithm for sampling. The complexity of the proximal algorithm can be obtained by combining the above convergence results for ASF and the complexity results we develop for RGO. The rest of the section is organized as follows. In Section 5.1 we develop an efficient algorithm for RGO associated with semi-smooth potentials via rejection sampling. This is combined with ASF to obtain an efficient sampling algorithm from semi-smooth potentials in Section 5.2. In Section 5.3, we further extend the algorithm and results to the second setting we are interested in which involves composite potentials (smooth + semi-smooth).

### 5.1 The restricted Gaussian oracle for semi-smooth potentials

The bottleneck of using the ASF (Algorithm 1) in sampling tasks with general distributions is the availability of RGO implementations. In this section, we address this issue for convex semi-smooth potentials by developing an efficient algorithm for the corresponding RGO.

Our algorithm of RGO for $f$ is based on rejection sampling. We use a special proposal which is a Gaussian distribution centered at the approximate minimizer of

$$f_y^\eta(x) := f(x) + \frac{1}{2\eta}\|x-y\|^2 \tag{27}$$

for a given $y$. With this proposal and a sufficiently small $\eta > 0$, the expected number of rejection sampling steps to obtain one effective sample turns out to be bounded from above by a dimension-free constant. To bound the complexity of the rejection sampling, we develop a novel technique

to estimate a modified Gaussian integral (see Proposition 5.4). In Section 3.1 we use the proximal bundle subroutine Algorithm 2 to optimize $f_y^\eta(x)$ to certain precision and establish the iteration-complexity for it. This is achieved by choosing a proper precision for the approximate minimizer of $f_y^\eta$ that balances the complexity of optimization and the efficiency of rejection sampling.

In this section we consider the RGO for $f$, namely, the sampling task from $\exp(-f_y^\eta(x))$ when the minimizer of $f_y^\eta(x)$ is not available. We use the optimization algorithm (Algorithm 2) developed in Section 3.1 to compute a $\delta$-solution to (4). Based on this $\delta$-solution, we modify Algorithm **??** so that the expected number of runs for the rejection sampling is still properly bounded.

To this end, let $J, \tilde{x}_J, x_J$ be the outputs of Algorithm 2 and define

$$h_1 := \frac{1}{2\eta}\|\cdot - x_J\|^2 + f_y^\eta(\tilde{x}_J) - \delta, \tag{28a}$$

$$h_2 := \frac{1}{2\eta}\|\cdot - x^*\|^2 + \frac{L_\alpha}{\alpha+1}\|\cdot - x^*\|^{\alpha+1} + f_y^\eta(x^*). \tag{28b}$$

Note that $h_2$ is only used for analysis and thus the fact it depends on $x^*$ is not an issue.

Algorithm 4 describes the implementation of RGO for $f$ based on Algorithm 2 and rejection sampling.

**Algorithm 4** Rejection Sampling without Proximal Map

1. Run Algorithm 2 to compute $x_J$ and $\tilde{x}_J$
2. Generate $X \sim \exp(-h_1(x))$
3. Generate $U \sim \mathcal{U}[0,1]$
4. If
$$U \le \frac{\exp(-f_y^\eta(X))}{\exp(-h_1(X))},$$
then accept/return $X$; otherwise, reject $X$ and go to step 2.

**Lemma 5.3.** *Assume $f$ is convex and $L_\alpha$-semi-smooth and let $f_y^\eta$ be as in* (27)*. Then, for every $x \in \mathbb{R}^d$,*

$$h_1(x) \le f_y^\eta(x) \le h_2(x) \tag{29}$$

*where $h_1$ and $h_2$ are as in* (28)*.*

**Proof**: Using Lemma 3.1(a)-(b) and the definition of $f_j^\eta$, we have

$$\begin{aligned} f(\tilde{x}_J) - f(x) + \frac{1}{2\eta}\|x - x_J\|^2 &\le f(\tilde{x}_J) - f_J(x) + \frac{1}{2\eta}\|x - x_J\|^2 \\ &\le f(\tilde{x}_J) - f_J^\eta(x_J) + \frac{1}{2\eta}\|x - y\|^2 \le \delta - \frac{1}{2\eta}\|\tilde{x}_J - y\|^2 + \frac{1}{2\eta}\|x - y\|^2. \end{aligned}$$

The first inequality in (29) holds in view of the definition of $h_1$ in (28a). By the definition of $f_y^\eta$ in (27) we get

$$\begin{aligned} f_y^\eta(x) - f_y^\eta(x^*) =& f(x) - f(x^*) + \frac{1}{2\eta}\|x - y\|^2 - \frac{1}{2\eta}\|x^* - y\|^2 \\ =& f(x) - f(x^*) + \frac{1}{2\eta}\|x - x^*\|^2 + \frac{1}{\eta}\langle x - x^*, x^* - y\rangle. \end{aligned} \tag{30}$$

It follows from Lemma 3.1(d) and (60) with $(u, v) = (x, x^*)$ that

$$f(x) - f(x^*) + \frac{1}{\eta}\langle x^* - y, x - x^*\rangle \le \frac{L_\alpha}{\alpha+1}\|x - x^*\|^{\alpha+1},$$

which together with (30) implies that

$$f_y^\eta(x) - f_y^\eta(x^*) \le \frac{L_\alpha}{\alpha+1}\|x - x^*\|^{\alpha+1} + \frac{1}{2\eta}\|x - x^*\|^2.$$

Using the above inequality and the definition of $h_2$ in (28b), we conclude that the second inequality in (29) holds. ∎

From the expression of $h_1$ in (28a), it is clear the proposal distribution is a Gaussian centered at $x_J$. To achieve a tight bound on the expected runs of the rejection sampling, we use a function $h_2$ which is not quadratic; the standard choice of quadratic function does not give as tight results due to the lack of smoothness. To use this $h_2$ in the complexity analysis, we need to estimate the integral $\int \exp(-h_2)$, which turns out to be *a highly nontrivial task*. Below we establish a technical result on a modified Gaussian integral, which will be used later to bound the integral $\int \exp(-h_2)$ and hence the complexity of the RGO rejection sampling in Algorithm 4.

