MIME: MIMICKING CENTRALIZED STOCHASTIC ALGORITHMS IN FEDERATED LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated learning (FL) is a challenging setting for optimization due to the heterogeneity of the data across different clients. This heterogeneity has been shown to cause a *client drift*, which can significantly degrade the performance of algorithms designed for the FL setting. In contrast, centralized learning with centrally collected data is not affected by such a drift and has seen great empirical and theoretical progress with innovations such as momentum and adaptivity. In this work, we propose a general algorithmic framework, MIME, which mitigates client drift and adapts arbitrary centralized optimization algorithms such as SGD and Adam to the federated learning setting. MIME uses a combination of *control-variates* and *server-level statistics* (e.g. momentum) at every client-update step to ensure that each local update mimics that of the centralized method run on iid data. Our thorough theoretical and empirical analyses strongly establish MIME's superiority over other baselines.

1 INTRODUCTION

Federated learning has become an important paradigm in large-scale machine learning where the training data remains distributed over a large number of clients, which may be mobile phones or network sensors (Konečný et al., 2016b;a; McMahan et al., 2017; Mohri et al., 2019; Kairouz et al., 2019). A centralized model, here referred to as a server model, is then trained, without ever transmitting client data over the network, thereby providing some basic levels of data privacy and security.

Two important settings are distinguished in Federated learning (Kairouz et al., 2019, Table 1): the *cross-device* and the *cross-silo* settings. The cross-silo setting corresponds to a relatively small number of reliable clients, typically organizations, such as medical or financial institutions. In contrast, in the *cross-device* federated learning setting, the number of clients may be extremely large and include, for example, all 3.5 billion active android phones (Holst, 2019). Thus, in that setting, we may never make even a single pass over the entire clients' data during training. The cross-device setting is further characterized by resource-poor clients communicating over a highly unreliable network. Together, the essential features of this setting give rise to unique challenges not present in the cross-silo setting. Here, we are interested in the cross-device setting, for which we will formalize and study stochastic optimization algorithms.

The de facto standard algorithm for this setting is FEDAVG (McMahan et al., 2017), which performs multiple SGD updates on the available clients, before communicating to the server. While this approach can reduce the total amount of communication required, performing multiple steps on the same client can lead to 'over-fitting' to its atypical local data, a phenomenon known as *client drift* (Karimireddy et al., 2020). Furthermore, algorithmic innovations such as momentum (Sutskever et al., 2013; Cutkosky and Orabona, 2019), adaptivity (Kingma and Ba, 2014; Zaheer et al., 2018; Zhang et al., 2019), and clipping (You et al., 2017; 2019; Zhang et al., 2020) are critical to the success of deep learning applications and need to be incorporated into the client updates, replacing the SGD update of FEDAVG. Perhaps due to such deficiencies, there exists a large gap in performance between the centralized setting, where data is centrally collected on the server, and the federated setting (Zhao et al., 2018; Hsieh et al., 2019; Hsu et al., 2019; Karimireddy et al., 2020).

To overcome such deficiencies, we propose a new framework, MIME, that mitigates client drift and adapts arbitrary centralized optimization algorithms, e.g. SGD with momentum or Adam, to the federated setting. In each local client update, MIME uses global statistics, e.g. momentum, and an

SVRG-style correction to mimic the updates of the centralized algorithm run on i.i.d. data. These global statistics are computed only at the server level and kept fixed throughout the local steps, thereby avoiding a bias due to the atypical local data of any single client.

Contributions. We summarize our main results below.

- We formalize the cross-device federated learning problem, and propose a new framework MIME that can adapt arbitrary centralized algorithms to this setting.
- We prove that incorporating *server momentum* into each *local* client update reduces client drift and leads to optimal statistical rates.
- Further, we quantify the usefulness of performing multiple local updates on a single client by carefully tracking the bias (client-drift) introduced. This is the first analysis showing improved rates by taking additional multiple steps for general smooth functions.
- Finally, we also propose a simpler variant, MIMELITE, with an empirical performance similar to MIME. We report the results of thorough experimental analysis demonstrating that both MIME and MIMELITE are faster than FEDAVG.

Related work. *Analysis of FedAvg:* Much of the recent work in federated learning has focused on analyzing FEDAVG. For identical clients, FEDAVG coincides with parallel SGD, for which Zinkevich et al. (2010) derived an analysis with asymptotic convergence. Sharper and more refined analyses of the same method, sometimes called local SGD, were provided by Stich (2019), and more recently by Stich and Karimireddy (2019), Patel and Dieuleveut (2019), Khaled et al. (2020), and Woodworth et al. (2020b), for identical functions. Their analysis was extended to heterogeneous clients in (Wang et al., 2019; Yu et al., 2019; Karimireddy et al., 2020; Khaled et al., 2020; Koloskova et al., 2020). Charles and Konečnỳ (2020) derived a tight characterization of FedAvg with quadratic functions and demonstrated the sensitivity of the algorithm to both client and server step sizes. Matching upper and lower bounds were recently given by Karimireddy et al. (2020) and Woodworth et al. (2020a) for general functions, proving that FEDAVG can be slower than even SGD for heterogeneous data, due to the *client-drift*. technically, it was a slightly modified FEDAVG version? P: FEDAVG as it is understood encompasses methods with server lr.

Comparison to SCAFFOLD: For the cross-silo setting where the number of clients is relatively slow, Karimireddy et al. (2020) proposed the SCAFFOLD algorithm, which uses control-variates (similar to SVRG) to correct for client drift. However, their algorithm crucially relies on *stateful clients* which repeatedly participate in the training process. In contrast, we focus on the cross-device setting where clients may be visited only once during training and where they are *stateless*. This is akin to the difference between the finite-sum and stochastic settings in traditional centralized optimization.

Improvements to FedAvg: Hsu et al. (2019) and Wang et al. (2020c) observed that using *server momentum* significantly improves over vanilla FEDAVG. This idea was generalized by Reddi et al. (2020), who replaced the server update with an arbitrary optimizer, e.g. Adam. However, these methods only modify the server update while using SGD for the client updates. MIME, on the other hand, ensures that every *local client update* resembles the optimizer e.g. MIME would apply momentum in every client update and not just at the server level. Beyond this, Li et al. (2018) proposed to add a regularizer to ensure client updates remain close. However, its usefulness is unclear (cf. App. B.2, Karimireddy et al., 2020; Wang et al., 2020b). Other orthogonal directions which can be combined with MIME include tackling computation heterogeneity, where some clients perform many more updates than others (Wang et al., 2020b), improving fairness by modifying the objective (Mohri et al., 2019; Li et al., 2019), incorporating differential privacy (Geyer et al., 2017; Agarwal et al., 2018; Thakkar et al., 2020), Byzantine adversaries (Pillutla et al., 2019; Wang et al., 2020a; He et al., 2020a), secure aggregation (Bonawitz et al., 2017; He et al., 2020b), etc. We refer the reader to the extensive survey by Kairouz et al. (2019) for additional discussion.

2 PROBLEM SETUP

This section formalizes the problem of cross-device federated learning. We first examine some key challenges of this setting (cf. Kairouz et al., 2019) to ensure our formalism captures the difficulty:

1. Communication cost between the server and the clients is a major concern and the source of bottleneck in federated learning; thus, a key metric for optimization in this setting is the number of communication rounds.

2. Each client is likely to participate at most once, due to the extremely large number of clients; furthermore, each individual client may have very little data of its own.
3. There may be a wide heterogeneity or *non-i.i.d.-ness* due to the difference of data distributions for the clients.

Thus,vour objective will be to minimize the following quantity within the fewest number of client-server communication rounds:

$$f(\boldsymbol{x}) = \mathbb{E}_{i\sim\mathcal{D}}\Big[f_i(\boldsymbol{x}) := \frac{1}{n_i}\sum_{\nu=1}^{n_i} f_i(\boldsymbol{x};\zeta_{i,\nu})\Big]\,. \tag{1}$$

Here, f_i denotes the loss function of client i and $\{\zeta_{i,1},\ldots,\zeta_{i,n_i}\}$ its local data. Since the number of clients is extremely large, while size of each local data is rather modest, we represent the former as an expectation and the latter as a finite sum. In each round, the algorithm samples a subset of clients (of size S) and performs some updates to the server model. There is some inherent tension between the second and the third challenge outlined above: if there exists a client with arbitrarily different data whom we may never encounter during training, then there is no hope to actually minimize f. Thus for (1) to be tractable, it is **necessary** to assume bounded dissimilarity between different f_i.

(A1) G^2**-BGD** or bounded gradient dissimilarity: there exists $G \geq 0$ such that

$$\mathbb{E}_{i\sim\mathcal{D}}[\|\nabla f_i(\boldsymbol{x}) - \nabla f(\boldsymbol{x})\|^2] \leq G^2\,,\ \forall \boldsymbol{x}\,.$$

Next, we also characterize the variance in the Hessians. Note that if $f_i(\cdot;\zeta)$ is L-smooth, (A2) is always satisfied with $\delta \leq 2L$ and hence is more of a definition rather than an assumption. Note that however, in realistic examples we expect the clients to be similar and hence that $\boldsymbol{\delta} \ll \boldsymbol{L}$.

(A2) δ**-BHD** or bounded Hessian dissimilarity: Almost surely, f is δ-weakly convex i.e. $\nabla^2 f_i(\boldsymbol{x}) \succeq -\delta I$ and the loss function of any client i satisfies

$$\|\nabla^2 f_i(\boldsymbol{x};\zeta) - \nabla^2 f(\boldsymbol{x})\| \leq \delta\,,\ \forall \boldsymbol{x}\,.$$

In addition, we assume that $f(\boldsymbol{x})$ is bounded from below by $f^\star$ and is L-smooth, as is standard.

3 Using momentum to reduce client drift

In this section we examine the tension between reducing communication by running multiple client updates each round, and degradation in performance due to client drift (Karimireddy et al., 2020). To simplify the discussion, we assume a single client is sampled each round and that clients use full-batch gradients.

Server-only approach. A simple way to avoid the issue of client drift is to take no local steps. We sample a client $i \sim \mathcal{D}$ and run SGDm with momentum parameter β and step size η:

$$\begin{aligned}\boldsymbol{x}_t &= \boldsymbol{x}_{t-1} - \eta\left((1-\beta)\nabla f_i(\boldsymbol{x}_{t-1}) + \beta \boldsymbol{m}_{t-1}\right),\\ \boldsymbol{m}_t &= (1-\beta)\nabla f_i(\boldsymbol{x}_{t-1}) + \beta \boldsymbol{m}_{t-1}\,.\end{aligned} \tag{2}$$

Here, the gradient $\nabla f_i(\boldsymbol{x}_t)$ is *unbiased* i.e. $\mathbb{E}[\nabla f_i(\boldsymbol{x}_t)] = \nabla f(\boldsymbol{x}_t)$ and hence we are guaranteed convergence. However, this strategy can be communication-intensive and we are likely to spend all our time waiting for communication with very little time spent on computing the gradients.

FedAvg approach. To reduce the overall communication rounds required, we need to make more progress in each round of communication. Starting from $\boldsymbol{y}_0 = \boldsymbol{x}_{t-1}$, FedAvg (McMahan et al., 2017) runs multiple SGD steps on the sampled client $i \sim \mathcal{D}$

$$\boldsymbol{y}_k = \boldsymbol{y}_{k-1} - \eta\nabla f_i(\boldsymbol{y}_{k-1}) \text{ for } k \in [K]\,, \tag{3}$$

and then a pseudo-gradient $\tilde{\boldsymbol{g}}_t = -(\boldsymbol{y}_K - \boldsymbol{x}_t)$ replaces $\nabla f_i(\boldsymbol{x}_{t-1})$ in the SGDm algorithm (2). This is referred to as server-momentum since it is computed and applied only at the server level (Hsu et al., 2019). However, such updates give rise to *client-drift* resulting in performance worse than the naive server-only strategy (2). This is because by using multiple local updates, (3) starts over-fitting to the local client data, optimizing $f_i(\boldsymbol{x})$ instead of the actual global objective $f(\boldsymbol{x})$. The net

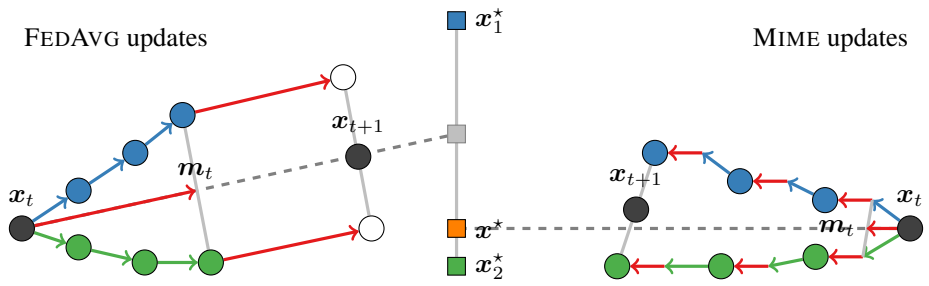

Figure 1: Client-drift in FEDAVG (left) and MIME (right) is illustrated for 2 clients with 3 local steps and momentum parameter $\beta = 0.5$. The local SGD updates of FEDAVG (shown using arrows for client 1 and client2) move towards the average of client optima $\frac{x_1^\star + x_2^\star}{2}$ which can be quite different from the true global optimum $x^\star$. Server momentum only speeds up the convergence to the wrong point in this case. In contrast, MIME uses unbiased momentum and applies it locally at every update. This keeps the updates of MIME closer to the true optimum $x^\star$.

effect is that FEDAVG moves towards an incorrect point (see Fig 1, left). If K is sufficiently large, approximately

$$\begin{aligned} y_K &\rightsquigarrow x_i^\star\,, \quad \text{where } x_i^\star := \arg\min_{x} f_i(x) \\ \Rightarrow \mathbb{E}_{i\sim\mathcal{D}}[\tilde{g}_t] &\rightsquigarrow (x_t - \mathbb{E}_{i\sim\mathcal{D}}[x_i^\star])\,. \end{aligned}$$

Further, the server momentum is based on $\tilde{g}_t$ and hence is also biased. Thus, it cannot correct for the client drift. We next see how a different way of using momentum can mitigate client drift.

