A Appendix

A.1 Derivation of the continuous-time representation of SGD

Consider the update step in mini-batch SGD

$$\theta_{k+1} = \theta_k - \alpha\eta\nabla f_{\mathcal{B}_k}(\theta_k), \tag{16}$$

where $\alpha\eta$ is the step-size, in which η is the maximal allowed step-size, and α is the adjustment factor as was also done in [14], and $\nabla f_{\mathcal{B}_k}(\theta_k) = \frac{1}{m_k}\sum_{i=1}^{m}\nabla f_i(\theta_k)$ is a mini-batch gradient of size m_k, with ∇f_i i.i.d. uniformly sampled from the data points $i \in [1, \dots, n]$. Let the empirical covariance of $\nabla f_i(\theta)$ be denoted as $\Sigma(\theta) = \frac{1}{n}\sum_{i=1}^{n}(\nabla f_i(\theta) - \nabla f(\theta))(\nabla f_i(\theta) - \nabla f(\theta))^T$, then by the assumption above, the covariance of $\nabla f_{\mathcal{B}_k}(\theta)$ is $\text{cov}(\nabla f_{\mathcal{B}_k}(\theta)) = \Sigma(\theta)/m_k$.
The update step in Eq. (16) can now be rewritten in the following way:

$$\theta_{k+1} = \theta_k - \alpha\nabla f(\theta_k)\eta + \alpha\eta(\nabla f(\theta_k) - \nabla f_{\mathcal{B}_k}(\theta_k)) \tag{17}$$

The last term is normally distributed with zero-mean (because $E[\nabla f_{\mathcal{B}_k}(\theta_k)] = \nabla f(\theta_k)$) and $\text{cov}[(\nabla f(\theta_k) - \nabla f_{\mathcal{B}_k}(\theta_k))\alpha\eta] = \text{cov}[\nabla f_{\mathcal{B}_k}(\theta_k)]\alpha^2\eta^2 = \frac{\Sigma(\theta_k)}{m_k}\alpha^2\eta^2$. Introducing the random variable $\Delta B_k \sim \mathcal{N}(0, \eta)$, we can rewrite the update as

$$\theta_{k+1} = \theta_k - \alpha\nabla f(\theta_k)\eta + \alpha\sqrt{\eta\frac{\Sigma(\theta_k)}{m_k}} \cdot \Delta B_k \tag{18}$$

Taking the limit by identifying $\eta \to dt$ and $\Delta B_k \to dB(t)$, From this, an SDE of the following form can be derived:

$$d\theta_t = -\alpha\nabla f(\theta_t)dt + \alpha\sqrt{\frac{\eta\Sigma(\theta_t)}{m_t}}dB_t. \tag{19}$$

The SDE derived in Eq. (19) is a continuous-time representation of the SGD in the sense that the SGD update step in Eq. (18) is the Euler-Maruyama discretization of the SDE in Eq. (19). For a more formal analysis, Li et al. also consider SGD as a discretization of an SDE in Theorem 1 within [14]. The weighting factor appears in Eq. (10) of [14]. Note, that the step-size to batch-size ratio $\frac{\eta}{m_t}$ does not appear there, but can be found in Eq. (5) in [11] (but here the weighting factor α doesn't appear).

A.2 Average decoupled dynamics of SDE

Let the loss function be the multi-dimensional quadratic

$$f(\theta) = \frac{1}{2}\theta^T A\theta, \tag{20}$$

where we assume that A is diagonalizable, i.e. $A = V^T\Lambda V$. Under the assumption that $\Sigma(\theta) = \Sigma$ is constant, the continuous-time model of SGD is

$$d\theta_t = -\alpha A\theta_t dt + \alpha\sqrt{\frac{\eta\Sigma}{m(t)}}dB_t. \tag{21}$$

Now we decouple the dimensions by transforming the SDE using Ito's Lemma with $Y_t = V\theta_t$,

$$dY_t = -\alpha V A\theta_t dt + \alpha V\sqrt{\frac{\eta\Sigma}{m(t)}}dB_t \tag{22}$$

$$= -\alpha VV^T\Lambda V\theta_t dt + \alpha V\sqrt{\frac{\eta\Sigma}{m(t)}}dB_t \tag{23}$$

$$= -\alpha\Lambda Y_t dt + \alpha\sqrt{\frac{\eta}{m(t)}}V\sqrt{\Sigma}dB_t. \tag{24}$$

With v_i being the i-th row of V, we can write down an one-dimensional SDE for each dimension

$$dY_{i,t} = -\alpha\lambda_i Y_{i,t}dt + \alpha\sqrt{\frac{\eta}{m(t)}}v_i^T\sqrt{\Sigma}dB_t. \tag{25}$$

If we denote $\sigma_i := v_i^T\Sigma$, we have

$$dY_{i,t} = -\alpha\lambda_i Y_{i,t}dt + \alpha\sqrt{\frac{\eta\sigma_i}{m(t)}}dB_t \quad \text{for } i = 1,\dots,d. \tag{26}$$