**Proposition 5.4.** *Let $\alpha \in [0, 1]$, $\eta > 0$, $a \ge 0$ and $d \ge 1$. If*

$$2a(\eta d)^{(\alpha+1)/2} \le 1, \tag{31}$$

*then*

$$\int_{\mathbb{R}^d} \exp\left(-\frac{1}{2\eta}\|x\|^2 - a\|x\|^{\alpha+1}\right) \mathrm{d}x \ge \frac{(2\pi\eta)^{d/2}}{2}. \tag{32}$$

**Proof**: Denote $r = \|x\|$, then

$$\mathrm{d}x = r^{d-1}\mathrm{d}r\mathrm{d}S^{d-1},$$

where $\mathrm{d}S^{d-1}$ is the surface area of the $(d-1)$-dimensional unit sphere. It follows that

$$\begin{aligned}\int_{\mathbb{R}^d} \exp\left(-\frac{1}{2\eta}\|x\|^2 - a\|x\|^{\alpha+1}\right) \mathrm{d}x &= \int_0^\infty \int \exp\left(-\frac{1}{2\eta}r^2 - ar^{\alpha+1}\right) r^{d-1}\mathrm{d}r\mathrm{d}S^{d-1} \\ &= \frac{2\pi^{d/2}}{\Gamma\left(\frac{d}{2}\right)} \int_0^\infty \exp\left(-\frac{1}{2\eta}r^2 - ar^{\alpha+1}\right) r^{d-1}\mathrm{d}r. \end{aligned} \tag{33}$$

In the above equation, we have used the fact that the total surface area of a $(d-1)$-dimensional unit sphere is $2\pi^{d/2}/\Gamma\left(\frac{d}{2}\right)$ where $\Gamma(\cdot)$ is the gamma function, i.e.,

$$\Gamma(z) = \int_0^\infty t^{z-1}e^{-t}\mathrm{d}t. \tag{34}$$

Defining

$$F_{d,\eta}(a) := \int_0^\infty \exp\left(-\frac{1}{2\eta}r^2 - ar^{\alpha+1}\right) r^d \mathrm{d}r, \tag{35}$$

to establish (32), it suffices to bound $F_{d-1,\eta}(a)$ from below.

It follows directly from the definition of $F_{d,\eta}$ in (35) that

$$\frac{\mathrm{d}F_{d-1,\eta}(a)}{\mathrm{d}a} = \int_0^\infty \exp\left(-\frac{1}{2\eta}r^2 - ar^{\alpha+1}\right)(-r^{\alpha+1})r^{d-1}\mathrm{d}r = -F_{d+\alpha,\eta}(a).$$

This implies $F_{d,\eta}$ is monotonically decreasing and thus $F_{d+\alpha,\eta}(a) \leq F_{d+\alpha,\eta}(0)$. As a result,

$$\frac{\mathrm{d}F_{d-1,\eta}(a)}{\mathrm{d}a} \geq -F_{d+\alpha,\eta}(0)$$

and therefore,

$$F_{d-1,\eta}(a) \geq F_{d-1,\eta}(0) - aF_{d+\alpha,\eta}(0). \tag{36}$$

Setting $t = r^2/(2\eta)$, we can write

$$\begin{aligned} F_{d,\eta}(0) &= \int_0^\infty \exp\left(-\frac{1}{2\eta}r^2\right) r^d \mathrm{d}r = \int_0^\infty e^{-t}(2\eta t)^{\frac{d-1}{2}}\eta \mathrm{d}t \\ &= 2^{\frac{d-1}{2}}\eta^{\frac{d+1}{2}}\int_0^\infty e^{-t}t^{\frac{d-1}{2}}\mathrm{d}t. \end{aligned} \tag{37}$$

In view of the definition of the gamma function (34), we obtain

$$F_{d,\eta}(0) = 2^{\frac{d-1}{2}}\eta^{\frac{d+1}{2}}\Gamma\left(\frac{d+1}{2}\right). \tag{38}$$

Applying the Wendel's double inequality (58) yields

$$\frac{\Gamma\left(\frac{d+\alpha+1}{2}\right)}{\Gamma\left(\frac{d}{2}\right)} \leq \left(\frac{d}{2}\right)^{\frac{\alpha+1}{2}}.$$

Using (36), (38), the above inequality and the assumption (31), we have

$$\begin{aligned} F_{d-1,\eta}(a) &\geq F_{d-1,\eta}(0) - aF_{d+\alpha,\eta}(0) \\ &= 2^{\frac{d}{2}-1}\eta^{\frac{d}{2}}\Gamma\left(\frac{d}{2}\right) - a2^{\frac{d+\alpha-1}{2}}\eta^{\frac{d+\alpha+1}{2}}\Gamma\left(\frac{d+\alpha+1}{2}\right) \\ &= 2^{\frac{d}{2}-1}\eta^{\frac{d}{2}}\Gamma\left(\frac{d}{2}\right)\left(1 - a2^{\frac{\alpha+1}{2}}\eta^{\frac{\alpha+1}{2}}\frac{\Gamma\left(\frac{d+\alpha+1}{2}\right)}{\Gamma\left(\frac{d}{2}\right)}\right) \\ &\geq 2^{\frac{d}{2}-1}\eta^{\frac{d}{2}}\Gamma\left(\frac{d}{2}\right)\left(1 - a(\eta d)^{\frac{\alpha+1}{2}}\right) \geq \frac{1}{4}(2\eta)^{\frac{d}{2}}\Gamma\left(\frac{d}{2}\right). \end{aligned}$$

The result (32) then follows from the above inequality and (33). ∎

We finally show that the number of rejections in Algorithm 4 is bounded from above by a small constant when $\delta$ is properly chosen. In particular, as shown in Proposition 5.5, it only gets worse by a factor of $\exp(\delta)$ and the factor does not depend on the dimension $d$. Hence, the implementation of RGO for $f$ is computationally efficient in practice.

**Proposition 5.5.** *Assume $f$ is convex and $L_\alpha$-semi-smooth and let $f_y^\eta$ be as in* (27)*. If in addition*

$$\eta \leq \frac{(\alpha+1)^{\frac{2}{\alpha+1}}}{(2L_\alpha)^{\frac{2}{\alpha+1}}d}, \tag{39}$$

*then the expected number of iterations in the rejection sampling of Algorithm 4 is at most $2\exp(\delta)$.*

**Proof**: It is a well-known result for rejection sampling that $X \sim \pi^{X|Y}(x \mid y)$ and the probability that $X$ is accepted is

$$\mathbb{P}\left(U \leq \frac{\exp(-f_y^\eta(X))}{\exp(-h_1(X))}\right) = \frac{\int_{\mathbb{R}^d} \exp(-f_y^\eta(x))\mathrm{d}x}{\int_{\mathbb{R}^d} \exp(-h_1(x))\mathrm{d}x}. \tag{40}$$

If follows directly from the definition of $h_2$ in (28b) that

$$\int_{\mathbb{R}^d} \exp(-h_2(x))\mathrm{d}x = \exp(-f_y^\eta(x^*)) \int_{\mathbb{R}^d} \exp\left(-\frac{1}{2\eta}\|x - x^*\|^2 - \frac{L_\alpha}{\alpha+1}\|x - x^*\|^{\alpha+1}\right)\mathrm{d}x$$

