Proceedings of Machine Learning Research vol 313:1–36, 2026 37th International Conference on Algorithmic Learning Theory

Differentially Private Bilevel Optimization

Guy Kornowski GUY.KORNOWSKI@WEIZMANN.AC.IL
Weizmann Institute of Science

Editors: Matus Telgarsky and Jonathan Ullman

Abstract

We present differentially private (DP) algorithms for bilevel optimization, a problem class that received significant attention lately in various machine learning applications. These are the first algorithms for such problems under standard DP constraints, and are also the first to avoid Hessian computations which are prohibitive in large-scale settings. Under the well-studied setting in which the upper-level is not necessarily convex and the lower-level problem is strongly-convex, our proposed gradient-based (ϵ, δ)-DP algorithm returns a point with hypergradient norm at most $\widetilde{\mathcal{O}}\left((\sqrt{d_{\rm up}}/\epsilon n)^{1/2} + (\sqrt{d_{\rm low}}/\epsilon n)^{1/3}\right)$ where n is the dataset size, and $d_{\rm up}/d_{\rm low}$ are the upper/lower level dimensions. Our analysis covers constrained and unconstrained problems alike, accounts for mini-batch gradients, and applies to both empirical and population losses. As an application, we specialize our analysis to derive a simple private rule for tuning a regularization hyperparameter.

Keywords: Differential privacy, bilevel optimization, first-order.

1. Introduction

Bilevel optimization (BO) is a fundamental framework for solving optimization objectives of hierarchical structure, in which constraints are defined themselves by an auxiliary optimization problem. Formally, it is defined as

$$\begin{aligned} \text{minimize}_{\mathbf{x}\in\mathcal{X}} \;\; & F(\mathbf{x}) := f(\mathbf{x}, \mathbf{y}^*(\mathbf{x})) \\ \text{subject to} \;\; & \mathbf{y}^*(\mathbf{x}) \in \arg\min_{\mathbf{y}} g(\mathbf{x}, \mathbf{y}) \;, \end{aligned} \tag{BO}$$

where $F : \mathbb{R}^{d_x} \to \mathbb{R}$ is referred to as the hyperobjective, $f : \mathbb{R}^{d_x} \times \mathbb{R}^{d_y} \to \mathbb{R}$ as the upper-level (or outer) objective, and $g : \mathbb{R}^{d_x} \times \mathbb{R}^{d_y} \to \mathbb{R}$ as the lower-level (or inner) objective. While BO is well studied for over half a century (Bracken and McGill, 1973), it has recently received significant attention due to its diverse applications in machine learning (ML). These include hyperparameter tuning (Bengio, 2000; Maclaurin et al., 2015; Franceschi et al., 2017, 2018; Lorraine et al., 2020; Engstrom et al., 2025), meta-learning (Andrychowicz et al., 2016; Bertinetto et al., 2018; Rajeswaran et al., 2019; Ji et al., 2020), neural architecture search (Liu et al., 2018), invariant learning (Arjovsky et al., 2019; Jiang and Veitch, 2022), and data reweighting (Grangier et al., 2023; Fan et al., 2024; Pan et al., 2025). In these applications, both the upper and lower level objectives typically represent some loss over data, and are given by empirical risk minimization (ERM) problems over a dataset $S = \{\xi_1, \dots, \xi_n\} \in \Xi^n$:[1]

$$f(\mathbf{x}, \mathbf{y}) := f_S(\mathbf{x}, \mathbf{y}) = \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{x}, \mathbf{y}, \xi_i) \;, \qquad g(\mathbf{x}, \mathbf{y}) := g_S(\mathbf{x}, \mathbf{y}) = \frac{1}{n}\sum_{i=1}^{n} g(\mathbf{x}, \mathbf{y}, \xi_i) \;, \tag{ERM}$$

1. It is possible for the datasets with respect to f and g to be distinct (e.g., validation/training), perhaps of different sizes. In this case, S is the entire dataset, and n is the total number of samples. Letting $f(\cdot; \xi_i) = 0$ or $g(\cdot; \xi_i) = 0$ for certain indices in order to exclude corresponding data points from either objective, will not affect our results.

where both objectives are often proxies of stochastic (i.e., population) losses with respect to a distribution $\mathcal{P}$ over Ξ :

$$f(\mathbf{x},\mathbf{y}) := f_{\mathcal{P}}(\mathbf{x},\mathbf{y}) = \mathbb{E}_{\xi\sim\mathcal{P}}[f(\mathbf{x},\mathbf{y};\xi)]~, \qquad g(\mathbf{x},\mathbf{y}) := g_{\mathcal{P}}(\mathbf{x},\mathbf{y}) = \mathbb{E}_{\xi\sim\mathcal{P}}[g(\mathbf{x},\mathbf{y};\xi)]~. \quad \text{(Pop)}$$

In this work, we study bilvel optimization under differential privacy (DP) (Dwork et al., 2006). As ML models are deployed in an ever-growing number of applications, protecting the privacy of the data on which they are trained is a major concern, and DP has become the gold-standard for privacy preserving ML (Abadi et al., 2016). Accordingly, DP optimization is extensively studied, with a vast literature focusing both on empirical and stochastic objectives under various settings (Chaudhuri et al., 2011; Kifer et al., 2012; Bassily et al., 2014; Wang et al., 2017; Bassily et al., 2019; Wang et al., 2019; Feldman et al., 2020; Tran and Cutkosky, 2022; Gopi et al., 2022; Arora et al., 2023; Carmon et al., 2023; Ganesh et al., 2024; Lowy et al., 2024).

Nonetheless, private algorithms for BO remain largely underexplored, and in particular, no gradient-based algorithm (aka first-order, which uses only gradient queries) that solves BO problems under DP, is known to date. This is no coincidence: until recently, no first-order methods with finite time guarantees were known even for non-private bilevel problems. This follows the fact (Ghadimi and Wang, 2018, Lemma 2.1) that under mild regularity assumptions, the so-called hypergradient takes the form:

$$\nabla F(\mathbf{x}) = \nabla_x f(\mathbf{x},\mathbf{y}^*(\mathbf{x})) - \nabla^2_{xy} g(\mathbf{x},\mathbf{y}^*(\mathbf{x}))[\nabla^2_{yy} g(\mathbf{x},\mathbf{y}^*(\mathbf{x}))]^{-1}\nabla_y f(\mathbf{x},\mathbf{y}^*(\mathbf{x}))~. \tag{1}$$

Consequently, directly applying a "gradient" method to F requires inverting Hessians of the lower-level problem at each time step, thus limiting applicability in contemporary high-dimensional applications. Following various approaches to tackle this challenge (see Section 1.2), recent breakthroughs were finally able to provide fully first-order methods for BO with non-asymptotic guarantees (Liu et al., 2022; Kwon et al., 2023; Yang et al., 2023; Chen et al., 2024). These recent algorithmic advancements show promising empirical results in large scale applications, even up to the LLM scale of $\sim 10^9$ parameters (Pan et al., 2025).

On the downside, as we will show, first-order bilevel techniques can lead to major privacy violations. Intuitively, this is due to possibly private information leaking between the inner and outer objectives, since $\nabla F(\mathbf{x})$ itself depends on $\mathbf{y}^*(\mathbf{x})$ (as seen in (1)). In fact, we show in Section 3 that even a single hypergradient computation can completely break privacy, even when the upper-level objective does not depend on the data at all.

The only prior method we are aware of for DP BO was proposed by Chen and Wang (2024), which falls short in two main aspects. First, their algorithm only provides *some* privacy guarantee which cannot be controlled by the user. Moreover, it requires evaluating and inverting Hessians at each step, namely it is not first-order, which limits scalability; see Section 1.2 for further discussion.

1.1. Our contributions

We present DP algorithms that solve BO problems whenever the the upper-level is smooth but not necessarily convex, and the lower-level problem is smooth and strongly-convex. These are the first bilevel algorithms of any sort under standard central DP constraints, and in particular, are the first private algorithms to do so using only first-order (i.e., gradient) queries of the upper- and lower-level objectives. Our contributions can be summarized as follows:

- **DP Bilevel ERM Algorithm (Theorem 4):** We present a (ϵ, δ)-DP first-order algorithm for the bilevel ERM problem (BO/ERM) that outputs with high probability a point with hypergradient norm bounded by

$$\|\nabla F_S(\mathbf{x})\| = \widetilde{\mathcal{O}}\left(\left(\frac{\sqrt{d_x}}{\epsilon n}\right)^{1/2} + \left(\frac{\sqrt{d_y}}{\epsilon n}\right)^{1/3}\right).$$

  Moreover, our algorithm is also applicable to the case where $\mathcal{X} \subsetneq \mathbb{R}^{d_x}$ is a non-trivial constraint set, which is common in applications of BO.[2] In the constrained setting, we obtain the same guarantee as above in terms of the projected hypergradient (see Section 2 for details).

- **Mini-batch DP Bilevel ERM Algorithm (Theorem 11):** Aiming for a more practical algorithm, we design a variant of our previous algorithm that relies on mini-batch gradients. For the bilevel ERM problem (BO/ERM), given any batch sizes $b_{\text{in}}, b_{\text{out}} \in \{1, \ldots, n\}$ for sampling gradients of the inner/outer problems respectively, our algorithm ensures (ϵ, δ)-DP and outputs with high probability a point with hypergradient norm bounded by

$$\|\nabla F_S(\mathbf{x})\| = \widetilde{\mathcal{O}}\left(\left(\frac{\sqrt{d_x}}{\epsilon n}\right)^{1/2} + \left(\frac{\sqrt{d_y}}{\epsilon n}\right)^{1/3} + \frac{1}{b_{\text{out}}}\right). \tag{2}$$

  Notably, (2) is independent of the inner-batch size, yet depends on the outer-batch size, which coincides with known results for "single"-level constrained nonconvex optimization (Ghadimi et al., 2016) (see Remark 12 for further discussion). Our mini-batch algorithm is also applicable in the constrained setting $\mathcal{X} \subsetneq \mathbb{R}^{d_x}$ with the same guarantee in terms of projected hypergradient.

- **Population loss guarantees (Theorem 13):** We further provide guarantees for stochastic objectives. In particular, we show that for the population bilevel problem (BO/Pop), our (ϵ, δ)-DP algorithm outputs with high probability a point with hypergradient norm bounded by

$$\|\nabla F_{\mathcal{P}}(\mathbf{x})\| = \widetilde{\mathcal{O}}\left(\left(\frac{\sqrt{d_x}}{\epsilon n}\right)^{1/2} + \left(\frac{\sqrt{d_y}}{\epsilon n}\right)^{1/3} + \left(\frac{d_x}{n}\right)^{1/2} + \frac{1}{n^{1/4}}\right),$$

  with an additional additive $1/b_{\text{out}}$ factor in the mini-batch setting.

- **Application to private hyperparameter tuning (Section 7):** We specialize our algorithmic framework to the classic problem of tuning a regularization hyperparameter, when fitting a regularized statistical model. Our analysis of this problem leads to a simple, privacy preserving update rule for tuning the amount of regularization. As far as we know, this is the first such method which avoids selecting a hyperparameter from a predetermined set of candidates (as studied by previous works), but rather updates the hyperparameter privately on the fly.

1.2. Related work

BO was introduced by Bracken and McGill (1973), and grew into a vast body of work, with classical results focusing on asymptotic guarantees for certain specific problem structures (Anandalingam and White, 1990; Ishizuka and Aiyoshi, 1992; White and Anandalingam, 1993; Vicente et al., 1994; Zhu, 1995; Ye and Zhu, 1997). There exist multiple surveys and books covering various approaches

2. For instance, in data reweighting $\mathcal{X}$ is the probability simplex, and in hyperparameter tuning it is the hyperparameter space, which is typically constrained (e.g., non-negative reals).

for these problems (Vicente and Calamai, 1994; Dempe, 2002; Colson et al., 2007; Bard, 2013; Sinha et al., 2017).

Ghadimi and Wang (2018) observed (1) under strong-convexity of the inner problem using the implicit function theorem, asserting that the hypergradient can be computed via inverse Hessians, which requires solving a linear system at each point. Many follow up works built upon this second-order approach with additional techniques such as variance reduction, momentum, Hessian sketches, projection-free updates, or incorporating external constraints (Amini and Yousefian, 2019; Yang et al., 2021; Khanduri et al., 2021; Guo et al., 2021; Ji et al., 2021; Chen et al., 2021; Akhtar et al., 2022; Chen et al., 2022; Tsaknakis et al., 2022; Hong et al., 2023; Jiang et al., 2023; Abolfazli et al., 2023; Merchav and Sabach, 2023; Xu and Zhu, 2023; Cao et al., 2024; Dagréou et al., 2024). Only recently, the groundbreaking result of Liu et al. (2022) proved finite-time convergence guarantees for a fully first-order method which is based on a penalty approach. This result was soon extended to stochastic objectives (Kwon et al., 2023), with the convergence rate later improved by (Yang et al., 2023; Chen et al., 2024, 2025), and also extended to constrained bilevel problems (Yao et al., 2024; Kornowski et al., 2024). The first-order penalty paradigm also shows promise for some bilevel problems in which the inner problem is not strongly-convex (Shen and Chen, 2023; Kwon et al., 2024; Lu and Mei, 2024), which is generally a highly challenging setting (Chen et al., 2024; Bolte et al., 2025). Moreover, Pan et al. (2025) provided an efficient implementation of this paradigm, showing its effectiveness for large scale applications.

As to DP optimization, there is an extensive literature both for ERM and stochastic losses, for either convex objectives (Chaudhuri et al., 2011; Kifer et al., 2012; Bassily et al., 2014; Wang et al., 2017; Bassily et al., 2019; Feldman et al., 2020; Gopi et al., 2022; Carmon et al., 2023) as well as for nonconvex objectives (Wang et al., 2019; Tran and Cutkosky, 2022; Arora et al., 2023; Ganesh et al., 2024; Lowy et al., 2024; Zhang et al., 2024; Kornowski et al., 2025).

To the best of our knowledge, the only existing result for DP BO prior to our work is the result of Chen and Wang (2024), which differs than ours in several aspect. Their proposed algorithm is second-order, requiring evaluating Hessians and inverting them at each time step, which we avoid altogether. Moreover, Chen and Wang (2024) study the *local* DP model (Kasiviswanathan et al., 2011), in which each user (i.e., ξ_i) does not reveal its individual information. Due to this more challenging setting, they only derive guarantees for *some finite* privacy budget $\epsilon < \infty$, even as the dataset size grows. We study the common central DP model, in which a trusted curator acts on the collected data and releases a private solution, and we are able to provide any desired privacy and accuracy guarantees with sufficiently many samples. Our work is the first to study BO in this common DP setting. We also note that Fioretto et al. (2021) studied the related problem of DP in Stackelberg games, which are certain bilevel programs which arise in game theory, aiming at designing coordination mechanisms that maintain each agent's privacy.

After the initial version of this work became available on arXiv (Kornowski, 2024), Lowy and Liu (2025) further studied DP BO, proposing *second*-order algorithms which at the computational cost of inverting Hessians, are able to achieve gradient bounds which, interestingly, remove the dependence on the inner dimension d_y presented in this work.

2. Preliminaries

Notation and terminology. We let bold-face letters (e.g., $\mathbf{x}$) denote vectors, and denote by $\mathbf{0}$ the zero vector (whenever the dimension is clear from context) and by $I_d \in \mathbb{R}^{d\times d}$ the identity matrix.

$[n] := \{1, 2, \ldots, n\}$, $\langle \cdot, \cdot \rangle$ denotes the standard Euclidean dot product, and $\|\cdot\|$ denotes either its induced norm for vectors or operator norm for matrices, and $\|f\|_\infty = \sup_{\mathbf{x}\in\mathcal{X}} |f(\mathbf{x})|$ denotes the sup-norm. We denote by $\mathrm{Proj}_{\mathbb{B}(\mathbf{z},R)}$ the projection onto the closed ball around $\mathbf{z}$ of radius R. $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ denotes a normal (i.e., Gaussian) random variable with mean $\boldsymbol{\mu}$ and covariance Σ. We use standard big-O notation, with $\mathcal{O}(\cdot)$ hiding absolute constants (independent of problem parameters), and $\widetilde{\mathcal{O}}(\cdot)$ also hiding poly-logarithmic terms. We denote $f \lesssim g$ if $f = \mathcal{O}(g)$, and $f \asymp g$ if $f \lesssim g$ and $g \lesssim f$. A function $f : \mathcal{X} \subseteq \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$ is L_0-Lipschitz if for all $\mathbf{x}, \mathbf{y} \in \mathcal{X} : \|f(\mathbf{x}) - f(\mathbf{y})\| \leq L_0 \|\mathbf{x} - \mathbf{y}\|$; L_1-smooth if ∇f exists and is L_1-Lipschitz; and L_2 Hessian-smooth if $\nabla^2 f$ exists and is L_2-Lipschitz (with respect to the operator norm). A twice-differentiable function f is μ-strongly-convex if $\nabla^2 f \succeq \mu I$, denoting by "$\succeq$" the standard PSD ("Loewner") order on matrices.

Differential privacy. Two datasets $S, S' \in \Xi^n$ are said to be neighboring if they differ by only one data point. A randomized algorithm $\mathcal{A} : \Xi^n \to \mathcal{R}$ is called (ϵ, δ) differentially private (or (ϵ, δ)-DP) for $\epsilon, \delta > 0$ if for any two neighboring datasets $S \sim S'$ and measurable $E \subseteq \mathcal{R}$ in the algorithm's range: $\Pr[\mathcal{A}(S) \in E] \leq e^\epsilon \Pr[\mathcal{A}(S') \in E] + \delta$ (Dwork et al., 2006). In Appendix A we recall some well known DP basics used in our analyses: composition and advanced composition, the Gaussian mechanism, and privacy amplification by subsampling.

Gradient mapping. Given a point $\mathbf{x} \in \mathbb{R}^d$, and some $\mathbf{v} \in \mathbb{R}^d$, $\eta > 0$, we denote $\mathcal{G}_{\mathbf{v},\eta}(\mathbf{x}) := \frac{1}{\eta}(\mathbf{x} - \mathcal{P}_{\mathbf{v},\eta}(\mathbf{x}))$, $\mathcal{P}_{\mathbf{v},\eta}(\mathbf{x}) := \arg\min_{\mathbf{u}\in\mathcal{X}} \left[\langle \mathbf{v}, \mathbf{u}\rangle + \frac{1}{2\eta}\|\mathbf{u} - \mathbf{x}\|^2\right]$. In particular, given an L-smooth function $F : \mathbb{R}^d \to \mathbb{R}$ and $\eta \leq \frac{1}{2L}$, we denote the projected gradient (also known as reduced gradient) and the gradient (or proximal) mapping, respectively, as

$$\mathcal{G}_{F,\eta}(\mathbf{x}) := \frac{1}{\eta}(\mathbf{x} - \mathcal{P}_{\nabla F,\eta}(\mathbf{x})) , \quad \mathcal{P}_{\nabla F,\eta}(\mathbf{x}) := \arg\min_{\mathbf{u}\in\mathcal{X}} \left[\langle \nabla F(\mathbf{x}), \mathbf{u}\rangle + \frac{1}{2\eta}\|\mathbf{u} - \mathbf{x}\|^2\right].$$

The projected gradient $\mathcal{G}_{F,\eta}(\mathbf{x})$ generalizes the gradient to the possibly constrained setting: for points $\mathbf{x} \in \mathcal{X}$ sufficiently far from the boundary of $\mathcal{X}$, $\mathcal{G}_{F,\eta}(\mathbf{x}) = \nabla F(\mathbf{x})$, namely it simply reduces to the gradient; see the textbooks (Nesterov, 2013; Lan, 2020) for additional details.