Now we apply Ito's Lemma once again to transform each SDE with $Z_{i,t} = \frac{1}{2}\lambda_i Y_{i,t}^2$:

$$dZ_{i,t} = \left[-\alpha\lambda_i^2 Y_{i,t}^2 + \frac{1}{2}\lambda_i\frac{\alpha^2\eta\sigma_i}{m(t)}\right]dt + \alpha\lambda_i Y_{i,t}\sqrt{\frac{\eta\sigma_i}{m(t)}}v_i^T dB_t \tag{27}$$

$$= \left[-2\alpha\lambda_i Z_{i,t} + \frac{1}{2}\lambda_i\frac{\alpha^2\eta\sigma_i}{m(t)}\right]dt + \alpha\lambda_i Y_{i,t}\sqrt{\frac{\eta\sigma_i}{m(t)}}v_i^T dB_t \tag{28}$$

Taking expectations on both sides and substituting $E[Z_{i,t}] = g_i(t)$ we finally arrive at a system of ODEs

$$E[dZ_{i,t}] = \left(-2\alpha\lambda_i E[Z_{i,t}] + \frac{1}{2}\sigma_i\lambda_i\frac{\alpha^2\eta}{m(t)}\right)dt \tag{29}$$

$$\frac{dg_i(t)}{dt} = -2\alpha\lambda_i g_i(t) + \frac{1}{2}\sigma_i\lambda_i\frac{\alpha^2\eta}{m(t)}, \qquad \text{for } i = 1,\dots,d. \tag{30}$$

A.3 Solving the HJB-equation

Recall the HJB-equation corresponding to the our control problem

$$\partial_t J(\underline{g},t) + \min_{m\in[m_{\min},m_{\max}]}\left\{(1-\gamma)m - 2\alpha\sum_{i=1}^d \lambda_i g_i\partial_{g_i}J(\underline{g},t) + \frac{1}{2}\frac{\alpha^2\eta}{m}\sum_{i=1}^d \sigma_i\lambda_i\partial_{g_i}J(\underline{g},t)\right\} = 0. \tag{31}$$

Since m takes only positive values, the optimal m^* depends on the sign of $\sum_{i=1}^d \sigma_i\lambda_i\partial_{g_i}J(\underline{g},t)$. If it is positive, then the expression in Eq. (31) is convex for positive m and we find m^* where the gradient vanishes, that is $m_t^* = \sqrt{\frac{1}{2}\frac{\alpha^2\eta}{(1-\gamma)}\sum_{i=1}^d \sigma_i\lambda_i\partial_{g_i}J(\underline{g},t)}$ (assuming that it is in $[m_{\min},m_{\max}]$). Otherwise the optimal m^* is just the smallest feasible batch-size $m_{\min}$. Thus, we have

$$m_t^* = \begin{cases}\min\left(m_{\max},\max\left(\sqrt{\frac{1}{2}\frac{\alpha^2\eta}{(1-\gamma)}\sum_{i=1}^d \sigma_i\lambda_i\partial_{g_i}J(\underline{g},t)},m_{\min}\right)\right) & \text{if } \sum_{i=1}^d \sigma_i\lambda_i\partial_{g_i}J(\underline{g},t) > 0\\ m_{\min} & \text{else.}\end{cases} \tag{32}$$

Irrespective what batch-size is chosen, the resulting PDE can be solved via the ansatz

$$J(\underline{g},t) = \sum_{i=1}^d g_i k_i(t) + l(t), \tag{33}$$

so we will have $\partial_{g_i}J(\underline{g},t) = k_i(t)$.

If $m_t^* = \sqrt{\frac{1}{2}\frac{\alpha^2\eta}{(1-\gamma)}\sum_{i=1}^d \sigma_i\lambda_i\partial_{g_i}J(\underline{g},t)}$, then the resulting PDE reads

$$\partial_t J(\underline{g},t) + \sqrt{2(1-\gamma)\alpha^2\eta\sum_{i=1}^d \sigma_i\lambda_i\partial_{g_i}J(\underline{g},t)} - 2\alpha\sum_{i=1}^d \lambda_i g_i\partial_{g_i}J(\underline{g},t) = 0 \tag{34}$$

with the boundary condition $J(\underline{g}, T) = \sum_{i=1}^{d} g_i k_i(T) + l(T) = \gamma \sum_{i=1}^{d} g_i$. This is $k_i(T) = 1$ for all $i = 0, \ldots, d$ and $l(T) = 0$. Plugging in the ansatz we get

$$\sum_{i=1}^{d} g_i k_i'(t) + l'(t) + \sqrt{2(1-\gamma)\alpha^2\eta \cdot \sum_{i=1}^{d} \sigma_i \lambda_i k_i(t)} - 2\alpha \sum_{i=1}^{d} \lambda_i g_i k_i(t) = 0 \tag{35}$$

$$\sum_{i=1}^{d} g_i (k_i'(t) - 2\alpha\lambda_i k_i(t)) + l'(t) + \sqrt{2(1-\gamma)\alpha^2\eta \sum_{i=1}^{d} \sigma_i \lambda_i k_i(t)} = 0. \tag{36}$$