Applying Proposition 5.4 to the above yields

$$\int_{\mathbb{R}^d} \exp(-h_2(x))\mathrm{d}x \geq \exp(-f_y^\eta(x^*))\frac{(2\pi\eta)^{d/2}}{2}.$$

Note that the condition (31) in Proposition 5.4 holds thanks to (39). By Lemma 5.3, the above inequality leads to

$$\int_{\mathbb{R}^d} \exp(-f_y^\eta(x))\mathrm{d}x \geq \int_{\mathbb{R}^d} \exp(-h_2(x))\mathrm{d}x \geq \exp(-f_y^\eta(x^*))\frac{(2\pi\eta)^{d/2}}{2}. \tag{41}$$

Using the definition of $h_1$ in (28a) and Lemma A.1, we have

$$\int_{\mathbb{R}^d} \exp(-h_1(x))\mathrm{d}x = \exp\left(-f_y^\eta(\tilde{x}_J) + \delta\right)(2\pi\eta)^{d/2}. \tag{42}$$

Using (40), (41) and the above identity, we conclude that

$$\mathbb{P}\left(U \leq \frac{\exp(-f_y^\eta(X))}{\exp(-h_1(X))}\right) \geq \frac{1}{2}\exp(-f_y^\eta(x^*) + f_y^\eta(\tilde{x}_J) - \delta) \geq \frac{1}{2}\exp(-\delta),$$

and the expected number of the iterations is

$$\frac{1}{\mathbb{P}\left(U \leq \frac{\exp(-f_y^\eta(X))}{\exp(-h_1(X))}\right)} \leq 2\exp(\delta).$$

■

### 5.2 Sampling from semi-smooth potentials

We now proceed to bound the total complexity to sample from a log-concave distribution $\nu$ in (1) with a semi-smooth potential $f$. We combine our efficient algorithm (Algorithm 4) of RGO for semi-smooth potentials in Section 5.1 and the convergent results for ASF, namely Theorems 5.1 and 5.2, to achieve this goal.

First, using Theorem 5.1 we establish the following result.

**Theorem 5.6.** *Assume $f$ is convex and $L_\alpha$-semi-smooth, then Algorithm 1, initialized with $\rho_0^X$ and stepsize $\eta \asymp 1/(L_\alpha^{\frac{2}{\alpha+1}} d)$, using Algorithm 4 as an RGO has the iteration-complexity bound*

$$\mathcal{O}\left(\frac{L_\alpha^{\frac{2}{\alpha+1}} d W_2^2(\rho_0^X, \nu)}{\varepsilon}\right) \tag{43}$$

*to achieve $\varepsilon$ error to the target $\nu \propto \exp(-f)$ in terms of KL divergence.*

**Proof**: The result follows directly from Theorem 5.1, Theorem 3.5 and Proposition 5.5 with the choice of stepsize $\eta \asymp 1/(L_\alpha^{\frac{2}{\alpha+1}} d)$. ∎

Next, using Theorem 5.2 we establish the following result.

**Theorem 5.7.** *Assume $f$ is convex and $L_\alpha$-semi-smooth and $\pi^X \propto \exp(-f)$ satisfies $C_{\mathrm{PI}}$-PI, then Algorithm 1, initialized with $\rho_0^X$ and stepsize $\eta \asymp 1/(L_\alpha^{\frac{2}{\alpha+1}} d)$, using Algorithm 4 as an RGO has the iteration-complexity bound*

$$\tilde{\mathcal{O}}\left(C_{\mathrm{PI}} L_\alpha^{\frac{2}{\alpha+1}} d\right) \tag{44}$$

*to achieve $\varepsilon$ error to the target $\nu \propto \exp(-f)$ in terms of Chi-squared divergence, and*

$$\tilde{\mathcal{O}}\left(C_{\mathrm{PI}} L_\alpha^{\frac{2}{\alpha+1}} q d R_{q,\nu}(\rho_0^X)\right) \tag{45}$$

*to achieve $\varepsilon$ error in terms of Rényi divergence $R_{q,\nu}$ ($q \geq 2$).*

**Proof**: The result is a direct consequence of Theorem 5.2, Theorem 3.5 and Proposition 5.5 with the choice of stepsize $\eta \asymp 1/(L_\alpha^{\frac{2}{\alpha+1}} d)$. ∎

### 5.3 Sampling from composite potentials

In this section, we consider sampling from a log-concave distribution $\nu \propto \exp(-f(x))$ associated with a composite potential $f$. In particular, we consider the setting where $f = f_1 + f_2$, and $f_1, f_2$ are convex, $L_1$-smooth and convex, $L_\alpha$-semi-smooth, respectively. Clearly, such a $f$ satisfies that, for every $u, v \in \mathbb{R}^d$,

$$\|f'(u) - f'(v)\| \leq L_\alpha \|u - v\|^\alpha + L_1 \|u - v\|. \tag{46}$$

We shall call such a potential $(L_1, L_\alpha)$-smooth-semi-smooth. Note that all the results in this section apply to any convex and $(L_1, L_\alpha)$-smooth-semi-smooth potentials. These potentials do not have to be a composite one as above and our algorithms do not rely on the decomposition of the potential.

This setting is a generalization of the semi-smooth setting studied in the previous sections since it reduces to the latter when $L_1 = 0$. It turns out that both Algorithm 1 and the implementation for RGO, Algorithm 4, developed for semi-smooth sampling can be applied directly to this new setting with properly chosen step sizes. Below we extend the analysis in Sections 5.1 and 5.2 to this more general setting and establish corresponding iteration-complexity results.

Before presenting the results, we make some observations of the $(L_1, L_\alpha)$-smooth-semi-smooth assumption (46). If $L_\alpha = 0$, then (46) becomes to $\|\nabla f(u) - \nabla f(v)\| \leq L_1 \|u - v\|$, and hence $f$ is $L_1$-smooth. If $L_1 = 0$, then (46) reduces to (2), and hence $f$ is $L_\alpha$-semi-smooth.

To bound the expected number of runs in Algorithm 4, we need to construct a different $h_2$ to bound $f_y^\eta$; the one in (28b) no longer works in the setting with composite potentials. Based on the property of $(L_1, L_\alpha)$-smooth-semi-smooth potentials, we construct $h_2$ as follows. The proof is postponed to Appendix B.1.