2.1. Setting

We impose the following assumptions which are standard in the BO literature.

Assumption 1 *For (BO) with either (ERM) or (Pop), we assume the following hold:*

i. $\mathcal{X} \subseteq \mathbb{R}^{d_x}$ *is a closed convex set;*

ii. $F(\mathbf{x}_0) - \inf_{\mathbf{x}\in\mathcal{X}} F(\mathbf{x}) \leq \Delta_F$ *for some initial point* $\mathbf{x}_0 \in \mathcal{X}$*;*

iii. f *is twice differentiable, and* L_1^f*-smooth;*

iv. For all $\xi \in \Xi :$ $f(\cdot, \cdot; \xi)$ *is* L_0^f*-Lipschitz (hence, so is* f*);*

v. g *is* L_2^g*-Hessian-smooth, and for all* $\mathbf{x} \in \mathcal{X} :$ $g(\mathbf{x}, \cdot)$ *is* μ_g*-strongly-convex;*

vi. For all $\xi \in \Xi :$ $g(\cdot, \cdot; \xi)$ *is* L_1^g*-smooth (hence, so is* g*).*

As mentioned, these assumptions are standard in the study of BO problems and are shared by nearly all of the previously discussed works. In particular, the strong convexity of $g(\mathbf{x},\cdot)$ ensures that $\mathbf{y}^*(\mathbf{x})$ is uniquely defined, which is generally required in establishing the regularity of the hyperobjective. Indeed, it is known that dropping this assumption, can, in general, lead to pathological behaviors not amenable for algorithmic guarantees (cf. Chen et al. 2024; Bolte et al. 2025 and discussions therein). For the purpose of differential privacy though, the strong convexity of $g(\mathbf{x},\cdot)$ raises a subtle issue. As the standard assumption in the DP optimization literature is that the component functions are Lipschitz, which allows privatizing gradients using sensitivity arguments, strongly-convex objectives cannot be Lipschitz over the entire Euclidean space.[3] Therefore, strongly-convex objectives are regularly analyzed in the DP setting under the additional assumption that the domain of interest is bounded. For bilevel problems, the domain of interest for $\mathbf{y}$ is the lower-level solution set, thus we impose the following assumption.

Assumption 2 *There exists a compact set $\mathcal{Y}\subset\mathbb{R}^{d_y}$ with $\{\mathbf{y}^*(\mathbf{x}):\mathbf{x}\in\mathcal{X}\}\subseteq\mathcal{Y}$, such that for all $\mathbf{x}\in\mathcal{X},\ \xi\in\Xi:\ g(\mathbf{x},\cdot;\xi)$ is L_0^g-Lipschitz over $\mathcal{Y}$.*

Remark 3 *Note that $\mathrm{diam}(\mathcal{Y})\leq 2L_0^g/\mu_g$: Indeed, fixing some $\mathbf{x}\in\mathcal{X}$, $g(\mathbf{x},\cdot;\xi)$ is L_0^g-Lipschitz over $\mathcal{Y}$ for all $\xi\in\Xi$, thus so is $g(\mathbf{x},\cdot)$. By μ_g-strong-convexity, we get for all $\mathbf{y}\in\mathcal{Y}:\ \mu_g\left\|\mathbf{y}-\mathbf{y}^*(\mathbf{x})\right\| \leq\|\nabla_y g(\mathbf{x},\mathbf{y})\|\leq L_0^g$. Hence $\mathcal{Y}\subseteq\mathbb{B}(\mathbf{y}^*(\mathbf{x}),L_0^g/\mu_g)$, which is of diameter $2L_0^g/\mu_g$.*

Following Assumptions 1 and 2, we denote $\ell:=\max\{L_0^f,L_1^f,L_0^g,L_1^g,L_2^g\},\ \kappa:=\ell/\mu_g$.

3. Warm up: Privacy can leak from lower to upper level

In this section, we provide an illuminating example of privacy breaking in first-order BO, which motivates the algorithmic approach presented later in the paper. In particular, we exemplify that even when the upper-level problem supposedly does not depend on the dataset at all, the bilevel structure can give rise to *hyper*gradients that are non-private.

Example 1 *Given any dataset $S=\{\xi_1,\ldots,\xi_n\}\subset\mathbb{R}^d$, consider the BO problem corresponding to the objectives over $\mathbb{R}^{2d}$:*

$$f(\mathbf{x},\mathbf{y})=\frac{1}{2}\left\|\mathbf{x}+\mathbf{y}\right\|^2~,\qquad g(\mathbf{x},\mathbf{y})=\frac{1}{2}\sum_{i=1}^n\left\|\mathbf{y}-\xi_i\right\|^2~.$$

These are smooth non-negative objectives, and g is strongly-convex with respect to $\mathbf{y}$, and so we see that Assumptions 1 and 2 hold. Further note that the upper-level objective f does not depend on the data S at all. Nonetheless, we will show that a single hypergradient computation can break privacy.

To see this, note that for any $\mathbf{x}\in\mathbb{R}^d$: $\mathbf{y}^*(\mathbf{x})=\arg\min_{\mathbf{y}}g(\mathbf{x},\mathbf{y})=\frac{1}{n}\sum_{i=1}^n\xi_i$ since the squared loss is minimized by the average. Furthermore, $\nabla_x f(\mathbf{x},\mathbf{y})=\mathbf{x}+\mathbf{y}$ and $\nabla^2_{xy}g\equiv\mathbf{0}$, so by (1) we get that $\nabla F(\mathbf{x})=\mathbf{x}+\mathbf{y}^*(\mathbf{x})=\mathbf{x}+\frac{1}{n}\sum_{i=1}^n\xi_i$. In particular, computing the hypergradient at the origin simply returns the dataset's mean $\nabla F(\mathbf{0})=\frac{1}{n}\sum_{i=1}^n\xi_i$, which is well-known to be a non-private statistic of the dataset S (cf. Kamath et al., 2019).

3. If $g(\mathbf{x},\cdot;\xi)$ were Lipschitz over $\mathbb{R}^{d_y}$ for all $\xi\in\Xi$, then so would $g(\mathbf{x},\cdot)$, contradicting strong convexity.

The algorithms to follow incorporate a fix to overcome such privacy leaks: instead of ever computing $\mathbf{y}^*(\mathbf{x})$ (as non-private algorithms do up to exponentially small precision), we estimate it using an auxiliary private method, and use the resulting private point $\widetilde{\mathbf{y}}^*(\mathbf{x})$ to estimate the hypergradient at $\mathbf{x}$.[4] This leads to non-negligible bias in the hypergradient estimators compared to non-private first-order BO, that we need to account for in our analysis. We further discuss this in Section 4.1.

4. Algorithm for DP bilevel ERM

In this section, we consider the bilevel ERM problem (BO/ERM), for which we denote the hyperobjective by F_S. Our algorithm is presented in Algorithm 1. We prove the following result:

Theorem 4 *Assume 1 and 2 hold, and that* $\alpha \leq \ell\kappa^3 \min\{\frac{1}{2\kappa}, \frac{L_0^g}{L_0^f}, \frac{L_1^g}{L_1^f}, \frac{\Delta_F}{\ell\kappa}\}$. *Then there is a parameter assignment* $\lambda \asymp \ell\kappa^3\alpha^{-1}$, $\sigma^2 \asymp \ell^2\kappa^2 T\log(T/\delta)\epsilon^{-2}n^{-2}$, $\eta \asymp \ell^{-1}\kappa^{-3}$, $T \asymp \Delta_F\ell\kappa^3\alpha^{-2}$, *such that running Algorithm 1 satisfies* (ϵ,δ)*-DP, and returns* $\mathbf{x}_{\text{out}}$ *such that with probability at least* $1-\gamma$:

$$\|\mathcal{G}_{F_S,\eta}(\mathbf{x}_{\text{out}})\| \leq \alpha = \widetilde{\mathcal{O}}\left(K_1\left(\frac{\sqrt{d_x}}{\epsilon n}\right)^{1/2} + K_2\left(\frac{\sqrt{d_y}}{\epsilon n}\right)^{1/3}\right),$$

where $K_1 = \mathcal{O}(\Delta_F^{1/4}\ell^{3/4}\kappa^{5/4})$, $K_2 = \mathcal{O}(\Delta_F^{1/6}\ell^{5/6}\kappa^{11/6})$.

Remark 5 *Recall that when* $\mathcal{X} = \mathbb{R}^{d_x}$, *then* $\mathcal{G}_{F_S,\eta}(\mathbf{x}_{\text{out}}) = \nabla F_S(\mathbf{x}_{\text{out}})$.

Remark 6 *Algorithm 1 generalizes previously studied algorithms in two special cases. First, in the non-private case (when* $\epsilon = \infty$ *or* $\delta = 1$*) the algorithm can skip the private inner loop and simply solve the inner problems instead in* $\widetilde{\mathcal{O}}(1)$ *steps, thus reducing to the first-order non-private bilevel algorithms analyzed by (Kwon et al., 2023; Chen et al., 2024, 2025). Second, if the lower-level objective is independent of* $\mathbf{x}$, *the partial derivative* $\nabla_x g \equiv \mathbf{0}$ *vanishes and* $\widetilde{\mathbf{y}}_t \approx \arg\min_{\mathbf{y}} g(\mathbf{x}_t,\mathbf{y})$ *remains constant independently of* t. *In this case, Algorithm 1 simply reduces to DP-GD with respect to the first variable of* f, *and indeed, the* $\mathcal{O}((\sqrt{d_x}/\epsilon n)^{1/2})$ *gradient bound we obtain matches the known result of DP-GD for smooth-nonconvex "single-level" optimization (Wang et al., 2017).*

4.1. Analysis overview

In this section, we will go over the main ideas that appear in the proof of Theorem 4, all of which are provided in full detail in Appendix B. We start by introducing some useful notation: Given $\lambda > 0$, we denote the penalty function

$$\mathcal{L}_\lambda(\mathbf{x},\mathbf{y}) := f(\mathbf{x},\mathbf{y}) + \lambda\left[g(\mathbf{x},\mathbf{y}) - g(\mathbf{x},\mathbf{y}^*(\mathbf{x}))\right],$$
$$\text{and} \quad \mathcal{L}_\lambda^*(\mathbf{x}) := \mathcal{L}_\lambda(\mathbf{x},\mathbf{y}^\lambda(\mathbf{x})), \quad \mathbf{y}^\lambda(\mathbf{x}) := \arg\min_{\mathbf{y}} \mathcal{L}_\lambda(\mathbf{x},\mathbf{y}).$$

The starting point of our analysis is the following result:

Lemma 7 (Kwon et al. 2023; Chen et al. 2024, 2025) *If* $\lambda \geq 2L_1^f/\mu_g$, *then: (i)* $\|\mathcal{L}_\lambda^* - F\|_\infty = \mathcal{O}(\ell\kappa/\lambda)$; *(ii)* $\|\nabla\mathcal{L}_\lambda^* - \nabla F\|_\infty = \mathcal{O}(\ell\kappa^3/\lambda)$; *(iii)* $\mathcal{L}_\lambda^*$ *is* $\mathcal{O}(\ell\kappa^3)$*-smooth (independently of* λ*).*

4. To be precise, in order to also avoid Hessian computations, $\nabla F(\mathbf{x})$ will be estimated by the difference of gradients at two private points close to $\mathbf{y}^*(\mathbf{x})$.

Algorithm 1 DP Bilevel

Input: Initialization $(\mathbf{x}_0, \mathbf{y}_0) \in \mathcal{X} \times \mathcal{Y}$, privacy budget (ϵ, δ), penalty $\lambda > 0$, noise level $\sigma^2 > 0$, step size $\eta > 0$, iteration budget $T \in \mathbb{N}$.

for $t = 0, \ldots, T-1$ **do**

Apply $(\frac{\epsilon}{\sqrt{18T}}, \frac{\delta}{3(T+1)})$-DP-Loc-GD (Algorithm 2) to solve ▷ Strongly-convex problems

$$\widetilde{\mathbf{y}}_t \approx \arg\min_{\mathbf{y}} g(\mathbf{x}_t, \mathbf{y}) , \quad \widetilde{\mathbf{y}}_t^{\lambda} \approx \arg\min_{\mathbf{y}} [f(\mathbf{x}_t, \mathbf{y}) + \lambda \cdot g(\mathbf{x}_t, \mathbf{y})]$$

$\widetilde{\mathbf{g}}_t = \nabla_x f(\mathbf{x}_t, \widetilde{\mathbf{y}}_t^{\lambda}) + \lambda \left(\nabla_x g(\mathbf{x}_t, \widetilde{\mathbf{y}}_t^{\lambda}) - \nabla_x g(\mathbf{x}_t, \widetilde{\mathbf{y}}_t)\right) + \nu_t$, where $\nu_t \sim \mathcal{N}(\mathbf{0}, \sigma^2 I_{d_x})$

$\mathbf{x}_{t+1} = \arg\min_{\mathbf{u}\in\mathcal{X}} \left\{\langle \widetilde{\mathbf{g}}_t, \mathbf{u}\rangle + \frac{1}{2\eta} \|\mathbf{u} - \mathbf{x}_t\|^2\right\}$ ▷ If $\mathcal{X} = \mathbb{R}^{d_x}$, then $\mathbf{x}_{t+1} = \mathbf{x}_t - \eta\widetilde{\mathbf{g}}_t$

end

$t_{\text{out}} = \arg\min_{t\in\{0,\ldots,T-1\}} \|\mathbf{x}_{t+1} - \mathbf{x}_t\|$.

Output: $\mathbf{x}_{t_{\text{out}}}$.

Algorithm 2 DP-Loc-GD

Input: Objective $h : \mathbb{R}^{d_y} \to \mathbb{R}$, initialization $\mathbf{y}_0 \in \mathcal{Y}$, privacy budget (ϵ', δ'), number of rounds $M \in \mathbb{N}$, noise level $\sigma_{\text{GD}}^2 > 0$, step sizes $(\eta_t)_{t=0}^{T-1}$, iteration budget $T_{\text{GD}} \in \mathbb{N}$, radii $(R_m)_{m=0}^{M-1} > 0$.

$\mathbf{y}_0^0 = \mathbf{y}_0$

for $m = 0, \ldots, M-1$ **do**

for $t = 0, \ldots, T_{\text{GD}} - 1$ **do**

$\mathbf{y}_{t+1}^m = \text{Proj}_{\mathbb{B}(\mathbf{y}_0^m, R_m)} [\mathbf{y}_t^m - \eta_t (\nabla h(\mathbf{y}_t^m) + \nu_t)]$, where $\nu_t^m \sim \mathcal{N}(\mathbf{0}, \sigma_{\text{GD}}^2 I_{d_y})$

end

$\mathbf{y}_0^{m+1} = \frac{1}{T} \sum_{t=0}^{T-1} \mathbf{y}_t^m$

end

Output: $\mathbf{y}_{\text{out}} = \mathbf{y}_0^M$.

In other words, the lemma shows that for sufficiently large penalty λ, $\mathcal{L}_\lambda^*$ is a smooth approximation of the hyperobjective F, and that it suffices to minimize the gradient norm of $\mathcal{L}_\lambda^*$ in order to get a hypergradient guarantee in terms of ∇F. Moreover, had we computed $\mathbf{y}^*(\mathbf{x}), \mathbf{y}^\lambda(\mathbf{x})$, we note that $\nabla\mathcal{L}_\lambda^*$ can be obtained in a first-order fashion, since by construction $\mathcal{L}_\lambda^*(\mathbf{x}) = \arg\min_{\mathbf{y}} \mathcal{L}_\lambda(\mathbf{x}, \mathbf{y})$, and therefore by first-order optimality:

$$\begin{aligned}\nabla\mathcal{L}_\lambda^*(\mathbf{x}) &= \nabla_x\mathcal{L}_\lambda^*(\mathbf{x}, \mathbf{y}^\lambda(\mathbf{x})) + \nabla_x\mathbf{y}^\lambda(\mathbf{x})^\top \underbrace{\nabla_y\mathcal{L}_\lambda(\mathbf{x}, \mathbf{y}^\lambda(\mathbf{x}))}_{=\mathbf{0}} \\ &= \nabla_x f(\mathbf{x}, \mathbf{y}^\lambda(\mathbf{x})) + \lambda \left(\nabla_x g(\mathbf{x}, \mathbf{y}^\lambda(\mathbf{x})) - \nabla_x g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\right) . \end{aligned} \tag{3}$$

This strategy raises a privacy concern, as we exemplified in Section 3: Since $\mathbf{y}^*(\mathbf{x}), \mathbf{y}^\lambda(\mathbf{x})$ are required in order to compute $\nabla\mathcal{L}_\lambda^*(\mathbf{x})$, and are defined as the minimizers of $g(\mathbf{x}, \cdot), \mathcal{L}_\lambda(\mathbf{x}, \cdot)$ which are data-dependent, we cannot simply compute them under DP. In other words, even deciding *where* to invoke the gradient oracles, can already leak user information, hence breaking privacy before the gradients are even computed. We therefore must resort to approximating them using an auxiliary private method, for which we use DP-Loc(alized)-GD (Algorithm 2).[5]

In our analysis, we crucially rely on the fact that $g(\mathbf{x},\cdot)$ $\mathcal{L}_\lambda(\mathbf{x},\cdot)$ are both strongly-convex, which implies that optimizing them produces $\widetilde{\mathbf{y}}_t, \widetilde{\mathbf{y}}_t^\lambda$ such that the *distances* to the minimizers, namely $\|\widetilde{\mathbf{y}}_t - \mathbf{y}^*(\mathbf{x}_t)\|, \|\widetilde{\mathbf{y}}_t^\lambda - \mathbf{y}^\lambda(\mathbf{x}_t)\|$, are small. The distance bound is key, since then (3) allows using the smoothness of f, g to translate the distance bounds into an inexact (i.e., biased) gradient oracle for $\nabla\mathcal{L}_\lambda^*(\mathbf{x}_t)$, computed at the private points $\widetilde{\mathbf{y}}_t, \widetilde{\mathbf{y}}_t^\lambda$. We are able to bound the gradient bias induced by privatizing the inner solvers, as follows:

Lemma 8 *If $\lambda \geq \max\{\frac{2L_1^g}{\mu_g}, \frac{L_0^f}{L_0^g}, \frac{L_1^f}{L_1^g}\}$, then the random variables $\widetilde{\mathbf{y}}_t, \widetilde{\mathbf{y}}_t^\lambda$ as defined in Algorithm 1 satisfy for all $t < T$ with probability at least $1-\gamma$:*

$$\left\|\nabla\mathcal{L}_\lambda^*(\mathbf{x}_t) - \left[\nabla_x f(\mathbf{x}_t, \widetilde{\mathbf{y}}_t^\lambda) + \lambda\left(\nabla_x g(\mathbf{x}_t, \widetilde{\mathbf{y}}_t^\lambda) - \nabla_x g(\mathbf{x}_t, \widetilde{\mathbf{y}}_t)\right)\right]\right\| \leq \beta = \widetilde{\mathcal{O}}\left(\frac{\lambda\ell\kappa\sqrt{d_y T}}{\epsilon n}\right) .$$