Since the last two terms are independent of g_i, the first term has to vanish, so we must have

$$k_i'(t) - 2\alpha\lambda_i k_i(t) = 0 \quad \text{for } i = 0, \ldots, d. \tag{37}$$

With the boundary condition $k_i(T) = 1$ we get the solution

$$k_i(t) = \gamma \cdot e^{-2\alpha\lambda_i(T-t)}, \quad \text{for } i = 1, \ldots, d, \tag{38}$$

from which we can conclude $\sum_{i=1}^{d} \sigma_i \lambda_i \partial_{g_i} J(\underline{g}, t) = \gamma \sum_{i=1}^{d} \sigma_i \lambda_i \cdot e^{-2\alpha\lambda_i(T-t)}$.
Similarly, if $m_t^* = m = const.$ (that is m_{min} or m_{max}), the resulting PDE reads

$$\partial_t J(\underline{g}, t) + (1-\gamma)m - 2\alpha \sum_{i=1}^{d} \lambda_i g_i \partial_{g_i} J(\underline{g}, t) + \frac{1}{2} \frac{\alpha^2\eta}{m} \sum_{i=1}^{d} \sigma_i \lambda_i \partial_{g_i} J(\underline{g}, t) = 0, \tag{39}$$

with the boundary condition $J(\underline{g}, T) = \gamma \sum_{i=1}^{d} g_i$. Similar to above, with the ansatz $J(\underline{g}, t) = \sum_{i=1}^{d} g_i k_i(t) + l(t)$ the PDE simplifies to

$$\sum_{i=1}^{d} g_i k_i'(t) + l'(t) + (1-\gamma)m - 2\alpha \sum_{i=1}^{d} \lambda_i g_i k_i(t) + \frac{1}{2} \frac{\alpha^2\eta}{m} \sum_{i=1}^{d} \sigma_i \lambda_i \partial_{g_i} J(\underline{g}, t) = 0 \tag{40}$$

$$\sum_{i=1}^{d} g_i (k_i'(t) - 2\alpha\lambda_i k_i(t)) + l'(t) + (1-\gamma)m + \frac{1}{2} \frac{\alpha^2\eta}{m} \sum_{i=1}^{d} \sigma_i \lambda_i \partial_{g_i} J(\underline{g}, t) = 0, \tag{41}$$

from which we can conclude the same solution for $k_i(t)$ as in Eq. (38).

Thus we have the following batch-size schedule: In the convex case,i.e. $\lambda_i > 0, \ \forall i$ (concave case, i.e. $\lambda_i < 0, \ \forall i$) $\gamma \sum_{i=1}^{d} \sigma_i \lambda_i \cdot e^{-2\alpha\lambda_i(T-t)} \underset{(<)}{>} 0$ for all $\underline{g}, t$.

$$m_t^* = \begin{cases} \min\left(m_{\text{max}}, \max\left(m_{\text{min}}, \sqrt{\frac{\alpha^2\eta}{2} \frac{\gamma}{(1-\gamma)} \sum_{i=1}^{d} \sigma_i \lambda_i e^{-2\alpha\lambda_i(T-t)}}\right)\right) & \text{if } f(\theta) \text{ is convex} \\ m_{\text{min}} & \text{if } f(\theta) \text{ is concave.} \end{cases} \tag{42}$$

If the objective $f(\theta)$ is non-convex, we can assume w.l.o.g. that the eigenvalues are ordered, such that $\lambda_1 < \ldots < \lambda_p \leq 0 < \lambda_p + 1 < \ldots < \lambda_d$. From the previous calculations we have that

$$\begin{aligned} \sum_{i=1}^{d} \sigma_i \lambda_i \partial_{g_i} J(\underline{g}, t) &= \sum_{i=1}^{d} \sigma_i \lambda_i \gamma \cdot e^{-2\alpha\lambda_i(T-t)} \\ &= \sum_{i=1}^{p} \sigma_i \lambda_i e^{-2\alpha\lambda_i(T-t)} + \sum_{i=p+1}^{d} \sigma_i \lambda_i e^{-2\alpha\lambda_i(T-t)} \end{aligned} \tag{43}$$

The first sum is strictly monotonic decreasing and the second sum is strictly monotonic increasing. Further we have for the expression above $\overset{t \to -\infty}{\to} -\infty$ and $\overset{t \to +\infty}{\to} +\infty$. This means that there must be

some finite t^*, for which

$$\left|\sum_{i=1}^{p} \sigma_i \lambda_i e^{-2\alpha\lambda_i(T-t^*)}\right| = \left|\sum_{i=p+1}^{d} \sigma_i \lambda_i e^{-2\alpha\lambda_i(T-t^*)}\right| \tag{44}$$

holds.

