# A First Approach to Universal Second-Order Acceleration for Convex Minimization

**Anonymous Author(s)**
Affiliation
Address
email

## Abstract

This work proposes a universal and adaptive second-order method for minimizing second-order smooth, convex functions. Our algorithm achieves $O(\sigma/\sqrt{T})$ convergence when the oracle feedback is stochastic with variance $\sigma^2$, and improves its convergence to $O(1/T^3)$ with deterministic oracles, where $T$ is the number of iterations. Our method also interpolates these rates without knowing the nature of the oracle apriori, which is enabled by a parameter-free adaptive step-size that is oblivious to the knowledge of smoothness modulus, variance bounds and the diameter of the constrained set. To our knowledge, this is the first universal algorithm with such global guarantees within the second-order optimization literature.

## 1 Introduction

Over the last few decades, first-order (convex) minimization methods have gained popularity for modern machine learning and optimization problems due to their efficient per-iteration cost and *global convergence* properties. The literature on first-order methods is rather dense and extensive with a concrete, thorough understanding of the optimal *global* convergence behavior. Focusing on the more relevant settings of smooth, convex minimization, the lower bounds have been well-established; $O(\sigma/\sqrt{T})$ when the gradient feedback is noisy with variance $\sigma^2$, and $O(1/T^2)$ under deterministic first-order oracles [38, 43]. Under slight variations of the aforementioned problem setting, there exists an extensive amount of work that enjoys the latter, "accelerated" rate [41, 42, 39, 50, 52, 29, 2, 31, 51, 13, 12, 26, 22].

On the contrary to its first-order analogue, the literature on *global convergence* of second-order, smooth methods is notably sparse with many open questions standing even in the simplest problem formulations. Following the pioneering works of Bennett [6], Kantorovich [24], Newton's method and its variations [30, 33] are considered as the staple of second-order methods in optimization. Although its powerful local convergence behavior has been repeatedly demonstrated [11, 28], studies on its global behavior are relatively limited. Prior attempts at tackling global convergence mostly make additional structural assumptions on the objective function [46, 34, 28] or assume extra regularity conditions on the Hessian [25] beyond the simplest smooth and convex setting. Over the last decade, we have witnessed important progress towards a more complete theory of globally-convergent second-order methods (more on this shortly), and yet there remains many important questions unanswered, which we will delve into in this paper.

To motivate the perspective in our technical endeavour, we take a small detour to introduce the idea of *universality*, which we particularly characterize as *adaptation to the level of noise in oracle feedback*. Enabled by the recent advances in online optimization, universal first-order algorithms essentially

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

attain the $O(\sigma/\sqrt{T} + 1/T^2)$ convergence, interpolating between stochastic and deterministic rates. There exist a plethora of algorithms that enjoy this rate under different sets of assumptions [29, 26, 16, 22] and we have a quite solid understanding of this optimal behavior. However, we observe that such universal results do not exist in second-order literature, hence, it is only natural to ask,

*Can we design a simple second-order method that will achieve accelerated universal rates beyond $O(\sigma/\sqrt{T} + 1/T^2)$?*

More recently, global sub-linear convergence rates for second-order methods have been characterized by [44] for second-order smooth and convex setting. Essentially, the so-called Cubic Regularized Newton's Method combines the quadratic Taylor approximation in the typical Newton update with a cubic regularization term. At the expense of solving a cubic problem, this method achieves $O(1/T^2)$ convergence rate. Shortly after, Nesterov [40] proposes an accelerated version of the cubic regularization idea with $O(1/T^3)$ value convergence, pioneering a new direction of research in the study of globally-convergent second-order methods [35]. This idea has been studied further for different settings in convex optimization [19, 20] with the same accelerated $O(1/T^3)$ rate and extended to non-convex realm [8, 9], obtaining the analogous rates of $O(1/T^{2/3})$ and $O(1/T^{1/3})$ for finding first-order and second-order stationary points, respectively, leading the way for further investigations [5, 15, 10].

Notice that accelerated cubic regularization is *sub*-optimal such that recent studies prove a respective lower-bound for second-order smooth, convex problems as $O(1/T^{7/2})$ [1, 4]. A different line of research, which we will call "bisection-type" methods, shrinks the gap between the upper and lower bounds for achieving an *almost*-optimal (more on this shortly) convergence [45]. Pioneered by Monteiro and Svaiter [36], these class of algorithms propose a conceptual method where the step-size of the algorithm *implicitly* depends on the next iterate. To resolve, the authors propose a bisection procedure that simultaneously finds a step-size/next iterate pair that satisfies the conditions of the iterative update, which enables the convergence rate of $O(1/T^{7/2})$, modulo the complexity of bisection procedure. This idea was very recently generalized for higher-order tensor methods [17]. Not so surprisingly, the same construction finds application in variational inequality (VI) and min-max optimization literature [7, 21].

Although there are promising results with an increasing interest into second-order –and also higher-order– methods, we identify three main shortcomings in the literature, which we will systematically address in the sequel. First, bisection-type methods achieve the optimal convergence rate however, the search procedure is computationally very prohibitive [45, 32] and the resulting algorithms are complicated with many interconnected components. Indeed, finding a computationally-efficient, optimal second-order method is still an open problem according to [45]. On the other hand, cubic regularization-based ideas propose a simple construction that achieves acceleration beyond $O(1/T^2)$ however, similar to bisection methods, they either require the knowledge of smoothness constant or need to execute a standard line-search procedure to estimate it locally. A common drawback for both approaches is that the algorithmic constructions are designed for handling *only* deterministic oracle information and it is an open question whether such frameworks could accommodate stochastic first and second-order information.

**Our contributions:** To address the aforementioned issues, we developed the first universal and adaptive second-order algorithm, EXTRANEWTON, for convex minimization. We summarize our contributions as follows:

1. We prove the global convergence rate of EXTRANEWTON that interpolates simultaneously between $\mathcal{O}(LD^2/T^3)$ for deterministic and $\mathcal{O}(\sigma/\sqrt{T})$ for stochastic oracles, achieving the first universal convergence result in the literature.
2. Our method is completely oblivious to any problem-dependent parameters including smoothness modulus, variance bounds on stochastic oracles, diameter of the constraint set and any possible bounds on the gradient and Hessian.

