Tuning-Free Coreset Markov Chain Monte Carlo via Hot DoG

Naitong Chen[1] **Jonathan H. Huggins**[2] **Trevor Campbell**[1]

[1]Department of Statistics, University of British Columbia, Vancouver, BC, Canada
[2]Department of Mathematics & Statistics and Faculty of Computing & Data Sciences, Boston University, Boston, MA, USA

Abstract

A Bayesian coreset is a small, weighted subset of a data set that replaces the full data during inference to reduce computational cost. The state-of-the-art coreset construction algorithm, *Coreset Markov chain Monte Carlo* (Coreset MCMC), uses draws from an adaptive Markov chain targeting the coreset posterior to train the coreset weights via stochastic gradient optimization. However, the quality of the constructed coreset, and thus the quality of its posterior approximation, is sensitive to the stochastic optimization learning rate. In this work, we propose a learning-rate-free stochastic gradient optimization procedure, *Hot-start Distance over Gradient* (Hot DoG), for training coreset weights in Coreset MCMC without user tuning effort. We provide a theoretical analysis of the convergence of the coreset weights produced by Hot DoG. We also provide empirical results demonstrate that Hot DoG provides higher quality posterior approximations than other learning-rate-free stochastic gradient methods, and performs competitively to optimally-tuned ADAM.

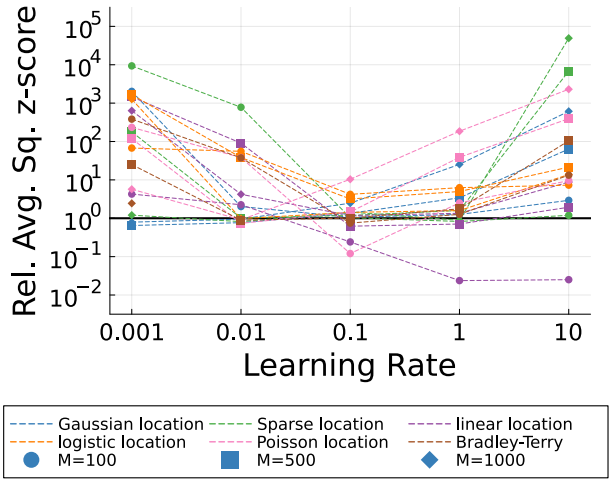


Figure 1: Relative Coreset MCMC posterior approximation error comparing ADAM (with different learning rates) and the proposed Hot DoG method (under our recommended setting). The metric plotted is the ratio of average squared z-scores (defined in Eq. (6)) under ADAM to those under Hot DoG. Values above the horizontal black line (10^0) indicate that the proposed Hot DoG method outperformed ADAM. Median values after 200,000 optimization iterations across 10 trials are used for the relative comparison for a variety of datasets, models, and coreset sizes.

1 INTRODUCTION

Bayesian inference provides a flexible framework for parameter estimation and uncertainty quantification in statistical models. Markov chain Monte Carlo [Robert and Casella, 2004; Robert and Casella, 2011; Gelman et al., 2013, Chs. 11 and 12], the standard methodology for performing Bayesian inference, involves simulating carefully constructed Markov chains whose stationary distribution is the target Bayesian posterior. In the large-scale data setting, this procedure can become prohibitively expensive, as it requires iterating over the entire data set to simulate the next state.

Bayesian coresets [Huggins et al., 2016] are a popular approach for speeding up Bayesian inference in the large-scale data setting. A Bayesian coreset is a weighted subset of data that replaces the full data set during inference, leveraging the insight that large datasets often exhibit a significant degree of redundancy.[1] With a carefully constructed coreset, one can significantly reduce the computational cost of in-

[1]A related approach, *data distillation*, constructs a small synthetic data set for downstream tasks. However, this approach often requires bespoke methods for non-real-valued data (see [Sachdeva and McAuley, 2023, Sec. 3]). In contrast, Bayesian coresets do not modify individual data points, and so are fully generic.

ference while still obtaining samples from a high quality approximation of the full Bayesian posterior. In fact, given a data set of N points, a coreset of size $\mathcal{O}(\log N)$ is sufficient for providing a near-exact posterior approximation in exponential family and other sufficiently simple models [Naik et al., 2022, Thms. 4.1 and 4.2; Chen et al., 2022, Prop. 3.1] and $\mathcal{O}(\text{polylog}\, N)$ is sufficient for more general cases [Campbell, 2024, Cor. 6.1].

Constructing a coreset involves picking the data points to include in the coreset and assigning each data point its corresponding weight. The state-of-the-art method, Coreset MCMC [Chen and Campbell, 2024], selects coreset points by sampling them uniformly from the full data set, and learns the weights using stochastic gradient optimization techniques, e.g., ADAM [Kingma and Ba, 2014], where the gradients are estimated using MCMC draws targeting the current coreset posterior. However, as we demonstrate in this paper, there are two issues with this approach. First, the quality of the constructed coreset is sensitive to the learning rate of the stochastic optimization algorithm. And second, gradient estimates using MCMC draws are affected strongly in early iterations by initialization bias, leading to poor optimization performance.

To address these challenges, we first propose *Hot-start Distance over Gradient* (Hot DoG), a tuning-free stochastic gradient optimization procedure that can be used for learning coreset weights in Coreset MCMC. Hot DoG is a stochastic gradient method combining techniques from Do(W)G [Ivgi et al., 2023, Khaled et al., 2023], ADAM [Kingma and Ba, 2014], and RMSProp [Hinton et al., 2012] to set learning rates automatically. Hot DoG also includes an automated warm-up phase prior to weight optimization, which guards against usage of low quality MCMC draws when estimating the objective function gradients. We then analyze the convergence behaviour of Hot DoG in a representative setting. Empirically, Fig. 1 demonstrates that Hot DoG under our recommended setting performs competitively to optimally-tuned ADAM across a wide range of models, datasets, and coreset sizes, and can be multiple orders of magnitude more accurate than ADAM using other learning rates. Beyond the results shown in Fig. 1, we provide an extensive empirical investigation of the reliability of Hot DoG in comparison to other methods across various synthetic and real experiments.

Algorithm 1 `CoresetMCMC`

Require: θ_0, κ_w, S, M
  ▷ Initialize coreset weights
  $w_{0m} = \frac{N}{M}, \quad m = 1, \cdots, M$
  for $t = 0, \ldots, T$ **do**
  ▷ Subsample the data
    $\mathcal{S}_t \leftarrow \text{Unif}(S, [N])$ (without replacement)
    ▷ Compute gradient estimate
    $\hat{g}_t \leftarrow g(w_t, \theta_t, \mathcal{S}_t)$ (Eq. (4))
    $w_{t+1} \leftarrow$ `stochastic_gradient_step`$(w_t, \hat{g}_t)$
    ▷ Step each Markov chain
    for $k = 1, \ldots, K$ **do**
      $\theta_{(t+1)k} \sim \kappa_{w_{t+1}}(\cdot \mid \theta_{tk})$
    end for
  end for

2 BACKGROUND

2.1 BAYESIAN CORESETS

We are given a data set $(X_n)_{n=1}^N$ of N observations, a log-likelihood $\ell_n := \log p(x_n \mid \theta)$ for observation n given $\theta \in \Theta$, and a prior density $\pi_0(\theta)$. We would like to sample from the Bayesian posterior with density

$$\pi(\theta) := \frac{1}{Z} \exp\left(\sum_{n=1}^N \ell_n(\theta)\right) \pi_0(\theta),$$

where Z is the unknown normalizing constant. A Bayesian coreset replaces the sum over N log-likelihood terms with a weighted sum over a subset of size M, where $M \ll N$. Without loss of generality, we assume that these are the first M points. The coreset posterior can then be written as

$$\pi_w(\theta) := \frac{1}{Z(w)} \exp\left(\sum_{m=1}^M w_m \ell_m(\theta)\right) \pi_0(\theta), \quad (1)$$

where $w \in \mathbb{R}_+^M$ is a vector of coreset weights. Recent coreset construction methods uniformly select M points to include in the coreset [Naik et al., 2022, Chen et al., 2022, Chen and Campbell, 2024], and then optimize the weights of those M points as a variational inference problem [Campbell and Beronov, 2019],

$$w^\star = \arg\min_{w \in \mathbb{R}^M} \mathrm{D}_{\mathrm{KL}}(\pi_w || \pi) \quad \text{s.t.} \quad w \in \mathcal{W}, \quad (2)$$

with objective function gradient

$$\nabla_w \mathrm{D}_{\mathrm{KL}}(\pi_w || \pi) \quad (3)$$
$$= \mathrm{Cov}_{\pi_w}\left(\begin{bmatrix} \ell_1(\theta) \\ \vdots \\ \ell_M(\theta) \end{bmatrix}, \sum_m w_m \ell_m(\theta) - \sum_n \ell_n(\theta)\right).$$

2.2 CORESET MCMC

The key challenge in solving Eq. (2) is that π_w does not admit tractable i.i.d. draws, and so unbiased estimates of the gradient in Eq. (3) are not readily available. Coreset MCMC [Chen and Campbell, 2024] is an adaptive algorithm that

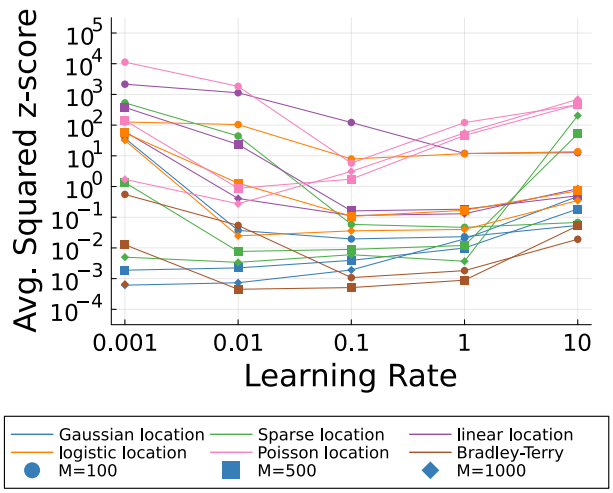


Figure 2: Coreset MCMC posterior approximation error (as defined in Eq. (6)) using ADAM with different learning rates for a variety of datasets, models, and coreset sizes. The lines indicate median values after 200,000 optimization iterations across 10 trials.

addresses this issue. The method first initializes weights $w_0 \in \mathbb{R}^M$ and $K \geq 2$ samples $\theta_0 = (\theta_{01}, \ldots, \theta_{0K}) \in \Theta^K$. At iteration $t \in \mathbb{N}$, given coreset weights w_t and samples $\theta_t \in \Theta^K$, it then updates the weights $w_t \to w_{t+1}$ using the stochastic gradient estimate based on the draws θ_t,

$$g(w_t, \theta_t, \mathcal{S}_t) = \tag{4}$$
$$\frac{1}{K-1}\sum_{k=1}^{K}\begin{bmatrix}\bar{\ell}_1(\theta_{tk})\\ \vdots \\ \bar{\ell}_M(\theta_{tk})\end{bmatrix}\left(\sum_m w_{tm}\bar{\ell}_m(\theta_{tk}) - \frac{N}{S}\sum_{s\in\mathcal{S}_t}\bar{\ell}_s(\theta_{tk})\right),$$

where $\mathcal{S}_t \subseteq [N]$ is a uniform subsample of indices of size S, and $\bar{\ell}_n(\theta_{tk}) = \ell_n(\theta_{tk}) - \frac{1}{K}\sum_{j=1}^{K}\ell_n(\theta_{tj})$. To complete the iteration, the method updates the samples by independently drawing $\theta_{(t+1)k} \sim \kappa_{w_{t+1}}(\theta_{tk}, \cdot)$ for each $k \in [K]$, where κ_w is a family of π_w-invariant Markov kernels. The pseudocode for Coreset MCMC is outlined in Algorithm 1.

3 TUNING-FREE CORESET MCMC

A key design choice when using Coreset MCMC is to specify how gradient estimates are used to optimize the weights. One can use ADAM [Kingma and Ba, 2014], which is used as the default optimizer for Coreset MCMC [Chen and Campbell, 2024]: at iteration t, with $\gamma_t > 0$ being the user-specified learning rate, we set

$$w_{t+1} \leftarrow \mathrm{proj}_{\geq 0}\left(w_t - \gamma_t \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}\right),$$

where $\hat{m}_t$ and $\hat{v}_t$ are exponential averages of past gradients $(\hat{g}_i)_{i=0}^t$ and their element-wise squares, and ϵ is a small constant. There are a wide range of other first-order stochastic methods available that could be used (e.g., vanilla stochastic gradient descent, AdaGrad [Duchi et al., 2011], etc.). However, like ADAM, most of these algorithms require setting a learning rate γ_t. And as we show in Fig. 2, the quality of samples obtained from Coreset MCMC can be highly sensitive to the selected learning rate. In particular, Fig. 2 shows that when using ADAM, no single learning rate applies well across all problems and coreset sizes; and for a given problem, the performance can vary by orders of magnitude as one varies the learning rate. Furthermore, the default ADAM learning rate of 10^{-3} [Kingma and Ba, 2014] provides poor results in most of the problems tested. As a result, careful tuning of the learning rate is required to obtain high quality posterior approximations. This usually involves a search on a log-scaled grid, which is computationally wasteful as the results for all but one of the parameter values are thrown out. Moreover, in practice determining which learning rate provides the best posterior approximation may not be straightforward, as we do not have access to estimates of the objective function.

A number of recent works in the literature propose learning-rate-free stochastic optimization methods to address this issue [Carmon and Hinder, 2022, Ivgi et al., 2023, Khaled et al., 2023, Defazio and Mishchenko, 2023, Mishchenko and Defazio, 2024]. Many of these methods are shown empirically to work competitively compared to optimally-tuned SGD on a wide range of large-scale, non-convex deep learning problems. Although different at first glance, all of these methods arise from the same insight. Suppose one would like to solve the stochastic optimization problem

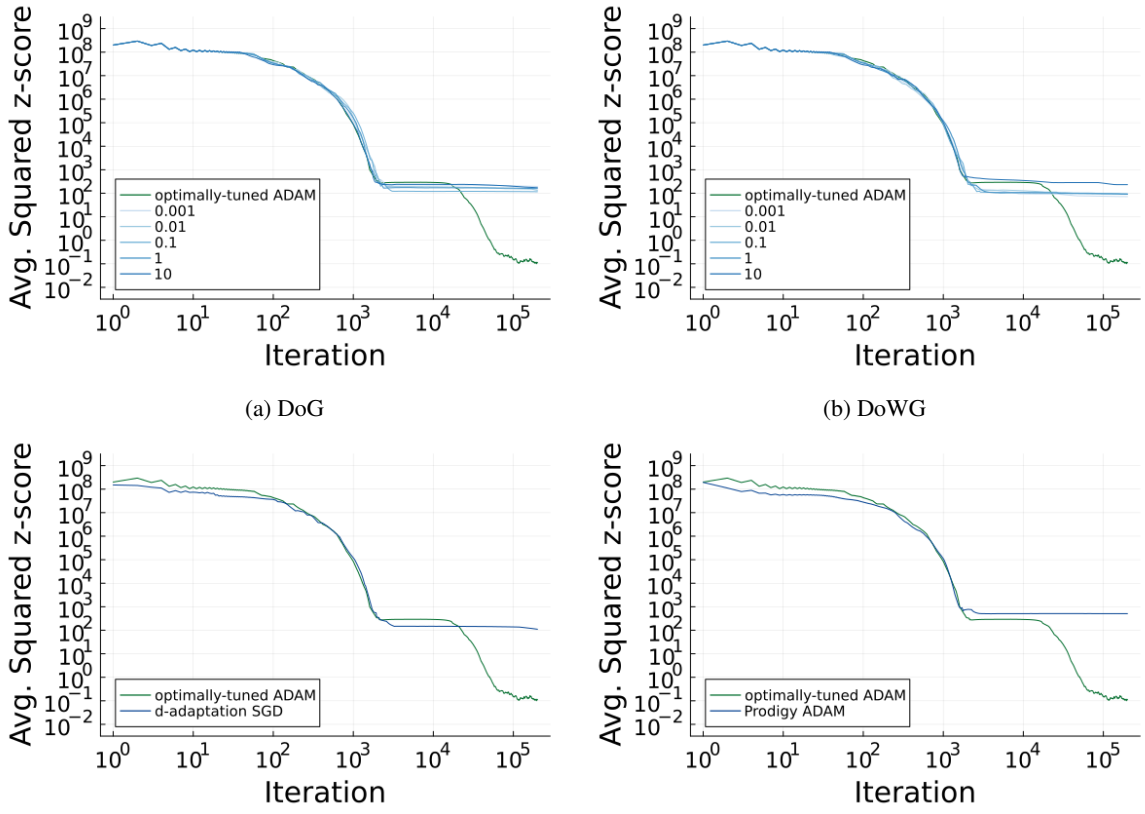
$$\min_{w\in\mathbb{R}^d} \mathbb{E}\left[f(w, \theta)\right],$$

where for all θ, $f(\cdot, \theta)$ is convex and we only have access to unbiased stochastic gradient $g_t = \partial f(w_t, \theta_t)$. Define the initial distance to the optimal solution $d_0 = \|w_0 - w^\star\|$ and the sum of all gradient norms $G_T = \sum_{t\leq T}\|g_t\|^2$. By setting the SGD learning rate $\gamma^\star = \frac{d_0}{\sqrt{G_T}}$, the average iterate $\bar{w} = \frac{1}{T}\sum_{t\leq T} w_t$ satisfies the optimal error bound

$$\mathbb{E}\left[f(\bar{w}, \theta)\right] - \mathbb{E}\left[f(w^\star, \theta)\right] \leq \frac{d_0\sqrt{G_T}}{T}$$

after T iterations [Carmon and Hinder, 2022, Orabona and Cutkosky, 2020]. Learning-rate-free methods therefore essentially try to estimate or bound the initial distance to the optimal solution d_0, which is unknown in practice. To the best of our knowledge, there are four state-of-the-art methods that do this in a manner that does not require multiple optimization runs, knowledge of unknown constants, or the ability to query the objective function: DoG [Ivgi et al., 2023], DoWG [Khaled et al., 2023], D-Adaptation [Defazio and Mishchenko, 2023] and prodigy [Mishchenko and Defazio, 2024]. DoG and DoWG run vanilla stochastic gradient descent (SGD),

$$w_{t+1} \leftarrow \mathrm{proj}_{\geq 0}(w_t - \gamma_t g_t),$$



(a) DoG

(b) DoWG

(c) D-Adaptation SGD

(d) prodigy ADAM

Figure 3: Traces of average squared coordinate-wise z-scores (defined in Eq. (6)) between the true and approximated posterior for a Bayesian linear regression example with $M = 1{,}000$ coreset points. We evaluate four learning-rate-free SGD methods: DoG and DoWG (with varying initial learning rate parameter), and D-Adaptation SGD and prodigy ADAM (with default initial lower bound 10^{-6}). The optimally-tuned ADAM baseline is shown in green. Results display the median after 200,000 optimization iterations across 10 trials.