For $t < t^*$ the negative eigenvalues dominate and for $t > t^*$ the positive eigenvalues dominate. Thus the batch-size schedule reads

$$m_t^* = \begin{cases} m_{\text{min}} & \text{if } t \leq t^* \\ \sqrt{\frac{\alpha^2\eta}{2}\frac{\gamma}{(1-\gamma)}\sum_{i=1}^{d}\sigma_i\lambda_i e^{-2\alpha\lambda_i(T-t)}} & \text{if } t > t^* \end{cases} \tag{45}$$

In the case where $d = 2$ and $\lambda_1 < 0 < \lambda_2$ we can express t^* explicitly as

$$t^* = T - \frac{1}{2\alpha(\lambda_2 - \lambda_1)} \ln\left(\frac{\sigma_2\lambda_2}{-\sigma_1\lambda_1}\right). \tag{46}$$

A.4 2D Saddle point with no running cost

We can also look at the case, in which we do not have a running cost in the cost-functional. Recall that we are looking at a saddle point in two dimensions with the eigenvalues $\lambda_1 < 0 < \lambda_2$ and the dynamics driven by the ODEs

$$\frac{dg_i(t)}{dt} = -2\alpha\lambda_i g_i(t) + \frac{1}{2}\frac{\alpha^2\eta}{m_t}\sigma_i\lambda_i \tag{47}$$

for $i = 1, 2$ with some initial condition $g_i(0) = g_{i,0}$.

In this case the optimal control problem reads

$$\min_{m\in[m_{\text{min}},m_{\text{max}}]} J^m(\underline{g}, t) \tag{48}$$

$$\text{with } J^m(\underline{g}, t) = \sum_{i=1}^{2} \Psi_i^m(t \to T, g_i) \tag{49}$$

$$\text{s.t. } J(\underline{g}, T) = \sum_{i=1}^{2} \Psi_i^m(T \to T, g_i) = \sum_{i=1}^{2} g_i \tag{50}$$

where $0 < m_{\text{min}} < m_{\text{max}} < \infty$ are some given constants, $\underline{g} = (g_1, g_2)$, and $\Psi_i^m(t \to T, g_i)$ refers to the corresponding forward flow map following the respective ODE in Eq. (47) with some batch-size schedule m_t and the initial condition $g_i := g_i(t)$ and ending at $g(T) =: \Psi_i(t \to T, g_i)$.

The corresponding HJB-equation reads

$$0 = \partial_t J(\underline{g}, t) + \min_{m\in[m_{\text{min}},m_{\text{max}}]} \left\{ -2\alpha \sum_{i=1}^{2} \lambda_i g_i \partial_{g_i} J(\underline{g}, t) + \frac{1}{2}\frac{\alpha^2\eta}{m}(\sigma_1\lambda_1\partial_{g_1}J(\underline{g}, t) + \sigma_2\lambda_2\partial_{g_2}J(\underline{g}, t)) \right\} \tag{51}$$

Only the last term depends on m and is of the form $\sim 1/m$. Thus we have a "bang-bang"-type of control with

$$m_t = \begin{cases} m_{\text{min}} & \text{if } \sigma_1\lambda_1\partial_{g_1}J(\underline{g}, t) + \sigma_2\lambda_2\partial_{g_2}J(\underline{g}, t) < 0 \\ m_{\text{max}} & \text{if } \sigma_1\lambda_1\partial_{g_1}J(\underline{g}, t) + \sigma_2\lambda_2\partial_{g_2}J(\underline{g}, t) > 0 \end{cases} \tag{52}$$

Independent of the value of m_t, the above PDE can be solved via the ansatz $J(\underline{g}, t) = g_1 k_1(t) + g_2 k_2(t) + l(t)$. With the boundary condition $J(\underline{g}, T) = g_1 k_1(T) + g_2 k_2(T) + l(T) \overset{!}{=} \sum_{i=1}^{2} g_i$, we eventually find that

$$k_1(t) = e^{-2\alpha\lambda_1(T-t)} \tag{53}$$

$$k_2(t) = e^{-2\alpha\lambda_2(T-t)}. \tag{54}$$

Since $\partial_{g_i} J(\underline{g}, t) = k_i(t)$, we see that

$$\lambda_1 \partial_{g_1} J(\underline{g}, t) + \lambda_2 \partial_{g_2} J(\underline{g}, t) = \lambda_1 e^{-2\alpha\lambda_1(T-t)} + \lambda_2 e^{-2\alpha\lambda_2(T-t)}. \tag{55}$$

Thus, there is a phase from $t = -\infty$ until some time t^*, in which the expression is negative followed by a phase from t^* until $t = \infty$, where the whole expression is positive. This transition time t^* can be found by setting $k_1(t) = k_2(t)$ and solving for t, which leads to

$$t^* = T - \frac{1}{2\alpha(\lambda_2 - \lambda_1)} \ln\left(\frac{\sigma_2\lambda_2}{-\sigma_1\lambda_1}\right). \tag{56}$$

Of course we are only interested in the case, when $0 < t^* < T$ because otherwise either m_{min} or m_{max} will be optimal for the entire optimization. This is only the case if $|\sigma_2\lambda_2| > |\sigma_1\lambda_1|$.
We can interpret t^* as the optimal transition time-point to get the minimal forward flow map $\Psi_1(0 \to T, g_{0,1}) + \Psi_2(0 \to T, g_{0,2})$.