Having constructed an inexact hypergradient oracle, we can privatize its response using the standard Gaussian mechanism. To do so, we recall that the noise variance required to ensure privacy is tied to the component Lipschitz constants, and note that $\mathcal{L}_\lambda^*(\mathbf{x})$ decomposes as the finite-sum

$$\mathcal{L}_\lambda^*(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^n \mathcal{L}_{\lambda,i}^*(\mathbf{x}) \ , \quad \mathcal{L}_{\lambda,i}^*(\mathbf{x}) := f(\mathbf{x}, \mathbf{y}^\lambda(\mathbf{x}); \xi_i) + \lambda\left[g(\mathbf{x}, \mathbf{y}^\lambda(\mathbf{x}); \xi_i) - g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}); \xi_i)\right] .$$

Bounding the Lipschitz constant of $\mathcal{L}_{\lambda,i}^*$ naively by applying the chain rule would result in the bound $\mathrm{Lip}(\mathbf{y}^*)(L_0^f + 2\lambda L_0^g) \lesssim \lambda \mathrm{Lip}(\mathbf{y}^*)L_0^g$, where $\mathrm{Lip}(\mathbf{y}^*)$ is the Lipschitz constant of $\mathbf{y}^*(\mathbf{x}) : \mathbb{R}^{d_x} \to \mathbb{R}^{d_y}$. Unfortunately, this bound grows with the penalty parameter λ, which will eventually be set large, and in particular, will grow with the dataset size n. We therefore need to derive a more nuanced analysis that allows obtaining a significantly tighter Lipschitz bound. We achieve this by showing that $\|\mathbf{y}^\lambda(\mathbf{x}) - \mathbf{y}^*(\mathbf{x})\| \lesssim 1/\lambda$, which cancels out the multiplication by λ, and thus prove the following tight Lipschitz bound:

Lemma 9 *$\mathcal{L}_{\lambda,i}^*$ is $\mathcal{O}(\ell\kappa)$-Lipschitz independently of λ.*

Finally, having constructed a private inexact stochastic oracle response for the smooth approximation $\mathcal{L}_\lambda^*$, we analyze an outer loop (update of $\mathbf{x}_t$), showing that is optimally robust to inexact and noisy gradients. Moreover, we employ an output rule which makes use of the already-privatized iterates, thus overcoming the need of additional noise in choosing the smallest gradient. This is a subtle technical challenge which is unique to the private setting, since non-private algorithms can simply validate the gradient norm and return the minimal one, yet in our case, using the Laplace mechanism to do so would spoil the convergence rate. In particular, we show that the corresponding trajectory and output rule lead a point with small (projected-)gradient norm, as stated below:

Proposition 10 *Suppose $h : \mathbb{R}^d \to \mathbb{R}$ is L-smooth, that $\|\widetilde{\nabla} h(\cdot) - \nabla h(\cdot)\| \leq \beta$, and consider the following update rule with $\eta = \frac{1}{2L}$: $\mathbf{x}_{t+1} = \arg\min_{\mathbf{u}\in\mathcal{X}} \left\{\langle \widetilde{\nabla} h(\mathbf{x}_t) + \nu_t, \mathbf{u}\rangle + \frac{1}{2\eta}\|\mathbf{x}_t - \mathbf{u}\|^2\right\}$, $\nu_t \sim$*

5. We can replace the inner solver by any DP method that guarantees with high probability the optimal rate for strongly-convex objectives, as we further discuss in Appendix C.

$\mathcal{N}(0,\sigma^2 I)$ with the output rule $\mathbf{x}_{\text{out}} := \mathbf{x}_{t_{\text{out}}},\ t_{\text{out}} := \arg\min_{t\in\{0,\dots,T-1\}} \|\mathbf{x}_{t+1}-\mathbf{x}_t\|$. If $\alpha > 0$ is such that $\alpha \geq C\max\{\beta, \sigma\sqrt{d\log(T/\gamma)}\}$ for a sufficiently large absolute constant $C > 0$, then with probability at least $1-\gamma$: $\|\mathcal{G}_{h,\eta}(\mathbf{x}_{\text{out}})\| \leq \alpha$ for $T = \mathcal{O}\big(\frac{L(h(\mathbf{x}_0)-\inf h)}{\alpha^2}\big)$.

Notably, for any $\alpha > 0$, the result above allows the bias β to be as large as $\Omega(\alpha)$ at all points, and the algorithm still reaches an α-stationary point. Overall, applying Proposition 10 to $h = \mathcal{L}_\lambda^*$, we see that the (projected-)gradient norm can be as small as $\max\{\beta, \sigma\sqrt{d_x}\}$, up to logarithmic terms. Accounting for the smallest possible inexactness β and noise addition σ that ensure the the inner and outer loops, respectively, are both sufficiently private, we conclude the proof of Theorem 4; the full details appear in Appendix B.

5. Mini-batch algorithm for DP bilevel ERM

In this section, we consider once again the bilevel ERM problem (BO/ERM), and provide Algorithm 3, which is a mini-batch variant of the previously discussed bilevel ERM algorithm. Given a mini-batch $B \subseteq S = \{\xi_1,\dots,\xi_n\}$ and a function $h : \mathbb{R}^d \times \Xi \to \mathbb{R}$, we let $\nabla h(\mathbf{z}; B) = \frac{1}{|B|}\sum_{\xi_i\in B}\nabla h(\mathbf{z};\xi_i)$ denote the mini-batch gradient. We prove the following result:

Theorem 11 *Assume 1 and 2 hold, and that $\alpha \leq \ell\kappa^3 \min\{\frac{1}{2\kappa}, \frac{L_0^g}{L_0^f}, \frac{L_1^g}{L_1^f}, \frac{\Delta_F}{\ell\kappa}\}$. Then running Algorithm 3 with parameters assigned as in Theorem 4 and any batch sizes $b_{\text{in}}, b_{\text{out}} \in [n]$ for sampling gradients, satisfies (ϵ,δ)-DP and returns $\mathbf{x}_{\text{out}}$ such that with probability at least $1-\gamma$:*

$$\|\mathcal{G}_{F_S,\eta}(\mathbf{x}_{\text{out}})\| \leq \alpha = \widetilde{\mathcal{O}}\left(K_1\Big(\frac{\sqrt{d_x}}{\epsilon n}\Big)^{1/2} + K_2\Big(\frac{\sqrt{d_y}}{\epsilon n}\Big)^{1/3} + K_3\cdot\frac{1}{b_{\text{out}}}\right),$$

where $K_1 = \mathcal{O}(\Delta_F^{1/4}\ell^{3/4}\kappa^{5/4}),\ K_2 = \mathcal{O}(\Delta_F^{1/6}\ell^{5/6}\kappa^{11/6}),\ K_3 = \mathcal{O}(\ell\kappa)$.

Algorithm 3 Mini-batch DP Bilevel

Input: Initialization $(\mathbf{x}_0,\mathbf{y}_0) \in \mathcal{X}\times\mathcal{Y}$, privacy budget (ϵ,δ), penalty $\lambda > 0$, noise level $\sigma^2 > 0$, step size $\eta > 0$, iteration budget $T \in \mathbb{N}$, batch sizes $b_{\text{in}}, b_{\text{out}} \in \mathbb{N}$.

for $t = 0,\dots,T-1$ **do**

Apply $(\frac{\epsilon}{\sqrt{18T}}, \frac{\delta}{3(T+1)})$-DP-Loc-SGD (Algorithm 4) to solve ▷ Strongly-convex problems

$$\widetilde{\mathbf{y}}_t \approx \arg\min_{\mathbf{y}} g(\mathbf{x}_t,\mathbf{y}), \qquad \widetilde{\mathbf{y}}_t^\lambda \approx \arg\min_{\mathbf{y}} [f(\mathbf{x}_t,\mathbf{y}) + \lambda\cdot g(\mathbf{x}_t,\mathbf{y})]$$

$\widetilde{\mathbf{g}}_t = \nabla_x f(\mathbf{x}_t, \widetilde{\mathbf{y}}_t^\lambda; B_t) + \lambda\big(\nabla_x g(\mathbf{x}_t,\widetilde{\mathbf{y}}_t^\lambda; B_t) - \nabla_x g(\mathbf{x}_t,\widetilde{\mathbf{y}}_t; B_t)\big) + \nu_t,\ B_t \sim S^{b_{\text{out}}},\ \nu_t \sim \mathcal{N}(\mathbf{0},\sigma^2 I_{d_x})$

$\mathbf{x}_{t+1} = \arg\min_{\mathbf{u}\in\mathcal{X}}\left\{\langle\widetilde{\mathbf{g}}_t,\mathbf{u}\rangle + \frac{1}{2\eta}\|\mathbf{u}-\mathbf{x}_t\|^2\right\}$ ▷ If $\mathcal{X} = \mathbb{R}^{d_x}$, then $\mathbf{x}_{t+1} = \mathbf{x}_t - \eta\widetilde{\mathbf{g}}_t$

end

$t_{\text{out}} = \arg\min_{t\in\{0,\dots,T-1\}}\|\mathbf{x}_{t+1}-\mathbf{x}_t\|$.

Output: $\mathbf{x}_{t_{\text{out}}}$.

Algorithm 4 DP-Loc-SGD

Input: Objective $h : \mathbb{R}^{d_y} \times \Xi \to \mathbb{R}$, initialization $\mathbf{y}_0 \in \mathcal{Y}$, privacy budget (ϵ', δ'), batch size $b_{\text{in}} \in \mathbb{N}$, number of rounds $M \in \mathbb{N}$, noise level $\sigma^2_{\text{SGD}} > 0$, step sizes $(\eta_t)_{t=0}^{T-1}$, iteration budget $T_{\text{SGD}} \in \mathbb{N}$, radii $(R_m)_{m=0}^{M-1} > 0$.

$\mathbf{y}_0^0 = \mathbf{y}_0$

for $m = 0, \ldots, M-1$ **do**

  for $t = 0, \ldots, T_{\text{SGD}} - 1$ **do**

    $\mathbf{y}_{t+1}^m = \text{Proj}_{\mathbb{B}(\mathbf{y}_0^m, R_m)} \left[\mathbf{y}_t^m - \eta_t \left(\nabla h(\mathbf{y}_t^m; B_t) + \nu_t\right)\right], \;\; B_t^m \sim S^{b_{\text{in}}}, \; \nu_t^m \sim \mathcal{N}(\mathbf{0}, \sigma^2_{\text{SGD}} I_{d_y})$

  end

  $\mathbf{y}_0^{m+1} = \frac{1}{T} \sum_{t=0}^{T-1} \mathbf{y}_t^m$

end

Output: $\mathbf{y}_{\text{out}} = \mathbf{y}_0^M$.

Remark 12 (Outer batch size dependence) *Algorithm 3 ensures privacy for any batch sizes, yet notably, the guaranteed gradient norm bound does not go to zero (as $n \to \infty$) for constant outer-batch size. The same phenomenon also holds for for "single"-level constrained nonconvex optimization, as noted by Ghadimi et al. (2016) (specifically, see Corollary 4 and related discussion therein). Accordingly, the inner-batch size b_{in} can be set whatsoever, while setting $b_{\text{out}} = \mathcal{O}(\max\{(\epsilon n/\sqrt{d_x})^{1/2}, (\epsilon n/\sqrt{d_y})^{1/3}\}) \ll n$ recovers the full-batch rate (in general, $\|\mathcal{G}(\mathbf{x}_{\text{out}})\| \overset{n\to\infty}{\longrightarrow} 0$ whenever $b_{\text{out}} \overset{n\to\infty}{\longrightarrow} \infty$). It is interesting to note that the $1/b_{\text{out}}$ term shows up in the analysis only as an upper bound on the sub-Gaussian norm of the mini-batch gradient estimator. Thus, in applications for which some (possibly small) batch size results in reasonably accurate gradients, the result above holds with the outer mini-batch gradient's standard deviation replacing $1/b_{\text{out}}$, which is to be expected anyhow for high probability guarantees.*

The difference between Algorithm 3 and Algorithm 1, is that both the inner and outer loops sample mini-batch gradients. The inner loop guarantee is the same regardless of the inner-batch size b_{in}, since for strongly-convex objectives it is possible to prove the same convergence rate guarantee for DP optimization in any case (as further discussed in Appendix C). As to the outer loop ($\mathbf{x}_{t+1}$ update rule), we apply standard concentration bounds to argue about the quality of the gradient estimates — hence the additive $1/b_{\text{out}}$ factor — and rely on our analysis of the outer loop with inexact gradients (which are now even less exact due to sampling stochasticity). We remark that compared to the well-known analysis of Ghadimi et al. (2016) for mini-batch constrained nonconvex optimization, we derive high probability bounds without requiring several re-runs of the algorithm. We further remark that mini-batch sampling with replacement is analyzed for simplicity, though the same guarantees (up to constants) can be derived for sampling without replacement, at the cost of a more involved analysis.

6. Generalizing from ERM to population loss

We now move to consider stochastic (population) objectives, namely the problem (BO/Pop). We denote the population hyperobjective by $F_{\mathcal{P}}$, and as before F_S denotes the empirical objective, where $S \sim \mathcal{P}^n$. We prove the following result:

Theorem 13 *Under Assumptions 1 and 2, if the preconditions of Theorem 4 hold, then Algorithm 1 is (ϵ,δ)-DP, and returns $\mathbf{x}_{\rm out}$ such that with probability at least $1-\gamma$:*

$$\|\mathcal{G}_{F_{\mathcal{P}},\eta}(\mathbf{x}_{\rm out})\|\leq\alpha=\widetilde{\mathcal{O}}\left(K_1\Big(\frac{\sqrt{d_x}}{\epsilon n}\Big)^{1/2}+K_2\Big(\frac{\sqrt{d_y}}{\epsilon n}\Big)^{1/3}+\mathrm{gen}_n\right),$$

where $K_1=\mathcal{O}(\Delta_F^{1/4}\ell^{1/4}\kappa^{5/4}),\ K_2=\mathcal{O}(\Delta_F^{1/6}\ell^{5/6}\kappa^{11/6})$, and

$$\mathrm{gen}_n=\widetilde{\mathcal{O}}\left(\frac{\ell\kappa(\ell\kappa+1)(\sqrt{d_x}+\kappa)}{\sqrt{n}}+\frac{\ell^{1/2}\kappa^{3/2}(\ell\kappa+1)}{n^{1/4}}\right).$$

Similarly, if the preconditions of Theorem 11 hold, then for any batch sizes $b_{\rm in},b_{\rm out}\in[n]$, Algorithm 3 is (ϵ,δ)-DP, and returns $\mathbf{x}_{\rm out}$ such that with probability at least $1-\gamma$:

$$\|\mathcal{G}_{F_{\mathcal{P}},\eta}(\mathbf{x}_{\rm out})\|\leq\alpha=\widetilde{\mathcal{O}}\left(K_1\Big(\frac{\sqrt{d_x}}{\epsilon n}\Big)^{1/2}+K_2\Big(\frac{\sqrt{d_y}}{\epsilon n}\Big)^{1/3}+\frac{\ell\kappa}{b_{\rm out}}+\mathrm{gen}_n\right).$$

The proof of Theorem 13 relies on establishing uniform convergence of the empirical hypergradients to the population hypergradients (Lemma 27), which as far as we know does not appear in prior literature and therefore may be of independent interest. This further implies uniform convergence of projected gradients, since the gradient mapping is non-expansive.

7. Application: private regularization hyperparameter tuning

In this section, we specialize our private bilevel framework to the well-studied problem of tuning a regularization hyperparameter. Specifying our analysis to this task results in a simple private algorithm, which as we will soon explain, offers an alternative paradigm for private hyperparameter tuning compared to existing works on this topic (Liu and Talwar, 2019; Papernot and Steinke, 2022).

We start by introducing the setting. Given a labeled dataset $S=(\mathbf{a}_i,b_i)_{i=1}^n\subset\mathbb{R}^{d_a}\times\mathbb{R}$, we aim to fit a parametric model $\{h_\theta:\mathbb{R}^{d_a}\to\mathbb{R}\mid\theta\in\Theta\subseteq\mathbb{R}^{d_\theta}\}$ (e.g., linear model, or neural network) with respect to a loss function $\ell oss:\mathbb{R}\times\mathbb{R}\to\mathbb{R}_{\geq0}$. A classic approach is to split the data into two disjoint datasets, train and validation $S=S_{\rm tr}\cup S_{\rm val}$, choose a regularizer $R:\Theta\to\mathbb{R}_{\geq0}$ (e.g., $R(\theta)=\|\theta\|_2$ or $R(\theta)=\|\theta\|_1$) and minimize the validation loss of a model trained under regularization, while searching the best amount of regularization. This corresponds to the bilevel problem

$$\begin{aligned}&\mathrm{minimize}_{\omega\geq0}\quad F(\omega):=f(\omega,\theta^*_\omega)=\frac{1}{|S_{\rm val}|}\sum_{(\mathbf{a}_i,b_i)\in S_{\rm val}}\ell oss(h_{\theta^*_\omega}(\mathbf{a}_i),b_i)\\&\text{subject to}\qquad\theta^*_\omega:=\arg\min_{\theta\in\Theta}g(\omega,\theta)=\frac{1}{|S_{\rm tr}|}\sum_{(\mathbf{a}_i,b_i)\in S_{\rm tr}}\ell oss(h_\theta(\mathbf{a}_i),b_i)+\omega\cdot R(\theta)\,.\end{aligned}\tag{4}$$

The so-called model selection problem described above is extremely well studied, and strategies to solve it generally fall into two categories. The first approach is to discretize the domain of ω to a finite set Ω, train models associated to each $\omega\in\Omega$, and pick the model minimizing the validation loss (cf. Hastie et al., 2009, Section 7 for a detailed discussion). This raises a privacy issue, since even if each model is private, the very choice of the hyperparameter can break privacy. Therefore, privatizing this approach requires employing a private selection algorithm over the hyperparameter, as studied by Liu and Talwar (2019); Papernot and Steinke (2022).

A second approach to hyperparameter tuning is based on differentiable programming: instead of predetermining a finite set of possible values for ω, we can solve (4) using BO methods, namely "differentiate through ω", and update it accordingly on the fly. This approach was pioneered by Bengio (2000), and was further developed by multiple works (Maclaurin et al., 2015; Franceschi et al., 2017, 2018; Lorraine et al., 2020; Engstrom et al., 2025). However, we are not aware of a private variant of this methodology, and our goal is to provide one here.