Mime approach. FEDAVG experiences client drift because both the momentum and the client updates are biased. To fix the former, we compute momentum using only global statistics as in (2):

$$m_t = (1-\beta)\nabla f_i(x_{t-1}) + \beta m_{t-1}\,. \tag{4}$$

To reduce the bias in the local updates, we will apply this unbiased momentum every step:

$$y_k = y_{k-1} - \eta((1-\beta)\nabla f_i(y_{k-1}) + \beta m_{t-1}) \quad \text{for } k \in [K]\,, \text{ where } . \tag{5}$$

Note that the momentum term is kept fixed during the local updates i.e. there is no local momentum used, only global momentum is applied locally. Since m_{t-1} is a moving average of unbiased gradients computed over multiple clients, it intuitively is a good approximation of the general direction of the updates. By taking a convex combination of the local gradient with m_{t-1}, the update (5) is potentially also less biased. In this way MIME combines the communication benefits of taking multiple local steps and prevents client-drift (see Fig 1, right). Sec. D makes this intuition precise.

4 MIME FRAMEWORK

In this section we describe how to adapt arbitrary centralized algorithms (and not just SGDm) to the federated learning problem (1) while ensuring there is no client-drift. Algorithm 1 describes two variants MIME and MIMELITE, which consists of three components i) a base algorithm we are trying to mimic, ii) how we compute the global statistics, and iii) the local client updates.

Base algorithm. We assume the centralized base algorithm we are imitating can be decomposed into two steps: an *update step* $\mathcal{U}$ which updates the parameters x, and a *statistics step* $\mathcal{V}(\cdot)$ which keeps track of global statistics s. Each step of the base algorithm $\mathcal{B} = (\mathcal{U}, \mathcal{V})$ uses a gradient g to

$$\begin{aligned} x &\leftarrow x - \eta\,\mathcal{U}(g, s)\,, \\ s &\leftarrow \mathcal{V}(g, s)\,. \end{aligned} \tag{BASEALG}$$

$\mathcal{V}$ may track multiple statistics which we represent collectively as s. While SGDm (2) is clearly of this form, Appendix C shows this for other algorithms like Adam, etc.

Algorithm 1 **Mime** and **MimeLite**

input: initial $\boldsymbol{x}$ and $\boldsymbol{s}$, learning rate η and base algorithm $\mathcal{B} = (\mathcal{U}, \mathcal{V})$
for each round $t = 1, \cdots, T$ **do**
  sample subset $\mathcal{S}$ of clients
  communicate $(\boldsymbol{x}, \boldsymbol{s})$ to all clients $i \in \mathcal{S}$
  communicate $\boldsymbol{c} \leftarrow \frac{1}{|\mathcal{S}|}\sum_{j\in\mathcal{S}} \nabla f_j(\boldsymbol{x})$ (only for Mime)
  on client $i \in \mathcal{S}$ **in parallel do**
    initialize local model $\boldsymbol{y}_i \leftarrow \boldsymbol{x}$
    for $k = 1, \cdots, K$ **do**
      sample mini-batch ζ from local data
      $\boldsymbol{y}_i \leftarrow \boldsymbol{y}_i - \eta\mathcal{U}(\nabla f_i(\boldsymbol{y}_i;\zeta) - \nabla f_i(\boldsymbol{x};\zeta) + \boldsymbol{c}, \boldsymbol{s})$ (**Mime**)
      $\boldsymbol{y}_i \leftarrow \boldsymbol{y}_i - \eta\mathcal{U}(\nabla f_i(\boldsymbol{y}_i;\zeta), \boldsymbol{s})$ (**MimeLite**)
    end for
    compute full local-batch gradient $\nabla f_i(\boldsymbol{x})$
    communicate $(\boldsymbol{y}_i, \nabla f_i(\boldsymbol{x}))$
  end on client
  $\boldsymbol{s} \leftarrow \mathcal{V}\Big(\frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}} \nabla f_i(\boldsymbol{x}),\ \boldsymbol{s}\Big)$ (update optimization statistics)
  $\boldsymbol{x} \leftarrow \frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}} \boldsymbol{y}_i$ (update server parameters)
end for

Compute statistics globally, apply locally. When updating the statistics of the base algorithm, we use only the gradient computed at the server parameters. Further, they remain fixed throughout the local updates of the clients. This ensures that these statistics remain unbiased and representative of the global function $f(\cdot)$. At the end of the round, the server performs

$$\boldsymbol{s} \leftarrow \mathcal{V}\Big(\frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}} \nabla f_i(\boldsymbol{x}),\ \boldsymbol{s}\Big), \quad \text{where } \nabla f_i(\boldsymbol{x}) = \frac{1}{n_i}\sum_{\nu=1}^{n_i} \nabla f_i(\boldsymbol{x};\zeta_{i,\nu})\,. \tag{STATS}$$

Note that we use full-batch gradients computed at the server parameters $\boldsymbol{x}$, not client parameters $\boldsymbol{y}_i$.

Local client updates. Each client $i \in \mathcal{S}$ performs K updates using $\mathcal{U}$ of the base algorithm and a minibatch gradient. There are two variants possible corresponding to MIME and MIMELITE differentiated using colored boxes. Starting from $\boldsymbol{y}_i \leftarrow \boldsymbol{x}$, repeat the following K times

$$\begin{aligned} &\boldsymbol{y}_i \leftarrow \boldsymbol{y}_i - \eta\mathcal{U}(\boldsymbol{g}, \boldsymbol{s})\,, \text{ where} \\ &\boldsymbol{g} \leftarrow \nabla f_i(\boldsymbol{y}_i;\zeta) - \nabla f_i(\boldsymbol{x};\zeta) + \frac{1}{|\mathcal{S}|}\sum_{j\in\mathcal{S}} \nabla f_j(\boldsymbol{x}) \text{ or } \nabla f_i(\boldsymbol{y}_i;\zeta)\,. \end{aligned} \tag{CLTSTEP}$$

MIMELITE simply uses the local minibatch gradient whereas MIME uses an SVRG style correction (Johnson and Zhang, 2013). This is done to reduce the noise from sampling a local mini-batch. While this correction yields faster rates in theory (and in practice for convex problems), in deep learning applications we found that MIMELITE closely matches the performance of MIME.

Finally, there are two modifications made in practical FL: we weigh all averages across the clients by the number of datapoints n_i, and we perform K epochs instead of K steps (McMahan et al., 2017). The former modifies the objective (1) with f_i being weighted by n_i, and the latter has been empirically observed to perform better, but lacks strong justification (Wang et al., 2020b).

5 THEORETICAL ANALYSIS OF MIME

Table 1 summarizes the rates of server-only methods, FEDAVG, SCAFFOLD and MIME (new results highlighted in blue). Our theory focuses on MIME with base algorithm of SGD with momentum based variance reduction[1] (MimeMVR) since it obtains optimal rates.

[1]The momentum based variance reduction (MVR), introduced by Cutkosky and Orabona (2019), is a modification of the standard SGDm algorithm to make it amenable to analysis. All our theory uses MVR, while our experiments use SGDm.

Table 1: Number of communication rounds required to reach $\|\nabla f(\boldsymbol{x})\|^2 \leq \epsilon$ for L-smooth functions (log factors are ignored) with S clients sampled each round. G^2 bounds the gradient dissimilarity (A1), and δ bounds the Hessian dissimilarity (A2). FEDAVG is slower than the server-only methods due to additional drift terms. Convergence of SCAFFOLD depends on the total number of clients N which is potentially infinite. MIME matches the optimal statistical rates (first term in the rates) of the server-only methods while improving the optimization (second) term (typically $\delta \ll L$).

Algorithm	Non-convex	μ-Strongly convex
SERVER-ONLY		
SGD (Ghadimi and Lan, 2013)	$\frac{G^2}{S\epsilon^2} + \frac{L}{\epsilon}$	$\frac{G^2}{\mu S\epsilon} + \frac{L}{\mu}$
MVR (Cutkosky and Orabona, 2019)	$\left(\frac{G}{\sqrt{S}\epsilon}\right)^{\frac{3}{2}} + \frac{L}{\epsilon}$	–
FEDAVG [1]		
FedSGD (Karimireddy et al., 2020)	$\frac{G^2}{S\epsilon^2} + \frac{G}{\epsilon^{3/2}} + \frac{L}{\epsilon}$	$\frac{G^2}{\mu S\epsilon} + \frac{G}{\mu\sqrt{\epsilon}} + \frac{L}{\mu}$
SCAFFOLD[2] (Karimireddy et al., 2020)	$\left(\frac{N}{S}\right)^{\frac{2}{3}} \frac{L}{\epsilon}$	$\frac{N}{S} + \frac{L}{\mu}$
MIME [3]		
MimeSGD	$\frac{G^2}{S\epsilon^2} + \frac{\delta}{\epsilon}$	$\frac{G^2}{\mu S\epsilon} + \frac{\delta}{\mu}$
MimeMVR	$\left(\frac{G}{\sqrt{S}\epsilon}\right)^{\frac{3}{2}} + \frac{\delta}{\epsilon}$	–
Lower bound (Arjevani et al., 2019)	$\Omega\left(\frac{G}{\sqrt{S}\epsilon}\right)^{\frac{3}{2}}$	$\Omega\left(\frac{G^2}{S\epsilon}\right)$

[1] Requires $K \geq \frac{\sigma^2}{G^2}$ number of local updates with within-client variance of σ^2.
[2] In cross-device FL, the total number of clients (N) can be of the same order as number of rounds (since we only make few passes over all clients), or even ∞, making the bounds vacuous.
[3] Requires $K \geq L/\delta$ number of local updates.

Theorem I. *For L-smooth f with G^2 gradient dissimilarity (A1), δ Hessian dissimilarity (A2) and $F := (f(\boldsymbol{x}^0) - f^\star)$, the output of MimeMVR after T rounds $\boldsymbol{x}^{out}$ satisfies $\mathbb{E}\|\nabla f(\boldsymbol{x}^{out})\|^2 \leq \epsilon$ for*

- ***PL-Strongly convex without momentum:*** *for $\eta = \tilde{\mathcal{O}}\Big(\min\big(\frac{1}{\delta K + \mu K + L}, \frac{1}{\mu T}\big)\Big)$, $\beta = 0$, and*
$$T = \tilde{\mathcal{O}}\left(\frac{LG^2}{\mu S\epsilon} + \frac{L + \delta K}{\mu K} F \log\left(\frac{1}{\epsilon}\right)\right),$$
- ***Non-convex without momentum:*** *for $\eta = \mathcal{O}\Big(\min\big(\frac{1}{\delta K + L}, (\frac{SF}{G^2 T K^2})^{1/2}\big)\Big)$, $\beta = 0$, and*
$$T = \mathcal{O}\left(\frac{LG^2 F}{S\epsilon^2} + \frac{(L + \delta K)F}{\epsilon K}\right),$$
- ***Non-convex with momentum:*** *for $\eta = \mathcal{O}\Big(\min\big(\frac{1}{\delta K + L}, (\frac{SF}{G^2 T K^3})^{1/3}\big)\Big)$, $\beta = 1 - \mathcal{O}(\eta^2 \delta^2 K^2)$,*
$$T = \mathcal{O}\left(\left(\frac{(1+\delta)G^2 F}{S\epsilon^2}\right)^{3/4} + \frac{(L + \delta K)F}{\epsilon K}\right).$$

Table 1 shows that the rate (ignoring constants) of FEDAVG on non-convex functions is $\left(\frac{G^2}{S\epsilon^2} + \frac{G}{\epsilon^{3/2}} + \frac{L}{\epsilon}\right)$. This is *slower* than simply running SGD which obtains a rate of $\left(\frac{G^2}{S\epsilon^2} + \frac{L}{\epsilon}\right)$. In contrast, MimeSGD obtains a rate of $\left(\frac{G^2}{S\epsilon^2} + \frac{\delta}{\epsilon}\right)$ where $\delta \ll L$ thus improving upon both SGD and FedAvg. While asymptotically these three rates may seem equivalent, in machine learning we care about low accuracy settings where ϵ is not too small (Bottou, 2010) and so the lower order terms matter also matter, as is additionally evidenced by our experiments (Sec. 6). We can also compare with SCAFFOLD (Karimireddy et al., 2020) which obtains a rate of $\left(\frac{N}{S}\right)^{\frac{2}{3}} \frac{L}{\epsilon}$ where N is the total number of clients. While asymptotically this is a faster rate, N in the cross device setting is potentially infinite or at least comparable to the total number of training rounds, making these bounds vacuous. This too is reflected in our experiments (App. B.2).

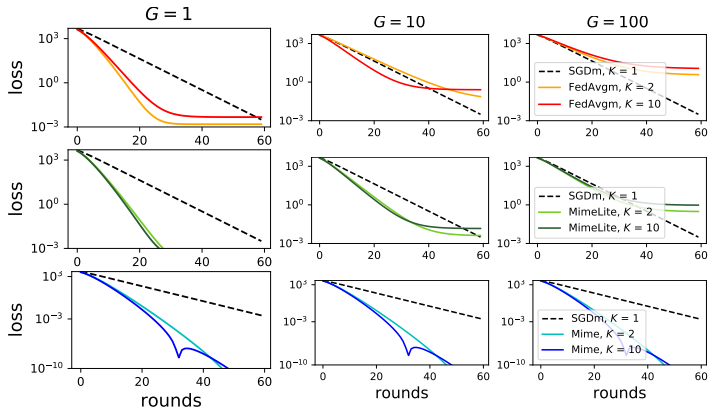

Figure 2: **SGDm** (dashed black), **FedSGDm** (top), **MimeLiteSGDm** (middle), and **MimeSGDm** (bottom) on simulated data, all with momentum ($\beta = 0.5$). FedAvg gets slower as the gradient-dissimilarity (G) increases (to the right). MimeLite shows a similar pattern, but is consistently better than FedAvg. Mime is significantly faster than both and is unaffected by heterogeneity (G).