Now let us turn to the ODEs in Eq. (47) to get another perspective on how to arrive at t^*.
We already know from our analysis of the HJB-equation that there are two phases in the optimization, in which two different constant batch-sizes are employed in each phase. Thus we can calculate the forward flow map $\sum_{i=1}^{2} \Psi_i^m(0 \to T, g_{0,i}) := \sum_{i=1}^{2} g_i(T)$ using some constant batch-size $m_t = m_1$ and the forward flow map $\sum_{i=1}^{2} \Psi_i^{m^*}(0 \to T, g_{0,i}) := \sum_{i=1}^{2} g_i^*(T)$ when using some batch-size schedule

$$m_t^* = \begin{cases} m_1 & \text{for } 0 \le t < t^* \\ m_2 & \text{for } t^* \le t < T \end{cases} \tag{57}$$

for another constant $m_2 > m_1$ and some transition time t^*. For any constant batch-size schedule $m_t = m$ and initial condition $g_i(t') = g_i'$, the solution to the ODE in Eq. (47) is

$$g_i(t) = \frac{\alpha\eta\sigma_i}{4m} + e^{-2\alpha\lambda_i(t-t')}\left(g_i' - \frac{\alpha\eta\sigma_i}{4m}\right). \tag{58}$$

Thus, running the ODE starting at $g(0) = g_{0,i}$ with $m_t = m_1$ in the interval $[0, T]$ simply leads to

$$g_i(T) = \frac{\alpha\eta\sigma_i}{4m_1} + e^{-2\alpha\lambda_i T}\left(g_{0,i} - \frac{\alpha\eta\sigma_i}{4m_1}\right). \tag{59}$$

Instead, if we start at $g(0) = g_{0,i}$ with $m_t = m_1$, but change the batch-size to $m_t = m_2$ for $t \in [t^*, T]$, we will get

$$g_i^*(T) = \frac{\alpha\eta\sigma_i}{4m_2}\left(1 - e^{-2\alpha\lambda_i(T-t^*)}\right) + \frac{\alpha\eta\sigma_i}{4m_1} e^{-2\alpha\lambda_i(T-t^*)} + e^{-2\alpha\lambda_i T}\left(g_{0,i} - \frac{\alpha\eta\sigma_i}{4m_1}\right) \tag{60}$$

Now, if we calculate $\sum_{i=1}^{2} g_i(T) - \sum_{i=1}^{2} g_i^*(T)$ we get

$$\sum_{i=1}^{2} g_i(T) - \sum_{i=1}^{2} g_i^*(T) = \frac{\alpha\eta\sigma}{4}\left(\frac{1}{m_1} - \frac{1}{m_2}\right)\left(\sigma_1\left(1 - e^{-2\alpha\lambda_1(T-t^*)}\right) + \sigma_2\left(1 - e^{-2\alpha\lambda_2(T-t^*)}\right)\right). \tag{61}$$

We see that this expression is maximized for any $m_2 > m_1$ with

$$t^* = T - \frac{1}{2\alpha(\lambda_2 - \lambda_1)} \ln\left(\frac{\sigma_2\lambda_2}{-\sigma_1\lambda_1}\right). \tag{62}$$

A.5 Experiment setup

To empirically validate the results, the following experiment was conducted.

A loss function $f(x)$ of the form

$$f(x) = \sum_{i=1}^{n} f_i(x) \tag{63}$$

was chosen, with $f(x) = \frac{1}{2}x^T Ax$, $A = \text{diag}(\lambda_1, \lambda_2)$, and $f_i(x) = \frac{1}{2}(x + \xi_i)^T A(x + \xi_i)$ with i.i.d. $\xi_i \sim \mathcal{N}([0, 0]^T, \Sigma)$. Then we have

$$\mathbb{E}[\nabla f_i(x)] = \frac{1}{n}\sum_{i=1}^{n} A(x + \xi_i) = Ax = \nabla f(x), \tag{64}$$

and

$$\begin{aligned}
\text{cov}(\nabla f_i(x)) &= E[(Ax - A(x + \xi_i))(Ax - A(x + \xi_i))^T] && (65)\\
&= E[A\xi_i\xi_i^T A^T] && (66)\\
&= AE[\xi_i\xi_i^T]A^T && (67)\\
&= A\Sigma A^T. && (68)
\end{aligned}$$

We chose a diagonal covariance matrix for ξ, i.e. $\Sigma = \text{diag}(\sigma_1, \sigma_2)$. Then we have that

$$\text{cov}(\nabla f_i(x)) = \text{diag}(\lambda_1^2\sigma_1, \lambda_2\sigma_2^2). \tag{69}$$

Furthermore, we have

$$\begin{aligned}
\text{cov}\left(\frac{1}{m}\sum_{i=1}^{m} \nabla f_i(x)\right) &= \frac{1}{m^2}\text{cov}\left(\sum_{i=1}^{m} \nabla f_i(x)\right) && (70)\\
&= \frac{1}{m}\text{cov}(\nabla f_i(x)). && (71)
\end{aligned}$$

Thus, in the experiment we are looking at the following SGD update step

$$\begin{aligned}
x_{k+1} &= x_k - \alpha\eta \cdot \nabla f_i(x_k) && (72)\\
&= x_k - \alpha\eta \cdot A(x_k + \xi_k), && (73)
\end{aligned}$$