with learning rate schedules

$$\gamma_t = \frac{r_t}{\sqrt{G_t}}(\text{DoG}),\ \gamma_t = \frac{r_t^2}{\sqrt{\sum_{i\leq t} r_i^2 \|g_i\|^2}}(\text{DoWG}), \quad (5)$$

where r_0 is set to some small constant and, for $t \geq 1$,

$$r_t = \max_{i\leq t} \|w_t - w_0\|.$$

For D-Adaptation and prodigy, r_t in Eq. (5) is replaced with a lower bound d_t on d_0, which is updated using estimated correlations between the g_t and step direction $w_0 - w_t$:

$$d_{t+1} = \max\left\{ \frac{\sum_{i=0}^t d_i \langle g_i, w_0 - w_i\rangle}{\left\|\sum_{i=0}^t d_i g_i\right\|}, d_t \right\}.$$

D-Adaptation replaces r_t in Eq. (5) (DoG) with d_t, while prodigy replaces r_t in Eq. (5) (DoWG) with d_t. Both D-Adaptation and prodigy have SGD and ADAM-based variants. All four methods have been shown empirically to match the performance of optimally-tuned SGD.

Fig. 3 shows the results from direct applications of DoG, DoWG, D-Adaptation (SGD), and prodigy (ADAM) to Coreset MCMC. We see that the quality of posterior approximation from all of four methods are orders of magnitude worse than optimally-tuned ADAM. With θ_0 initialized far away from high density regions of π_{w_0}, the initial gradient estimates are large in magnitude, which leads to small learning rates. The accumulation of these large gradient norms in the learning rate denominator eventually causes the learning rate to vanish, halting the progress of coreset weight optimization. We address these problems in the next section.

Before concluding this section, we note that there are other approaches for making SGD free of learning rate tuning: some methods involve using stochastic versions of line search [Vaswani et al., 2019, Paquette and Scheinberg, 2020], and others do the same for the Polyak step size [Loizou et al., 2021]. These methods are not applicable in our setting as they require evaluating the objective function. Recall that due to the unknown $Z(w)$ term in Eq. (1), we do not have access to estimates of the objective function.

Algorithm 2 `HotDoG`

Require: $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}, r = 10^{-3}$
T, θ_0, w_0
$v_0 \leftarrow \mathbf{0}, m_0 \leftarrow \mathbf{0}, d_0 \leftarrow \mathbf{0}, c \leftarrow 0, h \leftarrow$ `false`
for $t = 1, \ldots, T$ **do**
  if h **then**
    $c \leftarrow c + 1$
    $\mathcal{S}_t \leftarrow \mathrm{Unif}(S, [N])$ (without replacement)
    $\hat{g}_t = g(w_{t-1}, \theta_{t-1}, \mathcal{S}_t)$ (Eq. (4))
    $v_t \leftarrow \beta_2 v_{t-1} + (1-\beta_2)\hat{g}_t^2$
    $m_t \leftarrow \beta_1 m_{t-1} + (1-\beta_1)\hat{g}_t$
    $d_t \leftarrow \beta_1 d_{t-1} + (1-\beta_1)\max\{|w_{t-1} - w_0|, d_{t-1}\}$
    $\hat{v}_t \leftarrow v_t/(1-\beta_2^c)$
    $\hat{m}_t \leftarrow m_t/(1-\beta_1^c)$
    $\hat{d}_t \leftarrow$ (r **if** t==1 **else** $d_t/(1-\beta_1^{c-1})$)
    $w_t \leftarrow w_{t-1} - \hat{d}_t \left(\mathrm{diag}\left((c\,(\hat{v}_t + \epsilon))^{\frac{1}{2}}\right)\right)^{-1} \odot \hat{m}_t$
  else
    $w_t \leftarrow w_{t-1}, v_t \leftarrow v_{t-1}, m_t \leftarrow m_{t-1}, d_t \leftarrow d_{t-1}$
  end if
  for $k = 1, \ldots, K$ **do**
    $\theta_{tk} \sim \kappa_{w_t}(\cdot \mid \theta_{(t-1)k})$ ▷ record ℓ_{tk}
  end for
  ▷ Hot-start test
  $h \leftarrow$ (true **if** h **else** `HotStartTest`$\left((\ell_{ik})_{i=1,k=1}^{t,K}, t\right)$)
end for
return w_T

4 HOT DOG

In this section, we develop our novel Markovian optimization method, *Hot-start DoG* (Hot DoG), presented in Algorithm 2. Our method extends the original DoG optimizer in two ways: (1) we add a tuning-free hot-start test that automatically detects when the Markov chains have properly mixed and stochastic gradient estimates are stable, at which point we start coreset weight optimization; and (2) we apply acceleration techniques to DoG.

4.1 HOT-START TEST

Poorly initialized Markov chain states θ_0 can be detrimental to the performance of learning-rate-free methods in Coreset MCMC. Fig. 5, especially Figs. 5c to 5e show that this is likely due to the bias of initial gradient estimates. When θ_0 is initialized far away from high density regions of π_{w_0}, the initial gradient estimates can have norms that are orders of magnitude larger than those computed using i.i.d. draws. This leads to a quickly vanishing learning rate in Eq. (5). Therefore, it is crucial to hot-start the Markov chains to ensure they are properly mixed before training the coreset weights. There are MCMC convergence diagnostics that could be used for this purpose (e.g, $\hat{R}$ [Vehtari et al., 2021]); many work only with real-valued variables, and are overly stringent for our application. We require a test that works for general coreset posteriors of the form Eq. (1) and checks only that gradient estimates have stabilized reasonably.

Algorithm 3 `HotStartTest`

Require: $(\ell_{ik})_{i=1,k=1}^{t,K}$, t, $c = 0.5$
$n = \mathtt{ceil}(t/3)$
for $k = 1, \ldots, K$ **do**
  $s_{k1}^2 \leftarrow \frac{1}{n-2} \min_{a,b\in\mathbb{R}} \sum_{i=n+1}^{2n} (a + bi - \ell_{ik})^2$
  $s_{k2}^2 \leftarrow \frac{1}{n-2} \min_{a,b\in\mathbb{R}} \sum_{i=2n+1}^{t} (a + bi - \ell_{ik})^2$
  $u_k \leftarrow \frac{\left|\left(\frac{1}{n}\sum_{i=n+1}^{2n} \ell_{ik}\right) - \left(\frac{1}{n}\sum_{i=2n+1}^{t} \ell_{ik}\right)\right|}{\max\{s_{k1}, s_{k2}\}}$
end for
return (true **if** $\mathtt{median}(u_1, \ldots, u_K) < c$ **else** false)

To address this challenge, we propose keeping the weights fixed at their initialization (i.e., $w_{t+1} \leftarrow w_t$) until a hot-start test passes. For the test, for each Markov chain $k \in [K]$, we split the iterates $i = 1, \ldots, t$ into 3 segments, each of equal length $n = \lceil t/3 \rceil$. We compute the average log-potentials for the two latter segments m_{k1}, m_{k2}, and the standard deviations of residual errors s_{k1}, s_{k2} from a linear fit

$$m_{ki} = \frac{\sum_{j=in+1}^{(i+1)n} \ell_{jk}}{n}, \; s_{ki}^2 = \frac{\min_{a,b} \sum_{j=in+1}^{(i+1)n} (a + bj - \ell_{jk})^2}{n-2}.$$

Here $\ell_{jk} = \sum_{m'=1}^{M} w_{0m'} \ell_{m'}(\theta_{jk})$ is the log-potential for chain k at iteration j. Our test monitors the difference between m_{k1} and m_{k2} relative to s_{k1} and s_{k2}. A small difference in the averages indicates that the chains have stabilized. The residual standard errors allows us to remove trends from the noise computation. We define, for each $k \in [K]$,

$$u_k = \frac{|m_{k1} - m_{k2}|}{\max\{s_{k1}, s_{k2}\}},$$

and use the median of $(u_k)_{k=1}^K$ as our test statistic. This test statistic is checked against a threshold c; the test passes when the median test statistic is less than c. Algorithm 3 shows the pseudocode for the hot-start test. We find in practice setting $c = 0.5$ works well in general.

4.2 ACCELERATION

To accelerate DoG, we begin by noting that the denominator of the DoG learning rate in Eq. (5) is similar to that of AdaGrad [Duchi et al., 2011] in that it is a cumulative sum of some function of the gradient. Therefore, we can leverage the idea used in RMSProp [Hinton et al., 2012] for accelerating AdaGrad to accelerate DoG. In particular, at iteration t, we can replace $\sum_{i \le t} \|\hat{g}_i\|^2$ with $t\hat{v}_t$, the bias-corrected exponential moving average of the squared gradient. This allows us to exponentially decrease the weights of past gradient norms. As a result, the effect of the early $\|\hat{g}_t\|^2$ terms on the learning rate diminishes over time, resulting in less

conservative learning rates. To account for situations where the gradient estimates differ in scale across dimensions, we apply the above acceleration technique in a coordinate-wise fashion and obtain the following update rule for $\hat{v}_t$:

$$v_t = \beta_2 v_{t-1} + (1-\beta_2)\hat{g}_t^2, \quad \hat{v}_t = \frac{v_t}{1-\beta_2^t},$$

where $\beta_2 \in (0,1)$ is the exponential decay rate, $v_0 = 0$, and $\hat{g}_t^2$ denotes the vector with each entry of $\hat{g}_t$ squared. We further apply the same idea to r_t, the maximum distance traveled from w_0, and $\hat{g}_t$, the gradient estimate itself. We use $\beta_1 \in (0,1)$ to denote the exponential decay rate for these two quantities. Our final proposed optimization procedure is outlined in Algorithm 2. Note that in Algorithm 2, all computations are coordinate-wise.

In Hot DoG, we set the exponential decay rates, β_1 and β_2, to be the same as those in Kingma and Ba [2014], and we set the initial learning rate r to a small constant (default 10^{-3}) following the recommendation of Ivgi et al. [2023].

4.3 CONVERGENCE ANALYSIS

In this subsection, we present a theoretical analysis of the convergence of the coreset weights produced by Hot DoG. We begin by stating the set of assumptions, under which our analysis is conducted. These assumptions are stated formally stated in Appendix C.2. As required by Algorithm 2, we have that $|\beta_1| < 1$, $|\beta_2| < 1$, and $\epsilon, r > 0$. We further impose a set of assumptions about the feasible region $\mathcal{W}$ of the coreset weights. Namely, we assume (1) the coreset weights are non-negative and their sum is bounded above by a constant B (Assumptions C.2 and C.1), and (2) the existence of an exact coreset $w^\star \in \mathcal{W}$ in the sense that $\mathrm{D}_{\mathrm{KL}}(\pi_{w^\star}||\pi) = 0$. Both of these assumptions greatly simplify the analysis without sacrificing the representative nature of our assumed model. A typical choice for the coreset weight bound is to set $B = N$, where N is the total number of observations. In terms of the optimal coreset, past work has shown that it provides a near-exact approximation with high probability for the wide class of strongly log-concave models [Naik et al., 2022, Thm. 4.2]. Under Assumption C.2, which is similar to Assumption 3.1 in Chen and Campbell [2024], we do not expect the convergence result to change in a meaningful way, aside from there being a persistent error corresponding to the optimal coreset error.

Finally, we state our assumptions regarding the stochastic gradient (Eq. (4)), which estimates Eq. (3). We assume that the stochastic gradients are uniformly bounded above by a constant U (Assumption C.3). Now note that in Eq. (4), Monte Carlo error from the MCMC samples θ_t contributes to the stochasticity. We additionally impose a mixing condition on the Markov chains (Assumption C.4), and assume that the Monte Carlo error is controlled (Assumption C.5).

We now present our main theorem in Theorem 4.1. The proof of Theorem 4.1 can be found in Appendix C.3. Our result shows that Hot DoG produces coreset weights that converge to the optimum in expectation at a sublinear rate. This convergence rate is consistent with ADAM and other learning-rate-free stochastic gradient methods discussed in the paper (see for example [Ivgi et al., 2023, Thm. 3.10 and [Mishchenko and Defazio, 2024, Thm. 2]).

Theorem 4.1 (Hot DoG convergence)**.** *Suppose Assumptions C.1 to C.5 hold. As $t \to \infty$,*

$$\mathbb{E}\|w_t - w^\star\|^2 = O\left(\frac{1}{\sqrt{t}}\right).$$

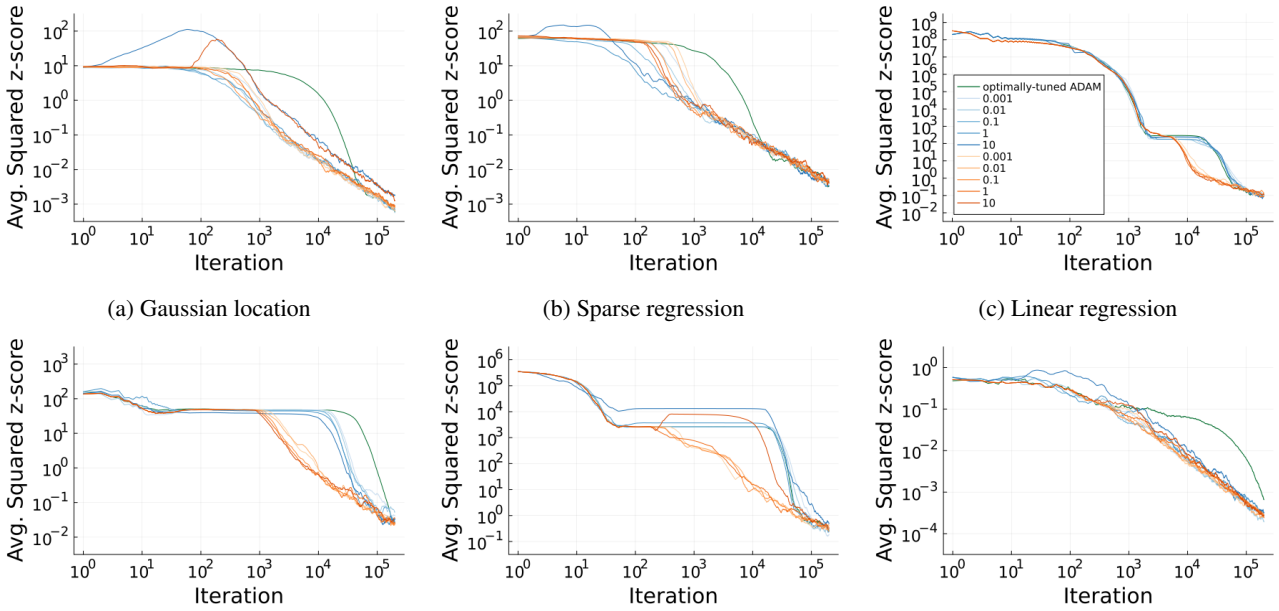
It is worth noting that whether to employ the hot-start test does not alter the convergence rate of Hot DoG as shown in Theorem 4.1. Instead, the hot-start test can lead to a more favourable constant in the convergence rate. As we discussed in Section 4.1, the hot-start test helps avoid updating the coreset weights using initial gradient estimates that may have unusually large norms. In terms of our analysis, by holding off optimizing w until the hot-start test passes, we can obtain a tighter bound on the gradient norm (i.e., a smaller U in Assumption C.3). This results in a smaller constant in the convergence rate given in Theorem 4.1, ultimately leading to improved finite-time performance.