Finally, by incorporating momentum based variance reduction, MimeMVR obtains a rate of $\big((\frac{G}{\sqrt{S}\epsilon})^{\frac{3}{2}} + \frac{\delta}{\epsilon}\big)$, matching the lower bound of $\Omega\big(\frac{G}{\sqrt{S}\epsilon}\big)^{\frac{3}{2}}$ by Arjevani et al. (2019). This result is is highly non-trivial and involves three crucial ingredients: i) computing the momentum at the server level to ensure that it remains unbiased and then applying it locally during every client update to reduce variance, ii) carefully keeping track of the bias introduced via additional local steps, and iii) an SVRG correction to allow using mini-batches. Our experiments (Sec. 6) verify that the first two theoretical insights are indeed applicable in deep learning settings as well, whereas the latter seems to matter more in convex settings. See App. D where we make this discussion more concrete and Appendices G–H for detailed proofs and theorem statements.

6 Experimental analysis

We run experiments on simulated and real datasets to confirm our theory. Our main findings are i) Mime outperforms FedAvg across all settings, ii) its SVRG correction is useful for convex problems, and iii) momentum significantly improves performance for non-convex problems.

We consider four algorithms: Server-Only, FedAvg, Mime, and MimeLite. Each of these adapt base optimizers SGD, SGDm, and Adam. The Server-Only method computes a full batch gradient on each of the sampled clients and uses their aggregate directly in the base optimizer (akin to (2)). For FedAvg, we follow Reddi et al. (2020) who run multiple epochs of SGD on each client sampled, and then aggregate the net client updates. This aggregated update is used as a pseudo-gradient in the base optimizer (called server optimizer). The learning rate for the server optimizer is fixed to 1 as in (Wang et al., 2020c). This is done to ensure all algorithms have the same number of hyper-parameters. Finally, Mime and MimeLite follow Algorithm 1 and also run a fixed number of epochs on the client. Aggregation is weighted by the number of samples on the clients.

6.1 Simulated convex experiments

Our simulated experiments use two clients each with a simple scalar quadratic loss, as in (Karimireddy et al., 2020). We use full-batch gradients with both clients participating every round. The simulated data has Hessian dissimilarity $\delta = 1$ (A2) and smoothness $L = 2$. We vary the gradient dissimilarity (A1) as $G \in [1, 10, 100]$. All the algorithms use momentum with $\beta = 0.5$ and their learning rates were tuned up to a tolerance of 5E-3 to ensure lowest loss after 60 rounds. The results are collected in Fig. 2. When G is small, we see that FedAvg can outperform the Server-Only (SGDm) baseline, though its loss quickly plateaus. On increasing G, FedAvg becomes even slower. MimeLite differs from FedAvg only in how the momentum is used. In all settings, it slightly outperforms FedAvg though even it sees a substantial slow down as we increase G. This reflects our theory which predicts that for convex cases, momentum does not give significant gains. Mime, on the other hand, is substantially faster than all other methods and is even unaffected by changing G. Thus, in this simple convex setting, the SVRG correction completely eliminates client drift.

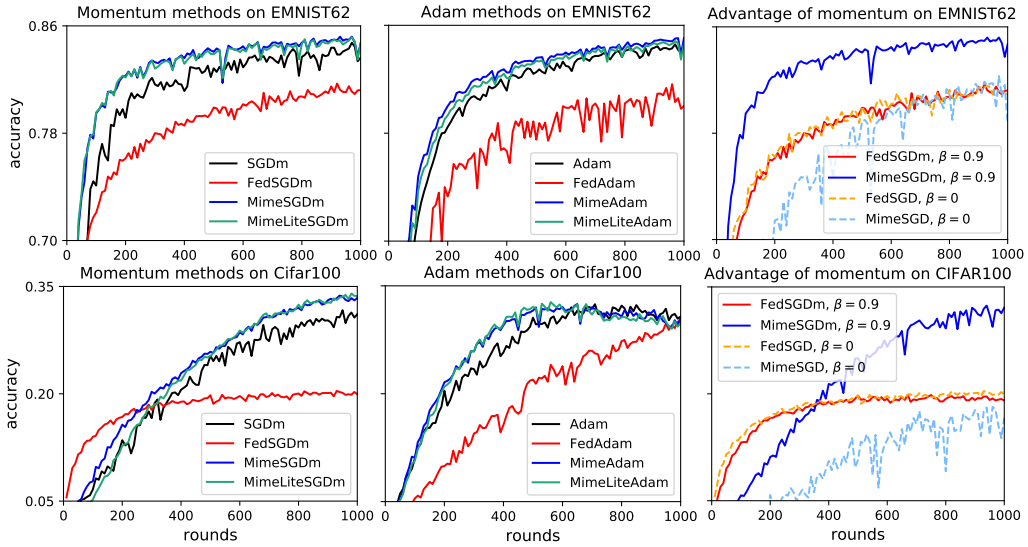

Figure 3: **Server-only**, **FedAvg**, **Mime**, and **MimeLite** with SGDm (left) and Adam (middle) run on (top) EMNIST62 and a 2 hidden layer (300u-100) MLP and (bottom) Resnet20 run on Cifar100. Mime and MimeLite have very similar performance and are consistently the best. FedAvg is even worse than the server-only baselines. Also, Mime makes better use of momentum than FedAvg, with a large increase in performance (right).

6.2 Real world experiments

We run real world deep learning experiments on EMNIST62 with a 2 layer MLP model and on Cifar100 with ResNet20, both accessed through Tensorflow Federated (TFF, 2020a). All methods run 10 local epochs, batch size 20, and the learning rates for all methods were individually tuned. We refer to Appendix A for additional details and results. Fig 3 shows the results.

Mime > MimeLite > Server-only > FedAvg. Mime and MimeLite have the best performance. FedAvg is slower than even the naive server-only methods which make no local updates. This perfectly mirrors our theory that Mime > server-only > FedAvg. The performance of MimeLite is because SVRG correction may not be necessary in deep learning (Defazio and Bottou, 2019).

With momentum > without momentum. Fig. 3 (right) examines the impact of momentum on FedAvg and Mime. Momentum slightly improves the performance of FedAvg, whereas it has a significant impact on the performance of Mime. This is also in line with our theory and confirms that Mime's strategy of applying it locally at every client update makes better use of momentum.

Fixed statistics > updated statistics. Finally, we check how the performance of Mime changes if instead of keeping the momentum fixed throughout a round, we let it change. The momentum is reset at the end of the round ignoring the changes the clients make to it. Appendix B.1 shows that this consistently *worsens* the performance, confirming that it is better to keep the statistics fixed. Together, the above observations validate all aspects of Mime (and MimeLite) design: compute statistics at the server level, and apply them unchanged at every client update.

7 Conclusion

Our work initiated a formal study of the cross-device federated learning problem. We argued that the natural heterogeneity among the clients gives rise to client drift and significantly hampers the performance of approaches such as FedAvg. We then showed how momentum can be an excellent tool to overcome this client drift if used correctly. Based on this observation, we introduced a new framework Mime which not only overcomes client drift, but also adapts arbitrary centralized algorithms such as Adam to the federated setting without any additional hyper-parameters. We demonstrated the superiority of Mime via strong convergence guarantees and empirical evaluations.

REFERENCES

Naman Agarwal, Ananda Theertha Suresh, Felix X. Yu, Sanjiv Kumar, and Brendan McMahan. cpSGD: Communication-efficient and differentially-private distributed SGD. In *Proceedings of NeurIPS*, pages 7575–7586, 2018.

Yossi Arjevani, Yair Carmon, John C Duchi, Dylan J Foster, Nathan Srebro, and Blake Woodworth. Lower bounds for non-convex stochastic optimization. *arXiv preprint arXiv:1912.02365*, 2019.

Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1175–1191. ACM, 2017.

Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečnỳ, Stefano Mazzocchi, H Brendan McMahan, et al. Towards federated learning at scale: System design. *arXiv preprint arXiv:1902.01046*, 2019.

Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

Sebastian Caldas, Peter Wu, Tian Li, Jakub Konečnỳ, H Brendan McMahan, Virginia Smith, and Ameet Talwalkar. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*, 2018.

Zachary Charles and Jakub Konečnỳ. On the outsized importance of learning rates in local update methods. *arXiv preprint arXiv:2007.00878*, 2020.

Ashok Cutkosky and Harsh Mehta. Momentum improves normalized SGD. *arXiv preprint arXiv:2002.03305*, 2020.

Ashok Cutkosky and Francesco Orabona. Momentum-based variance reduction in non-convex SGD. In *Advances in Neural Information Processing Systems*, pages 15210–15219, 2019.

Aaron Defazio and Léon Bottou. On the ineffectiveness of variance reduced optimization for deep learning. In *Advances in Neural Information Processing Systems*, pages 1753–1763, 2019.

Roy Frostig, Matthew James Johnson, and Chris Leary. Compiling machine learning programs via high-level tracing. *Systems for Machine Learning*, 2018.

Robin C Geyer, Tassilo Klein, and Moin Nabi. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*, 2017.

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

Haiku. *JAX Resnet model in Haiku*, 2020. URL https://github.com/deepmind/dm-haiku/blob/master/haiku/_src/nets/resnet.py.

Andrew Hard, Kurt Partridge, Cameron Nguyen, Niranjan Subrahmanya, Aishanee Shah, Pai Zhu, Ignacio Lopez Moreno, and Rajiv Mathews. Training keyword spotting models on non-iid data with federated learning. *arXiv preprint arXiv:2005.10406*, 2020.

Lie He, Sai Praneeth Karimireddy, and Martin Jaggi. Byzantine-robust learning on heterogeneous datasets via resampling. *arXiv preprint arXiv:2006.09365*, 2020a.

Lie He, Sai Praneeth Karimireddy, and Martin Jaggi. Secure byzantine-robust machine learning. *arXiv preprint arXiv:2006.04747*, 2020b.

Arne Holst. Smartphone users worldwide 2016-2021, 2019.

Kevin Hsieh, Amar Phanishayee, Onur Mutlu, and Phillip B Gibbons. The Non-IID data quagmire of decentralized machine learning. *arXiv preprint arXiv:1910.00189*, 2019.

Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in neural information processing systems*, pages 315–323, 2013.

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 795–811. Springer, 2016.

Sai Praneeth Karimireddy, Quentin Rebjock, Sebastian U. Stich, and Martin Jaggi. Error feedback fixes SignSGD and other gradient compression schemes. In *36th International Conference on Machine Learning (ICML)*, 2019.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J Reddi, Sebastian U Stich, and Ananda Theertha Suresh. SCAFFOLD: Stochastic controlled averaging for on-device federated learning. In *37th International Conference on Machine Learning (ICML)*, 2020.

Ahmed Khaled, Konstantin Mishchenko, and Peter Richtárik. Tighter theory for local SGD on indentical and heterogeneous data. In *Proceedings of AISTATS*, 2020.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Anastasia Koloskova, Nicolas Loizou, Sadra Boreiri, Martin Jaggi, and Sebastian U Stich. A unified theory of decentralized SGD with changing topology and local updates. In *37th International Conference on Machine Learning (ICML)*, 2020.

Jakub Konečnỳ, H. Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*, 2016a.

Jakub Konečnỳ, H. Brendan McMahan, Felix X. Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016b.

kuangliu. *Train CIFAR10 with PyTorch*, 2020 (accessed June 4, 2020). URL https://github.com/kuangliu/pytorch-cifar.

Tian Li, Anit Kumar Sahu, Maziar Sanjabi, Manzil Zaheer, Ameet Talwalkar, and Virginia Smith. On the convergence of federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*, 2018.

Tian Li, Maziar Sanjabi, and Virginia Smith. Fair resource allocation in federated learning. *arXiv preprint arXiv:1905.10497*, 2019.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of AISTATS*, pages 1273–1282, 2017.

Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic federated learning. *arXiv preprint arXiv:1902.00146*, 2019.

Yurii Nesterov. *Lectures on convex optimization*, volume 137. Springer, 2018.

Kumar Kshitij Patel and Aymeric Dieuleveut. Communication trade-offs for synchronized distributed SGD with large step size. In *33rd Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

Krishna Pillutla, Sham M Kakade, and Zaid Harchaoui. Robust aggregation for federated learning. *arXiv preprint arXiv:1912.13445*, 2019.

Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020.

Sebastian U. Stich. Local SGD converges fast and communicates little. *International Conference on Learning Representations (ICLR)*, 2019.

Sebastian U. Stich and Sai Praneeth Karimireddy. The error-feedback framework: Better rates for SGD with delayed gradients and compressed communication. *arXiv preprint arXiv:1909.05350*, 2019.

Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147, 2013.

TFF. *Tensorflow Federated Datasets*, 2020a. URL https://www.tensorflow.org/federated/api_docs/python/tff/simulation/datasets.

TFF. *Tensorflow Federated Averaging Implementation*, 2020b. URL https://www.tensorflow.org/federated/api_docs/python/tff/learning/build_federated_averaging_process.

Om Thakkar, Swaroop Ramaswamy, Rajiv Mathews, and Françoise Beaufays. Understanding unintended memorization in federated learning. *arXiv preprint arXiv:2006.07490*, 2020.

Quoc Tran-Dinh, Nhan H. Pham, Dzung T. Phan, and Lam M. Nguyen. Hybrid stochastic gradient descent algorithms for stochastic nonconvex optimization. *arXiv preprint arXiv:1905.05920*, 2019.

Sharan Vaswani, Francis Bach, and Mark Schmidt. Fast and faster convergence of SGD for over-parameterized models and an accelerated perceptron. *arXiv preprint arXiv:1810.07288*, 2018.

Hongyi Wang, Kartik Sreenivasan, Shashank Rajput, Harit Vishwakarma, Saurabh Agarwal, Jy-yong Sohn, Kangwook Lee, and Dimitris Papailiopoulos. Attack of the tails: Yes, you really can backdoor federated learning. *arXiv preprint arXiv:2007.05084*, 2020a.

Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. *arXiv preprint arXiv:2007.07481*, 2020b.

Jianyu Wang, Vinayak Tantia, Nicolas Ballas, and Michael Rabbat. SlowMo: Improving communication-efficient distributed sgd with slow momentum. *International Conference on Learning Representations (ICLR)*, 2020c.

Shiqiang Wang, Tiffany Tuor, Theodoros Salonidis, Kin K. Leung, Christian Makaya, Ting He, and Kevin Chan. Adaptive federated learning in resource constrained edge computing systems. *IEEE Journal on Selected Areas in Communications*, 37(6):1205–1221, 2019.

Blake Woodworth, Kumar Kshitij Patel, and Nathan Srebro. Minibatch vs local SGD for heterogeneous distributed learning. *arXiv preprint arXiv:2006.04735*, 2020a.

Blake Woodworth, Kumar Kshitij Patel, Sebastian U Stich, Zhen Dai, Brian Bullins, H Brendan McMahan, Ohad Shamir, and Nathan Srebro. Is local SGD better than minibatch SGD? In *37th International Conference on Machine Learning (ICML)*, 2020b.

Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*, 2017.

Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. Large batch optimization for deep learning: Training bert in 76 minutes. In *International Conference on Learning Representations*, 2019.

Hao Yu, Sen Yang, and Shenghuo Zhu. Parallel restarted SGD with faster convergence and less communication: Demystifying why model averaging works for deep learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5693–5700, 2019.

Manzil Zaheer, Sashank Reddi, Devendra Sachan, Satyen Kale, and Sanjiv Kumar. Adaptive methods for nonconvex optimization. In *Advances in neural information processing systems*, pages 9793–9803, 2018.

Jingzhao Zhang, Sai Praneeth Karimireddy, Andreas Veit, Seungyeon Kim, Sashank J Reddi, Sanjiv Kumar, and Suvrit Sra. Why ADAM beats SGD for attention models. *arXiv preprint arXiv:1912.03194*, 2019.

Jingzhao Zhang, Tianxing He, Suvrit Sra, and Ali Jadbabaie. Why gradient clipping accelerates training: A theoretical justification for adaptivity. In *International Conference on Learning Representations*, 2020.

Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.

Martin Zinkevich, Markus Weimer, Lihong Li, and Alex J Smola. Parallelized stochastic gradient descent. In *Advances in neural information processing systems*, pages 2595–2603, 2010.

Appendix

A EXPERIMENTAL SETUP

A.1 DESCRIPTION OF DATASETS AND TASKS

We perform 4 tasks over 3 datasets: i) On the EMNIST62 (extended MNIST) dataset (Caldas et al., 2018) we run a convex logistic regression model, a fully connected MLP with 2 hidden layers (300u-100), and convolution model with two CNN layers and two dense layers with dropout. ii) we also run a ResNet20 on CIFAR100.

In all cases we report the top-1 test accuracy in our experiments. EMNIST uses the metadata indicating the original author of the characters to separate them into multiple clients yielding naturally partitioned dataset. Table 2 summarizes the statistics about the different datasets. Note that the average number of rounds a client participates in (computed as sampled clients$\times$number of rounds/number of clients) provides an indication of how much of the training data is seen with SHAKESPEARE being closest to the cross-silo setting and STACKOVERFLOW representing the most cross-device in nature.

Table 2: Details about the datasets used and experiment setting.

	EMNIST62	CIFAR100
Clients	3,400	500
Examples	671,585	50,000
Sampled clients	20	20
Batch size	20	20
Number of epochs	10	10
Avg. rounds each client participates	5.9	40

We use Tensorflow federated datasets (TFF, 2020a) to generate the datasets. Our federated learning simulation code is written in Jax (Frostig et al., 2018). Our Resnet18 model is based off of (Haiku, 2020) (Resnet v2), and following (Hsieh et al., 2019; Reddi et al., 2020) we replace batch norm with group norm with 2 groups. Black and white was reversed in EMNIST62 (i.e. subtracted from 1) to make them similar to MNIST. CIFAR100 used the usual pre-processing (normalization and centering), and data augmentation (random crop and horizontal flipping) following (kuangliu, 2020 (accessed June 4, 2020).

A.2 PRACTICALITY OF EXPERIMENTS

As we already discussed previously, we believe both the datasets and the tasks being studied here are quite realistic in nature. We now discuss our choice of other parameters in the experiment setup (number of training rounds, sampled clients, batch-size, etc.) Each round of federated learning takes 2–3 mins in the real world (Bonawitz et al., 2019) implying that training 1000 rounds takes **1.4–2 days**. This underscores the importance of ensuring that the algorithms for federated learning converge in as few rounds as possible, as well as have very easy to set default hyper-parameters. Thus in our experimental setup we keep all parameters other than the learning rate to their default values. In practice, this learning rate can be set by set using a small dataset on the server (as in (Hard et al., 2020)). The choice of batch size being 10 was made both keeping in mind the limited memory available to each client as well as to match prior work. Finally, while we limit ourselves to sampling 20 workers per round due to computational constraints, in real world FL thousands of devices are often available for training simultaneously each round (Bonawitz et al., 2019). They also note that the probability of each of these devices being available has clear patterns and is far from uniform sampling. Conducting a large scale experimental study which mimics these alternate forms of heterogeneity is an important direction for future work.

A.3 HYPERPARAMETER SEARCH

We use the EMNIST62 with MLP model as a 'test-bed' for exploring different algorithms given it being both a representative task of cross-device FL as well as being computationally efficient. All plots reported are for this setting. A more fine-grained search over hyperparameters to report the final test accuracies is made over the rest of the tasks/datasets.

For all SGDm methods, we pick momentum $\beta = 0.9$. For Adam methods, we fix $\beta_1 = 0.9$, $\beta_2 = 0.99$, and $\varepsilon = 1 \times 10^{-3}$ similar to (Reddi et al., 2020). None of the algorithms use weight decay, clipping etc. The learning rate is then tuned to obtain the best test accuracy.

For all experiments, unless explicitly mentioned otherwise, the learning rate is searched over a grid of

$$\eta \in [1 \times 10^1, 1, 1 \times 10^{-1}, 1 \times 10^{-2}, 1 \times 10^{-3}, 1 \times 10^{-4}, 1 \times 10^{-5}]\,.$$

A.4 COMPARISON WITH PREVIOUS RESULTS

As far as we are aware, (Reddi et al., 2020) is the only prior work which conducts a systematic experimental study of federated learning algorithms over multiple realistic datasets. The algorithms comparable across the two works (e.g. FedSGD, FedSGDm, and FedAdam) have qualitatively similar performance except with one exception: FedAdam consistently underperforms FedSGDm. We believe this difference is because (Reddi et al., 2020) additionally tune the server learning rate and the ϵ parameter for Adam. As we explain in Section A.2, we chose to keep these parameters to some default values to compare methods in the 'low-tuning' setting. We also point that while FedAdam struggles to perform in this setup, MimeAdam and MimeLiteAdam are very stable and even often outperform their SGDm counterparts. This is also the default/recommended behavior in TensorFlow Federated (TFF, 2020b) and (Wang et al., 2020c).

B ADDITIONAL EXPERIMENTS

B.1 EFFECT OF CHANGING STATISTICS

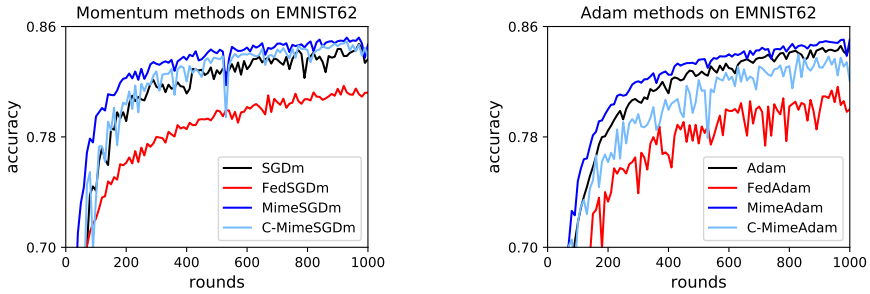

Figure 4: **Server-only**, **FedAvg**, **Mime**, and **C-Mime** with SGDm (left) and Adam (right) run on EMNIST62 with a 2 hidden layer (300u-100) MLP. C-Mime changes the statistics (momentum for SGDm, and first two moments for Adam) using the local client updates. These changes are discarded at the end of the round and the statistics are reset using only the server level gradients as in Mime. Clearly, C-Mime is always worse than Mime. This shows that adapting statistics during local client updates makes them too biased, and it best to keep them fixed during each round like Mime does.

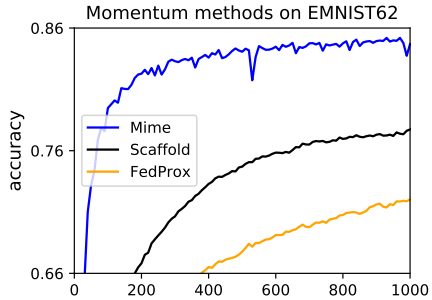

Figure 5: **Mime**, **SCAFFOLD** and **FedProx** with SGDm run on EMNIST62 with a 2 hidden layer (300u-100) MLP. For FedProx and SCAFFOLD, in addition to tuning the learning rate, we search for the best server momentum $\beta \in [0, 0.9, 0.99]$. FedProx uses an additional regularizer μ which we search over $[0.1, 0.5, 1]$ (note that FedProx with $\mu = 0$ is the same as FedAvg). The best test accuracy (which are plotted here) was by $\beta = 0$ for both and $\mu = 0.1$ for FedProx. Note that FedProx is the slowest method here (in fact it is even slower than FedAvg). The additional regularizer does not seem to reduce client drift while still slowing down convergence (Karimireddy et al., 2020; Wang et al., 2020b). SCAFFOLD is also slower than Mime in this setup. This is because SCAFFOLD was designed for the cross-silo setting and not corss-device setting. The large number of clients ($N = 3.4k$) means that each client is on averaged visited less than 6 times during the entire training (20 clients per round for 1k rounds). Hence, the client control variate stored is quite stale (from about 200 rounds ago) which slows down the convergence. This perfectly reflects our theoretical understanding that when the number of clients N is large relative to training rounds (which is true in the cross-device setting) SCAFFOLD is outperformed by MIME.

	FEDAVG			MIME			MIMELITE		
	SGD	SGDm	Adam	SGD	SGDm	Adam	SGD	SGDm	Adam
EMNIST62 logistic	66.0	66.8	67.5	67.2	67.3	**68.0**	66.1	67.3	**68.0**
EMNIST62 CNN	84.9	85.4	85.7	**86.0**	85.6	**86.0**	85.0	85.4	85.7

Table 3: Final test accuracy (larger is better) with fully tuned hyper-parameters. For FEDAVG we searched over both the client and server learning rates, whereas for MIME and MIMELITE, we search only over client (base) learning rate. This search is performed over a grid (0.002, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2, 0.4, 0.8). For momentum, we chose best of (0.9, 0.99) and for Adam, we varied β_1 in (0.9, 0.99), β_2 in (0.99, 0.999) and ϵ in (0.01, 0.001, 0.0001). In this setting as well MIME and MIMELITE outperform FEDAVG but with a smaller margin.

Table 4: Decomposing base algorithms into a parameter update ($\mathcal{U}$) and statistics tracking ($\mathcal{V}$).

Algorithm	Tracked statistics $\boldsymbol{s}$	Update step $\mathcal{U}$	Tracking step $\mathcal{V}$
SGD	–	$\boldsymbol{x} - \eta \boldsymbol{g}$	–
SGDm	$\boldsymbol{m}$	$\boldsymbol{x} - \eta((1-\beta)\boldsymbol{g} + \beta \boldsymbol{m})$	$\boldsymbol{m} = (1-\beta)\boldsymbol{g} + \beta \boldsymbol{m}$
RMSProp	$\boldsymbol{v}$	$\boldsymbol{x} - \frac{\eta}{\epsilon + \sqrt{\boldsymbol{v}}}\boldsymbol{g}$	$\boldsymbol{v} = (1-\beta)\boldsymbol{g}^2 + \beta \boldsymbol{v}$
Adam	$\boldsymbol{m}, \boldsymbol{v}$	$\boldsymbol{x} - \frac{\eta}{\epsilon + \sqrt{\boldsymbol{v}}}((1-\beta_1)\boldsymbol{g} + \beta_1 \boldsymbol{m})$	$\boldsymbol{m} = (1-\beta_1)\boldsymbol{g} + \beta_1 \boldsymbol{m}$ $\boldsymbol{v} = (1-\beta_2)\boldsymbol{g}^2 + \beta_2 \boldsymbol{v}$

B.2 Comparison with Scaffold and FedProx for cross-device FL.

B.3 High-tuning setting

C Additional algorithmic details

D Proof overview

In this section, we give proof sketches of the main components of Theorem I: i) how momentum reduces the effect of client drift, ii) how local steps can take advantage of Hessian similarity, and iii) why the SVRG correction improves constants.