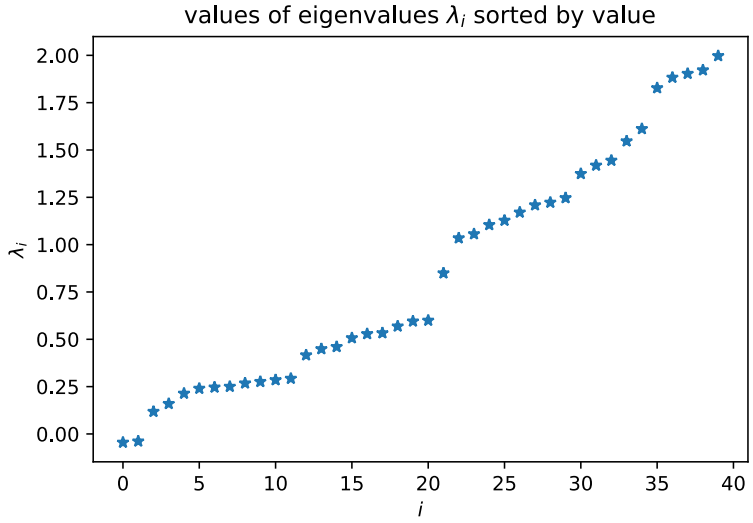

Figure 3: 40 Eigenvalue in the range of $[-0.04, 1.96]$ with two negative eigenvalues and 38 positive eigenvalues.

where ξ_k is due to the stochasticity of the data samples.
For the experiment, 100.000 samples were generated, s.t. $\mathbb{E}[\nabla f_i] = \nabla f$ and $\text{cov}(\nabla f_i) = \text{diag}(100, 1000)$. The parameters were chosen to be

$$\begin{aligned} \eta &= 1, \alpha = 0.3 \\ \gamma &= 0.99999 \\ x_0 &= [0, 5]^T \end{aligned} \tag{74}$$

A.6 More experiments

A.6.1 Optimizing non-convex function in higher dimensions

We also conducted another experiment for a loss function in higher dimensions. Specifically we chose $d = 40$, with $p = 2$ negative eigenvalues and $d - p = 38$ positive eigenvalues. The negative eigenvalues were uniformly sampled from $[-0.0001, -0.05]$ and the positive eigenvalue were uniformly sampled from $[0.1, 2]$. The exact distribution of eigenvalues for the experiment can be found in Fig. 3. The loss function was chosen to be $f(\theta) = \theta^T A \theta$ with $A = \text{diag}(\lambda_1, \ldots, \lambda_d)$. The covariance matrix Σ of ξ was chosen to be a diagonal matrix with the diagonal entries being i.i.d. uniformly distributed from $[100, 1000]$. A number of 100.000 samples were generated, distributed as $\sim \mathcal{N}(\underline{0}, \Sigma)$. The experiment was repeated for 1000 runs of each 500 iterations.
Other parameters, which were chosen are:

$$\begin{aligned} x_0 &\sim \mathcal{U}([-5, 5]) \\ \eta &= 1 \\ \alpha &= 0.1 \\ \gamma &= 0.9999 \end{aligned} \tag{75}$$

t^* was found numerically to be approx. 368.87. The non-constant schedules diverge first, but converge to a loss five orders of magnitudes lower compared to using the maximal batch-size from the beginning. Both non-constant schedules have almost the exact terminal loss, but the adaptive batch-size schedule evaluates only 48.3 % as many samples.

A.6.2 Optimizing convex function in higher dimensions

For the sake of completeness we also looked at applying the batch-size schedule on a convex noisy quadratic model.

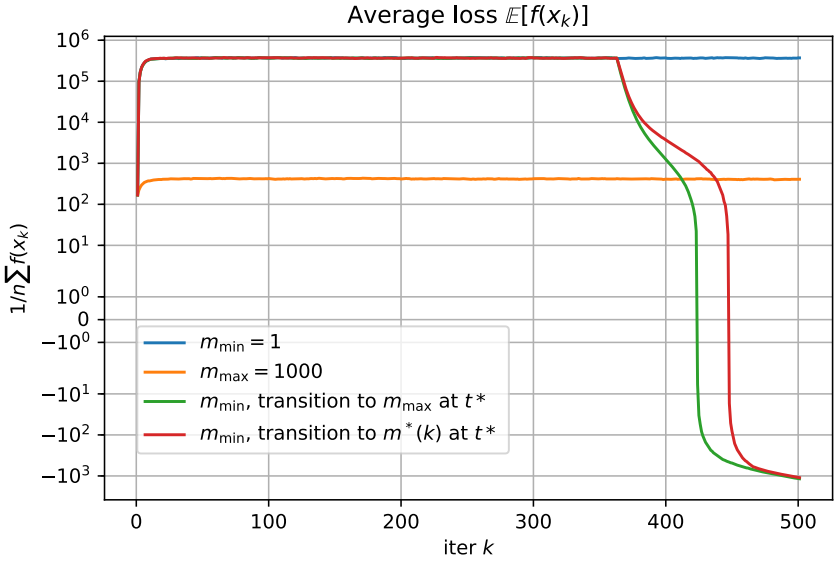

Figure 4: Average loss over 1000 runs, initialized with $f(x_0) = 187.56$. With a constant batch-size (either $m_{\min} = 1$ (blue) or $m_{\max} = 1000$ (red)) SGD is not able to find the descent direction. Whereas with an adaptive batch-size both schedules are able to find the descent direction. Using the maximal batch-size leads to a faster convergence (green), but the terminal value is approximately the same with both schedules.