5 EXPERIMENTS

In this section, we demonstrate the effectiveness of Hot DoG and compare our method against other learning-rate-free stochastic gradient methods: optimally-tuned ADAM from a log scale grid search, as well as prodigy ADAM [Mishchenko and Defazio, 2024], DoG [Ivgi et al., 2023], and DoWG [Khaled et al., 2023] over different initial parameters. We compare the quality of posterior approximations over different coreset sizes M and weight optimization procedures. Following Chen and Campbell [2024], we set the number of Markov chains to $K = 2$ and subsample size to $S = M$ in Eq. (4). We set κ_w to the hit-and-run slice sampler with doubling [Bélisle et al., 1993, Neal, 2003] for all real data experiments. For the Gaussian location model, we use a kernel that directly samples from π_w [Chen and Campbell, 2024, Sec. 3.4]; for the sparse regression example, we use Gibbs sampling [George and McCulloch, 1993].

We compare these algorithms using six different Bayesian models, the details of which are in Appendix A. We use Stan [Carpenter et al., 2017] to obtain full data inference results for real data experiments, and Gibbs sampling [George and McCulloch, 1993] for the sparse regression model with discrete variables. For all experiments, we measure the posterior approximation quality using the average squared z-score, which we define as

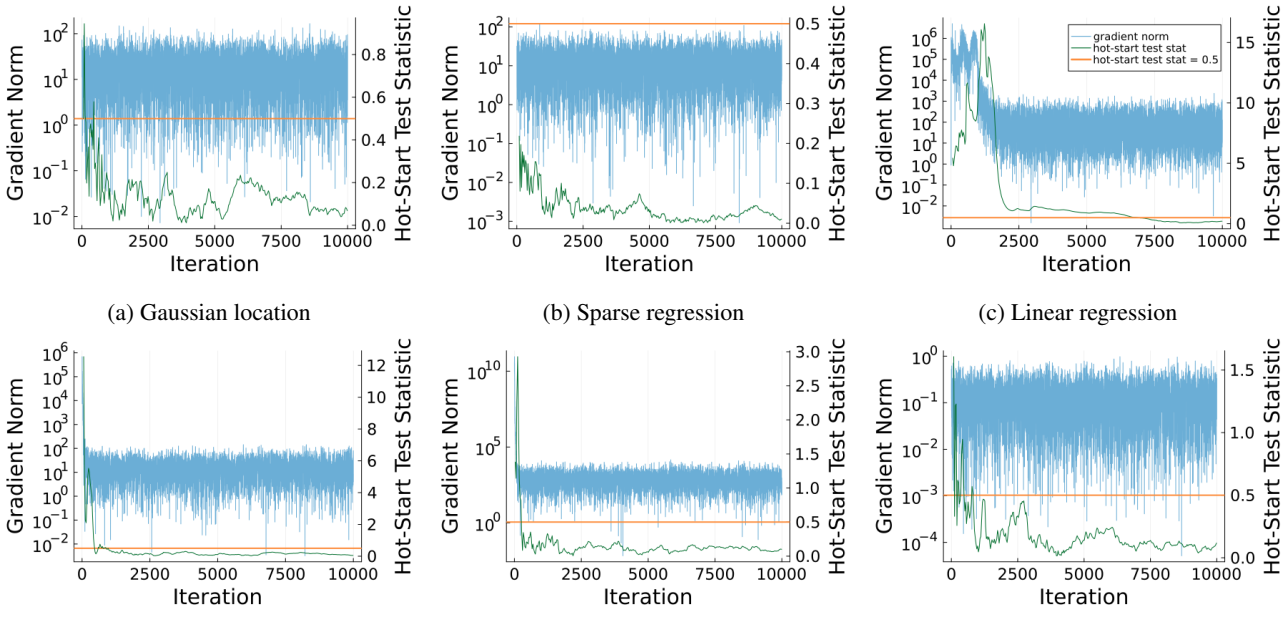
$$\frac{1}{D}\sum_{i=1}^{D}\left(\frac{\mu_i - \hat{\mu}_i}{\sigma_i}\right)^2. \tag{6}$$



(a) Gaussian location (b) Sparse regression (c) Linear regression

(d) Logistic regression (e) Poisson regression (f) Bradley-Terry

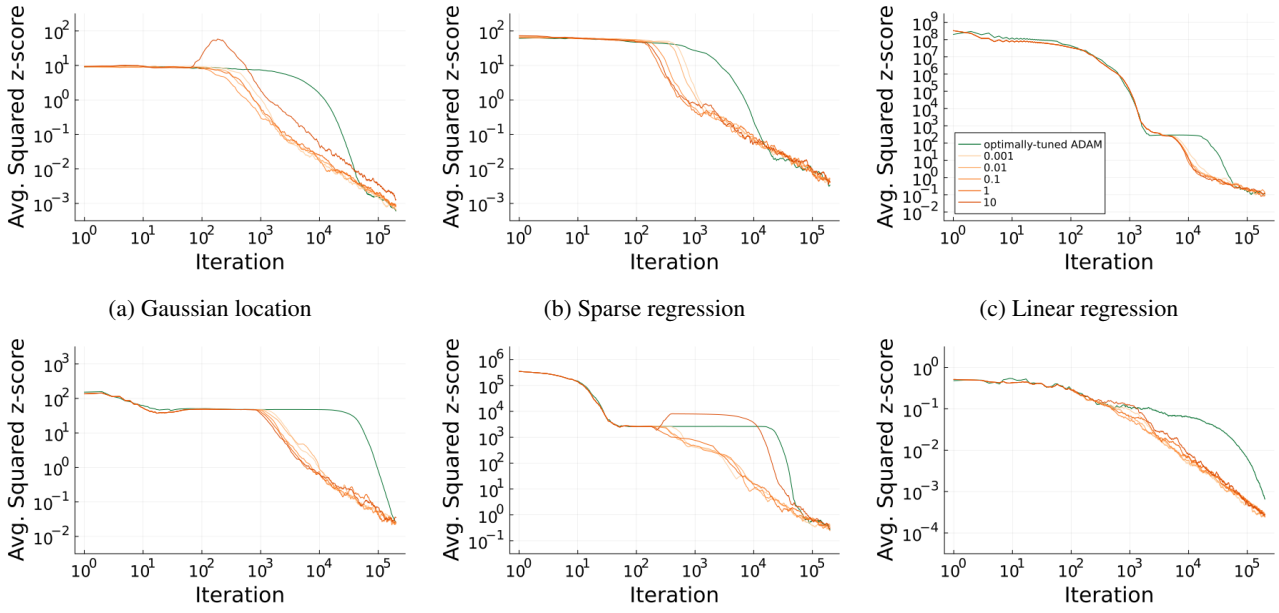
Figure 4: Traces of average squared coordinate-wise z-scores between the true and approximated posterior across all experiments, obtained using Hot DoG with and without hot-start test. All figures share the legend in Fig. 4c. The coreset size M is 1000 and each line represents a different initial learning rate parameter. The lines indicate the median from 10 runs. Orange lines indicate runs with hot-start test and blue lines without.



(a) Gaussian location (b) Sparse regression (c) Linear regression

(d) Logistic regression (e) Poisson regression (f) Bradley-Terry

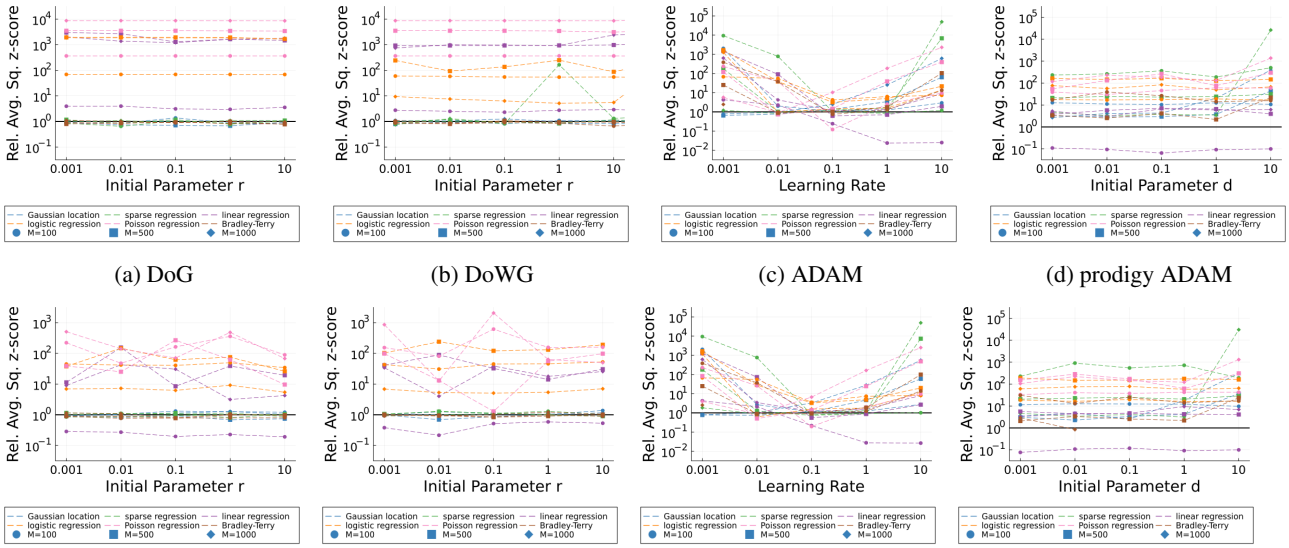
Figure 5: Trace of gradient estimate norms (blue) and hot-start test statistics (green) before weight optimization across all experiments with $M = 1000$. The orange horizontal line is the test statistic threshold $c = 0.5$.



(a) Gaussian location (b) Sparse regression (c) Linear regression

(d) Logistic regression (e) Poisson regression (f) Bradley-Terry

Figure 6: Traces of average squared coordinate-wise z-scores between the true and approximated posterior across all experiments, obtained from Hot DoG and optimally-tuned ADAM. All figures share the legend in Fig. 6c. The coreset size $M = 1000$ and each line represents a different initial learning rate parameter. The lines indicate the median from 10 runs.



(a) DoG (b) DoWG (c) ADAM (d) prodigy ADAM

(e) DoG with hot-start (f) DoWG with hot-start (g) ADAM with hot-start (h) prodigy ADAM with hot-start

Figure 7: Relative Coreset MCMC posterior approximation error comparing different optimization algorithms (labeled in the subfigure captions) and the proposed Hot DoG method (with fixed $r = 0.001$ and $c = 0.5$). The metric plotted is the ratio of average squared z-scores (defined in Eq. (6)) under the algorithm labeled in each subfigure caption to those under Hot DoG. Values above the horizontal black line (10^0) indicate that the proposed Hot DoG method outperformed the method it compared to. Median values after $200,000$ optimization iterations across 10 trials are used for the relative comparison for a variety of datasets, models, and coreset sizes.

In the above definition, D denotes the dimension of Θ; μ_i and σ_i are, respectively, the coordinate-wise mean and standard deviation estimated using the full data posterior, and $\hat{\mu}_i$ is the coordinate-wise mean estimated using draws from Coreset MCMC. This estimate is computed in a streaming fashion using the second half of all draws at the time; note this includes draws from π_{w_0} before the hot-start test passes.

Each algorithm was run on 8 single-threaded cores of a 2.1GHz Intel Xeon Gold 6130 processor with 32GB memory. Code for these experiments is available at `https://github.com/NaitongChen/automated-coreset-mcmc-experiments`. More experimental details and additional plots are in Appendices A and B.

Effect of hot-start test. Fig. 4 compares Hot DoG with and without the hot-start test for $M = 1000$ across all experiments; the same plots for other coreset sizes can be found in Appendix B. Without the hot-start test, the traces often hit a long plateau, before the effect of exponentially-weighted averaging is able to decay early large gradient norms. On the other hand, with burn-in, we begin by simulating from Markov chains targeting π_{w_0}, and start optimizing the coreset weights only after the hot-start test has passed. In terms of the number of log potential evaluations, Hot DoG with burn-in leaves the plateau sooner than without burn-in.

Fig. 5 examines the behaviour of the hot-start test in more detail, showing the traces of the gradient estimate norms $\|\hat{g}_t\|$ and test statistics $\texttt{median}(u_1, \ldots, u_K)$ across optimization iterations when using Hot DoG. Here we only show plots for $M = 1000$; the same plots for other coreset sizes can be found in Appendix B. In some experiments, the Markov chains are initialized reasonably well where the gradient norms are already stabilized, and the test passes almost immediately. In others, the Markov chains are initialized poorly and the gradient norms are large, but nevertheless, the hot-start test passes shortly after they stabilize. Across all experiments, a test statistic threshold of 0.5 worked well.

Robustness to fixed parameter r. Figure 6 provides an examination of the robustness of the proposed method to the fixed initial learning rate parameter r. Across all experiments, different values of r spanning multiple orders of magnitude result in similar posterior approximations across optimization iterations. Note that M is 1000 for all plots in Fig. 6. The same trends can be observed over different coreset sizes (see Appendix B). In practice, we follow the recommendation of Ivgi et al. [2023] and set $r = 0.001$.

Comparison with other related methods. Figure 7 shows a comparison between our method and DoG, DoWG, ADAM, as well as prodigy ADAM. We fix $r = 0.001$ and $c = 0.5$ for Hot DoG. Since the hot-start test itself can be applied to all methods, Hot DoG is compared against others both with and without burn-in. The posterior approximation quality of Hot DoG is orders of magnitude better than all other methods in many settings tested, and remain competitive otherwise. In particular, Hot DoG is capable of matching the performance of optimally-tuned ADAM without tuning.

6 CONCLUSION

This paper introduced Hot DoG, a learning-rate-free stochastic gradient method designed for learning coreset weights using Coreset MCMC. Our method extends DoG, but includes adjustments tailored to the Markovian setting of Coreset MCMC. In particular, Hot DoG includes a hot-start test detecting when to start training coreset weights as well as acceleration techniques. Our method is shown to produce coreset weights that converge to the optimum. The quality of coresets constructed by Hot DoG and their corresponding posterior approximation are robust to input parameters. Empirically, Hot DoG under our recommended setting ($r = 0.001$ and $c = 0.5$) produces better posterior approximations than other learning-rate-free stochastic gradient methods, and is competitive to optimally-tuned ADAM.

Acknowledgements

T.C. and N.C. were supported by an NSERC Discovery Grant RGPIN-2019-03962, and J.H.H. was partially supported by a National Science Foundation CAREER award IIS-2340586. We acknowledge the use of the ARC Sockeye computing platform from the University of British Columbia.

References

Claude Bélisle, Edwin Romeijn, and Robert Smith. Hit-and-run algorithms for generating multivariate distributions. *Mathematics of Operations Research*, 18(2):255–266, 1993.

Trevor Campbell. General bounds on the quality of Bayesian coresets. In *Advances in Neural Information Processing Systems*, 2024.

Trevor Campbell and Boyan Beronov. Sparse variational inference: Bayesian coresets from scratch. In *Advances in Neural Information Processing Systems*, 2019.

Yair Carmon and Oliver Hinder. Making SGD parameter-free. In *Conference on Learning Theory*, 2022.

Bob Carpenter, Andrew Gelman, Matthew Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. Stan: a probabilistic programming language. *Journal of Statistical Software*, 76(1):1—-32, 2017.

Naitong Chen and Trevor Campbell. Coreset Markov chain Monte Carlo. In *International Conference on Artificial Intelligence and Statistics*, 2024.

Naitong Chen, Zuheng Xu, and Trevor Campbell. Bayesian inference via sparse Hamiltonian flows. In *Advances in Neural Information Processing Systems*, 2022.

Aaron Defazio and Konstantin Mishchenko. Learning-rate-free learning by D-adaptation. In *International Conference on Machine Learning*, 2023.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12 (61):2121–2159, 2011.

Aprad Elo. *The Rating of Chessplayers, Past and Present*. Arco, 1st edition, 1978.