Improving the statistical term via momentum. Note that the statistical (first) term in Theorem I without momentum ($\beta = 0$) for the convex case is $\frac{LG^2}{\mu S \epsilon}$. This is (up to constants) optimal and cannot be improved. For the non-convex case however using $\beta = 0$ gives the usual rate of $\frac{LG^2}{S\epsilon^2}$. However, this can be improved to $\left(\frac{(1+\delta)G^2 F}{S\epsilon^2}\right)^{3/4}$ using momentum. This matches a similar improvement in the centralized setting (Cutkosky and Orabona, 2019; Tran-Dinh et al., 2019) and is in fact optimal (Arjevani et al., 2019). Let us examine why momentum improves the statistical term. Assume that we sample a single client i_t in round t and that we use full-batch gradients. Also let the local client update at step k round t be of the form

$$\boldsymbol{y} \leftarrow \boldsymbol{y} - \eta \boldsymbol{d}_k \,. \tag{6}$$

The ideal choice of update is of course $\boldsymbol{d}_k^\star = \nabla f(\boldsymbol{y})$ but however this is unattainable. Instead, MIME with momentum $\beta = 1 - a$ uses $\boldsymbol{d}_k^{\text{SGDm}} = \tilde{\boldsymbol{m}}_k \leftarrow a \nabla f_i(\boldsymbol{y}) + (1-a)\boldsymbol{m}_{t-1}$ where $\boldsymbol{m}_{t-1}$ is the momentum computed at the server. The variance of this update can then be bounded as

$$\begin{aligned}\mathbb{E}\|\tilde{\boldsymbol{m}}_k - \nabla f(\boldsymbol{y})\|^2 &\lesssim a^2\, \mathbb{E}\|\nabla f_{i_t}(\boldsymbol{y}) - \nabla f(\boldsymbol{y})\|^2 + (1-a)\, \mathbb{E}\|\boldsymbol{m}_{t-1} - \nabla f(\boldsymbol{y})\|^2 \\ &\approx a^2 G^2 + (1-a)\, \mathbb{E}\|\boldsymbol{m}_{t-1} - \nabla f(\boldsymbol{x}_{t-2})\|^2 \approx aG^2 \,.\end{aligned}$$

The last step follows by unrolling the recursion on the variance of $\boldsymbol{m}$. We also assumed that η is small enough that $\boldsymbol{y} \approx \boldsymbol{x}_{t-2}$. This way, momentum can reduce the variance of the update from G^2 to (aG^2) by using past gradients computed on different clients. To formalize the above sketch requires slightly modifying the momentum algorithm similar to (Cutkosky and Orabona, 2019), and is carried out in Appendix H.

Improving the optimization term via local steps. The optimization (second) term in Theorem I for the convex case is $\frac{\delta K + L}{\mu K}$ and for the non-convex case (with or without momentum) is $\frac{\delta K + L}{\epsilon K}$. In contrast, the optimization term of the server-only methods is L/μ and L/ϵ respectively. Since in most cases $\delta \ll L$, the former can be significantly smaller than the latter. This rate also suggests that the best choice of number of local updates is L/δ i.e. we should perform more client updates when they have more similar Hessians. This generalizes results of (Karimireddy et al., 2020) from quadratics to all functions.

This improvement is due to a careful analysis of the *bias* in the gradients computed during the local update steps. Note that for client parameters $\boldsymbol{y}_{k-1}$, the gradient $\mathbb{E}[\nabla f_{i_t}(\boldsymbol{y}_{k-1})] \neq \mathbb{E}[\nabla f(\boldsymbol{y}_{k-1})]$ since $\boldsymbol{y}_{k-1}$ was also computed using the same loss function f_{i_t}. In fact, only the first gradient computed at $\boldsymbol{x}_{t-1}$ is unbiased. Dropping the subscripts k and t, we can bound this bias as:

$$\begin{aligned}\mathbb{E}[\nabla f_i(\boldsymbol{y}) - \nabla f(\boldsymbol{y})] &= \mathbb{E}[\underbrace{\nabla f_i(\boldsymbol{y}) - \nabla f_i(\boldsymbol{x})}_{\approx \nabla^2 f_i(\boldsymbol{x})(\boldsymbol{y}-\boldsymbol{x})} + \underbrace{\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y}_i)}_{\approx \nabla^2 f(\boldsymbol{x})(\boldsymbol{x}-\boldsymbol{y}_i)}] + \underbrace{\mathbb{E}_i[\nabla f_i(\boldsymbol{x})] - \nabla f(\boldsymbol{x})}_{=0 \text{ since unbiased}} \\ &\approx \mathbb{E}[(\nabla^2 f_i(\boldsymbol{x}) - \nabla^2 f(\boldsymbol{x}))(\boldsymbol{y}_i - \boldsymbol{x})] \approx \delta\, \mathbb{E}[(\boldsymbol{y}_i - \boldsymbol{x})] \,.\end{aligned}$$

Thus, the Hessian dissimilarity (A2) control the bias, and hence the usefulness of local updates. This intuition can be made formal using Lemma 3.

Mini-batches via SVRG correction. In our previous discussion about momentum and local steps, we assumed that the clients compute full batch gradients and that only one client is sampled per

round. However, in practice a large number (S) of clients are sampled and further the clients use mini-batch gradients. The SVRG correction reduces this within-client variance since

$$\mathrm{Var}\Big(\nabla f_i(\boldsymbol{y}_i;\zeta) - \nabla f_i(\boldsymbol{x};\zeta) + \tfrac{1}{|\mathcal{S}|}\textstyle\sum_{i\in\mathcal{S}} \nabla f_i(\boldsymbol{x})\Big) \lesssim L^2\|\boldsymbol{y}_i - \boldsymbol{x}\|^2 + \frac{G^2}{S} \approx \frac{G^2}{S}\,.$$

Here, we used the smoothness of $f_i(\cdot;\zeta)$ and assumed that $\boldsymbol{y}_i \approx \boldsymbol{x}$ since we don't move too far within a single round. Thus, the SVRG correction allows us to use minibatch gradients in the local updates while still ensuring that the variance is of the order G^2/S.

E TECHNICALITIES

We examine some additional definitions and introduce some technical lemmas.

E.1 ADDITIONAL DEFINITIONS

We make precise a few definitions and explain some of their implications. Our assumption about the bound on the G gradient dissimilarity can easily be extended to (G, B) gradient dissimilarity used by (Karimireddy et al., 2019):

$$\mathbb{E}_i\|\nabla f_i(\boldsymbol{x})\|^2 \leq G^2 + B^2\|\nabla f(\boldsymbol{x})\|^2\,. \tag{7}$$

All the proofs in the paper extend in a straightforward manner to the above weaker notion. Since this notion does not present any novel technical challenge, we omit it in the rest of the proofs. Note however that the above weaker notion can potentially capture the fact that by increasing the model capacity, we can reduce G. In the extreme case, by taking a sufficiently over-parameterized model, it is possible to make $G = 0$ in certain settings (Vaswani et al., 2018). However, this comes both at a cost of increased resource requirements (i.e. higher memory and compute requirements per step) but can also result in other constants increasing (e.g. B and L). We next define the terms smoothness and strong-convexity which we repeatedly use in the paper.

(A3) f is **L-smooth** and satisfies:

$$\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\| \leq L\|\boldsymbol{x} - \boldsymbol{y}\|\,, \text{ for any } \boldsymbol{x}, \boldsymbol{y}\,. \tag{8}$$

The assumption (A3) also implies the following quadratic upper bound on f

$$f(\boldsymbol{y}) \leq f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x}\rangle + \frac{L}{2}\|\boldsymbol{y} - \boldsymbol{x}\|^2\,. \tag{9}$$

Further, if f is twice-differentiable, (A3) implies that $\|\nabla^2 f(\boldsymbol{x})\| \leq \beta$ for any $\boldsymbol{x}$.

(A4) f is **μ-PL strongly convex** (Karimi et al., 2016) for $\mu > 0$ if it satisfies:

$$\|\nabla f(\boldsymbol{x})\|^2 \geq 2\mu(f(\boldsymbol{x}) - f^\star)\,.$$

Note that PL-strong convexity is much weaker than the standard notion of strong-convexity (Karimi et al., 2016).

If f_i (and not just f) is twice-differentiable and satisfies (A3), then we have $\|\nabla^2 f_i(\boldsymbol{x}) - \nabla^2 f(\boldsymbol{x})\| \leq 2L$ for any $\boldsymbol{x}$.

E.2 SOME TECHNICAL LEMMAS

Now we cover some technical lemmas which are useful for computations later on. First, we state a relaxed triangle inequality true for the squared ℓ_2 norm.

Lemma 1 (relaxed triangle inequality)**.** *Let $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_\tau\}$ be τ vectors in $\mathbb{R}^d$. Then the following are true:*

1. $\|\boldsymbol{v}_i + \boldsymbol{v}_j\|^2 \leq (1 + c)\|\boldsymbol{v}_i\|^2 + (1 + \frac{1}{c})\|\boldsymbol{v}_j\|^2$ *for any $c > 0$, and*
2. $\|\sum_{i=1}^{\tau} \boldsymbol{v}_i\|^2 \leq \tau \sum_{i=1}^{\tau}\|\boldsymbol{v}_i\|^2$.

Proof. The proof of the first statement for any $c > 0$ follows from the identity:

$$\|\boldsymbol{v}_i + \boldsymbol{v}_j\|^2 = (1+c)\|\boldsymbol{v}_i\|^2 + (1+\tfrac{1}{c})\|\boldsymbol{v}_j\|^2 - \|\sqrt{c}\boldsymbol{v}_i + \tfrac{1}{\sqrt{c}}\boldsymbol{v}_j\|^2 .$$

For the second inequality, we use the convexity of $\boldsymbol{x} \to \|\boldsymbol{x}\|^2$ and Jensen's inequality

$$\left\|\frac{1}{\tau}\sum_{i=1}^{\tau}\boldsymbol{v}_i\right\|^2 \leq \frac{1}{\tau}\sum_{i=1}^{\tau}\|\boldsymbol{v}_i\|^2 . \qquad \square$$

Next we state an elementary lemma about expectations of norms of random vectors.

Lemma 2 (separating mean and variance)**.** *Let $\{\Xi_1, \ldots, \Xi_\tau\}$ be τ random variables in $\mathbb{R}^d$ which are not necessarily independent. First suppose that their mean is $\mathbb{E}[\Xi_i] = \xi_i$ and variance is bounded as $\mathbb{E}[\|\Xi_i - \xi_i\|^2] \leq \sigma^2$. Then, the following holds*

$$\mathbb{E}[\|\sum_{i=1}^{\tau}\Xi_i\|^2] \leq \|\sum_{i=1}^{\tau}\xi_i\|^2 + \tau^2\sigma^2 .$$

Now instead suppose that their conditional mean *is $\mathbb{E}[\Xi_i|\Xi_{i-1}, \ldots \Xi_1] = \xi_i$ i.e. the variables $\{\Xi_i - \xi_i\}$ form a martingale difference sequence, and the variance is bounded by $\mathbb{E}[\|\Xi_i - \xi_i\|^2] \leq \sigma^2$ as before. Then we can show the tighter bound*

$$\mathbb{E}[\|\sum_{i=1}^{\tau}\Xi_i\|^2] \leq 2\|\sum_{i=1}^{\tau}\xi_i\|^2 + 2\tau\sigma^2 .$$

Proof. For any random variable X, $\mathbb{E}[X^2] = (\mathbb{E}[X - \mathbb{E}[X]])^2 + (\mathbb{E}[X])^2$ implying

$$\mathbb{E}[\|\sum_{i=1}^{\tau}\Xi_i\|^2] = \|\sum_{i=1}^{\tau}\xi_i\|^2 + \mathbb{E}[\|\sum_{i=1}^{\tau}\Xi_i - \xi_i\|^2] .$$

Expanding the above expression using relaxed triangle inequality (Lemma 1) proves the first claim:

$$\mathbb{E}[\|\sum_{i=1}^{\tau}\Xi_i - \xi_i\|^2] \leq \tau\sum_{i=1}^{\tau}\mathbb{E}[\|\Xi_i - \xi_i\|^2] \leq \tau^2\sigma^2 .$$

For the second statement, ξ_i is not deterministic and depends on $\Xi_{i-1}, \ldots, \Xi_1$. Hence we have to resort to the cruder relaxed triangle inequality to claim

$$\mathbb{E}[\|\sum_{i=1}^{\tau}\Xi_i\|^2] \leq 2\|\sum_{i=1}^{\tau}\xi_i\|^2 + 2\,\mathbb{E}[\|\sum_{i=1}^{\tau}\Xi_i - \xi_i\|^2]$$

and then use the tighter expansion of the second term:

$$\mathbb{E}[\|\sum_{i=1}^{\tau}\Xi_i - \xi_i\|^2] = \sum_{i,j}\mathbb{E}\big[(\Xi_i - \xi_i)^\top(\Xi_j - \xi_j)\big] = \sum_{i}\mathbb{E}\big[\|\Xi_i - \xi_i\|^2\big] \leq \tau\sigma^2 .$$

The cross terms in the above expression have zero mean since $\{\Xi_i - \xi_i\}$ form a martingale difference sequence. $\square$

F Properties of functions with δ bounded Hessian dissimilarity

We now study two lemmas which hold for any functions which satisfy (A2). The first is closely related to the notion of smoothness (A3).

Lemma 3 (similarity)**.** *The following holds for any two functions $f_i(\cdot;\zeta)$ and $f(\cdot)$ satisfying* (A2)*, and any $\boldsymbol{x}, \boldsymbol{y}$:*

$$\|\nabla f_i(\boldsymbol{y};\zeta) - \nabla f_i(\boldsymbol{x};\zeta) + \nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\|^2 \leq \delta^2\|\boldsymbol{y} - \boldsymbol{x}\|^2 .$$

Proof. Consider the function $\Psi(\boldsymbol{z}) := f_i(\boldsymbol{z};\zeta) - f(\boldsymbol{z})$. By the assumption (A2), we know that $\|\nabla^2\Psi(\boldsymbol{z})\| \leq \delta$ for all $\boldsymbol{z}$ i.e. Ψ is δ-smooth. By standard arguments based on taking limits (Nesterov, 2018), this implies that

$$\|\nabla\Psi(\boldsymbol{y}) - \nabla\Psi(\boldsymbol{x})\| \leq \delta\|\boldsymbol{y} - \boldsymbol{x}\| \,.$$

Plugging back the definition of Ψ into the above inequality proves the lemma. □

Next, we see how weakly-convex functions satisfy a weaker notion of "averaging does not hurt".