**Lemma 5.8.** *Assume $f$ is convex and $(L_1, L_\alpha)$-smooth-semi-smooth and let $f_y^\eta$ be as in (27). Define*

$$h_2(x) := \frac{L_\alpha}{\alpha + 1} \|x - x^*\|^{\alpha+1} + \frac{1}{2\eta_{L_1}} \|x - x^*\|^2 + f_y^\eta(x^*) \tag{47}$$

*where*

$$\eta_{L_1} := \frac{\eta}{1 + \eta L_1} \tag{48}$$

*Then, $h_2(x) \geq f_y^\eta(x)$ for $\forall x \in \mathbb{R}^d$.*

With Lemma 5.8 in hand, we can bound the complexity of Algorithm 4 as follows. The proof is postponed to Appendix B.1.

**Proposition 5.9.** *If*

$$\eta \leq \min\left\{\frac{(\alpha+1)^{\frac{2}{\alpha+1}}}{(2L_\alpha)^{\frac{2}{\alpha+1}}d}, \frac{1}{L_1 d}\right\}, \tag{49}$$

*then the expected number of iterations of Algorithm 4 in rejection sampling is at most* $2\exp(1/2+\delta)$.

Through the above arguments, we show that our algorithm of the RGO designed for semi-smooth potentials is equally effective for $(L_1, L_\alpha)$-smooth-semi-smooth potentials. Combining Lemma 5.8 and Theorem 3.9 with the convergence results for ASF we obtain the following iteration-complexity bounds for sampling from $(L_1, L_\alpha)$-smooth-semi-smooth potentials. The proof is similar to that in Section 5.2 and is thus omitted.

**Theorem 5.10.** *Assume $f$ is a convex and $(L_1, L_\alpha)$-smooth-semi-smooth potential. Consider Algorithm 1, initialized with $\rho_0^X$ and stepsize $\eta \asymp \min\{1/(L_\alpha^{\frac{2}{\alpha+1}}d), 1/(L_1 d)\}$, using Algorithm 4 as a RGO.*
*1. Applying Theorem 5.1, the total complexity is*

$$\mathcal{O}\left(\frac{(L_\alpha^{\frac{2}{\alpha+1}} \vee L_1) d W_2^2(\rho_0^X, \nu)}{\varepsilon}\right)$$

*to achieve $\varepsilon$ error in terms of KL divergence.*
*2. If in addition, $\nu$ satisfies $(1/C_{\mathrm{PI}})$-PI, applying Theorem 5.2, the total complexity is*

$$\tilde{\mathcal{O}}\left((L_\alpha^{\frac{2}{\alpha+1}} \vee L_1) d C_{\mathrm{PI}}\right)$$

*to achieve $\varepsilon$ error in terms of Chi-squared divergence, and*

$$\tilde{\mathcal{O}}\left((L_\alpha^{\frac{2}{\alpha+1}} \vee L_1) q d C_{\mathrm{PI}} R_{q,\nu}(\rho_0^X)\right)$$

*to achieve $\varepsilon$ error in terms of Rényi divergence $R_{q,\nu}$ ($q \geq 2$).*

# 6 Proximal methods for online learning

**Algorithm 5** Online Bundle Method$(x_0, \lambda_1, \delta_1, \lambda_2, \delta_2, \ldots)$

0. Compute $\tau_1 = 4\lambda_1 M^2/(4\lambda_1 M^2 + \delta_1)$, and set $x_0^c = x_0$, $\tilde{x}_0 = x_0$, $\Gamma_0 = \gamma_0$, $u_0 = f_0(x_0)$ and $j = 1$;

1. Set functions $\Gamma_j = \tau\Gamma_{j-1} + (1-\tau)\gamma_j$, solve the subproblem

$$x_j := \operatorname*{argmin}_{u \in X} \left\{ \Gamma_j^\lambda(u) := \Gamma_j(u) + \frac{1}{2\lambda} V(u, x_{j-1}^c) \right\}, \tag{50}$$

and compute the optimal value $m_j := \Gamma_j^\lambda(x_j)$. Compute

$$\tilde{x}_j = \tau\tilde{x}_{j-1} + (1-\tau)x_j \tag{51}$$
$$u_j = \tau u_{j-1} + (1-\tau)f_j(x_j); \tag{52}$$

2. **If**

$$t_j := u_j - m_j \le \delta_j, \tag{53}$$

2.a) **then** perform a serious iteration, i.e., set $x_j^c \leftarrow x_j$, $\Gamma_j \leftarrow \gamma_j$ and $F_j \leftarrow f_j$, choose $\lambda_{j+1} > 0$ and $\delta_{j+1} > 0$, and compute

$$\tau_{j+1} = \frac{4\lambda_{j+1}M^2}{4\lambda_{j+1}M^2 + \delta_{j+1}}; \tag{54}$$

2.b) **else** perform a null iteration, i.e., set $x_j^c \leftarrow x_{j-1}^c$, $\lambda_{j+1} \leftarrow \lambda_j$, $\delta_{j+1} \leftarrow \delta_j$ and $\tau_{j+1} \leftarrow \tau_j$;

3. Set $j \leftarrow j+1$ and go to step 1.

**Lemma 6.1.** *Let $\ell_1$ be as in Proposition 3.8. For $j = \ell_0, \ell_0+1, \ldots, \ell_1 - 1$, we have $t_{j+1} \le \tau t_j + (1-\tau)\delta/2$.*

**Proof**: Using (50) and Lemma **??** with $\psi = 2\lambda\Gamma_j$, $a = x_j$, $b = x_{j-1}^c$ and $u = x_{j+1}$, we have

$$\Gamma_j(x_j) + \frac{1}{2\lambda}V(x_j, x_{j-1}^c) + \frac{1}{2\lambda}V(x_{j+1}, x_j) \le \Gamma_j(x_{j+1}) + \frac{1}{2\lambda}V(x_{j+1}, x_{j-1}^c).$$

It follows from the definition of $m_j$ in step 1 of OBM, the fact that $\Gamma_j = \tau_j\Gamma_{j-1} + (1-\tau_j)\gamma_{j-1}$ and the inequality above that

$$\begin{aligned} m_{j+1} = \Gamma_{j+1}^\lambda(x_{j+1}) &= \tau\Gamma_j^\lambda(x_{j+1}) + (1-\tau)\gamma_j^\lambda(x_{j+1}) \\ &\ge \tau m_j + (1-\tau)\gamma_j(x_{j+1}) + \frac{\tau}{2\lambda}V(x_{j+1}, x_j) \\ &\ge \tau m_j + (1-\tau)\gamma_j(x_{j+1}) + \frac{\tau}{2\lambda}\|x_{j+1} - x_j\|^2, \end{aligned}$$

where the last inequality is due to the fact that $V(x,y) \ge \|x-y\|^2$. The above inequality, (**??**) with $x = x_{j+1}$ and (**??**) imply that

$$m_{j+1} \ge \tau m_j + (1-\tau)\left( f_j(x_{j+1}) + \frac{L_\alpha^{\frac{2}{\alpha+1}}}{2[(\alpha+1)\delta]^{\frac{1-\alpha}{\alpha+1}}}\|x_{j+1} - x_j\|^2 - \frac{L_\alpha}{\alpha+1}\|x_{j+1} - x_j\|^{\alpha+1} \right),$$

which together with the definition of $t_j$ in (53) yields that

$$m_{j+1} \geq \tau u_j + (1-\tau) f_j(x_{j+1}) - \tau t_j + (1-\tau) \left( \frac{L_\alpha^{\frac{2}{\alpha+1}}}{2[(\alpha+1)\delta]^{\frac{1-\alpha}{\alpha+1}}} \|x_{j+1} - x_j\|^2 - \frac{L_\alpha}{\alpha+1} \|x_{j+1} - x_j\|^{\alpha+1} \right).$$