Andrew Gelman, John Carlin, Hal Stern, David Dunson, Aki Vehtari, and Donald Rubin. *Bayesian Data Analysis*. CRC Press, 3rd edition, 2013.

Edward George and Robert McCulloch. Variable selection via Gibbs sampling. *Journal of the American Statistical Association*, 88(423):881–889, 1993.

Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. Neural networks for machine learning lecture 6a: overview of mini-batch gradient descent, 2012.

Jonathan Huggins, Trevor Campbell, and Tamara Broderick. Coresets for scalable Bayesian logistic regression. In *Advances in Neural Information Processing Systems*, 2016.

Maor Ivgi, Oliver Hinder, and Yair Carmon. Dog is SGD's best friend: a parameter-free dynamic step size schedule. In *International Conference on Machine Learning*, 2023.

Ahmed Khaled, Konstantin Mishchenko, and Chi Jin. DoWG unleashed: an efficient universal parameter-free gradient descent method. In *Advances in Neural Information Processing Systems*, 2023.

Diederik Kingma and Jimmy Ba. Adam: a method for stochastic optimization. In *International Conference on Learning Representations*, 2014.

Nicolas Loizou, Sharan Vaswani, Issam Laradji, and Simon Lacoste-Julien. Stochastic polyak step-size for SGD: an adaptive learning rate for fast convergence. In *International Conference on Artificial Intelligence and Statistics*, 2021.

Konstantin Mishchenko and Aaron Defazio. Prodigy: an expeditiously adaptive parameter-free learner. In *International Conference on Machine Learning*, 2024.

Cian Naik, Judith Rousseau, and Trevor Campbell. Fast Bayesian coresets via subsampling and quasi-Newton refinement. In *Advances in Neural Information Processing Systems*, 2022.

Radford Neal. Slice sampling. *The Annals of Statistics*, 31 (3):705–767, 2003.

Francesco Orabona and Ashok Cutkosky. International Conference on Machine Learning tutorial on parameter-free stochastic optimization, 2020.

Courtney Paquette and Katya Scheinberg. A stochastic line search method with expected complexity analysis. *Society for Industrial and Applied Mathematics Journal on Optimization*, 30(1):349–376, 2020.

Christian Robert and George Casella. *Monte Carlo Statistical Methods*. Springer, 2nd edition, 2004.

Christian Robert and George Casella. A short history of Markov chain Monte Carlo: subjective recollections from incomplete data. *Statistical Science*, 26(1):102–115, 2011.

Noveen Sachdeva and Julian McAuley. Data distillation: a survey. *Transactions on Machine Learning Research*, 2023.

Sharan Vaswani, Aaron Mishkin, Issam Laradji, Mark Schmidt, Gauthier Gidel, and Simon Lacoste-Julien. Painless stochastic gradient: interpolation, line-search, and convergence rates. In *Advances in Neural Information Processing Systems*, 2019.

Aki Vehtari, Andrew Gelman, Daniel Simpson, Bob Carpenter, and Paul-Christian Bürkner. Rank-normalization, folding, and localization: an improved $\widehat{R}$ for assessing convergence of MCMC (with discussion). *Bayesian Analysis*, 16(2):667–718, 2021.

Tuning-Free Coreset Markov Chain Monte Carlo via Hot DoG (Supplementary Material)

Naitong Chen[1] **Jonathan H. Huggins**[2] **Trevor Campbell**[1]

[1]Department of Statistics, University of British Columbia, Vancouver, BC, Canada
[2]Department of Mathematics & Statistics and Faculty of Computing & Data Sciences, Boston University, Boston, MA, USA

A DETAILS OF EXPERIMENTS

A.1 MODEL SPECIFICATION

In this subsection, we describe the six examples (two synthetic and four real data) that we used for our experiments. Processed versions of all datasets used for the experiments are available at https://github.com/NaitongChen/automated-coreset-mcmc-experiments. For each of the regression models, we are given a set of points $(x_n, y_n)_{n=1}^N$, each consisting of features $x_n \in \mathbb{R}^p$ and response y_n.

Bayesian sparse linear regression: This is based on Example 4.1 from George and McCulloch [1993]. We use the model

$$\begin{aligned} \sigma^2 &\sim \text{InvGam}\left(\nu/2, \nu\lambda/2\right), \\ \forall i \in [p], \quad \gamma_i &\overset{\text{iid}}{\sim} \text{Bern}(q), \\ \beta_i \mid \gamma_i &\overset{\text{ind}}{\sim} \mathcal{N}\left(0, (\mathbb{1}(\gamma_i = 0)\tau + \mathbb{1}(\gamma_i = 1)c\tau)^2\right), \\ \forall n \in [N], \quad y_n \mid x_n, \beta, \sigma^2 &\overset{\text{ind}}{\sim} \mathcal{N}\left(x_n^\top \beta, \sigma^2\right), \end{aligned}$$

where we set $\nu = 0.1, \lambda = 1, q = 0.1, \tau = 0.1$, and $c = 10$. Here we model the variance σ^2, the vector of regression coefficients $\beta = \begin{bmatrix}\beta_1 & \dots & \beta_p\end{bmatrix}^\top \in \mathbb{R}^p$ and the vector of binary variables $\gamma = \begin{bmatrix}\gamma_1 & \dots & \gamma_p\end{bmatrix}^\top \in \{0,1\}^p$ indicating the inclusion of the p^{th} feature in the model. We set $N = 50{,}000$, $p = 10$, $\beta^\star = \begin{bmatrix}0 & 0 & 0 & 0 & 0 & 5 & 5 & 5 & 5 & 5\end{bmatrix}^\top$, and generate a synthetic dataset by

$$\begin{aligned} \forall n \in [N], \quad x_n &\overset{\text{iid}}{\sim} \mathcal{N}\left(0, I\right), \\ \epsilon_n &\overset{\text{iid}}{\sim} \mathcal{N}\left(0, 25^2\right), \\ y_n &= x_n^\top \beta^\star + \epsilon_n. \end{aligned}$$

Bayesian linear regression: We use the model

$$\begin{aligned} \begin{bmatrix}\beta & \log \sigma^2\end{bmatrix}^\top &\sim \mathcal{N}(0, I), \\ \forall n \in [N], y_n \mid x_n, \beta, \sigma^2 &\overset{\text{ind}}{\sim} \mathcal{N}\left(\begin{bmatrix}1 & x_n^\top\end{bmatrix}\beta, \sigma^2\right), \end{aligned}$$

where $\beta \in \mathbb{R}^{p+1}$ is a vector of regression coefficients and $\sigma^2 \in \mathbb{R}_+$ is the noise variance. Note that the prior here is not conjugate for the likelihood. The dataset consists of flight delay information from $N = 98{,}673$ observations and was constructed using flight delay data from https://www.transtats.bts.gov/Homepage.asp and historical weather information from https://www.wunderground.com/. We study the difference, in minutes, between the scheduled and actual departure times against $p = 10$ features including flight-specific and meteorological information.

Bayesian logistic regression: We use the model

$$\forall i \in [p+1], \quad \beta_i \overset{\text{iid}}{\sim} \text{Cauchy}(0,1),$$
$$\forall n \in [N], \quad y_n \overset{\text{ind}}{\sim} \text{Bern}\left(\left(1+\exp\left(-\begin{bmatrix}1 & x_n^\top\end{bmatrix}\beta\right)\right)^{-1}\right),$$

where $\beta = \begin{bmatrix}\beta_1 & \dots & \beta_{p+1}\end{bmatrix}^\top \in \mathbb{R}^{p+1}$ is a vector of regression coefficients. Here we use the same dataset as in linear regression, but instead model the relationship between whether a flight is cancelled using the same set of features. Note that of all flights included, only 0.058% were cancelled.

Bayesian Poisson regression: We use the model

$$\beta \sim \mathcal{N}(0, I),$$
$$\forall n \in [N], y_n \mid x_n, \beta \overset{\text{ind}}{\sim} \text{Poiss}\left(\log\left(1+e^{\begin{bmatrix}1 & x_n^\top\end{bmatrix}\beta}\right)\right),$$

where $\beta \in \mathbb{R}^{p+1}$ is a vector of regression coefficients. The dataset consists of $N = 15{,}641$ observations, and we model the hourly count of rental bikes against $p = 8$ features (e.g., temperature, humidity at the time, and whether or not the day is a workday). The original bike share dataset is available at https://archive.ics.uci.edu/dataset/275/bike+sharing+dataset.

The remaining two non-regression models are specified as follows.

Gaussian location: We use the model

$$\theta \sim \mathcal{N}(0, I),$$
$$\forall n \in [N], X_n \overset{\text{iid}}{\sim} \mathcal{N}(\theta, I),$$

where $\theta, X_n \in \mathbb{R}^d$. Here we model the mean θ. We set $N = 10{,}000, d = 20$ and generate a synthetic dataset by

$$\forall n \in [N], x_n \overset{\text{iid}}{\sim} \mathcal{N}(0, I).$$

Bradley-Terry model: We use the model

$$\theta \overset{\text{iid}}{\sim} \mathcal{N}(0, I),$$
$$\forall n \in [N], y_n \mid h_n, v_n, \theta \overset{\text{ind}}{\sim} \text{Bern}\left(\left(1+\exp\left((\theta_{v_n} - \theta_{h_n})/400\right)\right)^{-1}\right),$$

where $\theta \in \mathbb{R}^d$. The dataset was constructed using games statistics from https://www.nba.com/stats and consists of data of $N = 26,651$ NBA games between the 2004 and 2022 seasons. h_n and v_n are the home team and visitor team IDs for the n^{th} game in the dataset, and y_n denotes the outcome of the game ($y_n = 1$ if the home team won and $y_n = 0$ if the visitor team won). $\theta \in \mathbb{R}^d$ represents the Elo ratings or relative skill levels [Elo, 1978, Ch. 1] for each of the $d = 30$ teams. We model the Elo ratings using outcomes of pairwise comparisons between teams using game outcomes.

A.2 PARAMETER SETTINGS

For full-data inference results of all examples except for the sparse linear regression model, we ran Stan [Carpenter et al., 2017] with 10 parallel chains, each taking $100{,}000$ steps with the first $50{,}000$ discarded, for a combined $500{,}000$ draws. For full-data inference result of the sparse linear regression example, we use the Gibbs sampler developed by George and McCulloch [1993] to generate $200{,}000$ draws, with the first half discarded as burn-in.

To account for changes in w, for all real data experiments, we use the hit-and-run slice sampler with doubling [Bélisle et al., 1993, Neal, 2003]; for the Gaussian location model, we use a kernel that directly samples from π_w [Chen and Campbell, 2024, Sec. 3.4]. for the sparse regression, we use the Gibbs sampler developed by George and McCulloch [1993].

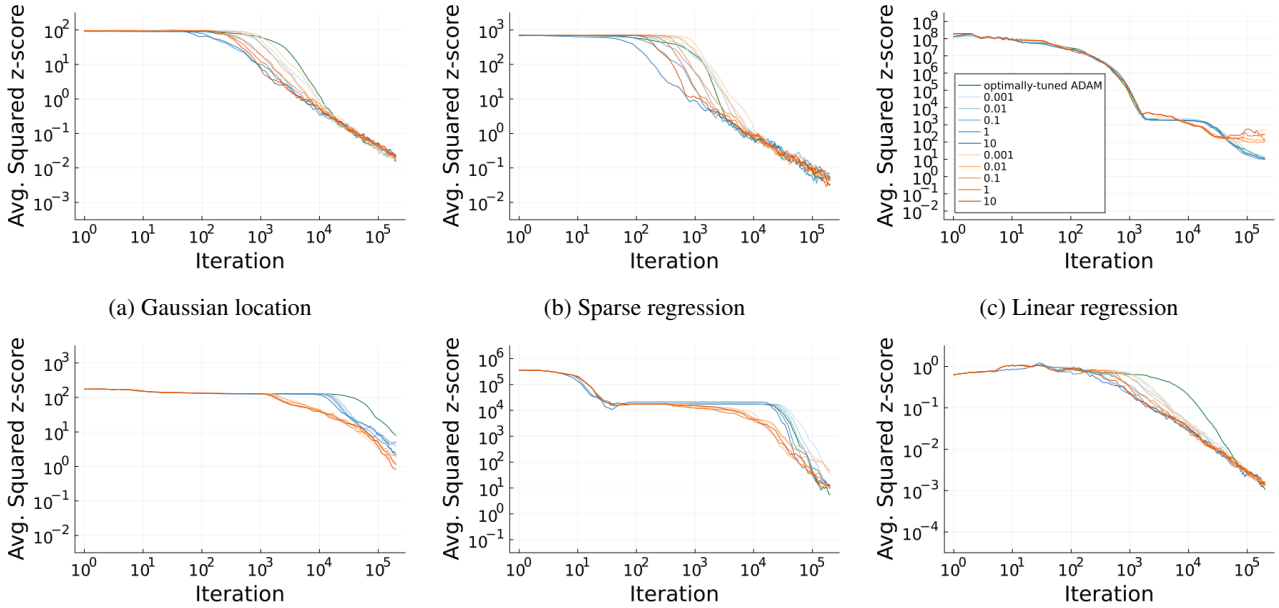
We use Stan [Carpenter et al., 2017] to obtain full data inference results for real data experiments, and Gibbs sampling [George and McCulloch, 1993] for the sparse regression model with discrete variables. The true posterior distribution for the Gaussian location model is available in closed form.

For ADAM, we test multiple learning rates over a log scale grid $\left(10^k\right)$ for $k = -3, -2, \ldots, 1$. For each experiment under each coreset size, the optimally-tuned ADAM is the one that obtained the lowest average squared z-score after $200,000$ iterations of weight optimization. For all learning-rate-free methods, we test different initial parameters (initial lower bound for prodigy ADAM and r_0 for Hot DoG, DoG, and DoWG) over a log scaled grid $\left(10^k\right)$ for $k = -3, -2, \ldots, 1$.

For the logistic regression example, to account for the class imbalance problem, we include all observations from the rare positive class if the coreset size is more than twice as big as the total number of observations with positive labels. Otherwise we sample our coreset to have 50% positive labels and 50% negative labels. Coreset points are uniformly subsampled for all other models.

B ADDITIONAL RESULTS

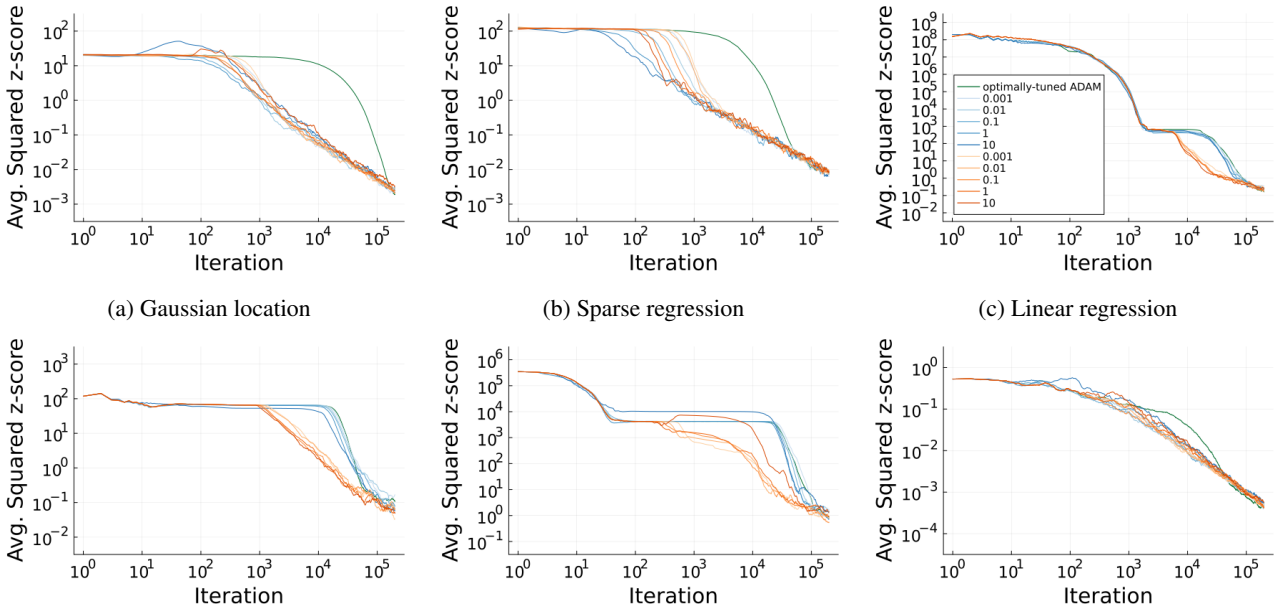
Figs. 4 to 6 in the main text show the traces of average squared coordinate-wise z-scores, as well as the gradient estimate norms and hot-start test statistics for Hot DoG when $M = 1000$. In this subsection, we show the same sets of plots for $M = 100$ and $M = 500$. Similarly to Fig. 4, Figs. 8 and 9 compare Hot DoG with and without hot-start test. Similarly to Fig. 5, Figs. 10 and 11 show the gradient estimate norms and hot-start test statistics during burn-in. Similarly to Fig. 6, Figs. 12 and 13 compare Hot DoG (with hot-start test) and optimally-tuned ADAM. We see that all plots show the same trends as the ones in Section 5, where $M = 1000$. As a result, we arrive at similar observations as in Section 5. In particular, Hot DoG with burn-in leaves the plateau sooner than without burn-in; the hot-start test passes and thus burn-in terminates shortly after gradient norms are stabilized; Hot DoG is robust to the fixed parameter r.