**Table 1:** A survey on first and second-order algorithms with key properties

| | **AGD** [41] | **UniXGrad** [26] | **Reg. Newton** [35] | **Accelerated Cubic Reg.** [40] | **ANPE**[1] [36] | **ExtraNewton [ours]** |
|---|---|---|---|---|---|---|
| *Convergence Rate* | $\frac{1}{T^2}$ | $\frac{\sigma}{\sqrt{T}} + \frac{1}{T^2}$ | $\frac{1}{T^2}$ | $\frac{1}{T^3}$ | $\frac{1}{T^{7/2}}$ | $\frac{\sigma}{\sqrt{T}} + \frac{1}{T^3}$ |
| *Bisection* | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ |
| *Adapts to L* | ✗ | ✓ | ✗ | Partial | ✗ | ✓ |
| *Adapts to Oracle* | ✗ | ✓ | ✗ | ✗ | ✗ | ✓ |

3. We design the first adaptive step-size, in the sense of [14, 48], that successfully incorporates second-order information "on-the-fly". While doing so, we bypass any bisection or linesearch procedure, and propose a simple, intuitive algorithmic framework.

From a technical point of view, what will allow us to achieve these results is the combination of three principal ingredients: *(i)* proposing appropriate adjustments to Extra-Gradient [27] that was originally designed for solving variational inequalities and min/max problems; *(ii)* an "optimistic" weighted iterate averaging scheme accompanied by an appropriate gradient rescaling strategy in the spirit of [51, 13, 26] which allows us to obtain an accelerated rate of convergence by means of a generalized online-to-batch conversion (Theorem 3.3), and *(iii)* the glue that holds these elements together is an adaptive learning rate inspired by [48, 26, 3] which automatically rescales aggregated gradients and second order information. In what follows, we shall explicate these arguments.

## 2 Problem setup

Throughout the sequel, we will be focusing on solving (constrained) convex minimization problems of the general form:

$$\begin{aligned} \text{minimize} \quad & f(x), \\ \text{subject to} \quad & x \in \mathcal{X}. \end{aligned} \tag{Opt}$$

Formally, in the above $\mathcal{X}$ is a convex and compact subset of a $d$- dimensional normed space $\mathcal{V} \cong \mathbb{R}^d$, and $f : \mathcal{V} \to \mathbb{R} \cup \{+\infty\}$ is a proper, lower semi-continuous, convex function with $\mathrm{dom} f = \{x \in \mathbb{R}^d : f(x) < +\infty\} \subset \mathcal{X}$. To that end, we make the following blanket assumptions for (Opt). First in order to avoid trivialities we assume that the said problem admits at least a solution, i.e.

**Assumption 2.1.** *The solution set* $\mathcal{X}^* = \arg\min_{x \in \mathcal{X}} f(x)$ *of* (Opt) *is non-empty.*

Furthermore, we assume that there exists a Lipschitz continuous selection $x \mapsto \nabla^2 f(x) \in \mathbb{R}^{d \times d}$, i.e.,

$$\|\nabla^2 f(x) - \nabla^2 f(x')\| \leq L\|x - x'\| \quad \forall x, x' \in \mathcal{X} \tag{H-smooth}$$

and in addition it satisfies the second order approximation:

$$f(x) = f(x') + \langle \nabla f(x'), x - x' \rangle + \frac{1}{2}\langle \nabla^2 f(x')(x - x'), x - x' \rangle + O\left(\|x - x'\|^3\right) \tag{Taylor}$$

To that end, combining (H-smooth) and (Taylor) we readily get the following inequality:

$$\|\nabla f(x) - \nabla f(x') - \nabla^2 f(x')(x - x')\| \leq \frac{L}{2}\|x - x'\|^2 \tag{1}$$

The above equivalences are well-established and hence we omit their proofs (we defer for a panoramic view to [53])

**Oracle feedback structure** From an algorithmic point of view, we aim to solve (Opt) by using methods that require access to a (stochastic) first and second order-oracle. Before we move forward with the methodology,we shall introduce the definitions and notations for this oracle model which

[1]Note that the bisection procedure is computationally prohibitive, we defer the reader to [45], p.304-305.

we will use in algorithm definitions and technical discussions. For notational simplicity we denote stochastic gradients by $g(\cdot)$ and stochastic Hessian by $H(\cdot)$ such that

$$\begin{aligned} \mathbb{E}\left[\, g(x) \mid x \,\right] &= \nabla f(x), \qquad & \mathbb{E}\left[\, \|g(x) - \nabla f(x)\|^2 \mid x \,\right] \leq \sigma_g^2 \\ \mathbb{E}\left[\, H(x) \mid x \,\right] &= \nabla^2 f(x), \qquad & \mathbb{E}\left[\, \|H(x) - \nabla^2 f(x)\|^2 \mid x \,\right] \leq \sigma_H^2 \end{aligned} \tag{2}$$

Due to space constraints, we will also define an operator that accommodates second-order information and its respective stochastic counterpart.

$$\begin{aligned} \mathbf{F}(x; x') &= \nabla f(x') + \nabla^2 f(x')(x - x') \\ \tilde{\mathbf{F}}(x; x') &= g(x') + H(x')(x - x') \end{aligned} \tag{3}$$

where $\mathbf{F}$ is essentially the gradient (with respect to $x$) of the second-order Taylor polynomial. By definition, the operator $\mathbf{F}$ satisfies the second-order smoothness property in Eq. (1)