Applying our algorithm to this problem turns out to carry some useful simplifications. Note that $\nabla_\omega f = 0$, $\nabla_\omega g(\omega, \theta) = R(\theta)$, thus letting $\theta_t \approx \arg\min_{\mathbf{w}} g(\omega_t, \theta)$ and $\theta_t^\lambda \approx \arg\min_\theta(f(\omega_t, \theta) + \lambda g(\omega_t, \theta))$ be private solutions to the inner problems,[6] our algorithm yields the following outer loop:

$$\begin{aligned}
\omega_{t+1} &= \arg\min_{u\geq 0} \left\{ u \left[\nabla_\omega f(\omega_t, \theta_t^\lambda) + \lambda(\nabla_\omega g(\omega_t, \theta_t^\lambda) - \nabla_\omega g(\omega_t, \theta_t)) + \mathcal{N}(0, \sigma^2) \right] + \tfrac{1}{2\eta}(u - \omega_t)^2 \right\} \\
&= \arg\min_{u\geq 0} \left\{ u \left[0 + \lambda(R(\theta_t^\lambda) - R(\theta_t)) + \mathcal{N}(0, \sigma^2) \right] + \tfrac{1}{2\eta}(u - \omega_t)^2 \right\} \\
&= \max \left\{ 0,\ \omega_t + \eta\lambda \cdot \mathcal{N}\left(R(\theta_t^\lambda) - R(\theta_t),\ \sigma^2/\lambda^2 \right) \right\} ,
\end{aligned}$$

where the last equality is easily verified by solving a one-dimensional quadratic over a half-line.

Overall, we obtain a simple closed-form update rule for privately tuning the regularization parameter ω, based on two private training subroutines (which can be solved in a black-box manner), according to a normal variable centered at the difference between their associated complexities measured by R.

8. Discussion

In this paper, we studied DP bilevel optimization, and proposed the first algorithms to solve this problem designed for the central DP model, while also being the first that use only gradient queries. Our provided guarantees hold both for constrained and unconstrained settings, cover empirical and population losses alike, and account for mini-batched gradients. As an application, we derived a private update rule for tuning a regularization hyperparameter when fitting a statistical model.

Our work leaves open several directions for future research. First, it is natural to ask what is the extent to which our results can be improved. Notably, even in the well-studied setting of *single*-level smooth-nonconvex DP optimization, there still exist gaps between known upper and lower bounds for minimizing the gradient norm (cf. Lowy et al. 2024 and discussion therein). The best known lower bound for such problems, which trivially applies also for DP BO which is a strictly more general problem setting, is $\Omega(\sqrt{d}/\epsilon n)$, hinting that our results are perhaps not tight. Moreover, Lowy and Liu (2025) recently presented improved gradient bounds for DP BO via second-order methods, and it is interesting to ask whether first-order methods for this setting, as studied in this work, can be improved.

To follow up on this question, we remark that a candidate strategy to improve the convergence rate in this work would be to use variance reduction, as Arora et al. (2023) used variance reduction

6. Note that θ_t can be interpreted as a regularized model over the training data, while θ_t^λ simply corresponds to a model trained by mixing in a bit of the validation loss (weighted down by $1/\lambda$).

to derive faster convergence to private stationarity for single-level DP nonconvex optimization. Applying this for DP BO as the outer loop would require, according to our analysis, to evaluate the cost of inexact gradients in variance-reduced methods, which is left for future work.

Another direction for future work is extending the analysis in Section 7 to derive update rules for privately tuning multiple hyperparameters simultaneously. It is interesting to note that the complexity of our results scales polynomially with the upper-level dimension, which corresponds to the number of hyperparameters, whereas preforming a grid search scales exponentially with the number of hyperparameters. The trade-off is, of course, convergence to local stationarity instead of global optimality.

Lastly, another open direction is understanding whether mini-batch algorithms can avoid the additive $1/b_{\text{out}}$ factor in the *unconstrained* case $\mathcal{X} = \mathbb{R}^{d_x}$. As previously discussed, for constrained problems, even single-level nonconvex algorithms suffer from this batch dependence (Ghadimi et al., 2016). However, for unconstrained problems, Ghadimi and Lan (2013) showed that setting a smaller stepsize, on the order of α^2/σ^2, converges to a point with gradient bounded by α after $\mathcal{O}(\alpha^{-4})$ steps, even for $b_{\text{out}} = 1$. Applying this to DP bilevel unconstrained optimization seems feasible, and requires accounting for the larger privacy loss due to the slower convergence rate (compared to $\mathcal{O}(\alpha^{-2})$ in our case).

Acknowledgments

This research is supported through an Azrieli Foundation graduate fellowship. We would like to express our gratitude to Vitaly Feldman, Kunal Talwar and the Apple MLR team for the encouraging environment in which this research was initiated. Specifically, we thank Kunal Talwar for insightful discussions, which included pointing out the utility of localizing DP-SGD for optimal high probability guarantees.

References

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

Nazanin Abolfazli, Ruichen Jiang, Aryan Mokhtari, and Erfan Yazdandoost Hamedani. An inexact conditional gradient method for constrained bilevel optimization. *arXiv preprint arXiv:2306.02429*, 2023.

Zeeshan Akhtar, Amrit Singh Bedi, Srujan Teja Thomdapu, and Ketan Rajawat. Projection-free stochastic bi-level optimization. *IEEE Transactions on Signal Processing*, 70:6332–6347, 2022.

Mostafa Amini and Farzad Yousefian. An iterative regularized incremental projected subgradient method for a class of bilevel optimization problems. In *2019 American Control Conference (ACC)*, pages 4069–4074. IEEE, 2019.

G Anandalingam and DJ White. A solution method for the linear static stackelberg problem using penalty functions. *IEEE Transactions on automatic control*, 35(10):1170–1173, 1990.

Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. Learning to learn by gradient descent by gradient descent. *Advances in neural information processing systems*, 29, 2016.

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Raman Arora, Raef Bassily, Tomás González, Cristóbal A Guzmán, Michael Menart, and Enayat Ullah. Faster rates of convergence to stationary points in differentially private optimization. In *International Conference on Machine Learning*, pages 1060–1092. PMLR, 2023.

Borja Balle, Gilles Barthe, and Marco Gaboardi. Privacy amplification by subsampling: Tight analyses via couplings and divergences. *Advances in neural information processing systems*, 31, 2018.

Jonathan F Bard. *Practical bilevel optimization: algorithms and applications*, volume 30. Springer Science & Business Media, 2013.

Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th annual symposium on foundations of computer science*, pages 464–473. IEEE, 2014.

Raef Bassily, Vitaly Feldman, Kunal Talwar, and Abhradeep Guha Thakurta. Private stochastic convex optimization with optimal rates. *Advances in neural information processing systems*, 32, 2019.

Yoshua Bengio. Gradient-based optimization of hyperparameters. *Neural computation*, 12(8):1889–1900, 2000.

Luca Bertinetto, Joao F Henriques, Philip HS Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. *arXiv preprint arXiv:1805.08136*, 2018.

Jérôme Bolte, Quoc-Tùng Lê, Edouard Pauwels, and Samuel Vaiter. Geometric and computational hardness of bilevel programming. *Mathematical Programming*, pages 1–36, 2025.

Jerome Bracken and James T McGill. Mathematical programs with optimization problems in the constraints. *Operations research*, 21(1):37–44, 1973.

Jincheng Cao, Ruichen Jiang, Nazanin Abolfazli, Erfan Yazdandoost Hamedani, and Aryan Mokhtari. Projection-free methods for stochastic simple bilevel optimization with convex lower-level problem. *Advances in Neural Information Processing Systems*, 36, 2024.

Yair Carmon, Arun Jambulapati, Yujia Jin, Yin Tat Lee, Daogao Liu, Aaron Sidford, and Kevin Tian. Resqueing parallel and private stochastic convex optimization. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 2031–2058. IEEE, 2023.

Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(3), 2011.

Lesi Chen, Jing Xu, and Jingzhao Zhang. On finding small hyper-gradients in bilevel optimization: Hardness results and improved analysis. In *Proceedings of Thirty Seventh Conference on Learning Theory*, volume 247 of *Proceedings of Machine Learning Research*, pages 947–980. PMLR, 2024.

Lesi Chen, Yaohua Ma, and Jingzhao Zhang. Near-optimal nonconvex-strongly-convex bilevel optimization with fully first-order oracles. *Journal of Machine Learning Research*, 26(109):1–56, 2025.

Tianyi Chen, Yuejiao Sun, and Wotao Yin. Closing the gap: Tighter analysis of alternating stochastic gradient methods for bilevel problems. *Advances in Neural Information Processing Systems*, 34: 25294–25307, 2021.

Tianyi Chen, Yuejiao Sun, Quan Xiao, and Wotao Yin. A single-timescale method for stochastic bilevel optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 2466–2488. PMLR, 2022.

Ziqin Chen and Yongqiang Wang. Locally differentially private decentralized stochastic bilevel optimization with guaranteed convergence accuracy. In *Forty-first International Conference on Machine Learning*, 2024.

Benoît Colson, Patrice Marcotte, and Gilles Savard. An overview of bilevel optimization. *Annals of operations research*, 153:235–256, 2007.

Mathieu Dagréou, Thomas Moreau, Samuel Vaiter, and Pierre Ablin. A lower bound and a near-optimal algorithm for bilevel empirical risk minimization. In *International Conference on Artificial Intelligence and Statistics*, pages 82–90. PMLR, 2024.

Stephan Dempe. *Foundations of bilevel programming*. Springer Science & Business Media, 2002.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Third Theory of Cryptography Conference*, pages 265–284. Springer, 2006.

Cynthia Dwork, Guy N Rothblum, and Salil Vadhan. Boosting and differential privacy. In *2010 IEEE 51st annual symposium on foundations of computer science*, pages 51–60. IEEE, 2010.

Logan Engstrom, Andrew Ilyas, Benjamin Chen, Axel Feldmann, William Moses, and Aleksander Madry. Optimizing ml training with metagradient descent. *arXiv preprint arXiv:2503.13751*, 2025.

Simin Fan, Matteo Pagliardini, and Martin Jaggi. Doge: Domain reweighting with generalization estimation. In *Forty-first International Conference on Machine Learning*, 2024.

Vitaly Feldman and Jan Vondrak. High probability generalization bounds for uniformly stable algorithms with nearly optimal rate. In *Conference on learning theory*, pages 1270–1279. PMLR, 2019.

Vitaly Feldman, Tomer Koren, and Kunal Talwar. Private stochastic convex optimization: optimal rates in linear time. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 439–449, 2020.

Ferdinando Fioretto, Lesia Mitridati, and Pascal Van Hentenryck. Differential privacy for stackelberg games. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 3480–3486, 2021.

Luca Franceschi, Michele Donini, Paolo Frasconi, and Massimiliano Pontil. Forward and reverse gradient-based hyperparameter optimization. In *International Conference on Machine Learning*, pages 1165–1173. PMLR, 2017.

Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimiliano Pontil. Bilevel programming for hyperparameter optimization and meta-learning. In *International conference on machine learning*, pages 1568–1577. PMLR, 2018.

Arun Ganesh, Daogao Liu, Sewoong Oh, and Abhradeep Guha Thakurta. Private (stochastic) non-convex optimization revisited: Second-order stationary points and excess risks. *Advances in Neural Information Processing Systems*, 36, 2024.

Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM journal on optimization*, 23(4):2341–2368, 2013.

Saeed Ghadimi and Mengdi Wang. Approximation methods for bilevel programming. *arXiv preprint arXiv:1802.02246*, 2018.

Saeed Ghadimi, Guanghui Lan, and Hongchao Zhang. Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization. *Mathematical Programming*, 155(1):267–305, 2016.

Sivakanth Gopi, Yin Tat Lee, and Daogao Liu. Private convex optimization via exponential mechanism. In *Conference on Learning Theory*, pages 1948–1989. PMLR, 2022.

David Grangier, Pierre Ablin, and Awni Hannun. Bilevel optimization to learn training distributions for language modeling under domain shift. In *NeurIPS 2023 Workshop on Distribution Shifts: New Frontiers with Foundation Models*, 2023.

Zhishuai Guo, Quanqi Hu, Lijun Zhang, and Tianbao Yang. Randomized stochastic variance-reduced methods for multi-task stochastic bilevel optimization. *arXiv preprint arXiv:2105.02266*, 2021.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer, 2009.

Mingyi Hong, Hoi-To Wai, Zhaoran Wang, and Zhuoran Yang. A two-timescale stochastic algorithm framework for bilevel optimization: Complexity analysis and application to actor-critic. *SIAM Journal on Optimization*, 33(1):147–180, 2023.

Yo Ishizuka and Eitaro Aiyoshi. Double penalty method for bilevel optimization problems. *Annals of Operations Research*, 34(1):73–88, 1992.

Kaiyi Ji, Jason D Lee, Yingbin Liang, and H Vincent Poor. Convergence of meta-learning with task-specific adaptation over partial parameters. *Advances in Neural Information Processing Systems*, 33:11490–11500, 2020.

Kaiyi Ji, Junjie Yang, and Yingbin Liang. Bilevel optimization: Convergence analysis and enhanced design. In *International conference on machine learning*, pages 4882–4892. PMLR, 2021.

Ruichen Jiang, Nazanin Abolfazli, Aryan Mokhtari, and Erfan Yazdandoost Hamedani. A conditional gradient-based method for simple bilevel optimization with convex lower-level problem. In *International Conference on Artificial Intelligence and Statistics*, pages 10305–10323. PMLR, 2023.

Yibo Jiang and Victor Veitch. Invariant and transportable representations for anti-causal domain shifts. *Advances in Neural Information Processing Systems*, 35:20782–20794, 2022.

Chi Jin, Praneeth Netrapalli, Rong Ge, Sham M Kakade, and Michael I Jordan. A short note on concentration inequalities for random vectors with subgaussian norm. *arXiv preprint arXiv:1902.03736*, 2019.

Gautam Kamath, Jerry Li, Vikrant Singhal, and Jonathan Ullman. Privately learning high-dimensional distributions. In *Conference on Learning Theory*, pages 1853–1902. PMLR, 2019.

Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011.

Prashant Khanduri, Siliang Zeng, Mingyi Hong, Hoi-To Wai, Zhaoran Wang, and Zhuoran Yang. A near-optimal algorithm for stochastic bilevel optimization via double-momentum. *Advances in neural information processing systems*, 34:30271–30283, 2021.

Daniel Kifer, Adam Smith, and Abhradeep Thakurta. Private convex empirical risk minimization and high-dimensional regression. In *Conference on Learning Theory*, pages 25–1. JMLR Workshop and Conference Proceedings, 2012.

Guy Kornowski. Differentially private bilevel optimization. *arXiv preprint arXiv:2409.19800*, 2024.

Guy Kornowski, Swati Padmanabhan, Kai Wang, Zhe Zhang, and Suvrit Sra. First-order methods for linearly constrained bilevel optimization. *Advances in neural information processing systems*, 2024.

Guy Kornowski, Daogao Liu, and Kunal Talwar. Improved sample complexity for private nonsmooth nonconvex optimization. *International Conference on Machine Learning*, 2025.

Jeongyeol Kwon, Dohyun Kwon, Stephen Wright, and Robert D Nowak. A fully first-order method for stochastic bilevel optimization. In *International Conference on Machine Learning*, pages 18083–18113. PMLR, 2023.

Jeongyeol Kwon, Dohyun Kwon, Stephen Wright, and Robert D Nowak. On penalty methods for nonconvex bilevel optimization and first-order stochastic approximation. In *The Twelfth International Conference on Learning Representations*, 2024.

Guanghui Lan. *First-order and stochastic optimization methods for machine learning*, volume 1. Springer, 2020.

Bo Liu, Mao Ye, Stephen Wright, Peter Stone, and Qiang Liu. Bome! bilevel optimization made easy: A simple first-order approach. *Advances in neural information processing systems*, 35: 17248–17262, 2022.

Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018.

Jingcheng Liu and Kunal Talwar. Private selection from private candidates. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 298–309, 2019.

Jonathan Lorraine, Paul Vicol, and David Duvenaud. Optimizing millions of hyperparameters by implicit differentiation. In *International conference on artificial intelligence and statistics*, pages 1540–1552. PMLR, 2020.

Andrew Lowy and Daogao Liu. Differentially private bilevel optimization: Efficient algorithms with near-optimal rates. *Neural Information Processing Systems*, 2025.

Andrew Lowy, Jonathan Ullman, and Stephen Wright. How to make the gradients small privately: Improved rates for differentially private non-convex optimization. In *Forty-first International Conference on Machine Learning*, 2024.

Zhaosong Lu and Sanyou Mei. First-order penalty methods for bilevel optimization. *SIAM Journal on Optimization*, 34(2):1937–1969, 2024.

Dougal Maclaurin, David Duvenaud, and Ryan Adams. Gradient-based hyperparameter optimization through reversible learning. In *International conference on machine learning*, pages 2113–2122. PMLR, 2015.

Song Mei, Yu Bai, and Andrea Montanari. The landscape of empirical risk for nonconvex losses. *The Annals of Statistics*, 46(6A):2747–2774, 2018.

Roey Merchav and Shoham Sabach. Convex bi-level optimization problems with nonsmooth outer objective function. *SIAM Journal on Optimization*, 33(4):3114–3142, 2023.

Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

Rui Pan, Dylan Zhang, Hanning Zhang, Xingyuan Pan, Minrui Xu, Jipeng Zhang, Renjie Pi, Xiaoyu Wang, and Tong Zhang. Scalebio: Scalable bilevel optimization for llm data reweighting. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 31959–31982, 2025.

Nicolas Papernot and Thomas Steinke. Hyperparameter tuning with renyi differential privacy. In *International Conference on Learning Representations*, 2022.

Aravind Rajeswaran, Chelsea Finn, Sham M Kakade, and Sergey Levine. Meta-learning with implicit gradients. *Advances in neural information processing systems*, 32, 2019.

Ohad Shamir. A variant of azuma's inequality for martingales with subgaussian tails. *arXiv preprint arXiv:1110.2392*, 2011.

Han Shen and Tianyi Chen. On penalty-based bilevel gradient descent method. In *International Conference on Machine Learning*, pages 30992–31015. PMLR, 2023.

Ankur Sinha, Pekka Malo, and Kalyanmoy Deb. A review on bilevel optimization: From classical to evolutionary approaches and applications. *IEEE transactions on evolutionary computation*, 22 (2):276–295, 2017.

Hoang Tran and Ashok Cutkosky. Momentum aggregation for private non-convex erm. *Advances in Neural Information Processing Systems*, 35:10996–11008, 2022.

Ioannis Tsaknakis, Prashant Khanduri, and Mingyi Hong. An implicit gradient-type method for linearly constrained bilevel problems. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5438–5442. IEEE, 2022.

Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

Luis Vicente, Gilles Savard, and Joaquim Júdice. Descent approaches for quadratic bilevel programming. *Journal of Optimization theory and applications*, 81(2):379–399, 1994.

Luis N Vicente and Paul H Calamai. Bilevel and multilevel programming: A bibliography review. *Journal of Global optimization*, 5(3):291–306, 1994.

Di Wang, Minwei Ye, and Jinhui Xu. Differentially private empirical risk minimization revisited: Faster and more general. *Advances in Neural Information Processing Systems*, 30, 2017.

Di Wang, Changyou Chen, and Jinhui Xu. Differentially private empirical risk minimization with non-convex loss functions. In *International Conference on Machine Learning*, pages 6526–6535. PMLR, 2019.

Douglas J White and G Anandalingam. A penalty function approach for solving bi-level linear programs. *Journal of Global Optimization*, 3:397–419, 1993.

Siyuan Xu and Minghui Zhu. Efficient gradient approximation method for constrained bilevel optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 12509–12517, 2023.

Junjie Yang, Kaiyi Ji, and Yingbin Liang. Provably faster algorithms for bilevel optimization. *Advances in Neural Information Processing Systems*, 34:13670–13682, 2021.

Yifan Yang, Peiyao Xiao, and Kaiyi Ji. Achieving o (ε-1.5) complexity in hessian/jacobian-free stochastic bilevel optimization. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pages 39491–39503, 2023.