(a) Gaussian location (b) Sparse regression (c) Linear regression

(d) Logistic regression (e) Poisson regression (f) Bradley-Terry

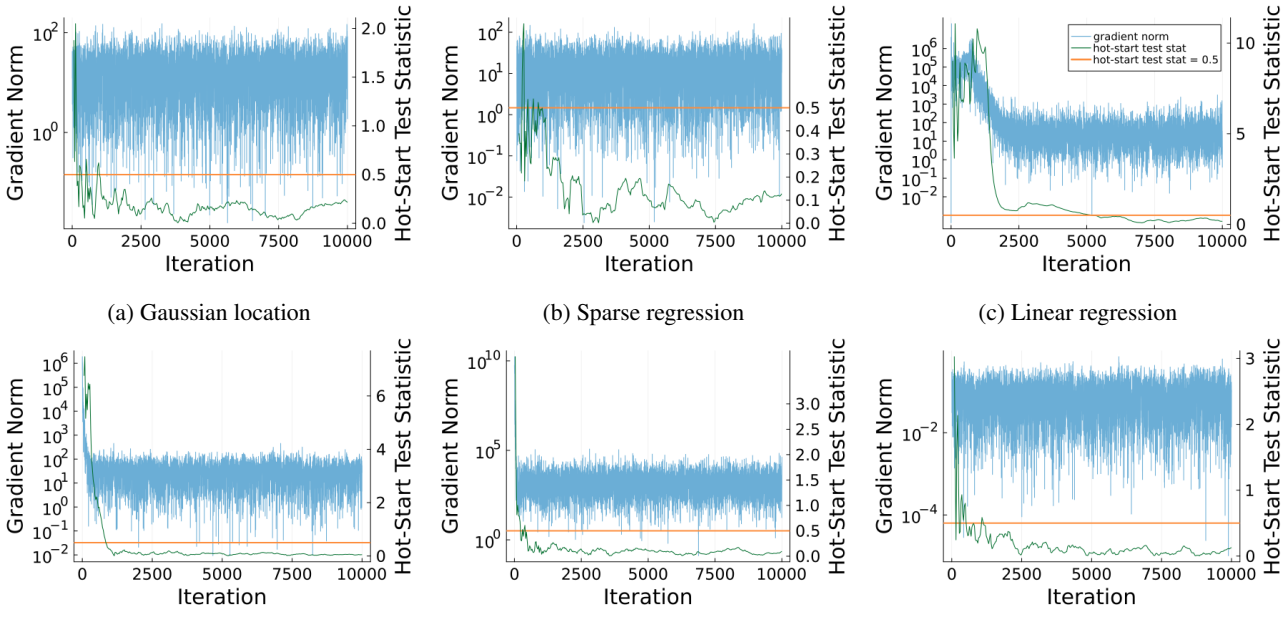
Figure 8: Traces of average squared coordinate-wise z-scores between the true and approximated posterior across all experiments, obtained using Hot DoG with and without hot-start test. All figures share the legend in Fig. 8c. The coreset size M is 100 and each line represents a different initial learning rate parameter. The lines indicate the median from 10 runs. Orange lines indicate runs with hot-start test and blue lines without.



(a) Gaussian location (b) Sparse regression (c) Linear regression

(d) Logistic regression (e) Poisson regression (f) Bradley-Terry

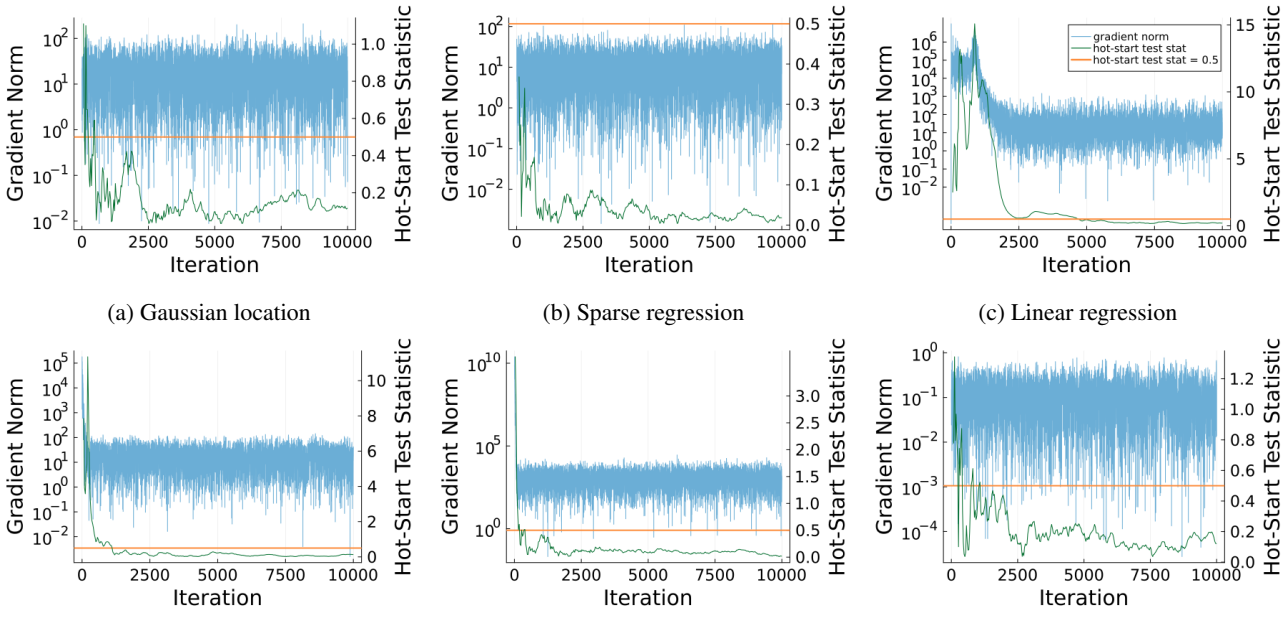
Figure 9: Traces of average squared coordinate-wise z-scores between the true and approximated posterior across all experiments, obtained using Hot DoG with and without hot-start test. All figures share the legend in Fig. 9c. The coreset size M is 500 and each line represents a different initial learning rate parameter. The lines indicate the median from 10 runs. Orange lines indicate runs with hot-start test and blue lines without.



(a) Gaussian location (b) Sparse regression (c) Linear regression

(d) Logistic regression (e) Poisson regression (f) Bradley-Terry

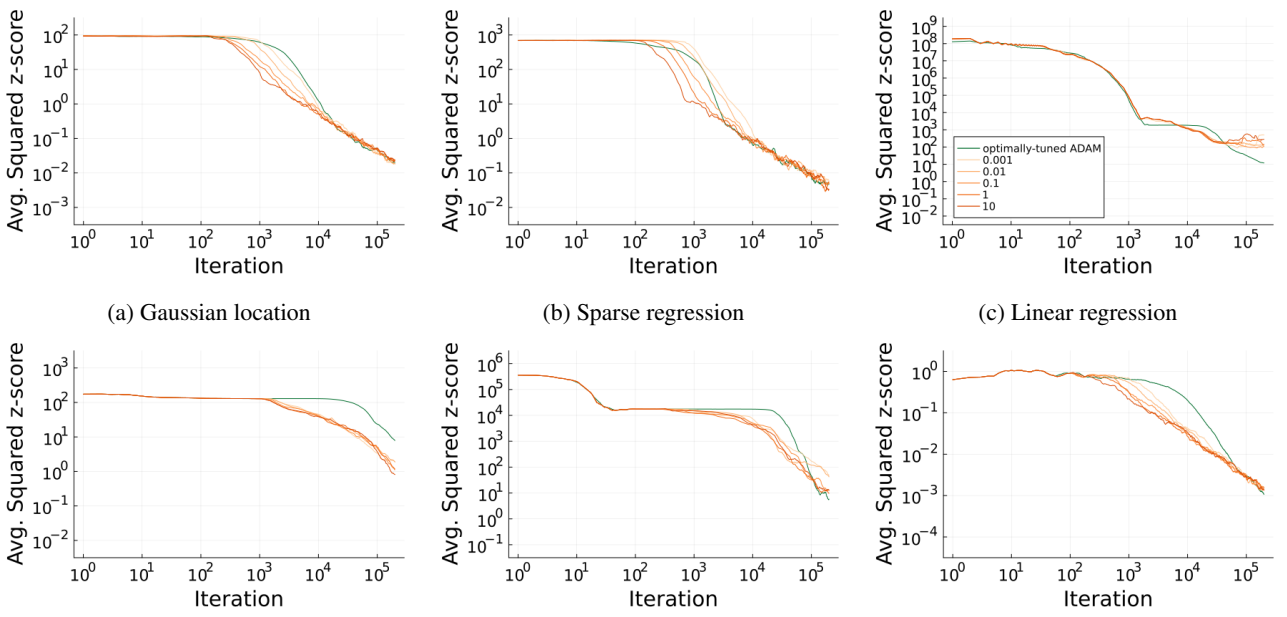
Figure 10: Trace of gradient estimate norms (blue) and hot-start test statistics (green) before weight optimization across all experiments with $M = 100$. The orange horizontal line is the test statistic threshold $c = 0.5$.



(a) Gaussian location (b) Sparse regression (c) Linear regression

(d) Logistic regression (e) Poisson regression (f) Bradley-Terry

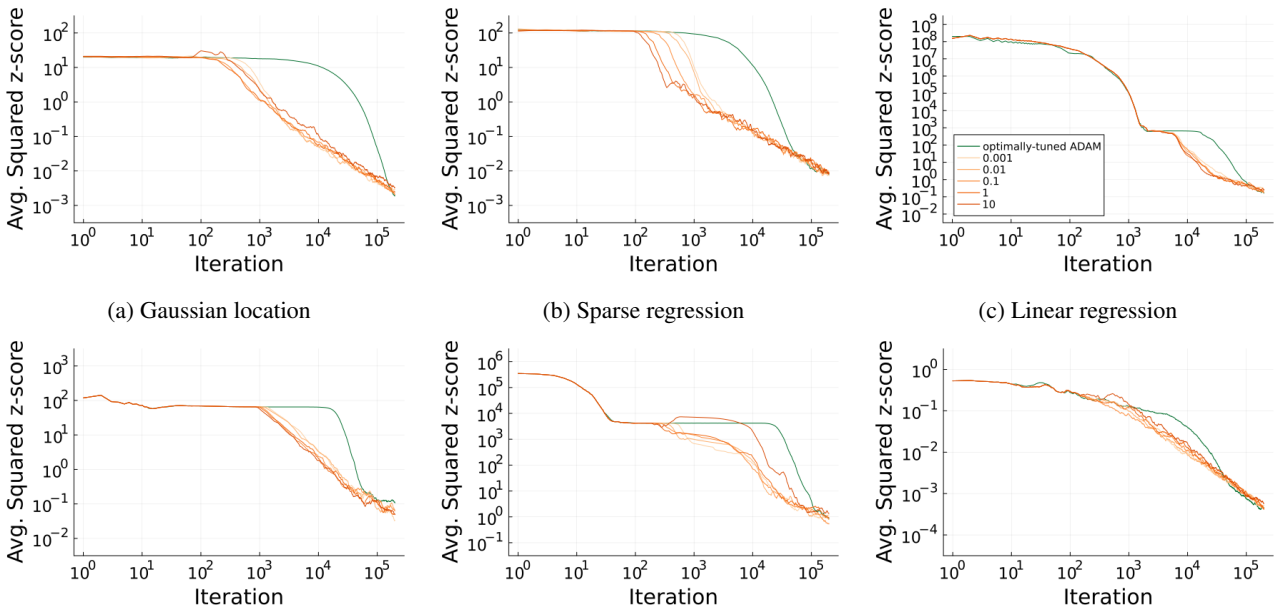
Figure 11: Trace of gradient estimate norms (blue) and hot-start test statistics (green) before weight optimization across all experiments with $M = 500$. The orange horizontal line is the test statistic threshold $c = 0.5$.



(a) Gaussian location (b) Sparse regression (c) Linear regression

(d) Logistic regression (e) Poisson regression (f) Bradley-Terry

Figure 12: Traces of average squared coordinate-wise z-scores between the true and approximated posterior across all experiments, obtained from Hot DoG and optimally-tuned ADAM. All figures share the legend in Fig. 12c. The coreset size $M = 100$ and each line represents a different initial learning rate parameter. The lines indicate the median from 10 runs.



(a) Gaussian location (b) Sparse regression (c) Linear regression

(d) Logistic regression (e) Poisson regression (f) Bradley-Terry

Figure 13: Traces of average squared coordinate-wise z-scores between the true and approximated posterior across all experiments, obtained from Hot DoG and optimally-tuned ADAM. All figures share the legend in Fig. 13c. The coreset size $M = 500$ and each line represents a different initial learning rate parameter. The lines indicate the median from 10 runs.

C HOT DOG CONVERGENCE

C.1 UPDATE RULE

The Hot DoG update for coordinate $i \in [M]$ at iteration $n \in \mathbb{N}_+$ can be written as

$$m_{t,i} = \beta_1 m_{t-1,i} + g_i(w_{t-1}, \theta_{t-1}, \mathcal{S}_{t-1}) \tag{7}$$

$$v_{t,i} = \beta_2 v_{t-1,i} + (g_i(w_{t-1}, \theta_{t-1}, \mathcal{S}_{t-1}))^2 \tag{8}$$

$$w_{t,i} = w_{t-1,i} - \alpha_{t,i} \frac{m_{t,i}}{\sqrt{t \cdot (\epsilon + v_{t,i})}},$$

where

$$\alpha_{t,i} = \frac{1-\beta_1}{1-\beta_1^t} \frac{\sqrt{1-\beta_2^t}}{\sqrt{1-\beta_2}} \tilde{r}_{t,i}$$

$$\tilde{r}_{t,i} = (1-\beta_1^t)^{-1} \left((1-\beta_1) \left(\sum_{k=0}^{t-1} \beta_1^k \bar{r}_{t-k,i} \right) + \beta_1^t \bar{r}_{0,i} \right)$$

$$\bar{r}_{t,i} = \left(\max_{k \le t} \{ |w_{t,i} - w_{0,i}| \} \right) \vee r_\delta.$$

We initialize the algorithm such that $m_0 = 0$ and $v_0 = 0$. Define $s_t \in \mathbb{R}^N$ such that $\forall j \in \mathcal{S}_t, s_{tj} = \frac{N}{S}$ and 0 otherwise. The M-dimensional subsampled gradient estimate as defined in Eq. (4) then takes the form

$$g(w_{t-1}, \theta_{t-1}, \mathcal{S}_{t-1}) = G_{t-1}(w_{t-1} - w^\star) + H_{t-1}(1 - s_{t-1}), \tag{9}$$

where

$$G_{t-1} = \frac{1}{K-1} \sum_{k=1}^{K} \begin{bmatrix} \bar{\ell}_1(\theta_{(t-1)k}) \\ \vdots \\ \ell_M(\theta_{(t-1)k}) \end{bmatrix} \begin{bmatrix} \bar{\ell}_1(\theta_{(t-1)k}) \\ \vdots \\ \ell_M(\theta_{(t-1)k}) \end{bmatrix}^\top \in \mathbb{R}^{M \times M}, H_{t-1} = \frac{1}{K-1} \sum_{k=1}^{K} \begin{bmatrix} \bar{\ell}_1(\theta_{(t-1)k}) \\ \vdots \\ \bar{\ell}_M(\theta_{(t-1)k}) \end{bmatrix} \begin{bmatrix} \bar{\ell}_1(\theta_{(t-1)k}) \\ \vdots \\ \bar{\ell}_N(\theta_{(t-1)k}) \end{bmatrix}^\top \in \mathbb{R}^{M \times N}.$$

Note that in Eq. (9), both matrix-vector products on the right hand side give us vectors of dimension M, which aligns with the desired dimension of the gradient estimate. We also define here two quantities that improves the readability of proofs presented in following subsections:

$$R_{t-1} = \left[\frac{\alpha_{t,1}}{\sqrt{t \cdot (\epsilon + v_{t,1})}} \quad \cdots \quad \frac{\alpha_{t,M}}{\sqrt{t \cdot (\epsilon + v_{t,M})}} \right]^\top, \qquad \Delta_{t-j} = w_{t-j-1} - w_{t-j} = \alpha_{t-j} \odot \frac{m_{t-j}}{\sqrt{(t-j) \cdot (\epsilon + v_{t-j})}}. \tag{10}$$

C.2 ASSUMPTIONS

Assumption C.1 (Coreset weight constraint)**.** $\mathcal{W} = \{w \in \mathbb{R}^M : w_t \ge 0, \sum_{m=1}^M w_{tm} \le B\}$.