Using the Young's inequality $ab \leq a^p/p + b^q/q$ with

$$a = \frac{L_\alpha}{(\alpha+1)\delta^{\frac{1-\alpha}{2}}} \|x_{j+1} - x_j\|^{\alpha+1}, \quad b = \delta^{\frac{1-\alpha}{2}}, \quad p = \frac{2}{\alpha+1}, \quad q = \frac{2}{1-\alpha},$$

we obtain

$$\frac{L_\alpha}{\alpha+1} \|x_{j+1} - x_j\|^{\alpha+1} \leq \frac{L_\alpha^{\frac{2}{\alpha+1}}}{2[(\alpha+1)\delta]^{\frac{1-\alpha}{\alpha+1}}} \|x_{j+1} - x_j\|^2 + \frac{(1-\alpha)\delta}{2}.$$

Plugging the above inequality into (**??**), we have

$$m_{j+1} \geq u_{j+1} - \tau t_j - (1-\tau)\frac{(1-\alpha)\delta}{2},$$

which, in view of (8), implies that

$$t_{j+1} \leq \tau t_j + (1-\tau)\frac{(1-\alpha)\delta}{2}.$$

Therefore, the lemma directly follows. ∎

### 6.1 Regret Bound Analysis

In this section we present the proof of Theorem **??** which we present formally in the following result. Throughout the section, we assume the existence of an oracle for finding $\tilde{x}_j$ in OBM. Examples of such an oracle are provided in Section **??** (see Lemma **??**).

**Theorem 6.2.** *For any distance generating function $\omega : X \to \mathbb{R}$ and sequence of convex functions $f_1, \ldots, f_T$, OBM guarantees:*

- *For $\lambda_k = \frac{D}{M\sqrt{T}}$ and $\delta_k = \frac{cMD}{\sqrt{T}}$, we have* $\text{Regret}_T \leq \left(c + \frac{1}{2}\right)\left(MD\sqrt{T}\right)$.
- *For $\lambda_k = \frac{D}{M}$ and $\delta_k = \frac{cMD}{k}$, we have* $\text{Regret}_T \leq MD\left(\frac{1}{2} + c(\ln T + 1)\right)$.

Here $c$ is an absolute constant that allow us to tune oracle (51). The proof of Theorem 6.2 relies on a general result that bounds the regret of OBM (Lemma 6.3). Consider the sequences $\{\Gamma_j\}$, $\{\gamma_j\}$, $\{F_j\}$, $\{f_j\}$, $\{x_j\}$ and $\{\tilde{x}_j\}$ constructed by OBM, and let $\{j_k : k \geq 0\}$ denotes the sequence of serious iteration indices generated by OBM. Note that $j_0 = 0$, and $\lambda_k$ and $\delta_k$ correspond to the input of OBM. Define $z_0 := x_0$, $\tilde{z}_0 := x_0$ and, for every $k \geq 1$,

$$z_k := x_{j_k}, \quad \tilde{z}_k := \tilde{x}_{j_k}, \quad \tilde{f}_k := f_{j_k}, \quad \tilde{F}_k := F_{j_k}, \quad \tilde{\gamma}_k := \gamma_{j_k}, \quad \tilde{\Gamma}_k := \Gamma_{j_k}. \tag{55}$$

**Lemma 6.3.** *For every $T \geq 1$ and $z \in X$, we have*

$$\text{Regret}_T = \sum_{k=1}^{T} \tilde{F}_k(\tilde{z}_k) - \tilde{F}_k(z) \leq \sum_{k=1}^{T} \delta_k + \frac{1}{2}\sum_{k=2}^{T} \left(\frac{1}{\lambda_k} - \frac{1}{\lambda_{k-1}}\right) V(z, z_{k-1}) + \frac{1}{2\lambda_1} V(z, z_0).$$

Before proving the lemma, let us show how to conclude Theorem 6.2. We present all the choices of $(\lambda_k, \delta_k)$ and derive their corresponding regret bounds.

**A1.** For a general function $\omega$ (and hence a general Bregman divergence $V$) and a fixed $T$, let $D^2 := \max_{z\in X} V(z, z_0)$ and $\lambda_k = \frac{D}{M\sqrt{T}}$, $\delta_k = \frac{cMD}{\sqrt{T}}$ for some $c > 0$. It follows from Lemma 6.3 that

$$\text{Regret}_T \le \sum_{k=1}^{T} \delta_k + \frac{V(z, z_0)}{2\lambda_1} \le \left(c + \frac{1}{2}\right) MD\sqrt{T} = \mathcal{O}(\sqrt{T}).$$

**A2.** For a general Bregman divergence, let $\lambda_k = \frac{D}{M}$, $\delta_k = \frac{cMD}{k}$ for some $c > 0$, then it follows from Lemma 6.3 that

$$\text{Regret}_T \le \sum_{k=1}^{T} \delta_k + \frac{V(z, z_0)}{2\lambda_1} \le \sum_{k=1}^{T} \frac{cMD}{k} + \frac{MD}{2} \le \left[(\ln T + 1)c + \frac{1}{2}\right] MD = \mathcal{O}(\log T).$$

In addition to the previous choices of $(\lambda_k, \delta_k)$ we also have the standard subgradient method regret guarantee:

**A3.** Suppose $\omega(x) = \|x\|^2$, and let $D_X$ be the diameter of $X$ and $\lambda_k = \frac{D_X}{M\sqrt{k}}$, $\delta_k = \frac{cMD_X}{\sqrt{k}}$, for some $c > 0$. It follows from Lemma 6.3 that

$$\begin{aligned}\text{Regret}_T &\le \sum_{k=1}^{T} \delta_k + \frac{D_X^2}{2}\left[\sum_{k=2}^{T}\left(\frac{1}{\lambda_k} - \frac{1}{\lambda_{k-1}}\right) + \frac{1}{\lambda_1}\right] = \sum_{k=1}^{T} \delta_k + \frac{D_X^2}{2\lambda_T} \\ &\le \sum_{k=1}^{T} \frac{cMD_X}{\sqrt{k}} + \frac{MD_X\sqrt{T}}{2} \le \left(c + \frac{1}{2}\right) MD_X\sqrt{T} = \mathcal{O}(\sqrt{T}).\end{aligned}$$