## 3 Method

In this section, we shall establish our universal second-order framework. Our presentation evolves around three key components: choosing the appropriate algorithmic template with the key motivations behind it, solving implementability issues that commonly arise in higher-order methods and finally designing a universal algorithm that can handle deterministic and noisy oracle feedback simultaneously without having prior knowledge. Our point of departure is the popular Extra-Gradient (EG) template; originally introduced by Korpelevich [27] and further developed in Nemirovski [37],

$$\begin{aligned} X_{t+1/2} &= \Pi_{\mathcal{X}}\left(X_t - \gamma_t \nabla f(x_t)\right) \\ X_{t+1} &= \Pi_{\mathcal{X}}\left(X_t - \gamma_t \nabla f(x_{t+1/2})\right), \end{aligned} \tag{EG}$$

where $\Pi_{\mathcal{X}}(x) = \arg\min_{z \in \mathcal{X}} \|x - z\|^2$ is the standard Euclidean projection onto the set $\mathcal{X}$. In terms of output, the candidate solution returned by (EG) after $T$ iterations is the so-called "ergodic average"

$$\bar{X}_T = \frac{\sum_{t=1}^{T} b_t X_{t+1/2}}{\sum_{t=1}^{T} b_t} \tag{4}$$

Then, taking $b_t = \gamma_t$ and assuming the method's step-size $\gamma_t$ is chosen appropriately, $\bar{X}_T$ enjoys the following universal guarantee [23, 47]:

$$\mathbb{E}[f(\overline{X}_t) - f(x^*)] = \mathcal{O}\left(\frac{1}{T} + \frac{\sigma}{\sqrt{T}}\right) \tag{5}$$

where $\sigma$ signifies the effect of the noisy feedback. However, as it becomes apparent, the vanilla (EG) template is not capable of matching the iconic $1/T^2$ for the smooth deterministic case. It is well-established in the literature of smooth, convex minimization that iterate averaging (or momentum in the sense of Nesterov [41]) is essential for matching the $O(1/T^2)$ lower bounds. In fact, plain uniform averaging is not sufficient; one needs to introduce new iterates with *increasing* weights. Precisely, this is equivalent to computing an average by taking $b_t = O(t)$. However, we cannot fully characterize the acceleration machinery without what we like to call "gradient weighting". On top of (weighted) iterate averaging, gradients must be multiplied by the *same order of weights* to achieve acceleration, which is a recurring theme in the literature of accelerated and universal optimization [50, 52, 29, 2, 31, 51, 12, 26, 22].

Going back to discussion on (EG), Wang and Abernethy [51] and Kavis et al. [26] provide useful insights into acceleration within the context of (EG). Wang and Abernethy [51] identifies a 2-player game with a particular structure called FENCHELGAME framework, which essentially reduces to minimizing a smooth, convex function when the players cooperate. By introducing an "optimistic" weighted iterate averaging along with a complementary gradient weighting strategy, the framework recovers different acceleration schemes of Nesterov [41, 42, 39]. On a related front, Diakonikolas and Orecchia [13] proposes the first acceleration of (EG) by appropriately integrating the optimistic averaging idea [51] into the (EG) template as follows:

$$\tilde{X}_t = \frac{b_t X_t + \sum_{s=1}^{t-1} b_s X_{s+1/2}}{\sum_{s=1}^{t} b_s}, \qquad \bar{X}_{t+1/2} = \frac{\sum_{s=1}^{t} b_s X_{s+1/2}}{\sum_{s=1}^{t} b_s} \tag{6}$$

where $b_t = O(t)$ is the "iterate averaging" parameter. Later on, Kavis et al. [26] designs an adaptive, universal variant of accelerated Mirror-Prox following the same optimistic averaging idea as in Eq. (6). All in all, it is a recurring theme among accelerated algorithms to adopt weighted iterate averaging ($b_t = O(t)$) with proportionate gradient weighting, and not so surprisingly, prior work establishes clear connections between the degree of weighting and convergence rate. Cutkosky [12] designs a black-box reduction that accelerated a class of online algorithms and proves that the rate of convergence of the reduction is $O(1/\sum_{t=1}^{T} b_t)$ for $b_t \in [1, t]$. In retrospect, we aim at answering the following question;

*What algorithmic construction would enable acceleration beyond $O(1/T^2)$?*

### 3.1 Implicit algorithm

We give a first affirmative answer to the above question by presenting our implicit accelerated algorithm which is constructed upon (EG), and establish its convergence properties. Note that the implicitness of the scheme serves as a gentle introduction to the actual explicit second order acceleration, which shall follow. Formally, our scheme is given via the following recursion:

$$\begin{aligned} X_{t+1/2} &= \Pi_{\mathcal{X}}\left(X_t - \gamma_t a_t \mathbf{F}(\bar{X}_{t+1/2}; \tilde{X}_t)\right) \\ X_{t+1} &= \Pi_{\mathcal{X}}\left(X_t - \gamma_t a_t \nabla f(\bar{X}_{t+1/2})\right) \end{aligned} \tag{Implicit Second-Order}$$

with $\tilde{X}_t$ and $\bar{X}_{t+1/2}$ being defined as in (6) and the adaptive step-size policy $\gamma_t$ defined as:

$$\gamma_t = \frac{\gamma}{\sqrt{1 + \sum_{s=1}^{t-1} a_s^2 \|\nabla f(\bar{X}_{s+1/2}) - \mathbf{F}(\bar{X}_{s+1/2}; \tilde{X}_s)\|^2}}. \tag{7}$$

The implicit nature of (Implicit Second-Order) originates from $X_{t+1/2}$ update (which we shall refer to as (corrected) extrapolation step at times) since $\bar{X}_{t+1/2}$ depends upon $X_{t+1/2}$ itself. However, this scheme exhibits several key differences from the vanilla (EG), which constitute the fundamental parts of our second-order acceleration machinery. In particular, we have:

(i) integration of second-order updates for sharper extrapolation steps - first step of acceleration.
(ii) interplay between averaging ($b_t$) and gradient weighting ($a_t$) which allows more aggressive averaging - second step of acceleration.
(iii) adaptive step-size in the sense of Rakhlin and Sridharan [48] - key to adaptivity and universality.