Assumption C.2 (Exact coreset)**.** *There exists a* $w^\star \in \mathbb{R}^M, c^\star \in \mathbb{R}$ *such that* $w^\star \in \mathcal{W}$ *and*

$$\sum_{n=1}^{N} \ell_n(\cdot) = \sum_{m=1}^{M} w_m^\star \ell_m(\cdot) + c^\star \quad \pi_0 - a.e.v.$$

Assumption C.3 (Bounded gradient)**.** *There exists* $U > 0$ *such that*

$$\forall w_t \in \mathcal{W}, \theta_t \in \Theta^K, \mathcal{S}_t \subseteq [N] \quad \|g(w_t, \theta_t, \mathcal{S}_t)\|_\infty \le U.$$

Assumption C.4 (Markov gradient mixing)**.** *There exists* $\lambda > 0$ *such that*

$$\forall w_t \in \mathcal{W}, \theta_{t-1} \in \Theta^K \quad \mathbb{E}\left[G_t | w_t, \theta_{t-1}\right] \succeq \lambda I.$$

Assumption C.5 (Markov gradient noise boundedness)**.** *There exists* $0 < \bar{\lambda} < \infty$ *such that*

$$\forall w_t, w_{t-j} \in \mathcal{W}, \theta_{t-1}, \theta_{t-j-i} \in \Theta^K \quad \mathbb{E}\left[G_{t-j}^\top G_t \middle| w_t, \theta_{t-1}, w_{t-j}, \theta_{t-j-1}\right] \preceq \bar{\lambda} I.$$

C.3 CONVERGENCE PROOF

Proof of Theorem 4.1. We begin by applying the projected gradient update to get

$$\begin{aligned}\|w_t - w^\star\|^2 &= \left\| \mathrm{proj}_{\mathcal{W}} \left(w_{t-1} - \alpha_t \odot \frac{m_t}{\sqrt{t \cdot (\epsilon + v_t)}} \right) - w^\star \right\|^2 \\ &= \left\| \mathrm{proj}_{\mathcal{W}} \left(w_{t-1} - \alpha_t \odot \frac{m_t}{\sqrt{t \cdot (\epsilon + v_t)}} \right) - \mathrm{proj}_{\mathcal{W}}\, w^\star \right\|^2 \\ &\leq \left\| w_{t-1} - \alpha_t \odot \frac{m_t}{\sqrt{t \cdot (\epsilon + v_t)}} - w^\star \right\|^2 . \end{aligned} \tag{11}$$

Here $\odot$ denotes element-wise multiplication, and the fraction $\frac{m_t}{\sqrt{t\cdot(\epsilon+v_t)}}$ is also applied element-wise. The second equality follows because $w^\star \in \mathcal{W}$ by assumption. The inequality follows because $\mathcal{W}$ defined in Assumption C.1 is convex and closed, and hence $\mathrm{proj}_{\mathcal{W}}$ is a contraction. We unroll m_t by Eq. (7) and use R_{t-1} as defined in Eq. (10) to get

$$\alpha_t \odot \frac{m_t}{\sqrt{t \cdot (\epsilon + v_t)}} = \mathrm{diag}(R_{t-1}) \sum_{k=0}^{t-1} \beta_1^k g(w_{t-k-1}, \theta_{t-k-1}, \mathcal{S}_{t-k-1}). \tag{12}$$

By substituting Eqs. (9) and (12) into Eq. (11) and taking expectations on both sides, we get

$$\begin{aligned}&\mathbb{E}\|w_t - w^\star\|^2 \\ &\leq \mathbb{E}\left[\left\| \left((w_{t-1} - w^\star) - \mathrm{diag}(R_{t-1}) \sum_{k=0}^{t-1} \beta_1^k G_{t-k-1}(w_{t-k-1} - w^\star) \right) - \left(\mathrm{diag}(R_{t-1}) \sum_{k=0}^{t-1} \beta_1^k H_{t-k-1}(1 - s_{t-k-1}) \right) \right\|^2 \right] \\ &= \mathbb{E}\left[\left\| (w_{t-1} - w^\star) - \mathrm{diag}(R_{t-1}) \sum_{k=0}^{t-1} \beta_1^k G_{t-k-1}(w_{t-k-1} - w^\star) \right\|^2 \right] \\ &\quad - 2\mathbb{E}\left[\left((w_{t-1} - w^\star) - \mathrm{diag}(R_{t-1}) \sum_{k=0}^{t-1} \beta_1^k G_{t-k-1}(w_{t-k-1} - w^\star) \right)^\top \left(\mathrm{diag}(R_{t-1}) \sum_{k=0}^{t-1} \beta_1^k H_{t-k-1}(1 - s_{t-k-1}) \right) \right] \\ &\quad + \mathbb{E}\left[\left\| \mathrm{diag}(R_{t-1}) \sum_{k=0}^{t-1} \beta_1^k H_{t-k-1}(1 - s_{t-k-1}) \right\|^2 \right] \\ &= \mathbb{E}\left[\left\| (w_{t-1} - w^\star) - \mathrm{diag}(R_{t-1}) \sum_{k=0}^{t-1} \beta_1^k G_{t-k-1}(w_{t-k-1} - w^\star) \right\|^2 \right] + \mathbb{E}\left[\left\| \mathrm{diag}(R_{t-1}) \sum_{k=0}^{t-1} \beta_1^k H_{t-k-1}(1 - s_{t-k-1}) \right\|^2 \right]. \end{aligned} \tag{13}$$

In the above, the last equality follows due to unbiased subsampling, i.e., for all t, $\mathbb{E}[1-s_t]=0$. We now rewrite the first term in Eq. (13) as follows:

$$\begin{aligned}
&\mathbb{E}\left[\left\|(w_{t-1}-w^\star)-\operatorname{diag}(R_{t-1})\sum_{k=0}^{t-1}\beta_1^k G_{t-k-1}(w_{t-k-1}-w^\star)\right\|^2\right]\\
&=\mathbb{E}\left[\left\|(w_{t-1}-w^\star)-\operatorname{diag}(R_{t-1})\sum_{k=0}^{t-1}\beta_1^k G_{t-k-1}(w_{t-k-1}-w^\star+w_{t-1}-w_{t-1})\right\|^2\right]\\
&=\mathbb{E}\left[\left\|(w_{t-1}-w^\star)-\operatorname{diag}(R_{t-1})\sum_{k=0}^{t-1}\beta_1^k G_{t-k-1}(w_{t-1}-w^\star)-\operatorname{diag}(R_{t-1})\sum_{k=0}^{t-1}\beta_1^k G_{t-k-1}(w_{t-k-1}-w_{t-1})\right\|^2\right]\\
&=\mathbb{E}\left[\left\|\left(I-\operatorname{diag}(R_{t-1})\sum_{k=0}^{t-1}\beta_1^k G_{t-k-1}\right)(w_{t-1}-w^\star)-\operatorname{diag}(R_{t-1})\sum_{k=1}^{t-1}\beta_1^k G_{t-k-1}\left(\sum_{j=1}^{k}\Delta_{t-j}\right)\right\|^2\right],
\end{aligned}$$

where Δ_{t-j} is as defined in Eq. (10). The last equality above follows by rewriting $w_{t-k-1}-w_{t-1}$ as a telescoping sum. Now let $A_t=\left(I-\operatorname{diag}(R_{t-1})\sum_{k=0}^{t-1}\beta_1^k G_{t-k-1}\right)$, $b_t=\operatorname{diag}(R_{t-1})\sum_{k=1}^{t-1}\beta_1^k G_{t-k-1}\left(\sum_{j=1}^{k}\Delta_{t-j}\right)$, and $c_t=\operatorname{diag}(R_{t-1})\sum_{k=0}^{t-1}\beta_1^k H_{t-k-1}(1-s_{t-k-1})$. Eq. (13) then becomes

$$\begin{aligned}
&\mathbb{E}\|w_t-w^\star\|^2\\
&\le\mathbb{E}\left[\|A_t(w_{t-1}-w^\star)-b_t\|^2\right]+\mathbb{E}\left[\|c_t\|^2\right]\\
&=\mathbb{E}\left[\mathbb{E}\left[(w_{t-1}-w^\star)^\top A_t^\top A_t(w_{t-1}-w^\star)-2b_t^\top A_t(w_{t-1}-w^\star)+b_t^\top b_t\middle|w_{t-1}\right]\right]+\mathbb{E}\left[\|c_t\|^2\right]\\
&\le\mathbb{E}\left[(w_{t-1}-w^\star)^\top\mathbb{E}\left[A_t^\top A_t\middle|w_{t-1}\right](w_{t-1}-w^\star)+2\left|\mathbb{E}\left[b_t^\top A_t(w_{t-1}-w^\star)\middle|w_{t-1}\right]\right|+\mathbb{E}\left[\|b_t\|^2\middle|w_{t-1}\right]\right]+\mathbb{E}\left[\|c_t\|^2\right]\\
&\le\mathbb{E}\left[(w_{t-1}-w^\star)^\top\mathbb{E}\left[A_t^\top A_t\middle|w_{t-1}\right](w_{t-1}-w^\star)+2\sqrt{\mathbb{E}\left[\|b_t\|^2|w_{t-1}\right]}\sqrt{\mathbb{E}\left[\|A_t(w_{t-1}-w^\star)\|^2|w_{t-1}\right]}\right]\\
&\quad+\mathbb{E}\left[\mathbb{E}\left[\|b_t\|^2\middle|w_{t-1}\right]\right]+\mathbb{E}\left[\|c_t\|^2\right]\\
&=\mathbb{E}\left[(w_{t-1}-w^\star)^\top\mathbb{E}\left[A_t^\top A_t\middle|w_{t-1}\right](w_{t-1}-w^\star)+2\sqrt{\mathbb{E}\left[\|b_t\|^2|w_{t-1}\right]}\sqrt{(w_{t-1}-w^\star)^\top\mathbb{E}\left[A_t^\top A_t\middle|w_{t-1}\right](w_{t-1}-w^\star)}\right]\\
&\quad+\mathbb{E}\left[\mathbb{E}\left[\|b_t\|^2\middle|w_{t-1}\right]\right]+\mathbb{E}\left[\|c_t\|^2\right],
\end{aligned}$$

where the last inequality is by Cauchy-Schwartz. By Lemmas C.6 to C.8, we know that there exists $T^\star<\infty$ and $C_1,C_2>0$ such that for all $t>T^\star$,

$$\mathbb{E}\left[A_t^\top A_t\middle|w_{t-1}\right]\preceq\exp\left(-\frac{D}{\sqrt{t}}\right)I,\quad\mathbb{E}\left[\|b_t\|^2\middle|w_{t-1}\right]\le\frac{C_1}{t^2},\quad\mathbb{E}\left[\|c_t\|^2\right]\le\frac{C_2}{t}.$$

Here $D=\frac{\lambda(1-\beta_1)r_\delta}{2\sqrt{\epsilon+(1-\beta_2)^{-1}U^2}}$ is as defined in Lemma C.6. We know $e^{-D/\sqrt{t}}\le 1$. By Assumption C.1, we also have that for all $t\ge 1$, $\|w_{t-1}-w^\star\|^2\le\sum_{m=1}^{M}B^2=MB^2$. Therefore,

$$\begin{aligned}
\mathbb{E}\|w_t-w^\star\|^2&\le e^{-D/\sqrt{t}}\mathbb{E}\|w_{t-1}-w^\star\|^2+2\mathbb{E}\left[\sqrt{\frac{C_1}{t^2}}\sqrt{\exp\left(-\frac{D}{\sqrt{t}}\right)\|w_{t-1}-w^\star\|^2}\right]+\frac{C_1}{t^2}+\frac{C_2}{t}\\
&\le e^{-D/\sqrt{t}}\mathbb{E}\|w_{t-1}-w^\star\|^2+2\frac{B\sqrt{MC_1}}{t}+\frac{C_1}{t^2}+\frac{C_2}{t}\\
&\le e^{-D/\sqrt{t}}\mathbb{E}\|w_{t-1}-w^\star\|^2+\frac{2B\sqrt{MC_1}+C_1+C_2}{t}.
\end{aligned}$$

We unroll this recursion backward from t to $T^\star$ to get

$$\begin{aligned}
\mathbb{E}\|w_t-w^\star\|^2&\le e^{-D\sum_{\tau=T^\star+1}^{t}\frac{1}{\sqrt{\tau}}}\mathbb{E}\left[\|w_{T^\star}-w^\star\|^2\right]+\left(2B\sqrt{MC_1}+C_1+C_2\right)\sum_{\tau=T^\star+1}^{t}\frac{1}{\tau}e^{-D\sum_{u=\tau+1}^{t}\frac{1}{\sqrt{u}}}\\
&\le MB^2e^{-D\sum_{\tau=T^\star+1}^{t}\frac{1}{\sqrt{\tau}}}+\left(2B\sqrt{MC_1}+C_1+C_2\right)\sum_{\tau=T^\star+1}^{t}\frac{1}{\tau}e^{-D\sum_{u=\tau+1}^{t}\frac{1}{\sqrt{u}}},
\end{aligned}$$

where the last inequality again uses $\|w_{T^\star} - w^\star\|^2 \leq = MB^2$. Since $\frac{1}{\sqrt{\tau}}$ monotonically decreases in τ, we have that $\sum_{\tau=T^\star+1}^{t} \frac{1}{\tau} \geq \int_{T^\star+1}^{t} \frac{1}{\sqrt{\tau}} d\tau = 2\left(\sqrt{t} - \sqrt{T^\star+1}\right)$. Therefore, as $t \to \infty$,

$$\begin{aligned}
\mathbb{E}\|w_t - w^\star\|^2 &\leq MB^2 e^{-2D(\sqrt{t}-\sqrt{T^\star+1})} + (2B\sqrt{MC_1} + C_1 + C_2) \sum_{\tau=T^\star+1}^{t} \frac{1}{\tau} e^{-2D(\sqrt{t}-\sqrt{\tau+1})} \\
&\leq MB^2 e^{2D\sqrt{T^\star+1}} e^{-2D\sqrt{t}} + \left(2B\sqrt{MC_1} + C_1 + C_2\right) e^{-2D\sqrt{t}} \sum_{\tau=1}^{t} \frac{1}{\tau} e^{2D\sqrt{\tau+1}} \\
&= O\left(e^{-2D\sqrt{t}} + e^{-2D\sqrt{t}} \sum_{\tau=1}^{t} \frac{1}{\tau} e^{2D\sqrt{\tau+1}}\right).
\end{aligned}$$

It is obvious that $e^{-2D\sqrt{t}} = O\left(\frac{1}{\sqrt{t}}\right)$ as $t \to \infty$. It remains to show that $e^{-2D\sqrt{t}} \sum_{\tau=1}^{t} \frac{1}{\tau} e^{2D\sqrt{\tau+1}} = O\left(\frac{1}{\sqrt{t}}\right)$ as $t \to \infty$. We begin by noting that, since $\forall \tau \geq 1$, $\frac{\tau+1}{\tau} \leq 2$,

$$\sum_{\tau=1}^{t} \frac{1}{\tau} e^{2D\sqrt{\tau+1}} = \sum_{\tau=1}^{t} \frac{1}{\tau} e^{2D\sqrt{\tau+1}} \frac{\tau}{\tau+1} \frac{\tau+1}{\tau} \leq 2 \sum_{\tau=1}^{t} \frac{1}{\tau+1} e^{2D\sqrt{\tau+1}} = 2 \sum_{\tau=2}^{t+1} \frac{1}{\tau} e^{2D\sqrt{\tau}}.$$

We can then equivalently show $2e^{-2D\sqrt{t}} \sum_{\tau=2}^{t+1} \frac{1}{\tau} e^{2D\sqrt{\tau}} = O\left(\frac{1}{\sqrt{t}}\right)$ as $t \to \infty$. We know that there exists $T' < \infty$ such that for all $\tau \geq T'$, $\frac{1}{\tau} e^{2D\sqrt{\tau}}$ monotonically increases with τ. We therefore split the sum at $g(t)$, with $T' \leq g(t) \leq t$, to get

$$2e^{-2D\sqrt{t}} \sum_{\tau=2}^{t+1} \frac{1}{\tau} e^{2D\sqrt{\tau}} = 2e^{-2D\sqrt{t}} \sum_{\tau=2}^{g(t)-1} \frac{1}{\tau} e^{2D\sqrt{\tau}} + 2e^{-2D\sqrt{t}} \sum_{\tau=g(t)}^{t+1} \frac{1}{\tau} e^{2D\sqrt{\tau}}. \tag{14}$$