Wei Yao, Chengming Yu, Shangzhi Zeng, and Jin Zhang. Constrained bi-level optimization: Proximal lagrangian value function approach and hessian-free algorithm, 2024.

JJ Ye and DL Zhu. Exact penalization and necessary optimality conditions for generalized bilevel programming problems. *SIAM Journal on optimization*, 7(2):481–507, 1997.

Qinzi Zhang, Hoang Tran, and Ashok Cutkosky. Private zeroth-order nonsmooth nonconvex optimization. In *The Twelfth International Conference on Learning Representations*, 2024.

DL Zhu. Optimality conditions for bilevel programming problems. *Optimization*, 33(1):9–27, 1995.

Appendix A. Differential privacy preliminaries

We recall here some well-known DP facts. The basic composition property of DP states that the (possibly adaptive) composition of (ϵ_0, δ_0)-DP- and (ϵ_1, δ_1)-DP mechanisms, is $(\epsilon_0 + \epsilon_1, \delta_0 + \delta_1)$-DP. Advanced composition improves this scaling in typical parameter regimes of interest:

Lemma 14 (Advanced composition, Dwork et al., 2010) *For $\epsilon_0 < 1$, a T-fold (possibly adaptive) composition of (ϵ_0, δ_0)-DP mechanisms is (ϵ, δ)-DP for $\epsilon = \sqrt{2T\log(1/\delta_0)}\epsilon_0 + 2T\epsilon_0^2,\ \delta = (T+1)\delta_0$.*

We remark that advanced composition is typically used when $\epsilon_0 \lesssim \sqrt{\log(1/\delta_0)/T}$, thus the accumulated privacy scales as $\epsilon \asymp \sqrt{T}\epsilon_0$.

Lemma 15 (Gaussian mechanism) *Given a function $h : \Xi^b \to \mathbb{R}^d$, the Gaussian mechanism $\mathcal{M}(h) : \Xi^b \to \mathbb{R}^d$ defined as $\mathcal{M}(h)(S) := h(S) + \mathcal{N}(\mathbf{0}, \sigma^2 I_d)$ is (ϵ, δ)-DP for $\epsilon, \delta \in (0,1)$, as long as $\sigma^2 \geq \frac{2\log(5/4\delta)(\mathcal{S}_h)^2}{\epsilon^2}$, where $\mathcal{S}_h := \sup_{S\sim S'} \|h(S) - h(S')\|$ is the L_2-sensitivity of h.*

Lemma 16 (Privacy amplification, Balle et al., 2018) *Suppose $\mathcal{M} : \Xi^b \to \mathcal{R}$ is (ϵ_0, δ_0)-DP. Then given $n \geq b$, the mechanism $\mathcal{M}' : \Xi^n \to \mathcal{R},\ \mathcal{M}'(S) := \mathcal{M}(B)$ where $B \sim \mathrm{Unif}(\Xi)^b$, is (ϵ, δ)-DP for $\epsilon = \log(1 + (1 - (1 - 1/n)^b)(e^{\epsilon_0} - 1)),\ \delta = \delta_0$.*

Privacy amplification is typically used when $\epsilon_0 \leq 1$, under which the privacy after subsampling scales as $\epsilon \asymp \frac{b\epsilon_0}{n}$ (since $e^{\epsilon_0} - 1 \asymp \epsilon_0,\ (1 - 1/n)^b \asymp \frac{b}{n}$ and $\log(1 + \frac{b}{n}\epsilon_0) \asymp \frac{b}{n}\epsilon_0$).

Appendix B. Proofs

Throughout the proof section, we abbreviate $f_i(\cdot) = f(\cdot\,; \xi_i),\ g_i(\cdot) = g(\cdot\,; \xi_i),\ F = F_S$.

B.1. Proof of Lemma 8

Note that the two sub-problems solved by Algorithm 1 are strongly-convex and admit Lipschitz components over $\mathcal{Y}$; $g(\mathbf{x}, \cdot)$ by assumption, and $f + \lambda g$ by combining this with the smoothness/Lipschitzness of f, as follows:

Lemma 17 *If $\lambda \geq \max\{\frac{2L_1^g}{\mu_g}, \frac{L_0^f}{L_0^g}\}$ then for all $\mathbf{x} \in \mathcal{X}$: $f(\mathbf{x}, \cdot) + \lambda g(\mathbf{x}, \cdot)$ is $\frac{\lambda\mu_g}{2}$ strongly-convex, and moreover for all $i \in [n]$: $f_i(\mathbf{x}, \cdot) + \lambda g_i(\mathbf{x}, \cdot)$ is $2\lambda L_0^g$-Lipschitz.*

We therefore invoke the following guarantee, which provides the optimal result for strongly-convex DP ERM via DP-Loc-GD (Algorithm 2).

Theorem 18 *Suppose that $h : \mathbb{R}^{d_y} \to \mathbb{R}$ is a μ-strongly-convex function of the form $h(\mathbf{y}) = \frac{1}{n}\sum_{i=1}^n h(\mathbf{y}, \xi_i)$ where $h(\cdot, \xi_i)$ is L-Lipschitz for all $i \in [n]$. Suppose $\arg\min h =: \mathbf{y}^* \in \mathbb{B}(\mathbf{y}_0, R_0)$ and that $n \geq \frac{LR_0^{2/\log(d_y)}}{\mu\epsilon'}$. Then there is an assignment of parameters $M = \log_2\log(\frac{\mu\epsilon' n}{L})$, $\sigma_{\mathrm{GD}}^2 = \widetilde{\mathcal{O}}(L^2/\epsilon'^2)$, $\eta_t = \frac{1}{\mu(t+1)}$, $T_{\mathrm{GD}} = n^2$, $R_m = \widetilde{\Theta}\left(\sqrt{\frac{R_{m-1}L}{\mu\epsilon' n}} + \frac{L\sqrt{d_y}}{\mu\epsilon' n}\right)$ such that running Algorithm 2 satisfies (ϵ', δ')-DP, and outputs $\mathbf{y}_{\mathrm{out}}$ such that $\|\mathbf{y}_{\mathrm{out}} - \mathbf{y}^*\| = \widetilde{\mathcal{O}}\left(\frac{L\sqrt{d_y}}{\mu n\epsilon'}\right)$ with probability at least $1 - \gamma$.*

Although the rate in Theorem 18 appears in prior works such as (Bassily et al., 2014; Feldman et al., 2020), it is typically manifested through a bound in expectation (and in terms of function value) as opposed to with high probability, required for our purpose. We therefore, for the sake of completeness, provide a self-contained proof of Theorem 18 in Appendix C.

Applied to the functions $g(\mathbf{x}_t, \cdot)$ and $f(\mathbf{x}_t, \cdot) + \lambda g(\mathbf{x}_t, \cdot)$, and invoking Lemma 17, yields the following.

Corollary 19 *If* $\lambda \geq \max\{\frac{2L_1^g}{\mu_g}, \frac{L_0^f}{L_0^g}\}$, *then* $\widetilde{\mathbf{y}}_t$ *and* $\widetilde{\mathbf{y}}_t^\lambda$ *(as appear in Algorithm 1) satisfy with probability at least* $1-\gamma$:

$$\max\left\{\|\widetilde{\mathbf{y}}_t - \mathbf{y}^*(\mathbf{x}_t)\|\,,\ \|\widetilde{\mathbf{y}}_t^\lambda - \mathbf{y}^\lambda(\mathbf{x}_t)\|\right\} = \widetilde{\mathcal{O}}\left(\frac{L_0^g\sqrt{d_y T}}{\epsilon\mu_g n}\right)\,.$$

We are now ready to prove the main proposition of this section, which we restate below:

Lemma 20 *If* $\lambda \geq \max\{\frac{2L_1^g}{\mu_g}, \frac{L_0^f}{L_0^g}, \frac{L_1^f}{L_1^g}\}$, *then the random variables* $\widetilde{\mathbf{y}}_t, \widetilde{\mathbf{y}}_t^\lambda$ *as defined in Algorithm 1 satisfy for all* $t<T$ *with probability at least* $1-\gamma$:

$$\left\|\nabla\mathcal{L}_\lambda^*(\mathbf{x}_t) - \left[\nabla_x f(\mathbf{x}_t, \widetilde{\mathbf{y}}_t^\lambda) + \lambda\left(\nabla_x g(\mathbf{x}_t, \widetilde{\mathbf{y}}_t^\lambda) - \nabla_x g(\mathbf{x}_t, \widetilde{\mathbf{y}}_t)\right)\right]\right\| \leq \beta = \widetilde{\mathcal{O}}\left(\frac{\lambda\ell\kappa\sqrt{d_y T}}{\epsilon n}\right)\,.$$

Proof [Proof of Lemma 8] As in (3), we note that by construction $\mathcal{L}_\lambda^*(\mathbf{x}) = \arg\min_{\mathbf{y}} \mathcal{L}_\lambda(\mathbf{x}, \mathbf{y})$, therefore it holds that

$$\nabla\mathcal{L}_\lambda^*(\mathbf{x}) = \nabla_x f(\mathbf{x}, \mathbf{y}^\lambda(\mathbf{x})) + \lambda\left(\nabla_x g(\mathbf{x}, \mathbf{y}^\lambda(\mathbf{x})) - \nabla_x g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\right)\,.$$

Denoting $\mathbf{g}_t = \nabla_x f(\mathbf{x}_t, \widetilde{\mathbf{y}}_t^\lambda) + \lambda\left(\nabla_x g(\mathbf{x}_t, \widetilde{\mathbf{y}}_t^\lambda) - \nabla_x g(\mathbf{x}_t, \widetilde{\mathbf{y}}_t)\right)$, by the smoothness of f and g we see that

$$\|\mathbf{g}_t - \nabla\mathcal{L}_\lambda^*(\mathbf{x}_t)\| \leq L_1^f\left\|\widetilde{\mathbf{y}}_t^\lambda - \mathbf{y}^\lambda(\mathbf{x}_t)\right\| + \lambda L_1^g\left\|\widetilde{\mathbf{y}}_t^\lambda - \mathbf{y}^\lambda(\mathbf{x}_t)\right\| + \lambda L_1^g\|\widetilde{\mathbf{y}}_t - \mathbf{y}^*(\mathbf{x}_t)\|\,.$$

Applying Corollary 19 and union bounding over T, we can further bound the above as

$$\begin{aligned}\|\mathbf{g}_t - \nabla\mathcal{L}_\lambda^*(\mathbf{x}_t)\| &= \widetilde{\mathcal{O}}\left(\frac{L_1^f L_0^g\sqrt{d_y T}}{\epsilon\mu_g n} + \frac{\lambda L_1^g L_0^g\sqrt{d_y T}}{\epsilon\mu_g n} + \frac{\lambda L_1^g L_0^g\sqrt{d_y T}}{\epsilon\mu_g n}\right) \\ &= \widetilde{\mathcal{O}}\left(\frac{\lambda L_1^g L_0^g\sqrt{d_y T}}{\epsilon\mu_g n}\right) \\ &= \widetilde{\mathcal{O}}\left(\frac{\lambda\ell\kappa\sqrt{d_y T}}{\epsilon n}\right)\,,\end{aligned}$$

where the second bound holds for $\lambda \geq \frac{L_1^f}{L_1^g}$. ■

B.2. Proof of Lemma 9

We start by providing two lemmas, both of which borrow ideas that appeared in the smoothness analysis of Chen et al. (2024).

Lemma 21 $\mathbf{y}^\lambda(\mathbf{x}):\mathbb{R}^{d_x}\to\mathbb{R}^{d_y}$ *is* $\left(\frac{4L_1^g}{\mu_g}\right)$*-Lipschitz.*

Proof [Proof of Lemma 21] Differentiating $\nabla_y\mathcal{L}_\lambda(\mathbf{x},\mathbf{y}^\lambda(\mathbf{x}))=\mathbf{0}$ with respect the first argument gives

$$\nabla^2_{xy}\mathcal{L}_\lambda(\mathbf{x},\mathbf{y}^\lambda(\mathbf{x}))+\nabla\mathbf{y}^\lambda(\mathbf{x})\cdot\nabla_{yy}\mathcal{L}_\lambda(\mathbf{x},\mathbf{y}^\lambda(\mathbf{x}))=\mathbf{0}\ ,$$

hence

$$\nabla\mathbf{y}^\lambda(\mathbf{x})=-\nabla^2_{xy}\mathcal{L}_\lambda(\mathbf{x},\mathbf{y}^\lambda(\mathbf{x}))\cdot\left[\nabla_{yy}\mathcal{L}_\lambda(\mathbf{x},\mathbf{y}^\lambda(\mathbf{x}))\right]^{-1}\ .$$

Noting that $\nabla^2_{xy}\mathcal{L}_\lambda\preceq 2\lambda L_1^g$ and $\nabla^2_{yy}\mathcal{L}_\lambda\succeq\lambda\mu_g/2$ everywhere, hence $[\nabla^2_{yy}\mathcal{L}_\lambda]^{-1}\preceq 2/\lambda\mu_g$ we get that

$$\left\|\nabla\mathbf{y}^\lambda(\mathbf{x})\right\|\le 2\lambda L_1^g\cdot\frac{2}{\lambda\mu_g}=\frac{4L_1^g}{\mu_g}\ .$$

■

Lemma 22 *For all* $\mathbf{x}\in\mathcal{X}:\ \left\|\mathbf{y}^\lambda(\mathbf{x})-\mathbf{y}^*(\mathbf{x})\right\|\le\frac{L_0^f}{\lambda\mu_g}$.

Proof [Proof of Lemma 22] First, note that by definition of $\mathbf{y}^\lambda(\mathbf{x})$ it holds that

$$\mathbf{0}=\nabla_y\mathcal{L}_\lambda(\mathbf{x},\mathbf{y}^\lambda(\mathbf{x}))=\nabla_y f(\mathbf{x},\mathbf{y}^\lambda(\mathbf{x}))+\lambda\nabla_y g(\mathbf{x},\mathbf{y}^\lambda(\mathbf{x}))\ ,$$

hence

$$\nabla_y g(\mathbf{x},\mathbf{y}^\lambda(\mathbf{x}))=-\frac{1}{\lambda}\cdot\nabla_y f(\mathbf{x},\mathbf{y}^\lambda(\mathbf{x}))\ ,$$

so in particular by the Lipschitz assumption on f we see that

$$\left\|\nabla_y g(\mathbf{x},\mathbf{y}^\lambda(\mathbf{x}))\right\|\le\frac{L_0^f}{\lambda}\ .$$

By invoking the μ-strong convexity of g we further get

$$\left\|\mathbf{y}^\lambda(\mathbf{x})-\mathbf{y}^*(\mathbf{x})\right\|\le\frac{1}{\mu}\left\|\nabla_y g(\mathbf{x},\mathbf{y}^\lambda(\mathbf{x}))-\underbrace{\nabla_y g(\mathbf{x},\mathbf{y}^*(\mathbf{x}))}_{=\mathbf{0}}\right\|\le\frac{L_0^f}{\lambda\mu}\ .$$

■

We are now ready to prove the main result of this section, restated below.

Lemma 23 $\mathcal{L}^*_{\lambda,i}$ *is* $\mathcal{O}(\ell\kappa)$*-Lipschitz independently of* λ.

Proof [Proof of Lemma 9] For all $\mathbf{x} \in \mathcal{X}$ it holds that

$$\begin{aligned}\left\|\nabla \mathcal{L}^*_{\lambda,i}(\mathbf{x})\right\| &= \left\|\nabla_x f_i(\mathbf{x}, \mathbf{y}^\lambda(\mathbf{x})) + \lambda\left[\nabla_x g_i(\mathbf{x}, \mathbf{y}^\lambda(\mathbf{x})) - \nabla_x g_i(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\right]\right\| \\ &\le \left\|\nabla_x f_i(\mathbf{x}, \mathbf{y}^\lambda(\mathbf{x}))\right\| + \lambda\left\|\nabla_x g_i(\mathbf{x}, \mathbf{y}^\lambda(\mathbf{x})) - \nabla_x g_i(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\right\| , \end{aligned} \tag{5}$$

thus we will bound each of the summands above.

For the first term, since $\mathbf{y}^\lambda$ is $\frac{4L_1^g}{\mu_g}$ Lipschitz according to Lemma 21, and f_i is L_0^f-Lipschitz by assumption, the chain rule yields the bound

$$\left\|\nabla_x f_i(\mathbf{x}, \mathbf{y}^\lambda(\mathbf{x}))\right\| \le \frac{4L_1^g L_0^f}{\mu_g} \le 4\ell\kappa . \tag{6}$$

As to the second term, since g_i is L_1^g-smooth, we use Lemma 22 and get that

$$\lambda\left\|\nabla_x g_i(\mathbf{x}, \mathbf{y}^\lambda(\mathbf{x})) - \nabla_x g_i(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\right\| \le \lambda L_1^g \left\|\mathbf{y}^\lambda(\mathbf{x}) - \mathbf{y}^*(\mathbf{x})\right\| \le \frac{L_1^g L_0^f}{\mu_g} \le \ell\kappa . \tag{7}$$

Plugging Eqs. (6) and (7) into (5) completes the proof.

■

B.3. Proof of Proposition 10

As $\nu_0, \ldots, \nu_{T-1} \overset{iid}{\sim} \mathcal{N}(0, \sigma^2 I)$, a standard Gaussian norm bound (cf. Vershynin 2018, Theorem 3.1.1) ensures that with probability at least $1-\gamma$, for all $t \in \{0, 1, \ldots, T-1\}$: $\|\nu_t\|^2 \lesssim d\sigma^2 \log(T/\gamma) \lesssim \frac{\alpha^2}{64}$. We therefore condition the rest of the proof on the highly probable event under which this uniform norm bound indeed holds. We continue by introducing some notation. We denote $\widetilde{\nabla}_t = \widetilde{\nabla} h(\mathbf{x}_t) + \nu_t$, and $\delta_t := \widetilde{\nabla}_t - \nabla h(\mathbf{x}_t)$. We further denote

$$\begin{aligned}\mathbf{x}_t^+ &:= \arg\min_{\mathbf{u}\in\mathcal{X}} \left\{\langle \nabla h(\mathbf{x}_t), \mathbf{u}\rangle + \frac{1}{2\eta}\|\mathbf{x}_t - \mathbf{u}\|^2\right\} , \\ \mathcal{G}_t &:= \frac{1}{\eta}(\mathbf{x}_t - \mathbf{x}_t^+) , \\ \rho_t &:= \frac{1}{\eta}(\mathbf{x}_t - \mathbf{x}_{t+1}) .\end{aligned}$$

Note that by construction,

$$\mathcal{G}_t = \mathcal{G}_{h,\eta}(\mathbf{x}_t) := \frac{1}{\eta}\left(\mathbf{x}_t - \mathcal{P}_{\nabla h,\eta}(\mathbf{x}_t)\right) , \quad \mathcal{P}_{\nabla h,\eta}(\mathbf{x}_t) := \arg\min_{\mathbf{u}\in\mathcal{X}}\left[\langle \nabla h(\mathbf{x}_t), \mathbf{u}\rangle + \frac{1}{2\eta}\|\mathbf{u} - \mathbf{x}_t\|^2\right] ,$$

and that $\mathcal{G}_{t_{\text{out}}}$ is precisely the quantity we aim to bound. We start by proving some useful lemmas regarding the quantities defined above.