**Second-order updates:** First, we will consider the particular interpretation of (EG) as an approximation to the Proximal Point method [49] which serves as motivation for the accommodation of second-order information in our scheme.

$$X_{t+1} = X_t - \gamma_t \nabla f(X_{t+1}). \tag{PP}$$

In particular, (EG) tries to approximate $X_{t+1}$ by generating the extrapolated point $X_{t+1/2}$, and make use of the gradient at $X_{t+1/2}$ to take a step from $X_t$ to $X_{t+1}$. Therefore, if the algorithm is able to compute a sharper estimate in the extrapolation step, it should be able live up to the fame of (PP) and display faster convergence. To this end, we augment the extrapolation step by introducing second-order term. Essentially, our algorithm makes use of *second-order Taylor approximation*, as opposed to first-order expansion, only for the extrapolation step, trading-off sharper approximation with second-order information.

**Iterate averaging and gradient weighting:** Now, we turn our attention to the second component in our acceleration machinery; averaging and weighting. Recall that the acceleration framework of Cutkosky [12] guarantees a value convergence rate of $O(1/t^{p+1})$ when weighting factor satisfies $b_t = O(t^p)$ with $p \in [0, 1]$. We take this result one step beyond in two fronts; our algorithm exploits higher-order smoothness in order to extend this bound for $p \in [0, 2]$, implying the accelerated rate of $O(1/T^3)$. Second, we observe that previous work restricts the choice of gradient weights and averaging weights by taking $a_t \approx b_t$. We decouple those weights by allowing the sequences $a_t$ and $b_t$ to be *different*, which in turn equips us with more aggressive iterate averaging when necessary.

**Adaptive step-size:** As the final component, we study the adaptive step-size (7) from the parameter adaptation perspective (i.e., adaptation to the Lipschitz modulus) and expand on its universal properties in the next section. The vast literature on adaptive methods predominantly rely on constructions of AdaGrad-like decreasing step-size policies by accumulating the observed gradient norms in its denominator. The intuition behind this choice is that whenever the method approaches a solution, the vanishing gradients bring about stabilization, ensuring progress around the solution's neighborhood. However, this idea fails for (compactly) constrained problems; when the solution lies on the boundary. So inspired by [48], we overcome this issue by accumulating $\|\nabla f(\bar{X}_{t+1/2}) - \mathbf{F}(\bar{X}_{t+1/2}; \tilde{X}_t)\|^2$ which converges to 0 as $\bar{X}_{t+1/2} - \tilde{X}_t \to 0$; which in turn implies convergence of the algorithm. To our knowledge, this is the first adaptive step-size that accommodates second order information.

Having established the core components of our design, we are in position to present the first accelerated convergence rate guarantee for (Implicit Second-Order). Formally, this is given by the following.

**Theorem 3.1.** *Let $\{X_{t+1/2}\}_{t=1}^T$ be generated by* (Implicit Second-Order) *run with the adaptive step-size policy* (7) *where $a_t = t^2$, $b_t = t^3$. Assume that $f$ satisfies* (H-smooth) *and denote $\sigma = \max\{\sigma_g, \sigma_H\}$. Then, the following guarantee holds:*

$$f(\bar{X}_{T+1/2}) - f(x^*) \leq O\left(\frac{L\left(D^4/\gamma^2 + D\gamma\right)}{T^3}\right).$$

*When $\gamma = D$, we obtain the converge rate $O\left(\frac{LD^2}{T^3}\right)$.*

*Remark* 3.1. We emphasize that the above rate *does not* require any prior knowledge of problem paramaters such as $L$, $D$, time-horizon $T$ and any bounds on gradient/Hessian norms. In order to have better dependence on $D$ one could set $\gamma = D$, and our rate of $O(1/T^3)$ holds irrespective of $\gamma$.

### 3.2 Explicit algorithm

Despite the fact that (Implicit Second-Order) improves upon the accelerated rate of $O(1/T^2)$, one may easily observe that it exhibits the following drawbacks:

1. (Implicit Second-Order) is a conceptual algorithm and therefore, *not* implementable in practice.
2. A fortiori, it cannot provide rate interpolation guarantees as it does not have the machinery to simultaneously cope with deterministic and stochastic feedback.

As discussed earlier, a common strategy for overcoming this implicit construction is using a bisection/line-search procedure [21, 36, 7]. Depending on the context, this procedure serves two *distinct* purposes. Primarily, it tackles the implicit nature of the update rule by simultaneously finding a pair of $(\gamma_t, X_{t+1/2})$ and secondly, it enables adaptation to the second-order smoothness modulus. However, one may identify major setbacks with these approaches; first, it is not clear how to handle stochastic oracles for executing the search procedure, so it is not capable of satisfying any universal guarantees. Moreover, it yields a rather complicated procedure as a byproduct that has many moving parts. To that end, we propose an alternative approach which not only yields a simple scheme, but also provides a universal algorithm that is able to handle noisy feedback on-the-fly. Without further ado, we display our explicit algorithm, EXTRANEWTON, with appropriate modifications.

**Algorithm 1:** EXTRANEWTON

**Input:** $X_1 \in \mathcal{X}$, gradient weight $\{a_t\}_{t=1}^T$, averaging weight $\{b_t\}_{t=1}^T$, $\gamma > 0$

**for** $t = 1$ to $T$ **do**

$\quad X_{t+1/2} = \arg\min_{x\in\mathcal{X}} \langle a_t g(\tilde{X}_t), x\rangle + \frac{a_t B_t}{2b_t}\langle H(\tilde{X}_t)(x - X_t), x - X_t\rangle + \frac{1}{2\gamma_t}\|x - X_t\|^2$

$\quad X_{t+1} \quad = \arg\min_{x\in\mathcal{X}} \langle a_t g(\bar{X}_{t+1/2}), x\rangle + \frac{1}{2\gamma_t}\|x - X_t\|^2$

**end for**

Having established our main algorithmic template, Algorithm 1, we would like to provide a more detailed description of its core components.