We can bound the first term in Eq. (14) as follows

$$2e^{-2D\sqrt{t}} \sum_{\tau=2}^{g(t)-1} \frac{1}{\tau} e^{2D\sqrt{\tau}} \leq 2e^{2D\left(\sqrt{g(t)}-\sqrt{t}\right)} \sum_{\tau=2}^{g(t)-1} \frac{1}{\tau} \leq 2e^{2D\left(\sqrt{g(t)}-\sqrt{t}\right)} \left(\ln\left(g(t)\right) + 1\right). \tag{15}$$

Looking at the second term in Eq. (14), since $\tau \geq g(t)$ is large enough that the summand monotonically increases with τ,

$$2e^{-2D\sqrt{t}} \sum_{\tau=g(t)}^{t+1} \frac{1}{\tau} e^{2D\sqrt{\tau}} \leq 2e^{-2D\sqrt{t}} \int_{g(t)}^{t+1} \frac{e^{2D\sqrt{\tau}}}{\tau} d\tau = 4e^{-2D\sqrt{t}} \int_{\sqrt{g(t)}}^{\sqrt{t+1}} \frac{e^{2Ds}}{s} ds, \tag{16}$$

Where the last equality follows by setting $s = \sqrt{\tau}$, $\tau = s^2$, $d\tau = 2sds$. Now for the integral in Eq. (16), we integrate by parts by defining $y = \frac{1}{s}$ and $dv = e^{2Ds} ds$:

$$\begin{aligned}
\int_{\sqrt{g(t)}}^{\sqrt{t+1}} \frac{e^{2Ds}}{s} ds &= \frac{1}{2D\sqrt{t+1}} e^{2D\sqrt{t+1}} - \frac{1}{2D\sqrt{g(t)}} e^{2D\sqrt{g(t)}} + \frac{1}{2D} \int_{\sqrt{g(t)}}^{\sqrt{t+1}} \frac{e^{2Ds}}{s^2} ds \\
&\leq \frac{1}{2D\sqrt{t+1}} e^{2D\sqrt{t+1}} + \frac{e^{2D\sqrt{t+1}}}{2D} \int_{\sqrt{g(t)}}^{\sqrt{t+1}} \frac{1}{s^2} ds \\
&= \frac{1}{2D\sqrt{t+1}} e^{2D\sqrt{t+1}} + \frac{e^{2D\sqrt{t+1}}}{2D} \left(\frac{1}{\sqrt{g(t)}} - \frac{1}{\sqrt{t+1}}\right).
\end{aligned}$$

Substituting the above back into Eq. (16) to get

$$\begin{aligned}
2e^{-2D\sqrt{t}} \sum_{\tau=g(t)}^{t+1} \frac{1}{\tau} e^{2D\sqrt{\tau}} &\leq 4e^{-2D\sqrt{t}} \left(\frac{1}{2D\sqrt{t+1}} e^{2D\sqrt{t+1}} + \frac{e^{2D\sqrt{t+1}}}{2D} \left(\frac{1}{\sqrt{g(t)}} - \frac{1}{\sqrt{t+1}}\right)\right) \\
&= \frac{2}{D\sqrt{t+1}} e^{2D(\sqrt{t+1}-\sqrt{t})} + \frac{2}{D} e^{2D(\sqrt{t+1}-\sqrt{t})} \left(\frac{1}{\sqrt{g(t)}} - \frac{1}{\sqrt{t+1}}\right) \\
&\leq \frac{2e^{2D}}{D\sqrt{t+1}} + \frac{2e^{2D}}{D} \left(\frac{1}{\sqrt{g(t)}} - \frac{1}{\sqrt{t+1}}\right).
\end{aligned} \tag{17}$$

Let $g(t) = \frac{t}{2}$. Then $T' \leq g(t) \leq t$ is satisfied for all $t \geq 2T'$. We can then combine Eqs. (15) and (17), and have that for all $t \geq 2T'$,

$$2e^{-2D\sqrt{t}} \sum_{\tau=2}^{t+1} \frac{1}{\tau} e^{2D\sqrt{\tau}} \leq 2e^{2D\left(\sqrt{t/2}-\sqrt{t}\right)} \left(\ln\left(\frac{t}{2}\right)+1\right) + \frac{2e^{2D}}{D\sqrt{t+1}} + \frac{2e^{2D}}{D}\left(\frac{1}{\sqrt{t/2}} - \frac{1}{\sqrt{t+1}}\right)$$
$$= 2e^{-2D\left(1-\frac{1}{\sqrt{2}}\right)\sqrt{t}} \left(\ln t - \ln 2 + 1\right) + \frac{2e^{2D}}{D\sqrt{t+1}} + \frac{4e^{2D}}{D\sqrt{t}},$$

which is $O\left(\frac{1}{\sqrt{t}}\right)$ as $t \to \infty$. Therefore, we arrive at the desired result that as $t \to \infty$,

$$\mathbb{E}\|w_t - w^\star\|^2 \leq O\left(e^{-2D\sqrt{t}} + 2e^{-2D\left(1-\frac{1}{\sqrt{2}}\right)\sqrt{t}} \left(\ln t - \ln 2 + 1\right) + \frac{2e^{2D}}{D\sqrt{t+1}} + \frac{4e^{2D}}{D\sqrt{t}}\right) = O\left(\frac{1}{\sqrt{t}}\right).$$

□

C.4 USEFUL LEMMAS

We used several lemmas in the above proof of Theorem 4.1. In this subsection, we present the proof of these lemmas.

Lemma C.6. *Suppose Assumptions C.1 to C.5 hold. Define* $D = \frac{\lambda(1-\beta_1)r_\delta}{2\sqrt{\epsilon+(1-\beta_2)^{-1}U^2}}$. *There exists* $T < \infty$ *such that* $\forall t \geq T$,

$$\mathbb{E}\left[\left(I - \operatorname{diag}(R_{t-1})\sum_{k=0}^{t-1}\beta_1^k G_{t-k-1}\right)^\top \left(I - \operatorname{diag}(R_{t-1})\sum_{k=0}^{t-1}\beta_1^k G_{t-k-1}\right)\middle| w_{t-1}\right] \preceq \left(\exp\left(-\frac{D}{\sqrt{t}}\right)\right) I.$$

Proof of Lemma C.6. We begin by expanding the matrix product

$$\mathbb{E}\left[\left(I - \operatorname{diag}(R_{t-1})\sum_{k=0}^{t-1}\beta_1^k G_{t-k-1}\right)^\top \left(I - \operatorname{diag}(R_{t-1})\sum_{k=0}^{t-1}\beta_1^k G_{t-k-1}\right)\middle| w_{t-1}\right]$$
$$= I - 2\mathbb{E}\left[\operatorname{diag}(R_{t-1})\sum_{k=0}^{t-1}\beta_1^k G_{t-k-1}\middle| w_{t-1}\right] + \mathbb{E}\left[\left(\sum_{k=0}^{t-1}\beta_1^k G_{t-k-1}\right)^\top \operatorname{diag}(R_{t-1}^2)\left(\sum_{k=0}^{t-1}\beta_1^k G_{t-k-1}\right)\middle| w_{t-1}\right]. \quad (18)$$

We bound the first expectation from below and the second expectation from above.

We being by bounding $\mathbb{E}\left[\operatorname{diag}(R_{t-1})\sum_{k=0}^{t-1}\beta_1^k G_{t-k-1}\middle| w_{t-1}\right]$ from below. Following the update rule as specified in Appendix C.1, we expand the i^{th} entry of R_{t-1} as defined by Eq. (10) and get

$$R_{t-1,i} = \frac{\alpha_{t,i}}{\sqrt{t(\epsilon+v_{t,i})}} = \left(\frac{1-\beta_1}{1-\beta_1^t}\right)\left(\frac{\sqrt{1-\beta_2^t}}{\sqrt{1-\beta_2}}\right)\frac{1}{1-\beta_1^t}\left((1-\beta_1)\left(\sum_{k=0}^{t-1}\beta_1^k \bar{r}_{t-k,i}\right) + \beta_1^t \bar{r}_{0,i}\right)\frac{1}{\sqrt{t(\epsilon+v_{t,i})}}. \quad (19)$$

By Assumption C.3 and that $|\beta_2| < 1$, we can bound $v_{t,i}$ as defined in Eq. (8) by

$$v_{t,i} = \sum_{k=0}^{t-1}\beta_2^k g_i^2(w_{t-k-1}, \theta_{t-k-1}) \leq U^2 \sum_{k=0}^{t-1}\beta_2^k \leq U^2(1-\beta_2)^{-1}.$$

Together with $|\beta_1| < 1$ and that $\forall t, i$, $\bar{r}_{t,i} \geq r_\delta$,

$$R_{t-1,i} \geq \left(\frac{1-\beta_1}{1-\beta_1^t}\right)\left(\frac{\sqrt{1-\beta_2^t}}{\sqrt{1-\beta_2}}\right)\frac{r_\delta}{1-\beta_1^t}\frac{1}{\sqrt{t}\sqrt{\epsilon+(1-\beta_2)^{-1}U^2}} \geq \frac{(1-\beta_1)r_\delta}{\sqrt{t}\sqrt{\epsilon+(1-\beta_2)^{-1}U^2}}.$$

As a result, for all t, we have that $\text{diag}(R_{t-1}) \succeq \frac{(1-\beta_1)r_\delta}{\sqrt{t}\sqrt{\epsilon+(1-\beta_2)^{-1}U^2}} I$.

Now let $A = \text{diag}(R_{t-1}) - \frac{1}{2}\left(\min_{1\le i\le M} R_{t-1,i}\right) I$. We know A is diagonal and $A \succeq \frac{(1-\beta_1)r_\delta}{2\sqrt{t}\sqrt{\epsilon+(1-\beta_2)^{-1}U^2}} I$. We also know $Q \coloneqq \sum_{k=0}^{t-1} \beta_1^k G_{t-k-1} \succeq 0$ since G_t are sample covariance matrices. Together, using $\Lambda_{\min}$ to denote the minimum eigenvalue, we have $\Lambda_{\min}(AQ) = \Lambda_{\min}\left(A^{\frac{1}{2}} Q A^{\frac{1}{2}}\right) \ge 0$, and so $AQ \succeq 0$. Therefore,

$$\text{diag}(R_{t-1}) \sum_{k=0}^{t-1} \beta_1^k G_{t-k-1} \succeq \frac{1}{2}\left(\min_{1\le i\le M} R_{t-1,i}\right) \sum_{k=0}^{t-1} \beta_1^k G_{t-k-1} = \frac{(1-\beta_1)r_\delta}{2\sqrt{t}\sqrt{\epsilon+(1-\beta_2)^{-1}U^2}} \sum_{k=0}^{t-1} \beta_1^k G_{t-k-1}.$$

Using the above, we have that

$$\begin{aligned}
\mathbb{E}\left[\text{diag}(R_{t-1}) \sum_{k=0}^{t-1} \beta_1^k G_{t-k-1} \middle| w_{t-1}\right] &\succeq \frac{(1-\beta_1)r_\delta}{2\sqrt{t}\sqrt{\epsilon+(1-\beta_2)^{-1}U^2}} \sum_{k=0}^{t-1} \beta_1^k \mathbb{E}\left[G_{t-k-1}|w_{t-1}\right] \\
&= \frac{(1-\beta_1)r_\delta}{2\sqrt{t}\sqrt{\epsilon+(1-\beta_2)^{-1}U^2}} \sum_{k=0}^{t-1} \beta_1^k \mathbb{E}\left[\mathbb{E}\left[G_{t-k-1}|w_{t-k-1}, \theta_{t-k-2}\right]|w_{t-1}\right] \\
&\succeq \frac{\lambda(1-\beta_1)r_\delta}{2\sqrt{t}\sqrt{\epsilon+(1-\beta_2)^{-1}U^2}} \left(\sum_{k=0}^{t-1} \beta_1^k\right) I \\
&\succeq \frac{\lambda(1-\beta_1)r_\delta}{2\sqrt{t}\sqrt{\epsilon+(1-\beta_2)^{-1}U^2}} I, \qquad (20)
\end{aligned}$$

where the inequalities are due to Assumption C.4 and $|\beta_1| < 1$.

We now bound $\mathbb{E}\left[\left(\sum_{k=0}^{t-1} \beta_1^k G_{t-k-1}\right)^\top \text{diag}(R_{t-1}^2) \left(\sum_{k=0}^{t-1} \beta_1^k G_{t-k-1}\right) \middle| w_{t-1}\right]$. We similarly begin by bounding $R_{t-1,i}$ from the other direction. By Assumption C.1, $\bar{r}_{t,i} \le B$. Together with $v_{t,i} \ge 0$, and that $|\beta_1| < 1, |\beta_2| < 1$, we can bound Eq. (19) from above by

$$R_{t-1,i} \le \left(\frac{1-\beta_1}{1-\beta_1^t}\right)\left(\frac{\sqrt{1-\beta_2^t}}{\sqrt{1-\beta_2}}\right)\frac{B}{1-\beta_1^t}\frac{1}{\sqrt{t}\sqrt{\epsilon}} \le \frac{B}{t\epsilon(1-\beta_1)\sqrt{1-\beta_2}}. \qquad (21)$$

Again using $|\beta_1| < 1, |\beta_2| < 1$, and squaring $R_{t-1,i}$, we have that

$$\text{diag}(R_{t-1}^2) \preceq \frac{B^2}{t\epsilon(1-\beta_1)^2(1-\beta_2)} I. \qquad (22)$$

Therefore,

$$\begin{aligned}
&\mathbb{E}\left[\left(\sum_{k=0}^{t-1} \beta_1^k G_{t-k-1}\right)^\top \text{diag}(R_{t-1}^2) \left(\sum_{k=0}^{t-1} \beta_1^k G_{t-k-1}\right) \middle| w_{t-1}\right] \\
&\preceq \frac{B^2}{t\epsilon(1-\beta_1)^2(1-\beta_2)} \mathbb{E}\left[\left(\sum_{k=0}^{t-1} \beta_1^k G_{t-k-1}\right)^\top \left(\sum_{k=0}^{t-1} \beta_1^k G_{t-k-1}\right) \middle| w_{t-1}\right] \\
&= \frac{B^2}{t\epsilon(1-\beta_1)^2(1-\beta_2)} \sum_{k=0}^{t-1}\sum_{k'=0}^{t-1} \beta_1^k \beta_1^{k'} \mathbb{E}\left[G_{t-k-1}^\top G_{t-k'-1} \middle| w_{t-1}\right] \\
&= \frac{B^2}{t\epsilon(1-\beta_1)^2(1-\beta_2)} \sum_{k=0}^{t-1}\sum_{k'=0}^{t-1} \beta_1^k \beta_1^{k'} \mathbb{E}\left[\mathbb{E}\left[G_{t-k'-1}^\top G_{t-k-1} \middle| w_{t-k'-1}, \theta_{t-k'-2}, w_{t-k-1}, \theta_{t-k-2}\right] \middle| w_{t-1}\right] \\
&\preceq \frac{\bar{\lambda}B^2}{t\epsilon(1-\beta_1)^2(1-\beta_2)} \left(\sum_{k=0}^{t-1} \beta_1^k\right)^2 I \\
&\preceq \frac{\bar{\lambda}B^2}{t\epsilon(1-\beta_1)^4(1-\beta_2)} I, \qquad (23)
\end{aligned}$$

where the second last inequality is due to Assumption C.5, and the last inequality is by $|\beta_1| < 1$. Let $D' = \frac{\bar{\lambda} B^2}{\epsilon(1-\beta_1)^4(1-\beta_2)}$ and recall that $D = \frac{\lambda(1-\beta_1)r_\delta}{2\sqrt{\epsilon+(1-\beta_2)^{-1}U^2}}$. Together by Eqs. (20) and (23), we can bound Eq. (18) by

$$\mathbb{E}\left[\left(I - \operatorname{diag}(R_{t-1})\sum_{k=0}^{t-1}\beta_1^k G_{t-k-1}\right)^\top \left(I - \operatorname{diag}(R_{t-1})\sum_{k=0}^{t-1}\beta_1^k G_{t-k-1}\right)\middle| w_{t-1}\right] \preceq \left(1 - \frac{2D}{\sqrt{t}} + \frac{D'}{t}\right) I.$$