Lemma 24 *Under the event that $\|\nu_t\|^2 \le \frac{\alpha^2}{64}$ for all t, it holds that $\|\delta_t\| \le \frac{\alpha}{4}$.*

Proof By our assumptions on $\beta, \|\nu_t\|$, we get that

$$\|\delta_t\| \leq \|\widetilde{\nabla} h(\mathbf{x}_t) - \nabla h(\mathbf{x}_t)\| + \|\nu_t\| \leq \beta + \frac{\alpha}{8} \leq \frac{\alpha}{4} \, .$$

■

Lemma 25 *It holds that* $\langle \widetilde{\nabla}_t, \rho_t \rangle \geq \|\rho_t\|^2$.

Proof By definition, $\mathbf{x}_{t+1} = \arg\min_{\mathbf{u}\in\mathcal{X}} \left\{ \langle \widetilde{\nabla}_t, \mathbf{u} \rangle + \frac{1}{2\eta} \|\mathbf{x}_t - \mathbf{u}\|^2 \right\}$. Hence, by the first-order optimality criterion, for any $\mathbf{u} \in \mathcal{X}$:

$$\left\langle \widetilde{\nabla}_t + \frac{1}{\eta}(\mathbf{x}_{t+1} - \mathbf{x}_t), \mathbf{u} - \mathbf{x}_{t+1} \right\rangle \geq 0 \, .$$

In particular, setting $\mathbf{u} = \mathbf{x}_t$ yields

$$0 \leq \left\langle \widetilde{\nabla}_t + \frac{1}{\eta}(\mathbf{x}_{t+1} - \mathbf{x}_t), \mathbf{x}_t - \mathbf{x}_{t+1} \right\rangle = \left\langle \widetilde{\nabla}_t - \rho_t, \eta\rho_t \right\rangle = \eta \left(\left\langle \widetilde{\nabla}_t, \rho_t \right\rangle - \|\rho_t\|^2 \right) \, ,$$

which proves the claim since $\eta > 0$. ■

With the lemmas above in hand, we are now ready to prove Proposition 10. Note that by construction, the algorithm returns the index t with minimal $\|\rho_t\|$. Further note that $\|\rho_t - \mathcal{G}_t\| \leq \|\delta_t\|$ by Lemma 29, thus

$$\|\mathcal{G}_t\| \leq \|\rho_t\| + \|\delta_t\| \leq \|\rho_t\| + \frac{\alpha}{4} \, , \tag{8}$$

where the last inequality is due to Lemma 24, hence it suffices to bound $\|\rho_{t_{\text{out}}}\|$ (which is the quantity measured by the output rule). To that end, by smoothness, we have for any $t \in \{0, 1, \ldots, T-2\}$:

$$\begin{aligned}
h(\mathbf{x}_{t+1}) &\leq h(\mathbf{x}_t) + \langle \nabla h(\mathbf{x}_t), \mathbf{x}_{t+1} - \mathbf{x}_t \rangle + \frac{L}{2} \|\mathbf{x}_{t+1} - \mathbf{x}_t\|^2 \\
&= h(\mathbf{x}_t) - \eta \langle \nabla h(\mathbf{x}_t), \rho_t \rangle + \frac{L\eta^2}{2} \|\rho_t\|^2 \\
&= h(\mathbf{x}_t) - \eta \left\langle \widetilde{\nabla}_t, \rho_t \right\rangle + \frac{L\eta^2}{2} \|\rho_t\|^2 + \eta \langle \delta_t, \rho_t \rangle \\
&\leq h(\mathbf{x}_t) - \eta \|\rho_t\|^2 + \frac{L\eta^2}{2} \|\rho_t\|^2 + \eta \|\delta_t\| \cdot \|\rho_t\| \, ,
\end{aligned}$$

where the last inequality followed by applying Lemma 25 and Cauchy-Schwarz. Rearranging, and recalling that $\eta = \frac{1}{2L}$, hence $1 < 2 - L\eta$ and also $\frac{1}{\eta} = 2L$, we get that

$$\|\rho_t\|^2 - 2\|\delta_t\| \cdot \|\rho_t\| \leq (2 - L\eta) \|\rho_t\|^2 - 2\|\delta_t\| \cdot \|\rho_t\| \leq \frac{2\left(h(\mathbf{x}_t) - h(\mathbf{x}_{t+1})\right)}{\eta} = 4L\left(h(\mathbf{x}_t) - h(\mathbf{x}_{t+1})\right) \, .$$

Summing over $t \in \{0, 1 \ldots, T-1\}$, using the telescoping property of the right hand side, and dividing by T gives that

$$\frac{1}{T} \sum_{t=0}^{T-1} \|\rho_t\| \left(\|\rho_t\| - 2\|\delta_t\|\right) \leq \frac{4L\left(h(\mathbf{x}_0) - \inf h\right)}{T} \, . \tag{9}$$

Note that if for some $t \in \{0, 1, \ldots, T-1\}: \ \|\rho_t\| \leq \frac{3\alpha}{4}$ then we have proved our desired claim by (8) and the fact that $\|\rho_{t_{\text{out}}}\| = \min_t \|\rho_t\|$ by definition. On the other hand, assuming that $\|\rho_t\| > \frac{3\alpha}{4}$ for all t, invoking Lemma 24, we see that $\|\rho_t\| - 2\|\delta_t\| \geq \|\rho_t\| - \frac{\alpha}{2} \geq \frac{1}{3}\|\rho_t\|$, which implies $\|\rho_t\|(\|\rho_t\| - 2\|\delta_t\|) \geq \frac{1}{3}\|\rho_t\|^2$. Combining this with (9) yields

$$\|\rho_{t_{\text{out}}}\|^2 = \min_{t\in\{0,1,\ldots,T-1\}} \|\rho_t\|^2 \leq \frac{1}{T}\sum_{t=0}^{T-1}\|\rho_t\|^2 \leq \frac{12L\left(h(\mathbf{x}_0) - \inf h\right)}{T} ,$$

and the right side is bounded by $\frac{9\alpha^2}{16}$ for $T = \mathcal{O}\left(\frac{L(h(\mathbf{x}_0) - \inf h)}{\alpha^2}\right)$, finishing the proof by (8).

B.4. Proof of Theorem 4

We start by proving the privacy guarantee. Since $\mathcal{L}^*_{\lambda,i}$ is $\mathcal{O}(\ell\kappa)$-Lipschitz by Lemma 9, the sensitivity of $\nabla_x f(\mathbf{x}_t, \widetilde{\mathbf{y}}^\lambda_t) + \lambda\left(\nabla_x g(\mathbf{x}_t, \widetilde{\mathbf{y}}^\lambda_t) - \nabla_x g(\mathbf{x}_t, \widetilde{\mathbf{y}}_t)\right)$ is at most $\mathcal{O}(\ell\kappa)$. Hence, by setting $\sigma^2 = C\frac{\ell^2\kappa^2\log(T/\delta)T}{\epsilon^2 n^2}$ for a sufficiently large absolute constant $C > 0$, $\widetilde{\mathbf{g}}_t$ is $(\frac{\epsilon}{\sqrt{18T}}, \frac{\delta}{3(T+1)})$-DP. By basic composition, since each iteration also runs $(\frac{\epsilon}{\sqrt{18T}}, \frac{\delta}{3(T+1)})$-DP-Loc-GD twice, we see that each iteration of the algorithm is $(3 \cdot \frac{\epsilon}{\sqrt{18T}}, 3 \cdot \frac{\delta}{3(T+1)}) = (\frac{\epsilon}{\sqrt{2T}}, \frac{\delta}{(T+1)})$-DP. Noting that under our parameter assignment $\frac{\epsilon}{\sqrt{T}} \ll 1$, by advanced composition we get that throughout T iterations, the algorithm is overall (ϵ, δ)-DP as claimed.

We turn to analyze the utility of the algorithm. It holds that

$$\begin{aligned}
\|\mathcal{G}_{F,\eta}(\mathbf{x}_{t_{\text{out}}})\| &\leq \left\|\mathcal{G}_{F,\eta}(\mathbf{x}_{t_{\text{out}}}) - \mathcal{G}_{\mathcal{L}^*_\lambda,\eta}(\mathbf{x})\right\| + \left\|\mathcal{G}_{\mathcal{L}^*_\lambda,\eta}(\mathbf{x}_{t_{\text{out}}})\right\| \\
&\leq \|\nabla F(\mathbf{x}_{t_{\text{out}}}) - \nabla\mathcal{L}^*_\lambda(\mathbf{x})\| + \|\mathcal{G}_{\mathcal{L}^*_\lambda,\eta}(\mathbf{x}_{t_{\text{out}}})\| \\
&\lesssim \frac{\ell\kappa^3}{\lambda} + \|\mathcal{G}_{\mathcal{L}^*_\lambda,\eta}(\mathbf{x}_{t_{\text{out}}})\| \\
&\leq \frac{\alpha}{2} + \|\mathcal{G}_{\mathcal{L}^*_\lambda,\eta}(\mathbf{x}_{t_{\text{out}}})\| ,
\end{aligned} \tag{10}$$

where the second inequality is due to Lemma 29, the third due to Lemma 7.b, and the last by our assignment of λ. It therefore remains to bound $\|\mathcal{G}_{\mathcal{L}^*_\lambda,\eta}(\mathbf{x}_{t_{\text{out}}})\|$.

To that end, applying Proposition 10 to the function $h = \mathcal{L}^*_\lambda$, under our assignment of T — which accounts for the smoothness and initial sub-optimality bounds ensured by Lemma 7 — we get that $\|\mathcal{G}_{\mathcal{L}^*_\lambda,\eta}(\mathbf{x}_{t_{\text{out}}})\| \leq \frac{\alpha}{2}$, for α as small as

$$\alpha = \Theta\left(\max\{\beta, \sigma\sqrt{d_x\log(T/\gamma)}\}\right) \tag{11}$$

By Lemma 8, it holds that

$$\beta = \widetilde{\mathcal{O}}\left(\frac{\lambda\ell\kappa\sqrt{d_y T}}{\epsilon n}\right) = \widetilde{\mathcal{O}}\left(\frac{\ell^{5/2}\kappa^{11/2}\Delta_F^{1/2}\sqrt{d_y}}{\alpha^2\epsilon n}\right) \tag{12}$$

and we also have

$$\sigma\sqrt{d_x\log(T/\gamma)} = \widetilde{\mathcal{O}}\left(\frac{\ell^{3/2}\kappa^{5/2}\Delta_F^{1/2}\sqrt{d_x}}{\alpha\epsilon n}\right) . \tag{13}$$

Plugging (12) and (13) back into (11), and solving for α, completes the proof.

B.5. Proof of Theorem 11

Throughout this section, we abbreviate $b = b_{\text{out}}$. We will need the following lemma, which is the mini-batch analogue of Lemma 8 from the full-batch setting.

Lemma 26 *If $\lambda \geq \max\{\frac{2L_1^g}{\mu_g}, \frac{L_0^f}{L_0^g}, \frac{L_1^f}{L_1^g}\}$, then there is $\beta_b = \widetilde{\mathcal{O}}\left(\frac{\lambda\ell\kappa\sqrt{d_y T}}{\epsilon n} + \frac{\ell\kappa}{b}\right)$ such that with probability at least $1-\gamma/2$, $\mathbf{g}_t^B := \nabla_x f(\mathbf{x}_t, \widetilde{\mathbf{y}}_t^\lambda; B_t) + \lambda\left(\nabla_x g(\mathbf{x}_t, \widetilde{\mathbf{y}}_t^\lambda; B_t) - \nabla_x g(\mathbf{x}_t, \widetilde{\mathbf{y}}_t; B_t)\right)$ satisfies for all $t \in \{0, \ldots, T-1\}$: $\|\nabla\mathcal{L}_\lambda^*(\mathbf{x}_t) - \mathbf{g}_t^B\| \leq \beta_b$.*

Proof [Proof of Lemma 26] It holds that

$$\left\|\nabla\mathcal{L}_\lambda^*(\mathbf{x}_t) - \mathbf{g}_t^B\right\| \leq \left\|\nabla\mathcal{L}_\lambda^*(\mathbf{x}_t) - \mathbb{E}[\mathbf{g}_t^B]\right\| + \left\|\mathbf{g}_t^B - \mathbb{E}[\mathbf{g}_t^B]\right\| .$$

To bound the first summand, note that $\mathbb{E}[\mathbf{g}_t^B] = \nabla_x f(\mathbf{x}_t, \widetilde{\mathbf{y}}_t^\lambda) + \lambda\left(\nabla_x g(\mathbf{x}_t, \widetilde{\mathbf{y}}_t^\lambda) - \nabla_x g(\mathbf{x}_t, \widetilde{\mathbf{y}}_t)\right)$, and therefore with probability at least $1-\gamma/4$:

$$\left\|\nabla\mathcal{L}_\lambda^*(\mathbf{x}_t) - \mathbb{E}[\mathbf{g}_t^B]\right\| = \mathcal{O}\left(\frac{\lambda L_1^g L_0^g\sqrt{d_y T}}{\epsilon\mu_g n}\right) = \mathcal{O}\left(\frac{\lambda\ell\kappa\sqrt{d_y T}}{\epsilon n}\right) ,$$

following the same proof as Lemma 8 in Section B.1, by replacing Theorem 18 by the mini-batch Theorem 28 (whose guarantee holds regardless of the inner batch size) .

To bound the second summand, note that $\|\nabla_x f(\mathbf{x}_t, \widetilde{\mathbf{y}}_t^\lambda; \xi) + \lambda(\nabla_x g(\mathbf{x}_t, \widetilde{\mathbf{y}}_t^\lambda; \xi) - \nabla_x g(\mathbf{x}_t, \widetilde{\mathbf{y}}_t; \xi))\| \leq M = \mathcal{O}(\ell\kappa)$ for every $\xi \in \Xi$, by Lemma 9. Hence, $\mathbf{g}_t^B$ is the average of b independent vectors bounded by M, all with the same mean, and therefore a standard concentration bound (cf. Jin et al. 2019) ensures that $\|\mathbf{g}_t^B - \mathbb{E}[\mathbf{g}_t^B]\| = \widetilde{\mathcal{O}}(M/b)$ with probability at least $1-\gamma/4$, which completes the proof. ■

We can now prove the main mini-batch result:

Proof [Proof of Theorem 11] We start by proving the privacy guarantee. Since $\mathcal{L}_{\lambda,i}^*$ is $\mathcal{O}(\ell\kappa)$-Lipschitz by Lemma 9, the sensitivity of $\nabla_x f(\mathbf{x}_t, \widetilde{\mathbf{y}}_t^\lambda; B_t) + \lambda\left(\nabla_x g(\mathbf{x}_t, \widetilde{\mathbf{y}}_t^\lambda; B_t) - \nabla_x g(\mathbf{x}_t, \widetilde{\mathbf{y}}_t; B_t)\right)$ is at most $\mathcal{O}(\ell\kappa)$. Accordingly, the "unamplified" Gaussian mechanism (Lemma 15) ensures $(\tilde{\epsilon}, \tilde{\delta})$-DP for $\tilde{\epsilon} = \widetilde{\Theta}\left(\frac{\ell\kappa}{b\sigma}\right)$, and hence is amplified (Lemma 16) to (ϵ_0, δ_0)-DP for $\epsilon_0 = \widetilde{\Theta}\left(\frac{\ell\kappa}{b\sigma}\cdot\frac{b}{n}\right) = \frac{\epsilon}{\sqrt{18T}}$, the last holding for sufficiently large $\sigma^2 = \widetilde{\Theta}\left(\frac{\ell^2\kappa^2 T}{\epsilon^2 n^2}\right)$, and for $\delta_0 = \frac{\delta}{3(T+1)}$. Therefore, basic composition shows that each iteration of the algorithm is $(3\cdot\frac{\epsilon}{\sqrt{18T}}, 3\frac{\delta}{3(T+1)}) = (\frac{\epsilon}{\sqrt{2T}}, \frac{\delta}{T+1})$-DP. Since $\epsilon/\sqrt{T} \ll 1$ under our parameter assignment, advanced composition over the T iterations yields the (ϵ, δ)-DP guarantee.

We turn to analyze the utility of the algorithm. It holds that

$$\begin{aligned}
\|\mathcal{G}_{F,\eta}(\mathbf{x}_{t_{\text{out}}})\| &\leq \left\|\mathcal{G}_{F,\eta}(\mathbf{x}_{t_{\text{out}}}) - \mathcal{G}_{\mathcal{L}_\lambda^*,\eta}(\mathbf{x})\right\| + \left\|\mathcal{G}_{\mathcal{L}_\lambda^*,\eta}(\mathbf{x}_{t_{\text{out}}})\right\| \\
&\leq \|\nabla F(\mathbf{x}_{t_{\text{out}}}) - \nabla\mathcal{L}_\lambda^*(\mathbf{x})\| + \|\mathcal{G}_{\mathcal{L}_\lambda^*,\eta}(\mathbf{x}_{t_{\text{out}}})\| \\
&\lesssim \frac{\ell\kappa^3}{\lambda} + \|\mathcal{G}_{\mathcal{L}_\lambda^*,\eta}(\mathbf{x}_{t_{\text{out}}})\| \\
&\leq \frac{\alpha}{2} + \|\mathcal{G}_{\mathcal{L}_\lambda^*,\eta}(\mathbf{x}_{t_{\text{out}}})\| , \qquad (14)
\end{aligned}$$

where the second inequality is due to Lemma 29, the third due to Lemma 7, and the last by our assignment of λ. It therefore remains to bound $\|\mathcal{G}_{\mathcal{L}_\lambda^*,\eta}(\mathbf{x}_{t_{\text{out}}})\|$.

To that end, applying Proposition 10 to the function $h = \mathcal{L}_\lambda^*$, under our assignment of T — which accounts for the smoothness and initial sub-optimality bounds ensured by Lemma 7 — we get that $\|\mathcal{G}_{\mathcal{L}_\lambda^*,\eta}(\mathbf{x}_{t_{\text{out}}})\| \leq \frac{\alpha}{2}$, for α as small as

$$\alpha = \Theta\left(\max\{\beta_b, \sigma\sqrt{d_x \log(T/\gamma)}\}\right) \tag{15}$$

By Lemma 26, it holds that

$$\beta_b = \widetilde{\mathcal{O}}\left(\frac{\lambda\ell\kappa\sqrt{d_y T}}{\epsilon n} + \frac{\ell\kappa}{b}\right) = \widetilde{\mathcal{O}}\left(\frac{\ell^{5/2}\kappa^{11/2}\Delta_F^{1/2}\sqrt{d_y}}{\alpha^2 \epsilon n} + \frac{\ell\kappa}{b}\right) , \tag{16}$$

and we also have

$$\sigma\sqrt{d_x \log(T/\gamma)} = \widetilde{\mathcal{O}}\left(\frac{\ell^{3/2}\kappa^{5/2}\Delta_F^{1/2}\sqrt{d_x}}{\alpha\epsilon n}\right) . \tag{17}$$

Plugging (16) and (17) back into (15), and solving for α, completes the proof.

■

B.6. Proof of Theorem 13

The proof is based on the following lemma that establishes uniform convergence of hypergradients.