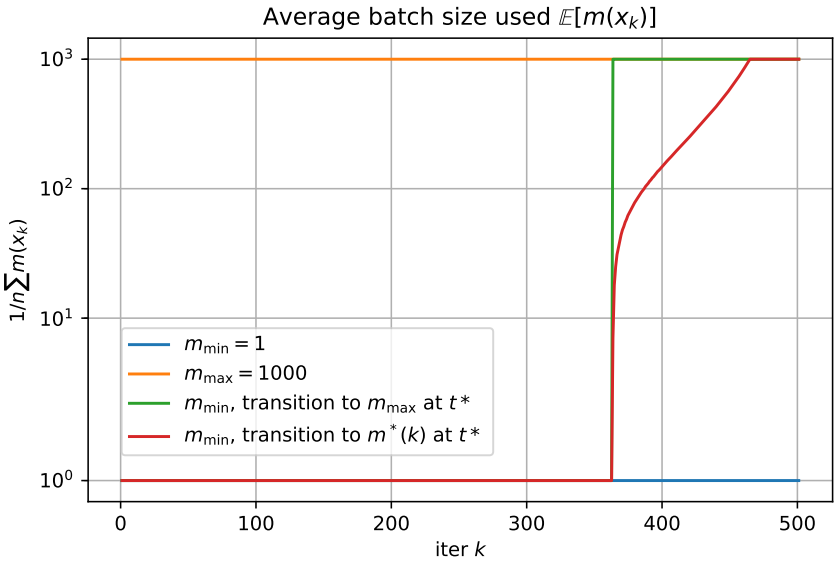

Figure 5: Average batch-size per iteration averaged over 1000 runs

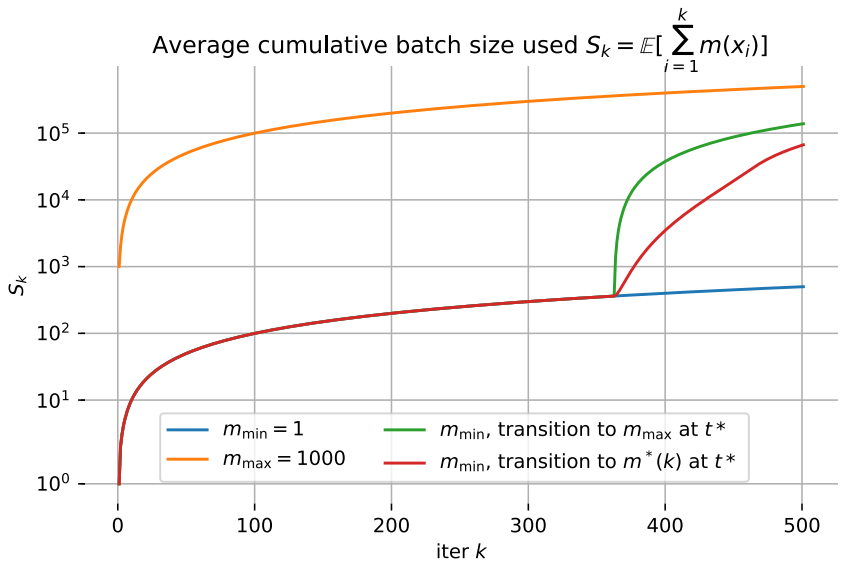

Figure 6: Average cumulative batch-sizes over 1000 runs. Note that while the terminal loss in Fig. 4 is approximately the same, the adaptive schedule evaluates only 48.3 % as many samples compared to the schedule jumping to $m_{\max}$ directly.