**Universal step-size** We modify our step-size (see Eq. (8)) in order to operate in the stochastic regime while making it noise-adaptive for achieving rate interpolation. Using the same optimistic weighted averaging scheme in Eq. (6), we define the universal counterpart of the adaptive step-size,

$$\gamma_t = \frac{\gamma}{\sqrt{1 + \sum_{s=1}^{t-1} a_s^2 \|g(\bar{X}_{s+1/2}) - \tilde{\mathbf{F}}(\bar{X}_{s+1/2}; \tilde{X}_s)\|^2}}. \tag{8}$$

Note that $\gamma_t$ is independent of any variable/randomness generated at iteration $t$; it accumulates $a_t^2 \|g(\bar{X}_{s+1/2}) - \tilde{\mathbf{F}}(\bar{X}_{s+1/2}; \tilde{X}_s)\|^2$ up to $t-1$. Therefore, the step-size is decoupled from the explicit update, *a priori*. Then, what remains is a new algorithmic design that will retain the accelerated convergence properties demonstrated by (Implicit Second-Order) while having an explicit construction that is capable of automatically adjusting to noise level in the oracle feedback.

Before expanding upon the technical details of our proof strategy, let us take our time to explain the consequences of our explicit design in comparison to (Implicit Second-Order).

**From implicit to explicit** To obtain the explicit algorithm, (*i*) we write the projection sub-problem in the $\arg\min$ form; (*ii*) replace $X_{t+1/2}$ with the free variable $x$; (*iii*) introduce *stochastic* evaluations into the quadratic, second-order term; then, (*iv*) simplify:

$$\frac{a_t B_t}{2b_t} \langle H(\tilde{X}_t)(\frac{b_t x + \sum_{s=1}^{t-1} b_s X_{s+1/2}}{B_t} - \tilde{X}_t), \frac{b_t x + \sum_{s=1}^{t-1} b_s X_{s+1/2}}{B_t} - \tilde{X}_t \rangle$$
$$\Downarrow$$
$$\frac{a_t b_t}{2B_t} \langle H(\tilde{X}_t)(x - X_t), x - X_t \rangle$$

Given the bisection-type conceptual methods [36, 21, 7], it is surprising how smoothly we could transition from implicit to explicit *once* we decouple the step-size from the current iteration *apriori*. Moreover, the resulting update rule for the extrapolation step retains the quadratic structure as the $X_{t+1}$ update rule.

Having analyzed the components of the explicit scheme, we will first present the universal convergence rates then provide a concise explanation of the proof strategy with particular emphasis on the principal components of the analysis.

**Theorem 3.2.** *Let $\{X_{t+1/2}\}_{t=1}^T$ be generated by Algorithm 1, run with the adaptive step-size policy (8) and $a_t = t^2, b_t = t^3$. Assume that $f$ satisfies (H-smooth), and that Assumptions (2) hold with $\sigma = \max\{\sigma_g, \sigma_H\}$. Then, the following universal guarantee holds:*

$$\mathbb{E}\left[ f(\bar{X}_{T+1/2}) - f(x^*) \right] \leq O\left( \frac{\sigma}{\sqrt{T}} + \frac{L\left(D^4/\gamma^2 + D\gamma\right)}{T^3} \right).$$

*When $\gamma = D$, we obtain the target rate $O\left(\frac{\sigma}{\sqrt{T}} + \frac{LD^2}{T^3}\right)$.*

*Remark* 3.2. Similar to Theorem 3.1, EXTRANEWTON achieves the preceding convergence rate independent of the knowledge of problem parameters.

Compatible with the (EG)-based algorithmic design, our proof has the following main steps

1. We perform an *offline* regret analysis of Alg. 1 and show adaptive regret bounds - see Prop. 3.1.
2. We prove an anytime online-to-batch conversion framework, which generalizes that of Cutkosky [12], through decoupling iterate averaging from gradient weighting - see Theorem 3.3.
3. Combining the adaptive regret bound with the conversion theorem immediately implies *universal, accelerated* value convergence of $O(\frac{\sigma}{\sqrt{T}} + \frac{LD^2}{T^3})$ - see Theorem 3.2.

Let us begin with clarifying what *offline regret* means for Algorithm 1. We define the (linear) regret considering the convention in both online learning [48, 12] and first-order acceleration literature [51, 26, 22]. We measure the performance of our decisions for the extrapolation sequence such that after playing $X_{t+1/2}$, our algorithm observes and suffers the linear (weighted) loss with respect to $a_t \nabla f(\bar{X}_{t+1/2})$. Hence, we define the regret as

$$\textsc{Reg}_{\mathrm{T}}(x) = \sum_{t=1}^{T} a_t \langle \nabla f(\bar{X}_{t+1/2}), X_{t+1/2} - x \rangle \tag{Reg}$$

where we run the algorithm for $T$ rounds. Next up, we provide our generalized conversion result.

**Theorem 3.3.** *Let* $\textsc{Reg}_{\mathrm{T}}(x^*)$ *denote the anytime regret for the decision sequence* $\{X_{t+1/2}\}_{t=1}^{T}$ *as in* (Reg)*, and define two sequences of non-decreasing weights* $a_t$ *and* $b_t$ *such that* $a_t, b_t \geq 1$*. As long as* $a_t/b_t$ *is ensured to be non-increasing,*

$$f(\bar{X}_T) - f(x^*) \leq \frac{\textsc{Reg}_{\mathrm{T}}(x^*)}{a_T \frac{B_T}{b_T}}$$

*Remark* 3.3. This conversion result holds independent of the order of smoothness of the objective as long as $f$ is convex. Moreover, it allows averaging parameter $b_t$ to be asymptotically larger than gradient weights $a_t$, enabling a more aggressive averaging strategy when necessary.

To complement the lower bound to the regret $\textsc{Reg}_{\mathrm{T}}(x^*)$, we will provide a template upper bound that will help us explain how we exploit second-order smoothness for a more aggressive weighting, hence the rate $O(1/T^3)$.