Since $D, D' > 0$, we have for all $t \geq \frac{D'^2}{D^2}$, $1 - \frac{2D}{\sqrt{t}} + \frac{D'}{t} \leq 1 - \frac{D}{\sqrt{t}} \leq \exp\left(-\frac{D}{\sqrt{t}}\right)$. Therefore, for all $t \geq \frac{D_2^2}{D_1^2}$, we have that

$$\mathbb{E}\left[\left(I - \operatorname{diag}(R_{t-1})\sum_{k=0}^{t-1}\beta_1^k G_{t-k-1}\right)^\top \left(I - \operatorname{diag}(R_{t-1})\sum_{k=0}^{t-1}\beta_1^k G_{t-k-1}\right)\middle| w_{t-1}\right] \preceq \left(\exp\left(-\frac{D}{\sqrt{t}}\right)\right) I.$$

□

Lemma C.7. *Suppose Assumptions C.1 to C.5 hold. We have that as $t \to \infty$,*

$$\mathbb{E}\left[\left\|\operatorname{diag}(R_{t-1})\sum_{k=1}^{t-1}\beta_1^k G_{t-k-1}\left(\sum_{j=1}^{k}\Delta_{t-j}\right)\right\|^2\middle| w_{t-1}\right] = O\left(\frac{1}{t^2}\right).$$

Proof of Lemma C.7. We begin by expanding the norm

$$\begin{aligned}
&\mathbb{E}\left[\left\|\operatorname{diag}(R_{t-1})\sum_{k=1}^{t-1}\beta_1^k G_{t-k-1}\left(\sum_{j=1}^{k}\Delta_{t-j}\right)\right\|^2\middle| w_{t-1}\right] \\
&\leq \mathbb{E}\left[\left(\max_{1\leq i\leq M} R_{t-1,i}\right)^2\left\|\sum_{k=1}^{t-1}\beta_1^k G_{t-k-1}\left(\sum_{j=1}^{k}\Delta_{t-j}\right)\right\|^2\middle| w_{t-1}\right] \\
&\leq \frac{B^2}{t\epsilon(1-\beta_1)^2(1-\beta_2)}\mathbb{E}\left[\sum_{k,k'=1}^{t-1}\beta_1^{k+k'}\left(\sum_{j=1}^{k}\Delta_{t-j}\right)^\top G_{t-k-1}^\top G_{t-k'-1}\left(\sum_{j=1}^{k'}\Delta_{t-j}\right)\middle| w_{t-1}\right] \\
&= \frac{B^2}{t\epsilon(1-\beta_1)^2(1-\beta_2)}\mathbb{E}\left[\sum_{k,k'=1}^{t-1}\beta_1^{k+k'}\left(w_{t-k-1}-w_{t-1}\right)^\top G_{t-k-1}^\top G_{t-k'-1}\left(w_{t-k'-1}-w_{t-1}\right)\middle| w_{t-1}\right], \qquad (24)
\end{aligned}$$

where the second inequality is by Eq. (22), and the last equality follows after writing $w_{t-k-1} - w_{t-1}$ as a telescoping sum. Using Assumption C.5, we can bound the expectation in Eq. (24) as follows:

$$\begin{aligned}
&\mathbb{E}\left[\sum_{k,k'=1}^{t-1}\beta_1^{k+k'}\left(w_{t-k-1}-w_{t-1}\right)^\top G_{t-k-1}^\top G_{t-k'-1}\left(w_{t-k'-1}-w_{t-1}\right)\middle| w_{t-1}\right] \\
&= \mathbb{E}\left[\sum_{k,k'=1}^{t-1}\beta_1^{k+k'}\left(w_{t-k-1}-w_{t-1}\right)^\top \mathbb{E}\left[G_{t-k-1}^\top G_{t-k'-1}\middle| w_{t-k-1}, \theta_{t-k-2}, w_{t-k'-1}, \theta_{t-k'-2}\right]\left(w_{t-k'-1}-w_{t-1}\right)\middle| w_{t-1}\right] \\
&\leq \bar{\lambda}\mathbb{E}\left[\sum_{k,k'=1}^{t-1}\beta_1^{k+k'}\left(w_{t-k-1}-w_{t-1}\right)^\top\left(w_{t-k'-1}-w_{t-1}\right)\middle| w_{t-1}\right] \\
&= \bar{\lambda}\mathbb{E}\left[\sum_{k,k'=1}^{t-1}\beta_1^{k+k'}\left(\sum_{j=1}^{k}\Delta_{t-j}\right)^\top\left(\sum_{j=1}^{k'}\Delta_{t-j}\right)\middle| w_{t-1}\right].
\end{aligned}$$

Therefore,

$$\begin{aligned}
&\mathbb{E}\left[\left\|\operatorname{diag}(R_{t-1})\sum_{k=1}^{t-1}\beta_1^k G_{t-k-1}\left(\sum_{j=1}^{k}\Delta_{t-j}\right)\right\|^2\middle| w_{t-1}\right] \\
&\leq \frac{\bar{\lambda}B^2}{t\epsilon(1-\beta_1)^2(1-\beta_2)}\mathbb{E}\left[\sum_{k,k'=1}^{t-1}\beta_1^{k+k'}\left(\sum_{j=1}^{k}\Delta_{t-j}\right)^\top\left(\sum_{j=1}^{k'}\Delta_{t-j}\right)\middle| w_{t-1}\right] \\
&\leq \frac{\bar{\lambda}B^2}{t\epsilon(1-\beta_1)^2(1-\beta_2)}\mathbb{E}\left[\sum_{k,k'=1}^{t-1}\beta_1^{k+k'}\left\|\sum_{j=1}^{k}\Delta_{t-j}\right\|\left\|\sum_{j=1}^{k'}\Delta_{t-j}\right\|\middle| w_{t-1}\right].
\end{aligned}$$

We now bound $\left\|\sum_{j=1}^{k}\Delta_{t-j}\right\|^2$.

$$\left\|\sum_{j=1}^{k}\Delta_{t-j}\right\|^2 = \sum_{j,j'=1}^{k}\Delta_{t-j}^\top\Delta_{t-j'} \leq \sum_{j,j'=1}^{k}\|\Delta_{t-j}\|\|\Delta_{t-j'}\|.$$

By Eqs. (10) and (7), we can write

$$\begin{aligned}
\|\Delta_{t-j}\|^2 &= \sum_{i=1}^{M} R_{n-j,i}^2 m_{n-j,i}^2 \\
&\leq \frac{B^2}{(t-j)\epsilon(1-\beta_1)^2(1-\beta_2)}\sum_{i=1}^{M} m_{n-j,i}^2 \\
&= \frac{B^2}{(t-j)\epsilon(1-\beta_1)^2(1-\beta_2)}\sum_{i=1}^{M}\left(\sum_{k=0}^{t-j-1}\beta_1^k g_i(w_{t-j-k-1},\theta_{t-j-k-1})\right)^2 \\
&\leq \frac{U^2B^2}{(t-j)\epsilon(1-\beta_1)^2(1-\beta_2)}\sum_{i=1}^{M}\left(\sum_{k=0}^{t-j-1}\beta_1^k\right)^2 \\
&\leq \frac{U^2B^2M}{(t-j)\epsilon(1-\beta_1)^4(1-\beta_2)},
\end{aligned}$$

where the first inequality is by Eq. (21), and the second inequality by Assumption C.3, and the last inequality by $|\beta_1|<1$. Let $D_1 = \frac{\bar{\lambda}B^2}{\epsilon(1-\beta_1)^2(1-\beta_2)}$, $D_2 = \frac{U^2B^2M}{\epsilon(1-\beta_1)^4(1-\beta_2)}$, we have

$$\begin{aligned}
&\mathbb{E}\left[\left\|\operatorname{diag}(R_{t-1})\sum_{k=1}^{t-1}\beta_1^k G_{t-k-1}\left(\sum_{j=1}^{k}\Delta_{t-j}\right)\right\|^2\middle| w_{t-1}\right] \\
&\leq \frac{D_1}{t}\sum_{k,k'=1}^{t-1}\beta_1^{k+k'}\sum_{j,j'=1}^{k}\frac{\sqrt{D_2}}{\sqrt{t-j}}\frac{\sqrt{D_2}}{\sqrt{t-j'}} = \frac{D_1D_2}{t}\left(\sum_{k=1}^{t-1}\beta_1^k\sum_{j=1}^{k}\frac{1}{\sqrt{t-j}}\right)^2.
\end{aligned}$$

If we can show that, as $t\to\infty$, $S(t) \coloneqq \sum_{k=1}^{t-1}\beta_1^k\sum_{j=1}^{k}\frac{1}{\sqrt{t-j}} = O\left(\frac{1}{\sqrt{t}}\right)$, then we have, as $t\to\infty$, $\frac{D_1D_2}{t}\left(\sum_{k=1}^{t-1}\beta_1^k\sum_{j=1}^{k}\frac{1}{\sqrt{t-j}}\right)^2 = O\left(\frac{1}{t^2}\right)$, thus concluding the proof. We now show that $S(t) = O\left(\frac{1}{\sqrt{t}}\right)$ as $t\to\infty$.

$$S(t) = \sum_{k=1}^{t-1}\beta_1^k\sum_{j=1}^{k}\frac{1}{\sqrt{t-j}} = \sum_{j=1}^{t-1}\sum_{k=j}^{t-1}\beta_1^k\frac{1}{\sqrt{t-j}} = \sum_{j=1}^{t-1}\frac{1}{\sqrt{t-j}}\sum_{k=j}^{t-1}\beta_1^k = \sum_{j=1}^{t-1}\frac{1}{\sqrt{t-j}}\frac{\beta_1^j(1-\beta_1^{t-j})}{1-\beta_1}.$$

We decompose the above into two sums to get

$$S(t) = \frac{1}{1-\beta_1}\sum_{j=1}^{t-1}\frac{\beta_1^j}{\sqrt{t-j}} - \frac{1}{1-\beta_1}\sum_{j=1}^{t-1}\frac{\beta_1^j\beta_1^{t-j}}{\sqrt{t-j}} = \frac{1}{1-\beta_1}\sum_{j=1}^{t-1}\frac{\beta_1^j}{\sqrt{t-j}} - \frac{\beta_1^t}{1-\beta_1}\sum_{j=1}^{t-1}\frac{1}{\sqrt{t-j}} \le \frac{1}{1-\beta_1}\sum_{j=1}^{t-1}\frac{\beta_1^j}{\sqrt{t-j}}.$$

Splitting the sum above at $\lfloor t/2 \rfloor$, we get that

$$S(t) = \frac{1}{1-\beta_1}\sum_{j=1}^{\lfloor t/2\rfloor}\frac{\beta_1^j}{\sqrt{t-j}} + \frac{1}{1-\beta_1}\sum_{j=\lfloor t/2\rfloor+1}^{t-1}\frac{\beta_1^j}{\sqrt{t-j}}.$$

In the first sum, since $j \le \lfloor t/2 \rfloor$, we know $t - j \ge t - \frac{t}{2} = \frac{t}{2}$. Then

$$\frac{1}{1-\beta_1}\sum_{j=1}^{\lfloor t/2\rfloor}\frac{\beta_1^j}{\sqrt{t-j}} \le \frac{1}{1-\beta_1}\frac{\sqrt{2}}{\sqrt{t}}\sum_{j=1}^{\lfloor t/2\rfloor}\beta_1^j \le \frac{\beta_1\sqrt{2}}{(1-\beta_1)^2\sqrt{t}}.$$

In the second sum, since $\lfloor t/2 \rfloor + 1 \le j \le t-1$, we know $t - j \ge 1$. Then

$$\frac{1}{1-\beta_1}\sum_{j=\lfloor t/2\rfloor+1}^{t-1}\frac{\beta_1^j}{\sqrt{t-j}} \le \frac{1}{1-\beta_1}\sum_{j=\lfloor t/2\rfloor+1}^{t-1}\beta_1^j \le \frac{1}{1-\beta_1}\sum_{j=\lfloor t/2\rfloor+1}^{\infty}\beta_1^j \le \frac{\beta_1^{\lfloor t/2\rfloor+1}}{(1-\beta_1)^2}.$$

Since $|\beta_1| < 1$, $\beta_1^{\lfloor t/2\rfloor+1}$ decays faster than $\frac{1}{\sqrt{t}}$ as $t \to \infty$. Therefore, we have that, as $t \to \infty$, $S(t) = O\left(\frac{1}{\sqrt{t}}\right)$. □

Lemma C.8. *Suppose Assumptions C.1 to C.5 hold. We have that as $t \to \infty$,*

$$\mathbb{E}\left[\left\|\operatorname{diag}(R_{t-1})\sum_{k=0}^{t-1}\beta_1^k H_{t-k-1}(1-s_{t-k-1})\right\|^2\right] = O\left(\frac{1}{t}\right).$$

Proof of Lemma C.8. We begin by expanding the norm

$$\begin{aligned}
&\mathbb{E}\left[\left\|\operatorname{diag}(R_{t-1})\sum_{k=0}^{t-1}\beta_1^k H_{t-k-1}(1-s_{t-k-1})\right\|^2\right] \\
&= \sum_{k,k'=0}^{t-1}\beta_1^{k+k'}\mathbb{E}\left[\left(H_{t-k-1}(1-s_{t-k-1})\right)^\top \operatorname{diag}(R_{t-1}^2)\left(H_{t-k'-1}(1-s_{t-k'-1})\right)\right] \\
&\le \frac{B^2}{t\epsilon(1-\beta_1)^2(1-\beta_2)}\sum_{k,k'=0}^{t-1}\beta_1^{k+k'}\mathbb{E}\left[(1-s_{t-k-1})^\top H_{t-k-1}^\top H_{t-k'-1}(1-s_{t-k'-1})\right] \\
&= \frac{B^2}{t\epsilon(1-\beta_1)^2(1-\beta_2)}\sum_{k=0}^{t-1}\beta_1^{2k}\mathbb{E}\left[(1-s_{t-k-1})^\top H_{t-k-1}^\top H_{t-k-1}(1-s_{t-k-1})\right] \\
&= \frac{B^2}{t\epsilon(1-\beta_1)^2(1-\beta_2)}\sum_{k=0}^{t-1}\beta_1^{2k}\mathbb{E}\left[\|H_{t-k-1}(1-s_{t-k-1})\|^2\right].
\end{aligned}$$

In the above, the inequality is by Eq. (22); the second last equality is due to unbiased subsampling and that when $k \ne k'$, $s_{t-k-1} \perp\!\!\!\perp s_{t-k'-1}$. If we can show $\forall t$, $\mathbb{E}\left[\|H_t(1-s_t)\|^2\right]$ is uniformly bounded above by some constant C, then we have

$$\begin{aligned}
&\mathbb{E}\left[\left\|\operatorname{diag}(R_{t-1})\sum_{k=0}^{t-1}\beta_1^k H_{t-k-1}(1-s_{t-k-1})\right\|^2\right] \\
&\le \frac{CB^2}{t\epsilon(1-\beta_1)^2(1-\beta_2)}\sum_{k=0}^{t-1}\beta_1^{2k} \\
&\le \frac{CB^2}{t\epsilon(1-\beta_1)^2(1-\beta_2)}\frac{1}{1-\beta_1^2},
\end{aligned}$$

where the last line is by $|\beta_1| < 1$. We can therefore conclude as $t \to \infty$, $\mathbb{E}\left[\left\|\operatorname{diag}(R_{t-1})\sum_{k=0}^{t-1}\beta_1^k H_{t-k-1}(1-s_{t-k-1})\right\|^2\right] = O\left(\frac{1}{t}\right)$. It now remains to show that $\forall t$, $\mathbb{E}\left[\|H_t(1-s_t)\|^2\right]$ is uniformly bounded above by a constant.

By Eq. (9), we have that

$$\begin{aligned}
\mathbb{E}\left[\|g(w_t,\theta_t,\mathcal{S}_t)\|^2\right] &= \mathbb{E}\left[\|G_t(w_t-w^\star)\|^2 + \|H_t(1-s_t)\|^2 + 2(w_t-w^\star)^\top G_t^\top H_t(1-s_t)\right] \\
&= \mathbb{E}\left[\|G_t(w_t-w^\star)\|^2 + \|H_t(1-s_t)\|^2 + 2(w_t-w^\star)^\top G_t^\top H_t \mathbb{E}\left[(1-s_t)|w_t,\theta_t\right]\right] \\
&= \mathbb{E}\left[\|G_t(w_t-w^\star)\|^2 + \|H_t(1-s_t)\|^2\right],
\end{aligned}$$

where the last equality is due to unbiased subsampling. Together with Assumption C.3, we have

$$\mathbb{E}\left[\|H_t(1-s_t)\|^2\right] \le \mathbb{E}\left[\|G_t(w_t-w^\star)\|^2 + \|H_t(1-s_t)\|^2\right] \le \mathbb{E}\left[\|g(w_t,\theta_t,\mathcal{S}_t)\|^2\right] \le MU^2,$$

thus concluding the proof. □