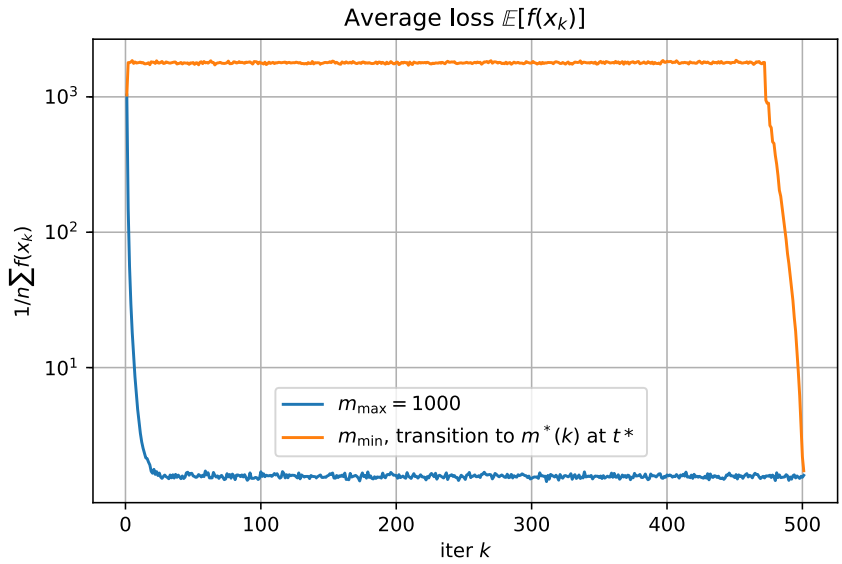

Figure 7: Average loss over 1000 runs. With a constant batch-size (either $m_{\min} = 1$ (blue) or $m_{\max} = 1000$ (red)) SGD is not able to find the descent direction. Whereas with an adaptive batch-size both schedules are able to find the descent direction. Using the maximal batch-size leads to a faster convergence (green), but the terminal value is approximately the same with both schedules.

We chose $d = 40$ and uniformly sampled the eigenvalues λ_i, $i = 1, \dots, d$ in the range of $[1, 10]$. The loss function was chosen to be $f(\theta) = \theta^T A \theta$ with $A = \text{diag}(\lambda_1, \dots, \lambda_d)$.
As above, the covariance matrix Σ of ξ was chosen to be a diagonal matrix with the diagonal entries being i.i.d. uniformly distributed from $[1, 10]$. A number of 100.000 samples were generated, distributed as $\sim \mathcal{N}(0, \Sigma)$. The experiment was repeated for 1000 runs of each 500 iterations. Other parameters were chosen the same as in Eq. (75).

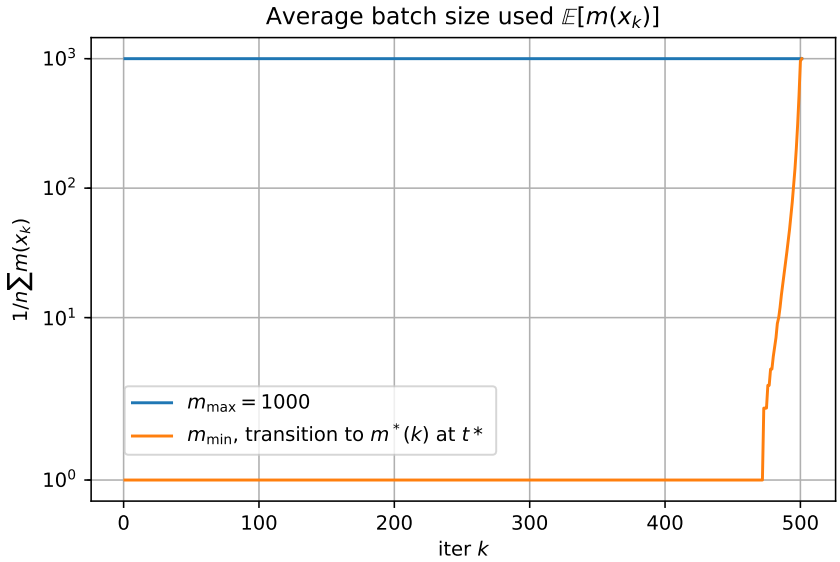

Figure 8: Average batch-size per iteration averaged over 1000 runs. The adaptive batch-size schedule increases exponentially only towards the end, but achieves almost the same terminal loss.

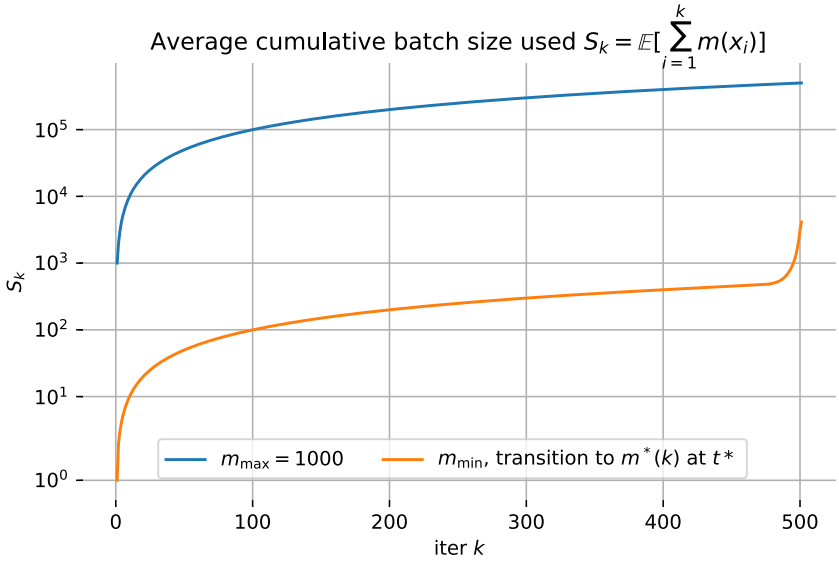

Figure 9: Average cumulative batch-sizes over 1000 runs. Note that while the terminal loss in Fig. 7 is approximately the same, the adaptive schedule evaluates only 0.8 % as many samples compared to using a batch-size of m_{max} from the beginning.