A Proof Details for Theorem 4.1

A.1 Proof of Lemma 1

First, we formulate a uniform-convergence bound, which closely resembles Theorem 1 from [25]. The only difference is that they consider PAC setting: sampling an i.i.d. dataset $\mathbf{Z}$ from a fixed distribution, and comparing the finite-sample objective $\hat{f}$ computed using $\mathbf{Z}$ with the true objective f, which is an expectation over the distribution. On the other hand, we consider an increasing sequence of datasets $\mathbf{Z}_t = \{z_i\}_{i=1}^{n_t}$, selected by a uniformly random permutation of the full dataset $\mathbf{Z}$. Note, that we assume the algorithm never observes the full dataset, only loading as much data as needed. Taking the limit of $N \to \infty$, the relationship between any subset $\mathbf{Z}_t$ and the full dataset Z becomes statistically equivalent to i.i.d. sampling from any fixed underlying distribution. Given that our goal is generalization to predicting on new data, that simplification is reasonable, although the analysis does go through in the strict optimization setting, where N is finite. However, even with this assumption, we still need to describe the relationship between two consecutive subsets in the sequence, which does not fit the i.i.d. sampling model. To that end, we can view $\mathbf{Z}_t$ as a fraction of elements from $\mathbf{Z}_{t+1}$, selected uniformly at random without replacement. We now describe the relationship between the two consecutive loss estimates in this sequence. Note, that in this section the big-$\mathcal{O}$ notation hides only fixed numeric constants.

Lemma 2 *With probability* $1-\delta$*, for all* $\mathbf{w}$ *and all* $0 \le t \le T$ *we have*

$$\hat{g}_{t+1}(\mathbf{w}) \le 2\,\hat{g}_t(\mathbf{w}) + \mathcal{O}\left(\frac{L^2B^2\log(T/\delta)}{\lambda\, n_t}\right). \quad (9)$$

Proof The proof is very similar to [25], except we replace standard Rademacher Complexity with Permutational Rademacher Complexity (PRC), proposed in [26]. Let us fix t, and consider a specific set of instances $\mathbf{Z}_{t+1}$, from which a random subset $\mathbf{Z}_t$ is sampled (without replacement). Following [25], for any $r > 0$ we define

$$\mathcal{H}_{t,r} \triangleq \left\{ h_{\mathbf{w}}^{t,r} = \frac{h_{\mathbf{w}}^t}{4^{k_{t,r}(\mathbf{w})}} \;:\; \mathbf{w} \in \mathbf{W} \right\},$$

where

$$k_{t,r}(\mathbf{w}) \triangleq \min\{k' \in \mathbb{Z}_+ \;:\; \hat{f}_{t+1}(\mathbf{w}) \le r4^{k'}\}$$

and

$$h_{\mathbf{w}}^t(z) \triangleq \ell_z(\mathbf{w}) - \ell_z(\widehat{\mathbf{w}}_{t+1}^*).$$

Our aim is to analyze the empirical average of the function values from $\mathcal{H}_{t,r}$ evaluated on a given instance set $\mathbf{Z}$:

$$\bar{h}_{\mathbf{Z}}^{t,r}(\mathbf{w}) = \frac{1}{|\,\mathbf{Z}\,|}\sum_{z\in\mathbf{Z}} h_{\mathbf{w}}^{t,r}(z).$$

We can translate the task of comparing $\hat{g}_{t+1}$ and $\hat{g}_t$ to describe it in terms of the function class $\mathcal{H}_{t,r}$:

$$\begin{aligned}\hat{g}_{t+1}(\mathbf{w}) - \hat{g}_t(\mathbf{w}) &= \hat{g}_{t+1}(\mathbf{w}) - (\hat{f}_t(\mathbf{w}) - \hat{f}_t(\widehat{\mathbf{w}}_t^*))\\ &\le \hat{g}_{t+1}(\mathbf{w}) - (\hat{f}_t(\mathbf{w}) - \hat{f}_t(\widehat{\mathbf{w}}_{t+1}^*))\\ &= 4^{k_r(\mathbf{w})}\left[\bar{h}_{\mathbf{Z}_{t+1}}^{t,r}(\mathbf{w}) - \bar{h}_{\mathbf{Z}_t}^{t,r}(\mathbf{w})\right].\end{aligned}$$