Lemma 27 *Suppose $\overline{\mathcal{X}} \subset \mathbb{R}^d_x$ is a subset of bounded diameter $\operatorname{diam}(\overline{\mathcal{X}}) \leq D$, and that $S \sim \mathcal{P}^n$. Then with probability at least $1-\gamma$ for all $\mathbf{x} \in \overline{\mathcal{X}}$:*

$$\|\nabla F_{\mathcal{P}}(\mathbf{x}) - \nabla F_S(\mathbf{x})\| = \widetilde{\mathcal{O}}\Bigg(\frac{(\ell^2\kappa^2 + \ell\kappa)\sqrt{d_x \log(D/\gamma)}}{\sqrt{n}} + \frac{(\ell^2\kappa^3 + \ell\kappa^2)\sqrt{\log(1/\gamma)}}{\sqrt{n}} + \frac{(\ell^{3/2}\kappa^{5/2} + \ell^{1/2}\kappa^{3/2})\log^{1/4}(1/\gamma)}{n^{1/4}}\Bigg) .$$

Proof [Proof of Lemma 27] We start by setting up notation. Given a dataset $S \sim \mathcal{P}^n$, recall that we denote by $F_{\mathcal{P}}/F_S$ the stochastic/empirical hyperobjectives, by $f_{\mathcal{P}}/f_S$ and $g_{\mathcal{P}}/g_S$ the stochastic/empirical upper- and lower-level objectives. We denote by $\mathbf{y}^*_{\mathcal{P}}(\mathbf{x}) = \arg\min_{\mathbf{y}} g_{\mathcal{P}}(\mathbf{x},\mathbf{y})$ and $\mathbf{y}^*_S(\mathbf{x}) = \arg\min_{\mathbf{y}} g_S(\mathbf{x},\mathbf{y})$. It holds that

$$\nabla F_{\mathcal{P}}(\mathbf{x}) = \nabla_x f_{\mathcal{P}}(\mathbf{x}, \mathbf{y}^*_{\mathcal{P}}(\mathbf{x})) + (\frac{d\mathbf{y}^*_{\mathcal{P}}(\mathbf{x})}{d\mathbf{x}})^\top \nabla_y f_{\mathcal{P}}(\mathbf{x}, \mathbf{y}^*_{\mathcal{P}}(\mathbf{x}))$$
$$\nabla F_S(\mathbf{x}) = \nabla_x f_S(\mathbf{x}, \mathbf{y}^*_S(\mathbf{x})) + (\frac{d\mathbf{y}^*_S(\mathbf{x})}{d\mathbf{x}})^\top \nabla_y f_S(\mathbf{x}, \mathbf{y}^*_S(\mathbf{x})) .$$

Thus,

$$\begin{aligned}\|\nabla F_{\mathcal{P}}(\mathbf{x})-\nabla F_S(\mathbf{x})\| &\leq \|\nabla_x f_{\mathcal{P}}(\mathbf{x},\mathbf{y}^*_{\mathcal{P}}(\mathbf{x}))-\nabla_x f_S(\mathbf{x},\mathbf{y}^*_S(\mathbf{x}))\| \\ &\quad+\|(\frac{d\mathbf{y}^*_{\mathcal{P}}(\mathbf{x})}{d\mathbf{x}})^\top\nabla_y f_{\mathcal{P}}(\mathbf{x},\mathbf{y}^*_{\mathcal{P}}(\mathbf{x}))-(\frac{d\mathbf{y}^*_S(\mathbf{x})}{d\mathbf{x}})^\top\nabla_y f_S(\mathbf{x},\mathbf{y}^*_S(\mathbf{x}))\| \\ &\leq \underbrace{\|\nabla_x f_{\mathcal{P}}(\mathbf{x},\mathbf{y}^*_{\mathcal{P}}(\mathbf{x}))-\nabla_x f_S(\mathbf{x},\mathbf{y}^*_S(\mathbf{x}))\|}_{(1)} \\ &\quad+\underbrace{\|\frac{d\mathbf{y}^*_{\mathcal{P}}(\mathbf{x})}{d\mathbf{x}}\|\cdot\|\nabla_y f_{\mathcal{P}}(\mathbf{x},\mathbf{y}^*_{\mathcal{P}}(\mathbf{x}))-\nabla_y f_S(\mathbf{x},\mathbf{y}^*_S(\mathbf{x}))\|}_{(2)} \\ &\quad+\underbrace{\|\nabla_y f_S(\mathbf{x},\mathbf{y}^*_S(\mathbf{x}))\|\cdot\|\frac{d\mathbf{y}^*_{\mathcal{P}}(\mathbf{x})}{d\mathbf{x}}-\frac{d\mathbf{y}^*_S(\mathbf{x})}{d\mathbf{x}}\|}_{(3)},\end{aligned}$$

where the last inequality used the fact that $|ab-cd|=|ab-ad+ad-cd|\leq|a||b-d|+|d||a-c|$. We turn to bound each summand with high probability, which immediately results in Lemma 27 by summing and union bounding.

Bound (1)**.** By smoothness, it holds that

$$\|\nabla_x f_{\mathcal{P}}(\mathbf{x},\mathbf{y}^*_{\mathcal{P}}(\mathbf{x}))-\nabla_x f_S(\mathbf{x},\mathbf{y}^*_S(\mathbf{x}))\|\leq\|\nabla_x f_{\mathcal{P}}(\mathbf{x},\mathbf{y}^*_{\mathcal{P}}(\mathbf{x}))-\nabla_x f_S(\mathbf{x},\mathbf{y}^*_{\mathcal{P}}(\mathbf{x}))\|+L_1^f\|\mathbf{y}^*_{\mathcal{P}}(\mathbf{x})-\mathbf{y}^*_S(\mathbf{x})\|.$$

To bound the first term, we note that the partial derivatives are bounded by L_0^f by Lipschitzness. Hence, we can apply a uniform convergence bound due to Mei et al. (2018, Theorem 1) and get with probability at least $1-\gamma/9$:

$$\|\nabla_x f_{\mathcal{P}}(\mathbf{x},\mathbf{y}^*_{\mathcal{P}}(\mathbf{x}))-\nabla_x f_S(\mathbf{x},\mathbf{y}^*_{\mathcal{P}}(\mathbf{x}))\|=\widetilde{\mathcal{O}}\left(L_0^f\sqrt{d_x\log(D/\gamma)/n}\right).$$

To bound the second term, we use strong-convexity to get

$$L_1^f\|\mathbf{y}^*_{\mathcal{P}}(\mathbf{x})-\mathbf{y}^*_S(\mathbf{x})\|\leq\frac{L_1^f}{\sqrt{\mu_g}}\sqrt{g_{\mathcal{P}}(\mathbf{x},\mathbf{y}^*_{\mathcal{P}}(\mathbf{x}))-g_{\mathcal{P}}(\mathbf{x},\mathbf{y}^*_S(\mathbf{x}))}.$$

Now, we note that since $\mathbf{y}^*_{\mathcal{P}}(\mathbf{x}),\mathbf{y}^*_S(\mathbf{x})$ are the stochastic/empirical minima of the strongly-convex stochastic optimization problem given by $g_{\mathcal{P}}$, a result due to Feldman and Vondrak (2019) bounds the generalization gap with probability at least $1-\gamma/9$ by

$$g_{\mathcal{P}}(\mathbf{x},\mathbf{y}^*_{\mathcal{P}}(\mathbf{x}))-g_{\mathcal{P}}(\mathbf{x},\mathbf{y}^*_S(\mathbf{x}))=\widetilde{\mathcal{O}}\left(\frac{(L_0^g)^2\log(1/\gamma)}{\mu_g n}+\frac{\sqrt{\log(1/\gamma)}}{\sqrt{n}}\right).$$

Overall, by union bounding and simplifying the condition dependent constants, we get

$$(1)=\widetilde{\mathcal{O}}\left(\frac{\ell\sqrt{d_x\log(D/\gamma)}}{\sqrt{n}}+\frac{\ell\kappa\sqrt{\log(1/\gamma)}}{\sqrt{n}}+\frac{\sqrt{\ell\kappa}\log^{1/4}(1/\gamma)}{n^{1/4}}\right).$$

Bound (2)**.** On one hand, by the implicit function theorem,

$$\|\frac{d\mathbf{y}^*_{\mathcal{P}}(\mathbf{x})}{d\mathbf{x}}\| = \|\nabla^2_{xy} g_{\mathcal{P}}(\mathbf{x}, \mathbf{y}^*_{\mathcal{P}}(\mathbf{x}))[\nabla^2_{yy} g_{\mathcal{P}}(\mathbf{x}, \mathbf{y}^*_{\mathcal{P}}(\mathbf{x}))]^{-1}\| \leq L_1^g/\mu_g \leq \kappa \,.$$

Moreover, repeating the argument used to bound (1), we get

$$\begin{aligned}\|\nabla_y f_{\mathcal{P}}(\mathbf{x}, \mathbf{y}^*_{\mathcal{P}}(\mathbf{x})) - \nabla_y f_S(\mathbf{x}, \mathbf{y}^*_S(\mathbf{x}))\| &\leq \|\nabla_y f_{\mathcal{P}}(\mathbf{x}, \mathbf{y}^*_{\mathcal{P}}(\mathbf{x})) - \nabla_y f_S(\mathbf{x}, \mathbf{y}^*_{\mathcal{P}}(\mathbf{x}))\| + L_1^f \|\mathbf{y}^*_{\mathcal{P}}(\mathbf{x}) - \mathbf{y}^*_S(\mathbf{x})\| \\ &= \widetilde{\mathcal{O}}\left(\frac{\ell\sqrt{d_x \log(D/\gamma)}}{\sqrt{n}} + \frac{\ell\kappa\sqrt{\log(1/\gamma)}}{\sqrt{n}} + \frac{\sqrt{\ell\kappa}\log^{1/4}(1/\gamma)}{n^{1/4}}\right) \,.\end{aligned}$$

Overall, we get

$$(2) = \widetilde{\mathcal{O}}\left(\frac{\ell\kappa\sqrt{d_x \log(D/\gamma)}}{\sqrt{n}} + \frac{\ell\kappa^2\sqrt{\log(1/\gamma)}}{\sqrt{n}} + \frac{\ell^{1/2}\kappa^{3/2}\log^{1/4}(1/\gamma)}{n^{1/4}}\right) \,.$$

Bound (3)**.** We first note that $\|\nabla_y f_S(\mathbf{x}, \mathbf{y}^*_S(\mathbf{x}))\| \leq L_0^f \leq \ell$. Moreover, due to the implicit function theorem, the basic estimate $|ab - cd| \leq |a||b-d| + |d||a-c|$, the fact $\|A^{-1} - B^{-1}\| \leq \frac{1}{\lambda_{\min}(A)\lambda_{\min}(B)}\|A - B\|$, and second-order smoothness of g, we get

$$\begin{aligned}\|\frac{d\mathbf{y}^*_{\mathcal{P}}(\mathbf{x})}{d\mathbf{x}} - \frac{d\mathbf{y}^*_S(\mathbf{x})}{d\mathbf{x}}\| &= \|\nabla^2_{xy} g_{\mathcal{P}}(\mathbf{x}, \mathbf{y}^*_{\mathcal{P}}(\mathbf{x}))[\nabla^2_{yy} g_{\mathcal{P}}(\mathbf{x}, \mathbf{y}^*_{\mathcal{P}}(\mathbf{x}))]^{-1} - \nabla^2_{xy} g_S(\mathbf{x}, \mathbf{y}^*_S(\mathbf{x}))[\nabla^2_{yy} g_S(\mathbf{x}, \mathbf{y}^*_S(\mathbf{x}))]^{-1}\| \\ &\leq \|\nabla^2_{xy} g_{\mathcal{P}}(\mathbf{x}, \mathbf{y}^*_{\mathcal{P}}(\mathbf{x}))\| \cdot \|[\nabla^2_{yy} g_{\mathcal{P}}(\mathbf{x}, \mathbf{y}^*_{\mathcal{P}}(\mathbf{x}))]^{-1} - [\nabla^2_{yy} g_S(\mathbf{x}, \mathbf{y}^*_S(\mathbf{x}))]^{-1}\| \\ &\quad + \|[\nabla^2_{yy} g_S(\mathbf{x}, \mathbf{y}^*_S(\mathbf{x}))]^{-1}\| \cdot \|\nabla^2_{xy} g_{\mathcal{P}}(\mathbf{x}, \mathbf{y}^*_{\mathcal{P}}(\mathbf{x})) - \nabla^2_{xy} g_S(\mathbf{x}, \mathbf{y}^*_S(\mathbf{x}))\| \\ &\leq (\frac{L_1^g}{\mu_g^2} + \frac{1}{\mu_g})\|\nabla^2 g_{\mathcal{P}}(\mathbf{x}, \mathbf{y}^*_{\mathcal{P}}(\mathbf{x})) - \nabla^2 g_S(\mathbf{x}, \mathbf{y}^*_S(\mathbf{x}))\| \\ &\leq (\frac{L_1^g}{\mu_g^2} + \frac{1}{\mu_g})(\|\nabla^2 g_{\mathcal{P}}(\mathbf{x}, \mathbf{y}^*_{\mathcal{P}}(\mathbf{x})) - \nabla^2 g_S(\mathbf{x}, \mathbf{y}^*_{\mathcal{P}}(\mathbf{x}))\| + L_2^g\|\mathbf{y}^*_{\mathcal{P}}(\mathbf{x}) - \mathbf{y}^*_S(\mathbf{x})\|) \,.\end{aligned}$$

Applying a Hessian uniform convergence bound due to Mei et al. (2018, Theorem 1.b) gives with probability at least $1 - \gamma/9$:

$$\|\nabla^2 g_{\mathcal{P}}(\mathbf{x}, \mathbf{y}^*_{\mathcal{P}}(\mathbf{x})) - \nabla^2 g_S(\mathbf{x}, \mathbf{y}^*_{\mathcal{P}}(\mathbf{x}))\| = \widetilde{\mathcal{O}}\left(L_1^g\sqrt{d_x \log(D/\gamma)/n}\right) \,.$$

The same argument that appeared in bounding (1) and (2) gives

$$L_2^g\|\mathbf{y}^*_{\mathcal{P}}(\mathbf{x}) - \mathbf{y}^*_S(\mathbf{x})\| = \widetilde{\mathcal{O}}\left(\frac{\ell\kappa\sqrt{\log(1/\gamma)}}{\sqrt{n}} + \frac{\sqrt{\ell\kappa}\log^{1/4}(1/\gamma)}{n^{1/4}}\right) \,.$$

Combining all the pieces and simplifying the condition dependent constants, we get overall

$$(3) = \widetilde{\mathcal{O}}\left(\frac{(\ell^2\kappa^2 + \ell\kappa)\sqrt{d_x \log(D/\gamma)}}{\sqrt{n}} + \frac{(\ell^2\kappa^3 + \ell\kappa^2)\sqrt{\log(1/\gamma)}}{\sqrt{n}} + \frac{(\ell^{3/2}\kappa^{5/2} + \ell^{1/2}\kappa^{3/2})\log^{1/4}(1/\gamma)}{n^{1/4}}\right) \,.$$

■

With Lemma 27 in hand, we turn to prove Theorem 13.

Proof [Proof of Theorem 13] Using the fact that the gradient mapping is non-expansive (Lemma 29), we see that with probability at least $1-\gamma/2$ the following holds:

$$\begin{aligned}\|\mathcal{G}_{F_{\mathcal{P}},\eta}(\mathbf{x}_{\text{out}})\| &\leq \|\mathcal{G}_{F_S,\eta}(\mathbf{x}_{\text{out}})\| + \|\mathcal{G}_{F_{\mathcal{P}},\eta}(\mathbf{x}_{\text{out}}) - \mathcal{G}_{F_S,\eta}(\mathbf{x}_{\text{out}})\| \\ &\leq \|\mathcal{G}_{F_S,\eta}(\mathbf{x}_{\text{out}})\| + \|\nabla F_{\mathcal{P}}(\mathbf{x}_{\text{out}}) - \nabla F_S(\mathbf{x}_{\text{out}})\| \\ &= \|\mathcal{G}_{F_S,\eta}(\mathbf{x}_{\text{out}})\| + \widetilde{\mathcal{O}}\Bigg(\frac{(\ell^2\kappa^2+\ell\kappa)\sqrt{d_x\log(1/\gamma)}}{\sqrt{n}} + \frac{(\ell^2\kappa^3+\ell\kappa^2)\sqrt{\log(1/\gamma)}}{\sqrt{n}} \\ &\qquad + \frac{(\ell^{3/2}\kappa^{5/2}+\ell^{1/2}\kappa^{3/2})\log^{1/4}(1/\gamma)}{n^{1/4}}\Bigg),\end{aligned}$$

where the alst inequality is by Lemma 27 with a domain bound $\|\mathbf{x}_{\text{out}} - \mathbf{x}_0\| \leq D$ for some sufficiently large D which is polynomial in all problem parameters (therefore only affecting log terms). Theorem 13 follows from applying Theorems 4 and 11 to bound $\|\mathcal{G}_{F_S,\eta}(\mathbf{x}_{\text{out}})\|$ and simplifying. ■

Appendix C. Optimal DP algorithm for strongly-convex objectives

The goal of this appendix is to provide a self contained analysis of a DP algorithm for strongly-convex optimization which achieves the optimal convergence rate with a a high probability guarantee. Any such algorithm can be used as the inner loop in our DP bilevel algorithm.

In particular, we analyze *localized* DP (S)GD. Although it would have been more natural to apply DP-SGD, this seems (at least according to our analysis) to yield an inferior rate with respect to the required high probability guarantee.[7] Indeed, for DP-(S)GD, previous works (such as Bassily et al. 2014) typically provide bounds in expectation, and then convert them into high-probability bounds via a black-box reduction, which applies several runs and selects the best run via the private noisy-min (via Laplace mechanism). The additional error incurred by this selection is of order $\frac{1}{n}$, which translates to $\frac{1}{\sqrt{n}}$ in terms of distance to the optimum, thus spoiling the fast rate of $\frac{1}{n}$ otherwise achieved in expectation for strongly-convex objectives. We therefore resort to localization (Feldman et al., 2020): by running projected-(S)GD over balls with shrinking radii, applying martingale concentration bounds enables us to show that the distance to optimum shrinks as $R_{m+1} \lesssim \sqrt{\frac{R_m}{n}} + \frac{1}{n}$, and thus with negligible overhead we eventually recover the optimal fast rate $R_M \lesssim \frac{1}{n}$ with high probability. Our analysis differs than previous localization analyses, as it does not require adapting the noise-level and step sizes throughout the rounds.

We prove the following (which easily implies also the full-batch Theorem 18):

Theorem 28 *Suppose that* $h:\mathbb{R}^{d_y}\times\Xi\to\mathbb{R}$ *is a* μ*-strongly-convex function of the form* $h(\mathbf{y}) = \frac{1}{n}\sum_{i=1}^n h(\mathbf{y},\xi_i)$ *where* $h(\cdot,\xi_i)$ *is L-Lipschitz for all* $i\in[n]$. *Suppose* $\arg\min h =: \mathbf{y}^* \in \mathbb{B}(\mathbf{y}_0, R_0)$, *and that* $n \geq \frac{LR_0^{\frac{2}{\log(d_y)}}}{\mu\epsilon'}$. *Then given any batch size* $b\in\{1,\dots,n\}$, *there is an assignment of parameters* $M = \log_2\log(\frac{\mu\epsilon' n}{L})$, $\sigma^2_{\text{SGD}} = \widetilde{\mathcal{O}}\left(\frac{L^2}{\epsilon'^2}\right)$, $\eta_t = \frac{1}{\mu(t+1)}$, $T_{\text{SGD}} = n^2$, $R_m =$

7. Even if we would have sought only expectation bounds with respect to the hyperobjective, the high probability bound with respect to the inner problem is key to being able to argue about the gradient inexactness thereafter.