**Proposition 3.1.** *Let* $\{X_{t+1/2}\}_{t=1}^{T}$ *be generated by Algorithm 1, run with a non-increasing step-size sequence* $\gamma_t$ *and non-decreasing sequences of weights* $a_t, b_t \geq 1$ *such that* $a_t/b_t$ *is also non-increasing. Then, the following guarantee holds:*

$$\mathbb{E}\left[\,\textsc{Reg}_{\mathrm{T}}(x^*)\,\right] \leq \frac{1}{2}\mathbb{E}\left[\frac{3D^2}{\gamma_{T+1}} + \sum_{t=1}^{T} \gamma_{t+1} a_t^2 \|g(\bar{X}_{t+1/2}) - \tilde{\mathbf{F}}(\bar{X}_{t+1/2}; \tilde{X}_t)\|^2 - \frac{\|X_{t+1/2} - X_t\|^2}{\gamma_{t+1}}\right]$$

It is important to note that the template inequality in Proposition 3.1 is agnostic to the design of our adaptive step-size as in Eq. (8) as well as the selection of the weights as we described in Theorem 3.2; it holds for Algorithm 1 in the most general parameter selection. It essentially applies to any non-increasing sequence of step-sizes and non-decreasing gradient weight sequence $a_t \geq 1$. To obtain it, we neither used convexity nor the smoothness of the objective $f$. In fact, the structure of the objective function, i.e., its convexity, will not be needed for upper-bounding the regret expression, and we use it only for the conversion in Theorem 3.3.

Now, let us explain how we make use of second-order smoothness for enjoying faster rates, and give a brief discussion of how the regret bound will look in its final form. First, we decompose the stochastic term $\|g(\bar{X}_{t+1/2}) - \tilde{\mathbf{F}}(\bar{X}_{t+1/2}; \tilde{X}_t)\|^2$ into deterministic feedback and noise. Then, we argue that *the noisy component* grows as $O(\sigma T^{5/2})$ where we define $\sigma = \max\{\sigma_g, \sigma_H\}$ for brevity. On the other hand, achieving the accelerated $O(1/T^3)$ component of the universal rate amounts to showing that the regret has a constant, $O(1)$, component. In the worst-case sense, however, *the deterministic component itself* grows as $O(T^{5/2})$, too. Fortunately, we identify that the negative term is "large enough" in magnitude to control the growth of the deterministic term, which permits a constant component of order $O(LD^2)$ for the regret bound.

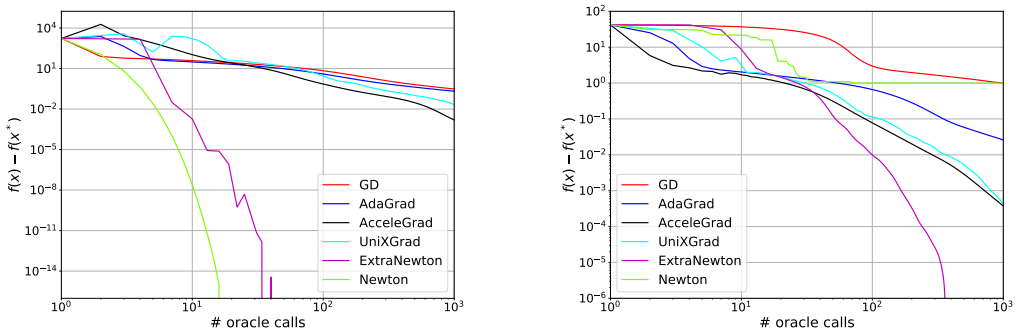
Although the regret bound of $O(LD^2 + \sigma T^{5/2})$ seems counter-intuitive from an online-learning perspective, it will make perfect sense when we discuss how second-order smoothness leads to "faster" conversion through more aggressive averaging. As a matter of fact, we will conclude our discussion with how second-order smoothness helps us accelerate. It turns out that using (H-smooth), iterate averaging as in Eq.(6) and compactness of $\mathcal{X}$, we can bound the negative term as,

$$-\frac{1}{\gamma_{t+1}}\|X_{t+1/2} - X_t\|^2 \leq -\frac{1}{L^2 D^2 \gamma_{t+1}} t^4 \|\nabla f(\bar{X}_{t+1/2}) - \mathbf{F}(\bar{X}_{t+1/2}; \tilde{X}_t)\|^2$$

Observe that to seamlessly combine the positive and negative terms, our analysis enforces that $a_t = O(t^2)$ and $b_t = \Omega(t^2)$. Then, the conversion implies a convergence rate of $\text{REG}_\text{T}(x^*)/T^3$, hence the recipe for acceleration. Therefore, the constant component of the regret amounts to $O(1/T^3)$ convergence rate, while the stochastic component of the regret implies $O(\sigma/\sqrt{T})$ rate, giving us the first universal acceleration beyond first-order smoothness.

# 4 Experiments

In this section, we will present practical performance of EXTRANEWTON against a set of algorithms, e.g., GD, SGD, ADAGRAD [14], ACCELEGRAD [31], UNIXGRAD [26], NEWTON'S, for least squares and logistic regression problems over a LIBSVM dataset, `a1a`. Our main objective is to convey two motivational ideas. First, when the objective has a favorable structure as in least squares, second-order method has cheap oracle costs (Hessian will be constant across the whole space) and display superior convergence behavior. Second, we want to demonstrate the improved rates of our algorithm against accelerated and non-accelerated first-order methods through the $\ell_2$-regularized logistic regression problem. In the plots, the x-axis denotes *# of oracle calls* which counts any gradient or Hessian computation as one oracle call.




**(a)** Least-squares regression on `a1a` **(b)** Logistic regression on `a1a`

**Figure 1:** Comparison of value convergence for regression problems with deterministic oracle access

When the problem is suitable, second-order methods show promising performance with truly superior run time. In Figure 1a, we display the result for least squares setting. Second-order methods are known to be suitable for quadratic problems, and our method exploits its hybrid construction to converge significantly faster than first-order methods, matching the behavior of NEWTON'S.

For the logistic regression problem, we regularize it with $g(x) = 1/2\|x\|^2$, but use a very small regularization constant to render the problem ill-conditioned, making things slightly more difficult for the algorithms [34, 35]. Although we implement NEWTON'S with line-search, we actually observed a sporadic convergence behavior; when the initial point is close to the solution it converges similarly to EXTRANEWTON, however when we initialize further away it doesn't converge. This non-convergent behavior has been known for NEWTON'S, even with line-search present [18]. On the contrary, EXTRANEWTON consistently converges; even if we perturb the initial step-size and make it adversarially large, it manages to recover due to its adaptive step-size.