Lemma 4 (averaging)**.** *Suppose f is δ-weakly convex. Then, for any $\gamma \geq \delta$, and a sequence of parameters $\{\boldsymbol{y}_i\}_{i\in\mathcal{S}}$ and $\boldsymbol{x}$:*

$$\frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}} f(\boldsymbol{y}_i) + \frac{\gamma}{2}\|\boldsymbol{x} - \boldsymbol{y}_i\|^2 \geq f(\bar{\boldsymbol{y}}) + \frac{\gamma}{2}\|\boldsymbol{x} - \bar{\boldsymbol{y}}\|^2 \,, \text{ where } \bar{\boldsymbol{y}} := \frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}} \boldsymbol{y}_i \,.$$

Proof. Since f is δ-weakly convex, $\Phi(\boldsymbol{z}) := f(\boldsymbol{z}) + \frac{\gamma}{2}\|\boldsymbol{z} - \boldsymbol{x}\|^2$ is convex. This proves the claim since $\frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}} \Phi(\boldsymbol{y}_i) \leq \Phi(\bar{\boldsymbol{y}})$. □

G ANALYSIS OF MIMESGD (WITHOUT MOMENTUM)

Let us rewrite the MimeSGD update using notation convenient for analysis. In each round t, we sample clients $\mathcal{S}^t$ such that $|\mathcal{S}^t| = S$. The server communicates the server parameters $\boldsymbol{x}^{t-1}$ as well as the average gradient across the sampled clients $\boldsymbol{c}^{t-1}$ defined as

$$\boldsymbol{c}^{t-1} = \frac{1}{S}\sum_{i\in\mathcal{S}^t} \nabla f_i(\boldsymbol{x}^{t-1})\,. \tag{10}$$

Note that computing $\boldsymbol{c}^{t-1}$ itself requires two rounds of communication. But from a theoretical viewpoint, this only changes the communication rounds required by a constant factor and hence we ignore this issue. Practically, we recommend using MimeLite if this additional rounds of communication are an issue.

Then each client $i \in \mathcal{S}^t$ makes a copy $\boldsymbol{y}^t_{i,0} = \boldsymbol{x}^{t-1}$ and perform K local client updates. In each local client update $k \in [K]$, the client samples a dataset $\zeta^t_{i,k}$ and

$$\boldsymbol{y}^t_{i,k} = \boldsymbol{y}^t_{i,k-1} - \eta(\nabla f_i(\boldsymbol{y}^t_{i,k-1};\zeta^t_{i,k}) - \nabla f_i(\boldsymbol{x}^{t-1};\zeta^t_{i,k}) + \boldsymbol{c}^{t-1})\,. \tag{11}$$

After K such local updates, the server then aggregates the new client parameters as

$$\boldsymbol{x}^t = \frac{1}{S}\sum_{i\in\mathcal{S}^t} \boldsymbol{y}^t_{i,K}\,. \tag{12}$$

Variance of update. Consider the local update at step k on client i, dropping superscript t

$$\boldsymbol{y}_{i,k} = \boldsymbol{y}_{i,k-1} - \eta\boldsymbol{d}_{i,k}, \text{ where } \boldsymbol{d}_{i,k} := \nabla f_i(\boldsymbol{y}_{i,k-1};\zeta_{i,k}) - \nabla f_i(\boldsymbol{x};\zeta_{i,k}) + \boldsymbol{c}\,.$$

Lemma 5. *Given that assumptions (A1) and (A2) are satisfied, each client update satisfies*

$$\mathbb{E}\|\boldsymbol{d}_{i,k}\|^2 \le \frac{3G^2}{S} + 3\delta^2\|\boldsymbol{y}_{i,k-1} - \boldsymbol{x}\|^2 + 3\|\nabla f(\boldsymbol{y}_{i,k-1})\|^2\,.$$

Proof. Starting from the definition of $\boldsymbol{d}_{i,k}$ and the relaxed triangle inequality,

$$\begin{aligned}
\|\boldsymbol{d}_{i,k}\|^2 &= \|\nabla f_i(\boldsymbol{y}_{i,k-1};\zeta_{i,k}) - \nabla f_i(\boldsymbol{x};\zeta_{i,k}) + \boldsymbol{c}\|^2 \\
&= \|\nabla f_i(\boldsymbol{y}_{i,k-1};\zeta_{i,k}) - \nabla f_i(\boldsymbol{x};\zeta_{i,k}) + \nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y}_{i,k-1}) + (\boldsymbol{c} - \nabla f(\boldsymbol{x})) + \nabla f(\boldsymbol{y}_{i,k-1})\|^2 \\
&\le 3\|\nabla f_i(\boldsymbol{y}_{i,k-1};\zeta_{i,k}) - \nabla f_i(\boldsymbol{x};\zeta_{i,k}) + \nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y}_{i,k-1})\|^2 + 3\|\boldsymbol{c} - \nabla f(\boldsymbol{x})\|^2 + 3\|\nabla f(\boldsymbol{y}_{i,k-1})\|^2 \\
&\le 3\delta^2\|\boldsymbol{y}_{i,k-1} - \boldsymbol{x}\|^2 + 3\|\boldsymbol{c} - \nabla f(\boldsymbol{x})\|^2 + 3\|\nabla f(\boldsymbol{y}_{i,k-1})\|^2\,.
\end{aligned}$$

We used Lemma 3 to bound the first term. Taking expectations on both sides to bound the second term via (A1) yields the lemma. □

Distance moved in each round. We show that the distance moved by a client in each round during the K updates can be controlled. To further reduce the burden of notation, we will drop he subscript i, k and refer $\boldsymbol{y}_{i,k-1}$ simply as $\boldsymbol{y}$ and $\boldsymbol{y}_{i,k}$ as $\boldsymbol{y}^+$.

Lemma 6. *For update following (11) for $\eta \le \frac{1}{4K\delta}$ satisfying (A1) and (A2), we have at any step k,*

$$\mathbb{E}\|\boldsymbol{y}^+ - \boldsymbol{x}\|^2 \le \left(1 + \tfrac{2}{K}\right)\|\boldsymbol{y} - \boldsymbol{x}\|^2 + 6K\eta^2\frac{G^2}{S} + 6K\eta^2\|\nabla f(\boldsymbol{y})\|^2\,.$$

Proof. Starting from the update (11) and the relaxed triangle inequality Lemma 1 with $c = K \ge 1$,

$$\begin{aligned}
\mathbb{E}\|\boldsymbol{y}^+ - \boldsymbol{x}\|^2 &= \mathbb{E}\|\boldsymbol{y} - \eta\boldsymbol{d} - \boldsymbol{x}\|^2 \\
&\le (1 + \tfrac{1}{c})\,\mathbb{E}\|\boldsymbol{y} - \boldsymbol{x}\|^2 + (1+c)\eta^2\,\mathbb{E}\|\boldsymbol{d}\|^2 \\
&\le (1 + \tfrac{1}{K})\,\mathbb{E}\|\boldsymbol{y} - \boldsymbol{x}\|^2 + 3(1+K)\eta^2\frac{G^2}{S} + 3(1+K)\eta^2\delta^2\|\boldsymbol{y} - \boldsymbol{x}\|^2 + 3(1+K)\eta^2\|\nabla f(\boldsymbol{y})\|^2 \\
&\le (1 + \tfrac{1}{K} + 6K\eta^2\delta^2)\,\mathbb{E}\|\boldsymbol{y} - \boldsymbol{x}\|^2 + 6K\eta^2\frac{G^2}{S} + 6K\eta^2\|\nabla f(\boldsymbol{y})\|^2\,.
\end{aligned}$$

The second to last step used the variance bound in Lemma 5. The proof now follows from the restriction on step-size since $16K^2\eta^2\delta^2 \le 1$. □

Progress in one client update. We now have the tools required to keep track of the progress made in one round.

Lemma 7. *For any constant $\mu \geq 0$ and each step of MimeSGD with step size $\eta \leq \min\left(\frac{1}{18L}, \frac{1}{756\delta K}, \frac{1}{42\mu K}\right)$, and given that (A1)–(A3) hold, we have*

$$\frac{\eta}{4}\,\mathbb{E}\|\nabla f(\boldsymbol{y}_{i,k-1}^t)\|^2 \leq A_{i,k-1}^t - A_{i,k}^t + \frac{(255KL\eta^2)G^2}{2S}\,,$$

where we define

$$A_{i,k}^t := \mathbb{E}[f(\boldsymbol{y}_{i,k}^t)] + \delta\Big(1+\tfrac{3}{K}\Big)^{K-k}\mathbb{E}\|\boldsymbol{y}_{i,k}^t - \boldsymbol{x}^{t-1}\|^2\,,\ \textit{and}$$

$$A_{i,k-1}^t := \mathbb{E}[f(\boldsymbol{y}_{i,k-1}^t)] + \delta(1-\mu\eta)\Big(1+\tfrac{3}{K}\Big)^{K-k+1}\mathbb{E}\|\boldsymbol{y}_{i,k-1}^t - \boldsymbol{x}^{t-1}\|^2\,.$$

Proof. The assumption that f is L-smooth implies a quadratic upper bound (9). Using this in our case, we have

$$\begin{aligned}\mathbb{E}[f(\boldsymbol{y}^+)] - \mathbb{E}[f(\boldsymbol{y})] &\leq -\eta\,\mathbb{E}[\langle\nabla f(\boldsymbol{y}), \boldsymbol{d}\rangle] + \frac{L\eta^2}{2}\,\mathbb{E}\|\boldsymbol{d}\|^2 \\ &= \underbrace{-\eta\,\mathbb{E}[\langle\nabla f(\boldsymbol{y}), \nabla f_i(\boldsymbol{y};\zeta) - \nabla f_i(\boldsymbol{x};\zeta) + \boldsymbol{c}\rangle]}_{\mathcal{T}_1} + \underbrace{\frac{L\eta^2}{2}\,\mathbb{E}\|\boldsymbol{d}\|^2}_{\mathcal{T}_2}\,.\end{aligned}$$

Let us examine the terms $\mathcal{T}_1$ and $\mathcal{T}_2$ separately. By our variance bound Lemma 5, we have that

$$\mathcal{T}_2 \leq \frac{3L\eta^2G^2}{2S} + \frac{3L\eta^2\delta^2}{2}\|\boldsymbol{y}-\boldsymbol{x}\|^2 + \frac{3L\eta^2}{2}\|\nabla f(\boldsymbol{y})\|^2\,.$$

To simplify $\mathcal{T}_1$, the biggest obstacle is that $\mathbb{E}[\nabla f_i(\boldsymbol{y};\zeta)] \neq \nabla f(\boldsymbol{y})$ since $\boldsymbol{y}$ itself depends on the sampling of the client i. Only the server gradient is unbiased and $\mathbb{E}[\boldsymbol{c}] = \nabla f(\boldsymbol{x})$. Instead we will use the similarity of the functions as in Lemma 3:

$$\begin{aligned}\mathcal{T}_1 &= -\eta\,\mathbb{E}[\langle\nabla f(\boldsymbol{y}), \nabla f_i(\boldsymbol{y};\zeta) - \nabla f_i(\boldsymbol{x};\zeta) + \nabla f(\boldsymbol{x})\rangle] \\ &\leq -\frac{\eta}{2}\,\mathbb{E}\|\nabla f(\boldsymbol{y})\|^2 + \frac{\eta}{2}\|\nabla f_i(\boldsymbol{y};\zeta) - \nabla f_i(\boldsymbol{x};\zeta) + \nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\|^2 \\ &\leq -\frac{\eta}{2}\,\mathbb{E}\|\nabla f(\boldsymbol{y})\|^2 + \frac{\eta\delta^2}{2}\,\mathbb{E}\|\boldsymbol{y}-\boldsymbol{x}\|^2\,.\end{aligned}$$

The first inequality above used that for any a, b, the following holds $-2ab = (a-b)^2 - a^2 - b^2 \leq (a-b)^2 - a^2$. The second used the similarity Lemma 3. Combining the terms $\mathcal{T}_1$ and $\mathcal{T}_2$ together, we have

$$\mathbb{E}[f(\boldsymbol{y}^+)] - \mathbb{E}[f(\boldsymbol{y})] \leq \frac{(3L\eta^2 - \eta)}{2}\,\mathbb{E}\|\nabla f(\boldsymbol{y})\|^2 + \frac{(\eta\delta^2 + 3L\eta^2\delta^2)}{2}\,\mathbb{E}\|\boldsymbol{y}-\boldsymbol{x}\|^2 + \frac{3L\eta^2G^2}{2S}\,.$$

To bound the distance between $\boldsymbol{y}$ and $\boldsymbol{x}$, we use Lemma 6 multiplied on both sides by $\delta\big(1+\frac{3}{K}\big)^{K-k}$. Note that $\delta \leq \delta\big(1+\frac{3}{K}\big)^{K-k} \leq 21\delta$. This gives us for any constant $\mu \geq 0$

$$\begin{aligned}\delta\Big(1+\tfrac{3}{K}\Big)^{K-k}\mathbb{E}\|\boldsymbol{y}^+ - \boldsymbol{x}\|^2 &\leq \delta\Big(1+\tfrac{3}{K}\Big)^{K-k}\Big(1+\tfrac{2}{K}\Big)\|\boldsymbol{y}-\boldsymbol{x}\|^2 + 6K\delta\Big(1+\tfrac{3}{K}\Big)^{K-k}\eta^2\frac{G^2}{S} + \\ &\qquad 6K\delta\Big(1+\tfrac{3}{K}\Big)^{K-k}\eta^2\|\nabla f(\boldsymbol{y})\|^2 \\ &\leq \delta(1-\mu\eta)\Big(1+\tfrac{3}{K}\Big)^{K-(k-1)}\|\boldsymbol{y}-\boldsymbol{x}\|^2 + 6K\delta\Big(1+\tfrac{3}{K}\Big)^{K-k}\eta^2\frac{G^2}{S} + \\ &\qquad 6K\delta\Big(1+\tfrac{3}{K}\Big)^{K-k}\eta^2\|\nabla f(\boldsymbol{y})\|^2 + \Big(1+\tfrac{3}{K}\Big)^{K-k}(\mu\eta\delta - \frac{\delta}{K})\|\boldsymbol{y}-\boldsymbol{x}\|^2 \\ &\leq \delta\big(1+\tfrac{3}{K}\big)^{K-(k-1)}\|\boldsymbol{y}-\boldsymbol{x}\|^2 + 126K\delta\eta^2\frac{G^2}{S} + 126K\delta\eta^2\|\nabla f(\boldsymbol{y})\|^2 \\ &\qquad + (21\mu\eta\delta - \tfrac{\delta}{K})\|\boldsymbol{y}-\boldsymbol{x}\|^2\,.\end{aligned}$$

Adding the two bounds, we get the following recursion

$$\underbrace{\mathbb{E}[f(\boldsymbol{y}^+)] + \delta\Big(1+\tfrac{3}{K}\Big)^{K-k}\mathbb{E}\|\boldsymbol{y}^+ - \boldsymbol{x}\|^2}_{=:A_{i,k}} \le \underbrace{\mathbb{E}[f(\boldsymbol{y})] + \delta\Big(1+\tfrac{3}{K}\Big)^{K-(k-1)}(1-\mu\eta)\|\boldsymbol{y}-\boldsymbol{x}\|^2}_{=:A_{i,k-1}}$$
$$+ \frac{(252K\delta\eta^2 + 3L\eta^2 - \eta)}{2}\mathbb{E}\|\nabla f(\boldsymbol{y})\|^2$$
$$+ \left(\frac{(\eta\delta^2 + 3L\eta^2\delta^2 + 42\mu\eta\delta)}{2} - \frac{\delta}{K}\right)\mathbb{E}\|\boldsymbol{y}-\boldsymbol{x}\|^2$$
$$+ \frac{(3L\eta^2 + 252K\delta\eta^2)G^2}{2S}$$

Now, note that our constraint on the step-size $\eta \le \min(\frac{1}{18L}, \frac{1}{756\delta K})$ implies that $252K\delta\eta^2 + 3L\eta^2 \le \frac{\eta}{2}$ and $K(\eta\delta^2 + 3L\eta^2\delta^2 + 42\mu\eta\delta) \le 2\delta$. Plugging this into the above bound and recalling that $\delta \le L$ finishes the proof. □

Convergence for PL strongly-convex functions. We will unroll the one step progress Lemma 7 to compute a linear rate.