To compare $\bar{h}_{\mathbf{Z}_t}^{t,r}$ with $\bar{h}_{\mathbf{Z}_{t+1}}^{t,r}$ we use Theorem 5 [26], which provides transductive risk bounds through expected PRC of function class $\mathcal{H}_{t,r}$, conditioned on set $\mathbf{Z}_{t+1}$:

$$\mathcal{Q}(\mathcal{H}_{t,r}, \mathbf{Z}_{t+1}) \triangleq \mathbb{E}\left[\hat{Q}_{n_t, n_t/2}(\mathcal{H}_{t,r}, \mathbf{Z}_t) \mid \mathbf{Z}_{t+1}\right].$$

Here, the randomness only comes from selecting $\mathbf{Z}_t$ as a subset of $\mathbf{Z}_{t+1}$. For any $\delta > 0$, with probability at least $1-\delta$,

$$\begin{aligned}&\sup_{\mathbf{w}}\left[\bar{h}_{\mathbf{Z}_{t+1}}^{t,r}(\mathbf{w}) - \bar{h}_{\mathbf{Z}_t}^{t,r}(\mathbf{w})\right]\\ &\quad\le \underbrace{\mathcal{Q}(\mathcal{H}_{t,r}, \mathbf{Z}_{t+1})}_{Y_1} + \underbrace{\sup_{h_{\mathbf{w}}^{t,r}, z} |h_{\mathbf{w}}^r(z)| \cdot \mathcal{O}\left(\sqrt{\frac{\log(1/\delta)}{n_t}}\right)}_{Y_2}.\end{aligned}$$

Note, that $\mathcal{Q}(\mathcal{H}_{t,r}, \mathbf{Z}_{t+1}) = \mathcal{O}(\mathcal{R}_{n_t}(\mathcal{H}_{t,r}))$ (see [26]), where $\mathcal{R}_{n_t}$ is the standard Rademacher Complexity. The remainder of the proof proceeds identically as in [25] (up to numerical constants), i.e. by bounding both terms Y_1 and Y_2 by

$$\mathcal{O}\left(LB\sqrt{\frac{r\log(1/\delta)}{\lambda n_t}}\right).$$

Note, that since the bound is obtained for every possible $\mathbf{Z}_{t+1}$, it will still hold with probability at least $1-\delta$ without conditioning on $\mathbf{Z}_{t+1}$.

Finally, as shown in [25], by setting r appropriately we obtain that w.p. $1-\delta$, for all $\mathbf{w}$

$$\hat{g}_{t+1}(\mathbf{w}) \le 2\,\hat{g}_t(\mathbf{w}) + \mathcal{O}\left(\frac{L^2B^2\log(1/\delta)}{\lambda\, n_t}\right).$$

Applying union bound to account for all values of t simultaneously, we obtain the desired result. □

We return to the proof of Lemma 1. Using Lipschitz and boundedness assumptions for the loss ℓ and mapping ϕ, as well as strong convexity of the regularized objective, we obtain initial tolerance of the loss estimate:

$$\begin{aligned}\hat{g}_0(\mathbf{w}_0) &= \hat{f}_0(\mathbf{w}_0) - \hat{f}_0(\widehat{\mathbf{w}}_0^*)\\ &\le \frac{1}{n_0}\sum_{i=1}^{n_0}\left(\ell_{z_i}(\mathbf{w}_0) - \ell_{z_i}(\widehat{\mathbf{w}}_0^*)\right)\\ &\le LB\|\,\widehat{\mathbf{w}}_0^*\| \le LB\sqrt{\frac{2\,\hat{g}_0(\mathbf{w}_0)}{\lambda}},\\ \hat{g}_0(\mathbf{w}_0) &\le \frac{2L^2B^2}{\lambda}.\end{aligned}$$