# 5 Conclusion

In this work, we present the *first* universal, second-order algorithm, EXTRANEWTON, which enjoys the value convergence rate of $O(\sigma/\sqrt{T} + 1/T^3)$. To that end, an important open question is whether we could design a method that achieves an improved rate interpolation guarantee $O(\sigma/\sqrt{T} + 1/T^{7/2})$ without depending on any line-search/bisection mechanism. We defer this to a future work.

# References


[1] Naman Agarwal and Elad Hazan. Lower bounds for higher-order convex optimization. In Sébastien Bubeck, Vianney Perchet, and Philippe Rigollet, editors, *Proceedings of the 31st Conference On Learning Theory*, volume 75 of *Proceedings of Machine Learning Research*, pages 774–792. PMLR, 06–09 Jul 2018. URL https://proceedings.mlr.press/v75/agarwal18a.html.

[2] Zeyuan Allen-Zhu and Lorenzo Orecchia. Linear coupling: An ultimate unification of gradient and mirror descent, 2016.

[3] Kimon Antonakopoulos, Elena Veronica Belmega, and Panayotis Mertikopoulos. Adaptive extra-gradient methods for min-max optimization and games. *CoRR*, abs/2010.12100, 2020. URL https://arxiv.org/abs/2010.12100.

[4] Yossi Arjevani, Ohad Shamir, and Ron Shiff. Oracle complexity of second-order methods for smooth convex optimization. *Mathematical Programming*, pages 1–34, 2019.

[5] Stefania Bellavia, Gianmarco Gurioli, and Benedetta Morini. Adaptive cubic regularization methods with dynamic inexact Hessian information and applications to finite-sum minimization. *IMA Journal of Numerical Analysis*, 41(1):764–799, 04 2020. ISSN 0272-4979. doi: 10.1093/imanum/drz076. URL https://doi.org/10.1093/imanum/drz076.

[6] Albert A. Bennett. Newton's method in general analysis. *Proceedings of the National Academy of Sciences*, 2(10):592–598, 1916. doi: 10.1073/pnas.2.10.592. URL https://www.pnas.org/doi/abs/10.1073/pnas.2.10.592.

[7] Brian Bullins and Kevin A. Lai. Higher-order methods for convex-concave min-max optimization and monotone variational inequalities, 2020. URL https://arxiv.org/abs/2007.04528.

[8] Coralia Cartis, Nicholas I. Gould, and Philippe L. Toint. Adaptive cubic regularisation methods for unconstrained optimization. part i: Motivation, convergence and numerical results. *Math. Program.*, 127 (2):245–295, apr 2011. ISSN 0025-5610.

[9] Coralia Cartis, Nicholas I. M. Gould, and Philippe L. Toint. Adaptive cubic regularisation methods for unconstrained optimization. part ii: Worst-case function- and derivative-evaluation complexity. *Math. Program.*, 130(2):295–319, dec 2011. ISSN 0025-5610. doi: 10.1007/s10107-009-0337-y. URL https://doi.org/10.1007/s10107-009-0337-y.

[10] Xi Chen, Bo Jiang, Tianyi Lin, and Shuzhong Zhang. Accelerating adaptive cubic regularization of newton's method via random sampling. *Journal of Machine Learning Research*, 23(90):1–38, 2022. URL http://jmlr.org/papers/v23/20-910.html.

[11] Andrew R. Conn, Nicholas I. M. Gould, and Philippe L. Toint. *Trust Region Methods*. Society for Industrial and Applied Mathematics, 2000. doi: 10.1137/1.9780898719857. URL https://epubs.siam.org/doi/abs/10.1137/1.9780898719857.

[12] Ashok Cutkosky. Anytime online-to-batch conversions, optimism, and acceleration. *the International Conference on Machine Learning (ICML)*, June 2019.

[13] Jelena Diakonikolas and Lorenzo Orecchia. Accelerated extra-gradient descent: A novel accelerated first-order method. In *ITCS*, 2018.

[14] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

[15] Jean-Pierre Dussault and Dominique Orban. Scalable adaptive cubic regularization methods. 2021. doi: 10.13140/RG.2.2.18142.15680. URL http://rgdoi.net/10.13140/RG.2.2.18142.15680.

[16] Alina Ene, Huy L. Nguyen, and Adrian Vladu. Adaptive gradient methods for constrained convex optimization and variational inequalities, 2021.

[17] Alexander Gasnikov, Pavel Dvurechensky, Eduard Gorbunov, Evgeniya Vorontsova, Daniil Selikhanovych, and César A. Uribe. Optimal tensor methods in smooth convex and uniformly convexoptimization. In Alina Beygelzimer and Daniel Hsu, editors, *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99 of *Proceedings of Machine Learning Research*, pages 1374–1391. PMLR, 25–28 Jun 2019. URL https://proceedings.mlr.press/v99/gasnikov19a.html.

[18] Florian Jarre and Philippe L. Toint. Simple examples for the failure of newton's method with line search for strictly convex minimization. *Math. Program.*, 158(1–2):23–34, jul 2016. ISSN 0025-5610. doi: 10.1007/s10107-015-0913-2. URL https://doi.org/10.1007/s10107-015-0913-2.

[19] Bo Jiang, Tianyi Lin, and Shuzhong Zhang. A unified scheme to accelerate adaptive cubic regularization and gradient methods for convex optimization, 2017. URL https://arxiv.org/abs/1710.04788.

[20] Bo Jiang, Tianyi Lin, and Shuzhong Zhang. A unified adaptive tensor approximation scheme to accelerate composite convex optimization. *SIAM Journal on Optimization*, 30(4):2897–2926, 2020. doi: 10.1137/19M1286025. URL https://doi.org/10.1137/19M1286025.

[21] Ruichen Jiang and Aryan Mokhtari. Generalized optimistic methods for convex-concave saddle point problems, 2022. URL https://arxiv.org/abs/2202.09674.