Theorem II. *Suppose that (A1)–(A4) are satisfied for $\mu > 0$. Then the updates of MimeSGD with step-size $\eta = \min(\eta_{\max}, \tilde{\mathcal{O}}\big(\frac{1}{\mu TK}\big))$ for $\eta_{\max} = \min\Big(\frac{1}{18L}, \frac{1}{756\delta K}, \frac{1}{42\mu K}\Big)$ satisfy*

$$\mathbb{E}\|\nabla f(\boldsymbol{x}^{out})\|^2 \le \tilde{\mathcal{O}}\left(\frac{LG^2}{\mu TS} + \frac{F}{\eta_{\max}}\exp\left(-\frac{\mu}{18L + 756\delta K + 42\mu K}TK\right)\right)$$

where we define $F := f(\boldsymbol{x}^0) - f^\star$, $\bar{\boldsymbol{y}}_k^t$ is chosen to be $\boldsymbol{y}_{i,k}^t$ for $i \in \mathcal{S}^t$ uniformly at random, and the output $\boldsymbol{x}^{out}$ to be $\bar{\boldsymbol{y}}_k^t$ with probability proportional to $(1-\frac{\eta\mu}{4})^{KT-kt}$.

Proof. Note that by PL strong convexity (A4), we have

$$\frac{\eta}{4}\|\nabla f(\boldsymbol{y})\|^2 \le \frac{\eta}{8}\|\nabla f(\boldsymbol{y})\|^2 + \frac{\eta\mu}{4}(f(\boldsymbol{y}) - f^\star)\,.$$

Using this, we can tighten the one step progress Lemma 7 as

$$\frac{\eta}{8}\mathbb{E}\|\nabla f(\boldsymbol{y}_{i,k-1}^t)\|^2 \le \underbrace{\Big(1-\frac{\mu\eta}{4}\Big)\mathbb{E}[f(\boldsymbol{y}_{i,k-1}^t) - f^\star] + \delta\Big(1-\frac{\mu\eta}{4}\Big)\Big(1+\tfrac{3}{K}\Big)^{K-k+1}\mathbb{E}\|\boldsymbol{y}_{i,k-1}^t - \boldsymbol{x}^{t-1}\|^2}_{=:\left(1-\frac{\mu\eta}{4}\right)\Phi_{i,k-1}^t}$$
$$- \underbrace{\mathbb{E}[f(\boldsymbol{y}_{i,k}^t) - f^\star] + \delta\Big(1+\tfrac{3}{K}\Big)^{K-k}\mathbb{E}\|\boldsymbol{y}_{i,k}^t - \boldsymbol{x}^{t-1}\|^2}_{=:\Phi_{i,k}^t} + \frac{(255KL\eta^2)G^2}{2S}\,,$$

Now take a weighted sum over the steps k using weights $(1-\frac{\eta\mu}{4})^{K-k}$

$$\frac{\eta}{8}\sum_{k\in[K]}(1-\tfrac{\eta\mu}{4})^{K-k}\mathbb{E}\|\nabla f(\boldsymbol{y}_{i,k-1}^t)\|^2 \le \Phi_{i,0}^t - \big(1-\tfrac{\mu\eta}{4}\big)^K\Phi_{i,K}^t + \sum_{k\in[K]}(1-\tfrac{\eta\mu}{4})^{K-k}\frac{(255KL\eta^2)G^2}{2S}\,.$$

By the initialization $\boldsymbol{y}_{i,0}^t = \boldsymbol{x}^{t-1}$ and hence $\Phi_{i,0}^t = \mathbb{E}\,f(\boldsymbol{x}^{t-1}) - f^\star$ and further by the averaging Lemma 4, we have

$$\frac{1}{S}\sum_{i\in\mathcal{S}}\Phi_{i,K}^t \ge \mathbb{E}\,f(\boldsymbol{x}^t) - f^\star\,.$$

Hence, on averaging over the clients we get the one round progress lemma

$$\frac{\eta}{8S}\sum_{k\in[K]}\sum_{i\in\mathcal{S}^t}(1-\tfrac{\eta\mu}{4})^{K-k}\mathbb{E}\|\nabla f(\boldsymbol{y}_{i,k-1}^t)\|^2 \le \mathbb{E}\,f(\boldsymbol{x}^{t-1}) - f^\star - \big(1-\tfrac{\mu\eta}{4}\big)^K(\mathbb{E}\,f(\boldsymbol{x}^t) - f^\star)$$
$$+ \sum_{k\in[K]}(1-\tfrac{\eta\mu}{4})^{K-k}\frac{(255KL\eta^2)G^2}{2S}\,.$$

Now further taking a weighted average over the rounds $t \in [T]$ with weights proportional to $(1 - \frac{\eta\mu}{4})^{tK}$ gives

$$\frac{\eta}{8S} \sum_{t\in[T]} \sum_{k\in[K]} \sum_{i\in\mathcal{S}^t} (1 - \tfrac{\eta\mu}{4})^{KT-kt}\, \mathbb{E}\|\nabla f(\boldsymbol{y}_{i,k-1}^t)\|^2 \leq \mathbb{E}\, f(\boldsymbol{x}^0) - f^\star + \sum_{t\in[T]} \sum_{k\in[K]} (1 - \tfrac{\eta\mu}{4})^{KT-kt} \frac{(255KL\eta^2)G^2}{2S}\,.$$

Finally, choosing the right step size, similar to Lemma 23 of (Karimireddy et al., 2020) yields the desired rate. □

Convergence for general functions. We will unroll the one step progress Lemma 7 to compute a sublinear rate.

Theorem III. *Suppose that* (A1)–(A3) *are satisfied. Then the updates of MimeSGD with step-size* $\eta = \min\left(\eta_{\max}, \frac{\sqrt{FS}}{\sqrt{255K^2LG^2T}}\right)$ *for* $\eta_{\max} = \min\left(\frac{1}{18L}, \frac{1}{756\delta K}\right)$ *satisfy*

$$\frac{1}{KTS} \sum_{k\in[K]} \sum_{t\in[T]} \sum_{i\in\mathcal{S}^t} \mathbb{E}\|\nabla f(\boldsymbol{y}_{i,k-1}^t)\|^2 \leq \mathcal{O}\left(\frac{G\sqrt{LF}}{\sqrt{TS}} + \frac{(L+\delta K)F}{TK}\right).$$

where we define $F := f(\boldsymbol{x}^0) - f^\star$.

Proof. By summing over the equations from Lemma 7 for all local steps in one round we obtain

$$\frac{\eta}{2} \sum_{k=1}^{K} \mathbb{E}\|\nabla f(\boldsymbol{y}_{i,k-1}^t)\|^2 \leq A_{i,0}^t - A_{i,K}^t + \frac{255K^2L\eta^2G^2}{2S}\,.$$

By the initialization $\boldsymbol{y}_{i,0}^t = \boldsymbol{x}^{t-1}$, hence $A_{i,0}^t = A_{j,0}^t = \mathbb{E}[f(\boldsymbol{x}^{t-1})]$ for all $i, j \in \mathcal{S}^t$. Furthermore, by Lemma 4

$$\frac{1}{|\mathcal{S}^t|} \sum_{i\in\mathcal{S}_t} A_{i,K}^t \geq \mathbb{E}[f(\boldsymbol{x}^t)] + \delta\|\boldsymbol{x}^{t-1} - \boldsymbol{x}^t\|^2 \geq \mathbb{E}[f(\boldsymbol{x}^t)] = A_{i,0}^{t+1}$$

This means that we can keep unrolling over all rounds, obtaining

$$\frac{\eta}{2S} \sum_{t=1}^{T} \sum_{k=1}^{K} \sum_{i\in\mathcal{S}^t} \mathbb{E}\|\nabla f(\boldsymbol{y}_{i,k-1}^t)\|^2 \leq A_{i,0}^1 - A_{i,K}^T + \frac{255TK^2L\eta^2G^2}{2S}\,.$$

By noting $A_{i,0}^1 - A_{i,K}^T = (f(\boldsymbol{x}^0) - f^\star) - (\mathbb{E}[f(\boldsymbol{x}^T) - f^\star) \leq F$ and the choice of the stepsize the theorem follows. □

H ANALYSIS OF MIMEMVR (WITH MOMENTUM BASED VARIANCE REDUCTION)

In this section we see how to use momentum based variance reduction (Cutkosky and Orabona, 2019) to reduce the variance of the updates and improve convergence. It should be noted that MVR does not exactly fit the MIME framework (BASEALG) since it requires computing gradients at two points on the same batch. However, it is straightforward to extend the idea of MIME to MVR as we will now do. We use MVR as a theoretical justification for why the usual momentum works well in practice. An interesting future direction would be to adapt the algorithm and analysis of (Cutkosky and Mehta, 2020), which does fit the framework of MIME.

MimeMVR algorithm. Now, we formally describe the MimeMVR algorithm. In each round t, we sample clients $\mathcal{S}^t$ such that $|\mathcal{S}^t| = S$. The server communicates the server parameters $\boldsymbol{x}^{t-1}$, the momentum $\boldsymbol{m}^{t-1}$ and the average gradient across the sampled clients $\boldsymbol{c}^{t-1}$ defined as

$$\boldsymbol{c}^{t-1} = \frac{1}{S} \sum_{i \in \mathcal{S}^t} \nabla f_i(\boldsymbol{x}^{t-2}) \,. \tag{13}$$

Note that both $\boldsymbol{c}^{t-1}$ and $\boldsymbol{m}^{t-1}$ use gradients and parameters from previous rounds (different from the previous section).

Then each client $i \in \mathcal{S}^t$ makes a copy $\boldsymbol{y}^t_{i,0} = \boldsymbol{x}^{t-1}$ and perform K local client updates. In each local client update $k \in [K]$, the client samples a dataset $\zeta^t_{i,k}$ and

$$\begin{aligned} \boldsymbol{y}^t_{i,k} &= \boldsymbol{y}^t_{i,k-1} - \eta \boldsymbol{d}^t_{i,k} \,, \text{ where} \\ \boldsymbol{d}^t_{i,k} &= a(\nabla f_i(\boldsymbol{y}^t_{i,k-1}; \zeta^t_{i,k}) - \nabla f_i(\boldsymbol{x}^{t-1}; \zeta^t_{i,k}) + \boldsymbol{c}^t - 1) + (1-a)\boldsymbol{m}^{t-1} \\ &\qquad + (1-a)(\nabla f_i(\boldsymbol{y}^t_{i,k-1}; \zeta^t_{i,k}) - \nabla f_i(\boldsymbol{x}^{t-1}; \zeta^t_{i,k})) \,. \end{aligned} \tag{14}$$

After K such local updates, the server then aggregates the new client parameters as

$$\boldsymbol{x}^t = \frac{1}{S} \sum_{j \in \mathcal{S}^t} \boldsymbol{y}^t_{j,K} \,. \tag{15}$$

The momentum term is updated at the end of the round for $a \geq 0$ as

$$\boldsymbol{m}^t = \underbrace{a(\tfrac{1}{S} \textstyle\sum_{j \in \mathcal{S}^t} \nabla f_j(\boldsymbol{x}^{t-1})) + (1-a)\boldsymbol{m}^{t-1}}_{\text{SGDm}} + \underbrace{(1-a)(\tfrac{1}{S} \textstyle\sum_{j \in \mathcal{S}^t} \nabla f_j(\boldsymbol{x}^{t-1}) - \nabla f_j(\boldsymbol{x}^{t-2}))}_{\text{correction}} \,. \tag{16}$$

As we can see, the momentum update of MVR can be broken down into the usual SGDm update, and a correction. Intuitively, this correction term is very small since f_i is smooth and $\boldsymbol{x}^{t-1} \approx \boldsymbol{x}^{t-2}$. Another way of looking at the update (16) is to note that if all functions are identical i.e. $f_j = f_k$ for any j, k, then (16) just becomes the usual gradient descent. Thus MimeMVR tries to maintain an exponential moving average of only the variance terms, reducing its bias. We refer to (Cutkosky and Orabona, 2019) for more detailed explanation of MVR.