**Proof**: [of Lemma 6.3] Recall that $t_j = u_j - m_j = u_j - \Gamma_j^{\lambda_j}(x_j)$, so the termination criterion of null iterations (53) is equivalent to $u_k - \tilde{\Gamma}_k^{\lambda_k}(z_k) \le \delta_k$, i.e.,

$$u_k - \tilde{\Gamma}_k(z_k) - \frac{1}{2\lambda_k} V(z_k, z_{k-1}) \le \delta_k. \tag{56}$$

It is easy to see from (50) and (55) that $z_k = \text{argmin}\left\{\tilde{\Gamma}_k(u) + \frac{1}{2\lambda_k} V(u, z_{k-1}) : u \in X\right\}$. Using the above inequality and Lemma **??** with $\psi = 2\lambda_k\tilde{\Gamma}_k$, $a = z_k$ and $b = z_{k-1}$, we have for every $k \ge 1$ and $z \in X$,

$$\tilde{\Gamma}_k(z_k) + \frac{1}{2\lambda_k} V(z_k, z_{k-1}) \le \tilde{\Gamma}_k(z) + \frac{1}{2\lambda_k} V(z, z_{k-1}) - \frac{1}{2\lambda_k} V(z, z_k),$$

and hence that

$$u_k - \tilde{\Gamma}_k(z) \le u_k - \tilde{\Gamma}_k(z_k) - \frac{1}{2\lambda_k} V(z_k, z_{k-1}) + \frac{1}{2\lambda_k} V(z, z_{k-1}) - \frac{1}{2\lambda_k} V(z, z_k).$$

It follows from (56) that

$$\tilde{F}_k(\tilde{z}_k) - \tilde{\Gamma}_k(z) \le u_k - \tilde{\Gamma}_k(z) \le \delta_k + \frac{1}{2\lambda_k}\left(V(z, z_{k-1}) - V(z, z_k)\right). \tag{57}$$

Summing the above inequality from $k = 1$ to $T$, we have

$$\begin{aligned}\sum_{k=1}^{T}\left(\tilde{F}_k(\tilde{z}_k) - \tilde{\Gamma}_k(z)\right) &\le \sum_{k=1}^{T} \delta_k + \sum_{k=1}^{T} \frac{1}{2\lambda_k}\left(V(z, z_{k-1}) - V(z, z_k)\right) \\ &\le \sum_{k=1}^{T} \delta_k + \frac{1}{2}\sum_{k=2}^{T}\left(\frac{1}{\lambda_k} - \frac{1}{\lambda_{k-1}}\right) V(z, z_{k-1}) + \frac{1}{2\lambda_1} V(z, z_0).\end{aligned}$$

We conclude from the fact that $\tilde{F}_k \ge \tilde{\Gamma}_k$ and the above inequality that the lemma holds. ∎

# 7 Conclusion

In this paper we presented a novel sampling algorithm from convex semi-smooth potentials or convex composite potentials with semi-smooth components. Our algorithm is based on the recent ASF which utilizes Gibbs sampling over an augmented distribution. In each iteration of the ASF, one needs to sample from a quadratically regularized version of the target potential, which is itself a challenging task due to the non-smoothness of the problem. In this work we presented a rejection sampling based scheme with a tailored proposal to sample from the regularized version of the target potential. Moreover, we developed a novel technique to bound the complexity of this scheme. By combining our scheme with the ASF we established a sampling algorithm for convex semi-smooth potentials or convex composite potentials with semi-smooth components, with better complexity than all the existing methods. In the future, we plan to investigate the sampling settings where the potential is semi-smooth or composite with a semi-smooth component, but not necessarily convex.

# Acknowledgement

This work was supported by NSF under grant 1942523 and 2008513.

# References

[1] David Applegate and Ravi Kannan. Sampling and integration of near log-concave functions. In *Proceedings of the twenty-third annual ACM symposium on Theory of computing*, pages 156–163, 1991.

[2] E. Bernton. Langevin Monte Carlo and JKO splitting. In *Conference On Learning Theory*, pages 1777–1798. PMLR, 2018.

[3] Dimitris Bertsimas and Santosh Vempala. Solving convex programs by random walks. *Journal of the ACM (JACM)*, 51(4):540–556, 2004.

[4] Nawaf Bou-Rabee and Martin Hairer. Nonasymptotic mixing of the MALA algorithm. *IMA Journal of Numerical Analysis*, 33(1):80–110, 2013.

[5] Niladri Chatterji, Jelena Diakonikolas, Michael I Jordan, and Peter Bartlett. Langevin Monte Carlo without smoothness. In *International Conference on Artificial Intelligence and Statistics*, pages 1716–1726. PMLR, 2020.

[6] Yongxin Chen, Sinho Chewi, Adil Salim, and Andre Wibisono. Improved analysis for a proximal algorithm for sampling. *arXiv preprint arXiv:2202.06386*, 2022.

[7] Yuansi Chen, Raaz Dwivedi, Martin J Wainwright, and Bin Yu. Fast MCMC sampling algorithms on polytopes. *The Journal of Machine Learning Research*, 19(1):2146–2231, 2018.

[8] Yuansi Chen, Raaz Dwivedi, Martin J Wainwright, and Bin Yu. Fast mixing of metropolized Hamiltonian Monte Carlo: Benefits of multi-step gradients. *J. Mach. Learn. Res.*, 21:92–1, 2020.

[9] Xiang Cheng and Peter Bartlett. Convergence of Langevin MCMC in KL-divergence. In *Algorithmic Learning Theory*, pages 186–211. PMLR, 2018.

[10] Xiang Cheng, Niladri S Chatterji, Peter L Bartlett, and Michael I Jordan. Underdamped Langevin MCMC: A non-asymptotic analysis. In *Conference on learning theory*, pages 300–323. PMLR, 2018.

[11] Sinho Chewi, Murat A Erdogdu, Mufan Bill Li, Ruoqi Shen, and Matthew Zhang. Analysis of Langevin Monte Carlo from Poincaré to Log-Sobolev. *arXiv preprint arXiv:2112.12662*, 2021.

[12] Arnak S Dalalyan. Theoretical guarantees for approximate sampling from smooth and log-concave densities. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(3):651–676, 2017.