[22] Pooria Joulani, Anant Raj, Andras Gyorgy, and Csaba Szepesvari. A simpler approach to accelerated optimization: iterative averaging meets optimism. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4984–4993. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/joulani20a.html.

[23] Anatoli Juditsky, Arkadi Semen Nemirovski, and Claire Tauvel. Solving variational inequalities with stochastic mirror-prox algorithm. *Stochastic Systems*, 1(1):17–58, 2011.

[24] L. V. Kantorovich. Functional analysis and applied mathematics. *Uspekhi Mat. Nauk*, 3:89–185, 1948.

[25] Sai Praneeth Karimireddy, Sebastian U. Stich, and Martin Jaggi. Global linear convergence of newton's method without strong-convexity or lipschitz gradients, 2018. URL https://arxiv.org/abs/1806.00413.

[26] Ali Kavis, Kfir Y. Levy, Francis Bach, and Volkan Cevher. Unixgrad: A universal, adaptive algorithm with optimal guarantees for constrained optimization. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 6260–6269. Curran Associates, Inc., 2019.

[27] G. M. Korpelevich. The extragradient method for finding saddle points and other problems. *Matecon*, 12: 747–756, 1976.

[28] Dmitry Kovalev, Konstantin Mishchenko, and Peter Richtárik. Stochastic newton and cubic newton methods with simple local linear-quadratic rates, 2019. URL https://arxiv.org/abs/1912.01597.

[29] Guanghui Lan. An optimal method for stochastic composite optimization. *Mathematical Programming*, 133(1-2):365–397, 2012.

[30] Kenneth Levenberg. A method for the solution of certain non – linear problems in least squares. *Quarterly of Applied Mathematics*, 2:164–168, 1944.

[31] Kfir Y Levy, Alp Yurtsever, and Volkan Cevher. Online adaptive methods, universality and acceleration. In *Neural and Information Processing Systems (NeurIPS)*, December 2018.

[32] Tianyi Lin and Michael. I. Jordan. Perseus: A simple high-order regularization method for variational inequalities, 2022. URL https://arxiv.org/abs/2205.03202.

[33] Donald W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):431–441, 1963. doi: 10.1137/0111030. URL https://doi.org/10.1137/0111030.

[34] Ulysse Marteau-Ferey, Francis R. Bach, and Alessandro Rudi. Globally convergent newton methods for ill-conditioned generalized self-concordant losses. In *NeurIPS*, 2019.

[35] Konstantin Mishchenko. Regularized newton method with global $o(1/k^2)$ convergence, 2021. URL https://arxiv.org/abs/2112.02089.

[36] Renato D. C. Monteiro and B. F. Svaiter. An accelerated hybrid proximal extragradient method for convex optimization and its implications to second-order methods. *SIAM Journal on Optimization*, 23(2): 1092–1125, 2013. doi: 10.1137/110833786. URL https://doi.org/10.1137/110833786.

[37] Arkadi Nemirovski. Prox-method with rate of convergence o(1/t) for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 15(1):229–251, 2004.

[38] Arkadii Nemirovskii, David Borisovich Yudin, and ER Dawson. Problem complexity and method efficiency in optimization. 1983.

[39] Yu Nesterov. Smooth minimization of non-smooth functions. *Math. Program.*, 103(1):127–152, may 2005. ISSN 0025-5610. doi: 10.1007/s10107-004-0552-5. URL https://doi.org/10.1007/s10107-004-0552-5.

[40] Yu. Nesterov. Accelerating the cubic regularization of newton's method on convex problems. *Mathematical Programming*, 2008.

[41] Yurii Nesterov. A method for solving the convex programming problem with convergence rate $o(1/k^2)$. *Dokl. Akad. Nauk SSSR*, 269:543–547, 1983.

[42] Yurii Nesterov. On an approach to the construction of optimal methods of minimization of smooth convex functions. *Ekonomika i Mateaticheskie Metody*, 24(3):509–517, 1988.

[43] Yurii Nesterov. Introductory lectures on convex optimization. 2004, 2003.

[44] Yurii Nesterov and Boris Polyak. Cubic regularization of newton method and its global performance. *Math. Program.*, 108:177–205, 08 2006. doi: 10.1007/s10107-006-0706-8.

[45] Yurii E. Nesterov. *Lectures on Convex Optimization*, volume 137 of *Springer Optimization and Its Applications*. Springer, 2018.

[46] Boris Polyak. Newton-kantorovich method and its global convergence. *Journal of Mathematical Sciences*, 133:1513–1523, 2006.

[47] Alexander Rakhlin and Karthik Sridharan. Optimization, learning, and games with predictable sequences. In *NIPS '13: Proceedings of the 27th International Conference on Neural Information Processing Systems*, 2013.

[48] Sasha Rakhlin and Karthik Sridharan. Optimization, learning, and games with predictable sequences. In *Advances in Neural Information Processing Systems*, pages 3066–3074, 2013.

[49] R. Tyrrell Rockafellar. Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization*, 14(5):877–898, 1976. doi: 10.1137/0314056. URL https://doi.org/10.1137/0314056.

[50] P Tseng. On accelerated proximal gradient methods for convex-concave optimization. *submitted to SIAM Journal on Optimization*, 01 2008.

[51] Jun-Kun Wang and Jacob D Abernethy. Acceleration through optimistic no-regret dynamics. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 3824–3834. Curran Associates, Inc., 2018.

[52] Lin Xiao. Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, 11(Oct):2543–2596, 2010.

[53] NESTEROV Yurii. Implementable tensor methods in unconstrained convex optimization. LIDAM Discussion Papers CORE 2018005, Université catholique de Louvain, Center for Operations Research and Econometrics (CORE), March 2018. URL https://ideas.repec.org/p/cor/louvco/2018005.html.

# Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 5
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Section 2 and Theorem 3.1, 3.3, 3.2
   (b) Did you include complete proofs of all theoretical results? [Yes] See the Appendix
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] We provide mean and standard deviation on the curves over multiple runs
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]