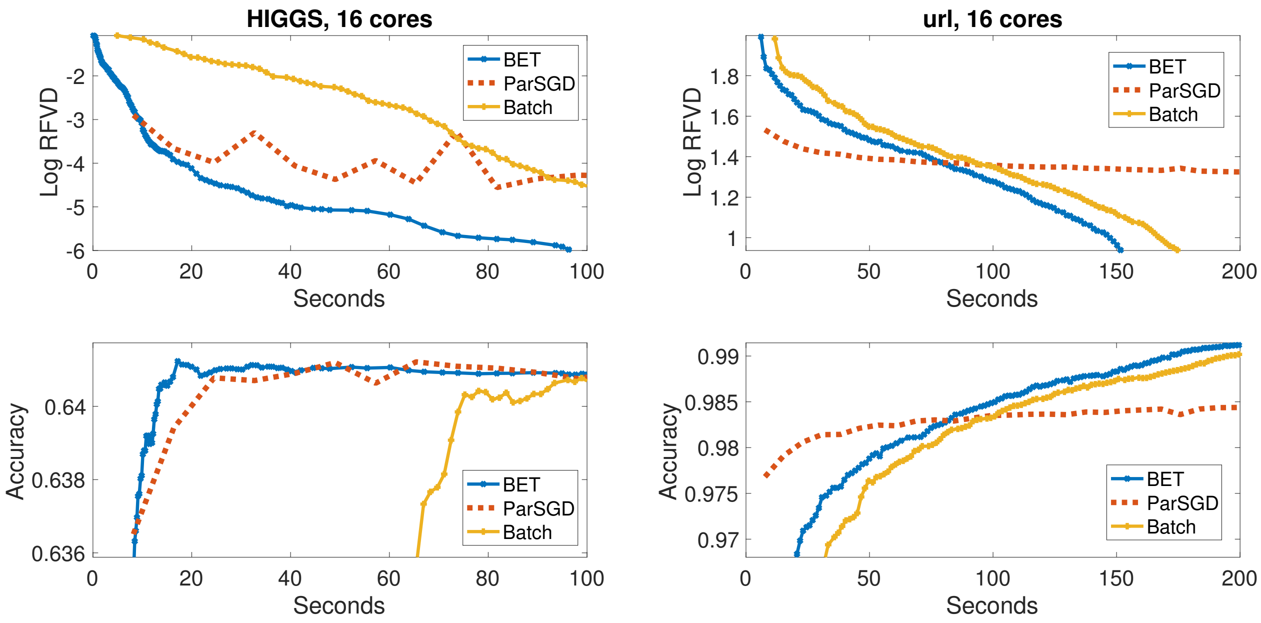

Figure 6: Comparing BET, Batch and Parallel SGD for HIGGS (left) and url (right) datasets, running on 16 cores. BET is as good as the best method in each case.

We used the fact that $\mathbf{w}_0$ is set to zero only for applying inequality $\| \mathbf{w}_0 \| \leq \| \widehat{\mathbf{w}}_0^* \|$ to drop the regularization terms (any initialization satisfying that requirement is acceptable).

Finally, Condition (7) regards the relationship between approximation error estimate $\hat{g}_T$ and full approximation error $\hat{g}$. This bound can be obtained by repeating the same argument as in Lemma 2. We can either assume $N \to \infty$ and use standard Rademacher complexity, as in Theorem 1, [25], or stay with the finite optimization model and apply PRC. Thus, we can set ϵ_0 to satisfy the conditions of Lemma 1. $\square$

A.2 Deriving Log Terms in Theorem 4.1

The number of iterations, $T = \mathcal{O}(\log(\epsilon_0/\epsilon))$, depends on ϵ_0. But in Lemma 1 we defined ϵ_0 using T. To address this, we have to find ϵ_0 satisfying:

$$\epsilon_0 \geq K \log\left(\frac{\log(\epsilon_0/\epsilon)}{\delta}\right),$$

with $K = \mathcal{O}(L^2B^2/\lambda)$. It is easy to show that for small enough ϵ it suffices to set

$$\begin{aligned}\epsilon_0 &\triangleq 2K \log\left(\frac{\log(1/\epsilon)}{\delta}\right) \\ &= \mathcal{O}\left(\frac{L^2B^2}{\lambda} \cdot (\log\log(1/\epsilon) + \log(1/\delta))\right).\end{aligned}$$

Thus, setting $n_0 = 1$, we obtain the final complexity bound in Theorem 4.1 as

$$\mathcal{O}\left(\frac{\kappa}{\lambda\epsilon} \cdot L^2B^2 \cdot (\log\log(1/\epsilon) + \log(1/\delta))\right).$$

B Additional Experiments

Dataset, size	Train/Test	Dim.	λ
HIGGS, 8GB	10.5M/0.5M	28	1e-10
url, 1GB	1.8M/0.5M	3.2M	1e-8

Table 2: A list of additional datasets and regularization used for the experiments.

In this section we look at two datasets with very different properties. First one, HIGGS, is large, but extremely low dimensional. In this case, given an overabundance of data, if we look at the accuracy plot (see Figure 6), the Batch algorithm takes much longer to converge than Parallel SGD. This follows from the fact that the task has low sample complexity, and a Batch method is wasting resources by training on too much data. The second dataset, url, is very high-dimensional, and in this case Batch has clear advantage over SGD. BET does as well as the best method on each dataset. In the case of HIGGS, BET simply converges to the optimum accuracy before it even reaches full dataset, thus saving on expensive iterations. For url, the batch expansion happens relatively early on in the optimization, and from that point on the algorithm is simply running full L-BFGS. Those two extreme cases show the versatility and robustness of our proposed meta-algorithm.