[13] Alain Durmus, Szymon Majewski, and Błażej Miasojedow. Analysis of Langevin Monte Carlo via convex optimization. *The Journal of Machine Learning Research*, 20(1):2666–2711, 2019.

[14] Alain Durmus, Eric Moulines, and Marcelo Pereyra. Efficient Bayesian computation by proximal Markov Chain Monte Carlo: when Langevin meets Moreau. *SIAM Journal on Imaging Sciences*, 11(1):473–506, 2018.

[15] Raaz Dwivedi, Yuansi Chen, Martin J Wainwright, and Bin Yu. Log-concave sampling: Metropolis-Hastings algorithms are fast! In *Conference on learning theory*, pages 793–797. PMLR, 2018.

[16] Martin Dyer, Alan Frieze, and Ravi Kannan. A random polynomial-time algorithm for approximating the volume of convex bodies. *Journal of the ACM (JACM)*, 38(1):1–17, 1991.

[17] Murat A Erdogdu and Rasa Hosseinzadeh. On the convergence of Langevin Monte Carlo: The interplay between tail growth and smoothness. *arXiv preprint arXiv:2005.13097*, 2020.

[18] Yoav Freund, Yi-An Ma, and Tong Zhang. When is the convergence time of Langevin algorithms dimension independent? a composite optimization viewpoint. *arXiv preprint arXiv:2110.01827*, 2021.

[19] Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian Data Analysis*. CRC press, 2013.

[20] Stuart Geman and Donald Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, (6):721–741, 1984.

[21] Ulf Grenander and Michael I Miller. Representations of knowledge in complex systems. *Journal of the Royal Statistical Society: Series B (Methodological)*, 56(4):549–581, 1994.

[22] Adam Tauman Kalai and Santosh Vempala. Simulated annealing for convex optimization. *Mathematics of Operations Research*, 31(2):253–266, 2006.

[23] Ravi Kannan, László Lovász, and Miklós Simonovits. Random walks and an $O^*(n^5)$ volume algorithm for convex bodies. *Random Structures & Algorithms*, 11(1):1–50, 1997.

[24] Werner Krauth. *Statistical mechanics: algorithms and computations*, volume 13. OUP Oxford, 2006.

[25] Yin Tat Lee, Ruoqi Shen, and Kevin Tian. Structured logconcave sampling with a restricted Gaussian oracle. In *Conference on Learning Theory*, pages 2993–3050. PMLR, 2021.

[26] Yin Tat Lee and Santosh S Vempala. Convergence rate of Riemannian Hamiltonian Monte Carlo and faster polytope volume computation. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1115–1121, 2018.

[27] Yin Tat Lee and Santosh Srinivas Vempala. Eldan's stochastic localization and the KLS hyperplane conjecture: an improved lower bound for expansion. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 998–1007. IEEE, 2017.

[28] Joseph Lehec. The Langevin Monte Carlo algorithm in the non-smooth log-concave case. *Available on arXiv:2101.10695*, 2021.

[29] Claude Lemaréchal. An extension of davidon methods to non differentiable problems. In *Nondifferentiable optimization*, pages 95–109. Springer, 1975.

[30] Claude Lemaréchal. Nonsmooth optimization and descent methods. 1978.

[31] Jiaming Liang and Yongxin Chen. A proximal algorithm for sampling from non-smooth potentials. *arXiv preprint arXiv:2110.04597*, 2021.

[32] Jiaming Liang and Renato D. C. Monteiro. A proximal bundle variant with optimal iteration-complexity for a large range of prox stepsizes. *SIAM Journal on Optimization*, 31(4):2955–2986, 2021.

[33] Jiaming Liang and Renato D. C. Monteiro. A unified analysis of a class of proximal bundle methods for hybrid convex composite optimization problems. *Available on arXiv:2110.01084*, 2021.

[34] László Lovász and Santosh Vempala. Fast algorithms for logconcave functions: Sampling, rounding, integration and optimization. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 57–68. IEEE, 2006.

[35] Robert Mifflin. A modification and an extension of Lemaréchal's algorithm for nonsmooth minimization. In *Nondifferential and variational techniques in optimization*, pages 77–90. Springer, 1982.

[36] Wenlong Mou, Nicolas Flammarion, Martin J Wainwright, and Peter L Bartlett. An efficient sampling algorithm for non-smooth composite potentials. *Available on arXiv:1910.00551*, 2019.

[37] Radford M Neal. MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2(11):2, 2011.

[38] Dao Nguyen, Xin Dang, and Yixin Chen. Unadjusted Langevin algorithm for non-convex weakly smooth potentials. *arXiv preprint arXiv:2101.06369*, 2021.

[39] Neal Parikh and Stephen Boyd. Proximal algorithms. *Foundations and Trends in optimization*, 1(3):127–239, 2014.

[40] Giorgio Parisi. Correlation functions and computer simulations. *Nuclear Physics B*, 180(3):378–384, 1981.

[41] Gareth O Roberts and Jeffrey S Rosenthal. Optimal scaling of discrete approximations to Langevin diffusions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60(1):255–268, 1998.

[42] Gareth O Roberts and Osnat Stramer. Langevin diffusions and Metropolis-Hastings algorithms. *Methodology and Computing in Applied Probability*, 4(4):337–357, 2002.

[43] Gareth O Roberts and Richard L Tweedie. Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli*, pages 341–363, 1996.

[44] Ralph Tyrrell Rockafellar. Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization*, 14(5):877–898, 1976.

[45] Adil Salim and Peter Richtárik. Primal dual interpretation of the proximal stochastic gradient Langevin algorithm. *Advances in Neural Information Processing Systems*, 33:3786–3796, 2020.

[46] Ruoqi Shen and Yin Tat Lee. The randomized midpoint method for log-concave sampling. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[47] Ruoqi Shen, Kevin Tian, and Yin Tat Lee. Composite logconcave sampling with a restricted Gaussian oracle. *Available on arXiv:2006.05976*, 2020.

[48] Jack W Sites Jr and Jonathon C Marshall. Delimiting species: a renaissance issue in systematic biology. *Trends in Ecology & Evolution*, 18(9):462–470, 2003.

[49] Cédric Villani. *Topics in optimal transportation*, volume 58. American Mathematical Soc., 2021.

[50] JG Wendel. Note on the gamma function. *The American Mathematical Monthly*, 55(9):563–564, 1948.

[51] Andre Wibisono. Sampling as optimization in the space of measures: The Langevin dynamics as a composite optimization problem. In *Conference on Learning Theory*, pages 2093–3027. PMLR, 2018.