$\widetilde{\Theta}\left(\sqrt{\frac{R_{m-1}L}{\mu\epsilon' n}} + \frac{L\sqrt{d_y}}{\mu\epsilon' n}\right)$ *such that running Algorithm 4 satisfies* (ϵ, δ)*-DP, and outputs* $\mathbf{y}_{\mathrm{out}}$ *such that* $\|\mathbf{y}_{\mathrm{out}} - \mathbf{y}^*\| = \widetilde{\mathcal{O}}\left(\frac{L\sqrt{d_y}}{\mu n\epsilon}\right)$ *with probability at least* $1-\gamma$*.*

Proof [Proof of Theorem 28] We start by proving the privacy guarantee. By the Lipschitz assumption, the sensitivity of $\nabla h(\cdot; B_t)$ is at most $\frac{2L}{b}$, thus the "unamplified" Gaussian mechanism (Lemma 15) ensures $(\tilde{\epsilon}, \tilde{\delta})$-DP with $\tilde{\epsilon} = \widetilde{\Theta}\left(\frac{L}{b\sigma}\right) = \widetilde{\Theta}\left(\frac{\epsilon'}{b}\right)$, and hence is amplified (Lemma 16) to (ϵ_0, δ_0)-DP for $\epsilon_0 = \widetilde{\Theta}\left(\frac{\epsilon'}{b}\cdot\frac{b}{n}\right) = \widetilde{\Theta}\left(\frac{\epsilon'}{n}\right) = \widetilde{\Theta}\left(\frac{\epsilon'}{\sqrt{T}}\right)$. Advanced composition (Lemma 14) therefore ensures that the overall algorithm is (ϵ', δ')-DP (note that this uses the fact that $M = \widetilde{\mathcal{O}}(1)$).

We turn to prove the utility of the algorithm. We first show that for all m :

$$\Pr\left[\mathbf{y}^* \in \mathbb{B}(\mathbf{y}_0^m, R_m)\right] \geq 1 - \frac{m\gamma}{M}\,. \tag{18}$$

We prove this by induction over m. The base case $m=0$ follows by the assumption $\mathbf{y}^* \in \mathbb{B}(\mathbf{y}_0, R_0)$. Denoting $\mathbf{e}_t^m := \nabla h(\mathbf{y}_t^m; B_t) - \nabla h(\mathbf{y}_t^m)$, using the inductive hypothesis that $\mathbf{y}^* \in \mathbb{B}(\mathbf{y}_0^m, R_m)$ with probability at least $1-\frac{m\gamma}{M}$, under this probably event we get

$$\begin{aligned}
\left\|\mathbf{y}_{t+1}^m - \mathbf{y}^*\right\|^2 &= \left\|\mathrm{Proj}_{\mathbb{B}(\mathbf{y}_0^m, R_m)}\left[\mathbf{y}_t^m - \eta_t(\nabla h(\mathbf{y}_t^m; B_t^m) + \nu_t^m)\right] - \mathbf{y}^*\right\|^2 \\
&\leq \|\mathbf{y}_t^m - \eta_t(\nabla h(\mathbf{y}_t^m; B_t^m) + \nu_t^m) - \mathbf{y}^*\|^2 \\
&= \|\mathbf{y}_t^m - \mathbf{y}^*\|^2 - 2\eta_t\left\langle \mathbf{y}_t^m - \mathbf{y}^*, \nabla h(\mathbf{y}_t^m; B_t^m) + \nu_t^m\right\rangle + \eta_t^2\left\|\nabla h(\mathbf{y}_t^m; B_t^m) + \nu_t^m\right\|^2 \\
&\leq \|\mathbf{y}_t^m - \mathbf{y}^*\|^2 - 2\eta_t\left\langle \mathbf{y}_t^m - \mathbf{y}^*, \nabla h(\mathbf{y}_t) + \mathbf{e}_t^m + \nu_t^m\right\rangle + 2\eta_t^2\left(\|\nu_t^m\|^2 + \|\nabla h(\mathbf{y}_t)\|^2\right) \\
&= \|\mathbf{y}_t^m - \mathbf{y}^*\|^2 - 2\eta_t\left\langle \mathbf{y}_t^m - \mathbf{y}^*, \nabla h(\mathbf{y}_t^m)\right\rangle \\
&\quad - 2\eta_t\left\langle \mathbf{y}_t^m - \mathbf{y}^*, \mathbf{e}_t^m + \nu_t^m\right\rangle + 2\eta_t^2\left(\|\nu_t^m\|^2 + \|\nabla h(\mathbf{y}_t)\|^2\right)\,.
\end{aligned}$$

Rearranging, and using the strong convexity and Lipschitz assumptions, we see that

$$\begin{aligned}
h(\mathbf{y}_t^m) - h(\mathbf{y}^*) &\leq \left\langle \mathbf{y}_t^m - \mathbf{y}^*, \nabla h(\mathbf{y}_t^m)\right\rangle - \frac{\mu}{2}\|\mathbf{y}_t^m - \mathbf{y}^*\|^2 \\
&\leq \left(\frac{1}{2\eta_t} - \frac{\mu}{2}\right)\|\mathbf{y}_t^m - \mathbf{y}^*\|^2 - \frac{1}{2\eta_t}\left\|\mathbf{y}_{t+1}^m - \mathbf{y}^*\right\|^2 \\
&\quad - \left\langle \mathbf{y}_t^m - \mathbf{y}^*, \mathbf{e}_t^m + \nu_t^m\right\rangle + \eta_t\left(\|\nu_t^m\|^2 + L^2\right)\,.
\end{aligned}$$

Averaging over t and using $\eta_t = \frac{1}{\mu(t+1)}$, which satisfies $\left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}} - \mu\right) \leq 0$ and $\frac{1}{T}\sum_{t=0}^T \eta_t \lesssim \frac{\log T}{\mu T}$, by Jensen's inequality, overall we get with probability at least $1-\frac{m\gamma}{M}$:

$$\begin{aligned}
h(\mathbf{y}_0^{m+1}) - h(\mathbf{y}^*) &= h\left(\frac{1}{T}\sum_{t=0}^{T-1}\mathbf{y}_t^m\right) - h(\mathbf{y}^*) \\
&\lesssim \underbrace{\left|\frac{1}{T}\sum_{t=0}^{T-1}\left\langle \mathbf{y}_t^m - \mathbf{y}^*, \mathbf{e}_t^m + \nu_t^m\right\rangle\right|}_{(I)} + \frac{L^2\log T}{\mu T} + \frac{\log T}{\mu T}\underbrace{\sum_{t=0}^{T-1}\|\nu_t^m\|^2}_{(II)}\,.
\end{aligned} \tag{19}$$

We now apply concentration inequalities to bound (I) and (II) with high probability, for which we will use basic properties of sub-Gaussian distributions (cf. Vershynin 2018, §3.4). To bound (I), note that for all $t:\ \mathbb{E}\mathbf{e}_t^m = \mathbb{E}\nu_t^m = \mathbf{0}$ and therefore $\mathbb{E}\left\langle \mathbf{y}_t^m - \mathbf{y}^*, \mathbf{e}_t^m + \nu_t^m \right\rangle = 0$. Moreover, $\mathbf{e}_t^m = \frac{1}{b}\sum_{\xi\in B_t^m}(\nabla h(\mathbf{x}_t^m;\xi) - \nabla h(\mathbf{y}_t^m))$ is the average of b independent vectors with norm bounded by at most $2L$, while $\nu_t^m \sim \mathcal{N}(\mathbf{0}, \sigma_{\text{SGD}}^2 I_{d_y}) = \mathcal{N}(\mathbf{0}, \widetilde{\mathcal{O}}(\frac{L^2}{\epsilon'^2}) \cdot I_{d_y})$, and also $\|\mathbf{y}_t^m - \mathbf{y}^*\| \leq R_m$ by the inductive hypothesis. By combining all of these observations, we see that $\langle \mathbf{y}_t^m - \mathbf{y}^*, \mathbf{e}_t^m + \nu_t^m \rangle$ is a $\mathcal{O}(R_m \cdot (\frac{L}{b} + \frac{L}{\epsilon'})) = \mathcal{O}(\frac{R_m L}{\epsilon'})$-sub-Gaussian random variable. By Azuma's inequality for sub-Gaussians (Shamir, 2011), we get that with probability at least $1 - \frac{\gamma}{2M}$:

$$(I) = \widetilde{\mathcal{O}}\left(\frac{\frac{R_m L}{\epsilon'}\log(\gamma/M)}{\sqrt{T}}\right) = \widetilde{\mathcal{O}}\left(\frac{R_m L}{\epsilon' n}\right)\ . \tag{20}$$

To bound (II), by concentration of the Gaussian norm (cf. Vershynin 2018, Theorem 3.1.1) and the union bound we can get that with probability at least $1 - \frac{\gamma}{2M}$:

$$(II) = \widetilde{\mathcal{O}}\left(d_y \sigma_{\text{SGD}}^2\right) = \widetilde{\mathcal{O}}\left(\frac{d_y L^2}{\epsilon'^2}\right)\ . \tag{21}$$

Plugging Eqs. (20) and (21) into (19), we overall get that with probability at least $1 - \frac{m\gamma}{M} - 2\cdot\frac{\gamma}{2M} = 1 - \frac{(m+1)\gamma}{M}$:

$$h(\mathbf{y}_0^{m+1}) - h(\mathbf{y}^*) = \widetilde{\mathcal{O}}\left(\frac{R_m L}{\epsilon' n} + \frac{d_y L^2}{\mu \epsilon'^2 n^2}\right)\ .$$

Applying the μ-strong-convexity of h, and sub-additivity of the square root, we get that

$$\left\|\mathbf{y}_0^{m+1} - \mathbf{y}^*\right\| \leq \sqrt{\frac{2(h(\mathbf{y}_0^{m+1}) - h(\mathbf{y}^*))}{\mu}} = \widetilde{\mathcal{O}}\left(\sqrt{\frac{R_m L}{\mu\epsilon' n}} + \frac{L\sqrt{d_y}}{\mu\epsilon' n}\right) \leq R_{m+1}\ .$$

We have therefore proven (18). In particular for $m = M$ we get that with probability at least $1 - \gamma$:

$$\|\mathbf{y}_{\text{out}} - \mathbf{y}^*\| \leq R_M\ , \tag{22}$$

hence it remains to bound R_M. We will prove, once again by induction over m, that

$$R_m = \widetilde{\mathcal{O}}\left(R_0^{\frac{1}{2^m}}\left(\frac{L}{\mu\epsilon' n}\right)^{1-\frac{1}{2^m}} + \frac{L}{\mu\epsilon' n}\sum_{i=1}^{m} d_y^{\frac{1}{2^i}}\right)\ . \tag{23}$$

The base $m = 0$ simply follows since the left hand side in (23) reduces to R_0. Denoting $A := \frac{L}{\mu\epsilon' n}$, by our assignment of R_{m+1}, the induction hypothesis and sub-additivitiy of the square root we get:

$$\begin{aligned}
R_{m+1} &= \widetilde{\mathcal{O}}\left(\sqrt{R_m A} + A\sqrt{d_y}\right) \\
&= \widetilde{\mathcal{O}}\left(A^{1/2}\left(R_0^{\frac{1}{2^{m+1}}} A^{\frac{1}{2}-\frac{1}{2^{m+1}}} + A^{1/2}\sum_{i=1}^{m} d_y^{\frac{1}{2^{i+1}}}\right) + A d_y^{1/2}\right) \\
&= \widetilde{\mathcal{O}}\left(R_0^{\frac{1}{2^{m+1}}} A^{1-\frac{1}{2^{m+1}}} + A\sum_{i=1}^{m+1} d_y^{\frac{1}{2^i}}\right)\ ,
\end{aligned}$$

therefore proving (23). In particular, for $m = M = \log_2 \log(\frac{\mu\epsilon' n}{L})$, which satisfies $\frac{1}{2^M} = \frac{1}{\log(\frac{\mu\epsilon' n}{L})} = \frac{1}{\log(1/A)}$ we get

$$\begin{aligned} R_M &= \widetilde{\mathcal{O}}\left(R_0^{\frac{1}{\log(1/A)}} A^{1+\frac{1}{\log(A)}} + MAd_y^{1/2}\right) \\ &= \widetilde{\mathcal{O}}\left(R_0^{\frac{1}{\log(1/A)}} A + Ad^{1/2}\right) \\ &= \widetilde{\mathcal{O}}\left(R_0^{\frac{1}{\log(\mu\epsilon' n/L)}} \frac{L}{\mu\epsilon' n} + \frac{Ld_y^{1/2}}{\mu\epsilon' n}\right) \\ &= \widetilde{\mathcal{O}}\left(\frac{L\sqrt{d_y}}{\mu\epsilon' n}\right) , \end{aligned}$$

where the last follows from our assumption on n. This completes the proof by (22). ■

Appendix D. Auxiliary lemma

We will recall a useful fact, which asserts that the mapping $\mathcal{G}_{\mathbf{v},\eta}(\mathbf{x})$ is non-expansive with respect to $\mathbf{v}$:

Lemma 29 *For any* $\mathbf{x}, \mathbf{v}, \mathbf{w} \in \mathbb{R}^d,\ \eta > 0 :\ \|\mathcal{G}_{\mathbf{v},\eta}(\mathbf{x}) - \mathcal{G}_{\mathbf{w},\eta}(\mathbf{x})\| \leq \|\mathbf{v} - \mathbf{w}\|$.

In our analysis we argue that in the bilevel setting, gradient estimates must be *inexact* to avoid privacy leaking between the two levels, and Lemma 29 allows us to control the error due to this inexactness. Lemma 29 is known (cf. Ghadimi et al. 2016), and we reprove it here for completeness.
Proof [Proof of Lemma 29] The proof is due to Ghadimi et al. (2016). By definition,

$$\begin{aligned} \mathcal{P}_{\mathbf{v},\eta}(\mathbf{x}) &= \arg\min_{\mathbf{u}\in\mathcal{X}} \left\{ \langle \mathbf{v}, \mathbf{u} \rangle + \frac{1}{2\eta} \|\mathbf{x} - \mathbf{u}\|^2 \right\} , \\ \mathcal{P}_{\mathbf{w},\eta}(\mathbf{x}) &= \arg\min_{\mathbf{u}\in\mathcal{X}} \left\{ \langle \mathbf{w}, \mathbf{u} \rangle + \frac{1}{2\eta} \|\mathbf{x} - \mathbf{u}\|^2 \right\} , \end{aligned}$$

hence by first order optimality criteria, for any $\mathbf{u} \in \mathcal{X}$:

$$\begin{aligned} \left\langle \mathbf{v} + \frac{1}{\eta}(\mathcal{P}_{\mathbf{v},\eta}(\mathbf{x}) - \mathbf{x}), \mathbf{u} - \mathcal{P}_{\mathbf{v},\eta}(\mathbf{x}) \right\rangle \geq 0 \,, \\ \left\langle \mathbf{w} + \frac{1}{\eta}(\mathcal{P}_{\mathbf{w},\eta}(\mathbf{x}) - \mathbf{x}), \mathbf{u} - \mathcal{P}_{\mathbf{w},\eta}(\mathbf{x}) \right\rangle \geq 0 \,. \end{aligned}$$

Plugging $\mathcal{P}_{\mathbf{w},\eta}(\mathbf{x})$ as $\mathbf{u}$ into the first inequality above, and $\mathcal{P}_{\mathbf{v},\eta}(\mathbf{x})$ into the second, shows that

$$\begin{aligned} 0 &\leq \left\langle \mathbf{v} + \frac{1}{\eta}(\mathcal{P}_{\mathbf{v},\eta}(\mathbf{x}) - \mathbf{x}), \mathcal{P}_{\mathbf{w},\eta}(\mathbf{x}) - \mathcal{P}_{\mathbf{v},\eta}(\mathbf{x}) \right\rangle , \\ 0 &\leq \left\langle \mathbf{w} + \frac{1}{\eta}(\mathcal{P}_{\mathbf{w},\eta}(\mathbf{x}) - \mathbf{x}), \mathcal{P}_{\mathbf{v},\eta}(\mathbf{x}) - \mathcal{P}_{\mathbf{w},\eta}(\mathbf{x}) \right\rangle = \left\langle -\mathbf{w} + \frac{1}{\eta}(\mathbf{x} - \mathcal{P}_{\mathbf{w},\eta}(\mathbf{x})), \mathcal{P}_{\mathbf{w},\eta}(\mathbf{x}) - \mathcal{P}_{\mathbf{v},\eta}(\mathbf{x}) \right\rangle . \end{aligned}$$

Summing the two inequalities yields

$$
\begin{aligned}
0 &\leq \left\langle \mathbf{v} - \mathbf{w} + \frac{1}{\eta}(\mathcal{P}_{\mathbf{v},\eta}(\mathbf{x}) - \mathcal{P}_{\mathbf{w},\eta}(\mathbf{x})), \mathcal{P}_{\mathbf{w},\eta}(\mathbf{x}) - \mathcal{P}_{\mathbf{v},\eta}(\mathbf{x}) \right\rangle \\
&= \langle \mathbf{v} - \mathbf{w}, \mathcal{P}_{\mathbf{w},\eta}(\mathbf{x}) - \mathcal{P}_{\mathbf{v},\eta}(\mathbf{x}) \rangle - \frac{1}{\eta} \left\| \mathcal{P}_{\mathbf{w},\eta}(\mathbf{x}) - \mathcal{P}_{\mathbf{v},\eta}(\mathbf{x}) \right\|^2 \\
&\leq \left\| \mathcal{P}_{\mathbf{w},\eta}(\mathbf{x}) - \mathcal{P}_{\mathbf{v},\eta}(\mathbf{x}) \right\| \left(\|\mathbf{v} - \mathbf{w}\| - \frac{1}{\eta} \left\| \mathcal{P}_{\mathbf{w},\eta}(\mathbf{x}) - \mathcal{P}_{\mathbf{v},\eta}(\mathbf{x}) \right\| \right) \ .
\end{aligned}
$$

Hence,

$$
\begin{aligned}
\|\mathbf{v} - \mathbf{w}\| &\geq \frac{1}{\eta} \left\| \mathcal{P}_{\mathbf{v},\eta}(\mathbf{x}) - \mathcal{P}_{\mathbf{w},\eta}(\mathbf{x}) \right\| \\
&= \left\| \frac{1}{\eta} \left(\mathcal{P}_{\mathbf{v},\eta}(\mathbf{x}) - \mathbf{x} \right) - \frac{1}{\eta} \left(\mathcal{P}_{\mathbf{w},\eta}(\mathbf{x}) - \mathbf{x} \right) \right\| \\
&= \left\| \mathcal{G}_{\mathbf{v},\eta}(\mathbf{x}) - \mathcal{G}_{\mathbf{w},\eta}(\mathbf{x}) \right\| \ .
\end{aligned}
$$

■