[52] Philip Wolfe. A method of conjugate subgradients for minimizing nondifferentiable functions. In *Nondifferentiable optimization*, pages 145–173. Springer, 1975.

[53] Zhuoran Yang, Yufeng Zhang, Yongxin Chen, and Zhaoran Wang. Variational transport: A convergent particle-based algorithm for distributional optimization. *Available on arXiv:2012.11554*, 2020.

# Contents

# A Technical results

**Lemma A.1** (Gaussian integral)**.** *For any $\eta > 0$,*

$$\int_{\mathbb{R}^d} \exp\left(-\frac{1}{2\eta}\|x\|^2\right) \mathrm{d}x = (2\pi\eta)^{d/2}.$$

The following lemma provides both lower and upper bounds on the ratio of gamma functions. Its proof can be found in [50].

**Lemma A.2** (Wendel's double inequality)**.** *For $0 < s < 1$ and $t > 0$, the gamma function defined as in* (34) *satisfies*

$$\left(\frac{t}{t+s}\right)^{1-s} \leq \frac{\Gamma(t+s)}{t^s\Gamma(t)} \leq 1,$$

*or equivalently,*

$$t^{1-s} \leq \frac{\Gamma(t+1)}{\Gamma(t+s)} \leq (t+s)^{1-s}. \tag{58}$$

The following lemma states a basic property for convex and $(L_1, L_\alpha)$-smooth-semi-smooth functions satisfying (46).

**Lemma A.3.** *Assume $f$ is convex and $(L_1, L_\alpha)$-smooth-semi-smooth. Then, for every $u, v \in \mathbb{R}^d$, we have*

$$f(u) - f(v) - \langle f'(v), u - v\rangle \leq \frac{L_\alpha}{\alpha+1}\|u-v\|^{\alpha+1} + \frac{L_1}{2}\|u-v\|^2. \tag{59}$$

*As a consequence, if $f$ is convex and $L_\alpha$-semi-smooth, then for every $u, v \in \mathbb{R}^d$, we have*

$$f(u) - f(v) - \langle f'(v), u - v\rangle \leq \frac{L_\alpha}{\alpha+1}\|u-v\|^{\alpha+1}. \tag{60}$$

**Proof**: We first consider the case when $f$ is convex and $(L_1, L_\alpha)$-smooth-semi-smooth. It is easy to see that

$$\begin{aligned} f(u) &= f(v) + \int_0^1 \langle f'(v+\tau(v-u)), u-v\rangle \mathrm{d}\tau \\ &= f(v) + \langle f'(v), u-v\rangle + \int_0^1 \langle f'(v+\tau(v-u)) - f'(v), u-v\rangle \mathrm{d}\tau. \end{aligned}$$

Using the above identity, the Cauchy-Schwarz inequality, and (46), we have

$$\begin{aligned} f(u) - f(v) - \langle f'(v), u-v\rangle &= \int_0^1 \langle f'(v+\tau(v-u)) - f'(v), u-v\rangle \mathrm{d}\tau \\ &\leq \int_0^1 \left\|f'(v+\tau(v-u)) - f'(v)\right\| \|u-v\| \mathrm{d}\tau \\ &\leq \int_0^1 \left(L_1\tau\|u-v\| + L_\alpha\tau^\alpha\|u-v\|^\alpha\right)\|u-v\|\mathrm{d}\tau \\ &= \int_0^1 \left(L_1\tau\|u-v\|^2 + L_\alpha\tau^\alpha\|u-v\|^{\alpha+1}\right)\mathrm{d}\tau \\ &= \frac{L_1}{2}\|u-v\|^2 + \frac{L_\alpha}{\alpha+1}\|u-v\|^{\alpha+1}. \end{aligned}$$

Moreover, if $f$ is convex and $L_\alpha$-semi-smooth, we easily obtain (60) by setting $L_1 = 0$ in (59). ∎

# B Missing proofs

## B.1 Missing proofs in Section 5.3

**Proof of Lemma 5.8:** It follows from the same argument as in the proof of Lemma 5.3 that (30) holds. Using (30), Lemma 3.1(d), and (59) with $(u, v) = (x, x^*)$, we conclude that

$$f_y^\eta(x) - f_y^\eta(x^*) \leq \frac{L_\alpha}{\alpha+1}\|x - x^*\|^{\alpha+1} + \frac{L_1}{2}\|x - x^*\|^2 + \frac{1}{2\eta}\|x - x^*\|^2.$$

The lemma immediately follows from the above inequality, and the definitions of $h_2$ and $\eta_{L_1}$ in (47) and (48), respectively. ∎

**Proof of Proposition 5.9:** It follows from (49) that (39) holds and hence that $\eta_{L_1}$ satisfies (39) in view of the definition of $\eta_{L_1}$ in (48). If follows directly from the definition of $h_2$ in (47) that

$$\int_{\mathbb{R}^d} \exp(-h_2(x))\mathrm{d}x = \exp(-f_y^\eta(x^*)) \int_{\mathbb{R}^d} \exp\left(-\frac{1}{2\eta_{L_1}}\|x - x^*\|^2 - \frac{L_\alpha}{\alpha+1}\|x - x^*\|^{\alpha+1}\right)\mathrm{d}x.$$

Using a similar argument as in the proof of Proposition 5.5, we have (40) holds and

$$\int_{\mathbb{R}^d} \exp(-f_y^\eta(x))\mathrm{d}x \geq \int_{\mathbb{R}^d} \exp(-h_2(x))\mathrm{d}x \geq \exp(-f_y^\eta(x^*))\frac{(2\pi\eta_{L_1})^{d/2}}{2}.$$

Moreover, it follows from the same argument as in the proof of Proposition 5.5 that (42) holds. Using (40), (42), the above inequality, and the definition of $\eta_{L_1}$ in (48), we have

$$\mathbb{P}\left(U \leq \frac{\exp(-f_y^\eta(X))}{\exp(-h_1(X))}\right) \geq \frac{1}{2}\exp\left(f_y^\eta(\tilde{x}_J) - f_y^\eta(x^*) - \delta\right)\left(\frac{\eta_{L_1}}{\eta}\right)^{d/2} \geq \frac{1}{2}\exp(-\delta)\left(\frac{1}{1+\eta L_1}\right)^{d/2}.$$

The above inequality and (49) imply that

$$\frac{1}{\mathbb{P}\left(U \leq \frac{\exp(-f_y^\eta(X))}{\exp(-h_1(X))}\right)} \leq 2\exp(\delta)(1+\eta L_1)^{d/2} \leq 2\exp(\delta)\left(1+\frac{1}{d}\right)^{d/2} \leq 2\exp(1/2+\delta),$$

where the last inequality is due to the fact that $(1+1/d)^d \leq e$. ∎