Momentum variance bound. We compute the variance of the server momentum $\boldsymbol{m}^{t-1}$. Define the variance term $V^t = \boldsymbol{m}^t - \nabla f(\boldsymbol{x}^{t-1})$. Then its expected norm can be bounded as follows.

Lemma 8. *For the momentum update* (16)*, given* (A1) *and* (A2)*, the following holds for any* $a \in [0,1]$ *and* $V^t := \boldsymbol{m}^t - \nabla f(\boldsymbol{x}^{t-1})$

$$\mathbb{E}\|V^t\|^2 \leq (1-a)\,\mathbb{E}\|V^{t-1}\|^2 + 2\delta^2\,\mathbb{E}\|\boldsymbol{x}^{t-1} - \boldsymbol{x}^{t-2}\|^2 + \frac{2a^2G^2}{S} \,.$$

Proof. Starting from the momentum update (16),

$$
\begin{aligned}
V^t &= (1-a)V^{t-1} \\
&\quad + (1-a)\left(\frac{1}{S}\sum_{j\in\mathcal{S}^t}(\nabla f_j(\boldsymbol{x}^{t-1}) - \nabla f_j(\boldsymbol{x}^{t-2})) - \nabla f(\boldsymbol{x}^{t-1}) + \nabla f(\boldsymbol{x}^{t-2})\right) \\
&\quad + a\left(\frac{1}{S}\sum_{j\in\mathcal{S}^t}(\nabla f_j(\boldsymbol{x}^{t-1}) - \nabla f(\boldsymbol{x}^{t-1})\right).
\end{aligned}
$$

Now, the term V^{t-1} does not have any information from round t and hence is statistically independent of the rest of the terms. Further, the rest of the terms have mean 0. Hence, we can separate out the zero mean noise terms from the V^{t-1} following Lemma 2 and then the relaxed triangle inequality Lemma 1 to claim

$$
\begin{aligned}
\mathbb{E}\|V^t\|^2 &= (1-a)^2\,\mathbb{E}\|V^{t-1}\|^2 \\
&\quad + 2(1-a)^2\left\|\frac{1}{S}\sum_{j\in\mathcal{S}^t}(\nabla f_j(\boldsymbol{x}^{t-1}) - \nabla f_j(\boldsymbol{x}^{t-2})) - \nabla f(\boldsymbol{x}^{t-1}) + \nabla f(\boldsymbol{x}^{t-2})\right\|^2 \\
&\quad + 2a^2\left\|\frac{1}{S}\sum_{j\in\mathcal{S}^t}(\nabla f_j(\boldsymbol{x}^{t-1}) - \nabla f(\boldsymbol{x}^{t-1})\right\|^2 \\
&\le (1-a)^2\,\mathbb{E}\|V^{t-1}\|^2 + 2(1-a)^2\delta^2\|\boldsymbol{x}^{t-1} - \boldsymbol{x}^{t-2}\|^2 + \frac{2a^2G^2}{S}.
\end{aligned}
$$

The inequality used the Hessian similarity Lemma 3 to bound the second term and the heterogeneity bound (A1) to bound the last term. Finally, note that $(1-a)^2 \le (1-a) \le 1$ for $a \in [0,1]$. □

Update variance bound. Now we examine the variance of our update in each local step $\boldsymbol{d}_{i,k}^t$.

Lemma 9. *For the client update (14), given (A1) and (A2), the following holds for any $a \in [0,1]$*

$$
\mathbb{E}\|\boldsymbol{d}_{i,k}^t - \nabla f(\boldsymbol{y}_{i,k-1}^t)\|^2 \le 3\,\mathbb{E}\|V^{t-1}\|^2 + 3\delta^2\,\mathbb{E}\|\boldsymbol{y}_{i,k-1}^t - \boldsymbol{x}^{t-2}\|^2 + \frac{3a^2G^2}{S}.
$$

Proof. Starting from the client update (14), we can rewrite it as

$$
\begin{aligned}
\boldsymbol{d}_{i,k}^t - \nabla f(\boldsymbol{y}_{i,k-1}^t) &= (1-a)V^{t-1} \\
&\quad + \left(\nabla f_i(\boldsymbol{y}_{i,k-1}^t;\zeta_{i,k}^t) - \nabla f_i(\boldsymbol{x}^{t-2};\zeta_{i,k}^t)\right) - \nabla f(\boldsymbol{y}_{i,k-1}^t + \nabla f(\boldsymbol{x}^{t-2})) \\
&\quad + a\left(\frac{1}{S}\sum_{j\in\mathcal{S}^t}\nabla f_j(\boldsymbol{x}^{t-2}) - \nabla f(\boldsymbol{x}^{t-2}\right).
\end{aligned}
$$

We can use the relaxed triangle inequality Lemma 1 to claim

$$
\begin{aligned}
\mathbb{E}\|\boldsymbol{d}_{i,k}^t - \nabla f(\boldsymbol{y}_{i,k-1}^t)\|^2 &= 3(1-a)^2\,\mathbb{E}\|V^{t-1}\|^2 \\
&\quad + 3(1-a)^2\left\|\nabla f_i(\boldsymbol{y}_{i,k-1}^t;\zeta_{i,k}^t) - \nabla f_i(\boldsymbol{x}^{t-2};\zeta_{i,k}^t)) - \nabla f(\boldsymbol{y}_{i,k-1}^t + \nabla f(\boldsymbol{x}^{t-2})\right\|^2 \\
&\quad + 3a^2\left\|\frac{1}{S}\sum_{j\in\mathcal{S}^t}(\nabla f_j(\boldsymbol{x}^{t-2}) - \nabla f(\boldsymbol{x}^{t-2})\right\|^2 \\
&\le 3\,\mathbb{E}\|V^{t-1}\|^2 + 3\delta^2\|\boldsymbol{y}_{i,k-1}^t - \boldsymbol{x}^{t-2}\|^2 + \frac{3a^2G^2}{S}.
\end{aligned}
$$

The last inequality used the Hessian similarity Lemma 3 to bound the second term and the heterogeneity bound (A1) to bound the last term. Also, $(1-a)^2 \le 1$ since $a \in [0,1]$. □

Distance moved in each step. We show that the distance moved by a client in each step during the client update can be controlled.

Lemma 10. *For MimeMVR updates (14) with $\eta \leq \frac{1}{6K\delta}$ and given (A1) and (A2), the following holds*

$$\Delta_{i,k}^t \leq \left(1 + \frac{1}{K}\right)\Delta_{i,k-1}^t + 18\eta^2 K a^2 \frac{G^2}{S} + 18\eta^2 K\, \mathbb{E}\|V^{t-1}\|^2 + 6\eta^2 K \|\nabla f(\boldsymbol{y}_{i,k-1}^t)\|^2 \,,$$

where we define $\Delta_{i,k}^t := \max\Big(\mathbb{E}\|\boldsymbol{y}_{i,k}^t - \boldsymbol{x}^{t-2}\|^2 \,, \mathbb{E}\|\boldsymbol{y}_{i,k}^t - \boldsymbol{x}^{t-1}\|^2, \mathbb{E}\|\boldsymbol{x}^{t-1} - \boldsymbol{x}^{t-2}\|^2\Big)$.

Proof. Starting from the MimeMVR update (14) and the relaxed triangle inequality with $c = 2K$,

$$\begin{aligned}
\mathbb{E}\|\boldsymbol{y}_{i,k}^t - \boldsymbol{x}^{t-2}\|^2 &= \mathbb{E}\|\boldsymbol{y}_{i,k-1}^t - \eta \boldsymbol{d}_{i,k}^t - \boldsymbol{x}^{t-2}\|^2 \\
&\leq \left(1 + \frac{1}{2K}\right)\mathbb{E}\|\boldsymbol{y}_{i,k-1}^t - \boldsymbol{x}^{t-2}\|^2 + (2K+1)\eta^2\, \mathbb{E}\|\boldsymbol{d}_{i,k}^t\|^2 \\
&\leq \left(1 + \frac{1}{2K}\right)\mathbb{E}\|\boldsymbol{y}_{i,k-1}^t - \boldsymbol{x}^{t-2}\|^2 + 6K\eta^2\, \mathbb{E}\|\boldsymbol{d}_{i,k}^t - \nabla f(\boldsymbol{y}_{i,k-1}^t)\|^2 \\
&\qquad + 6K\eta^2\, \mathbb{E}\|\nabla f(\boldsymbol{y}_{i,k-1}^t)\|^2 \\
&\leq \left(1 + \frac{1}{2K} + 18K\eta^2\delta^2\right)\mathbb{E}\|\boldsymbol{y}_{i,k-1}^t - \boldsymbol{x}^{t-2}\|^2 \\
&\qquad + 18K\eta^2\, \mathbb{E}\|V^{t-1}\|^2 + \frac{18K\eta^2 a^2 G^2}{S} + 6K\eta^2\, \mathbb{E}\|\nabla f(\boldsymbol{y}_{i,k-1}^t)\|^2 \,.
\end{aligned}$$

The last inequality used the update variance bound Lemma 9. We can simplify the expression further since $\eta \leq \frac{1}{6K\delta}$ implies $18K\eta^2\delta^2 \leq \frac{1}{2K}$. Similar computations for $\mathbb{E}\|\boldsymbol{y}_{i,k}^t - \boldsymbol{x}^{t-1}\|^2$ yield the lemma. □

Progress in one step. Now we have all the tools required to compute the progress made in each round.

Lemma 11. *For any step of MimeMVR with step size $\eta \leq \min\big(\frac{1}{L}, \frac{1}{40\delta K}\big)$ and momentum parameter $a = 1536\eta^2\delta^2K^2$. Then, given that (A1)–(A3) hold, we have*

$$\begin{aligned}
\mathbb{E}[f(\boldsymbol{y}_{i,k}^t)] + \frac{3\eta}{a}\,\mathbb{E}\|V^t\|^2 + \frac{8\eta\delta^2 K}{a}\left(1 + \frac{2}{K}\right)^{K-k}\Delta_{i,k}^t \\
\leq \mathbb{E}[f(\boldsymbol{y}_{i,k-1}^t)] + \frac{3\eta}{a}\,\mathbb{E}\|V^{t-1}\|^2 + \frac{8\eta\delta^2 K}{a}\left(1 + \frac{2}{K}\right)^{K-(k-1)}\Delta_{i,k-1}^t \\
- \frac{\eta}{4}\,\mathbb{E}\|\nabla f(\boldsymbol{y}_{i,k-1}^t)\|^2 + \frac{11136\eta^3\delta^2K^2G^2}{S} \,.
\end{aligned}$$

Proof. The assumption that f is L-smooth implies a quadratic upper bound (9).

$$\begin{aligned}
f(\boldsymbol{y}_{i,k}^t) - f(\boldsymbol{y}_{i,k-1}^t) &\leq -\eta\langle \nabla f(\boldsymbol{y}_{i,k-1}^t), \boldsymbol{d}_{i,k}^t\rangle + \frac{L\eta^2}{2}\|\boldsymbol{d}_{i,k}^t\|^2 \\
&= -\frac{\eta}{2}\|\nabla f(\boldsymbol{y}_{i,k-1}^t)\|^2 + \frac{L\eta^2 - \eta}{2}\|\boldsymbol{d}_{i,k}^t\|^2 + \frac{\eta}{2}\|\boldsymbol{d}_{i,k}^t - \nabla f(\boldsymbol{y}_{i,k-1}^t)\|^2 \,.
\end{aligned}$$

The second equality used the fact that for any a, b, $-2ab = (a-b)^2 - a^2 - b^2$. The second term can be removed since $\eta \leq \frac{1}{L}$. Taking expectation on both sides and using the update variance bound Lemma 9,

$$\begin{aligned}
\mathbb{E}\, f(\boldsymbol{y}_{i,k}^t) - \mathbb{E}\, f(\boldsymbol{y}_{i,k-1}^t) \leq -\frac{\eta}{2}\,\mathbb{E}\|\nabla f(\boldsymbol{y}_{i,k-1}^t)\|^2 + \frac{3\eta a^2 G^2}{2S} \\
+ \frac{3\eta}{2}\,\mathbb{E}\|V^{t-1}\|^2 + \frac{3\eta\delta^2}{2}\,\mathbb{E}\|\boldsymbol{y}_{i,k-1}^t - \boldsymbol{x}^{t-2}\|^2
\end{aligned}$$

Multiplying the momentum variance bound Lemma 8 by $\frac{3\eta}{a}$, we have

$$\frac{3\eta}{a}\,\mathbb{E}\|V^t\|^2 \le \frac{3\eta}{a}\,\mathbb{E}\|V^{t-1}\|^2 + \frac{6\eta\delta^2}{a}\,\mathbb{E}\|\boldsymbol{x}^{t-1}-\boldsymbol{x}^{t-2}\|^2 + \frac{6\eta a G^2}{S} - 3\eta\,\mathbb{E}\|V^{t-1}\|^2\,.$$

We will also multiply the distance bound Lemma 10 by $\frac{8\eta\delta^2 K}{a}\big(1+\frac{2}{K}\big)^{K-k}$. Note that for any $K \ge 1$ and $k \in [K]$, we have $1 \le \big(1+\frac{2}{K}\big)^{K-k} \le 8$. Then we get

$$\begin{aligned}\frac{8\eta\delta^2 K}{a}\left(1+\frac{2}{K}\right)^{K-k}\Delta_{i,k}^t \le{}& \frac{8\eta\delta^2 K}{a}\left(1+\frac{2}{K}\right)^{K-(k-1)}\Delta_{i,k-1}^t - \frac{8\eta\delta^2}{a}\Delta_{i,k-1}^t \\ &+ 1152\eta^3\delta^2K^2 a\frac{G^2}{S} + \frac{1152\eta^3\delta^2K^2}{a}\,\mathbb{E}\|V^{t-1}\|^2 + \frac{384\eta^3\delta^2K^2}{a}\|\nabla f(\boldsymbol{y}_{i,k-1}^t)\|^2\,,\end{aligned}$$

where recall that we defined $\Delta_{i,k}^t := \max\Big(\mathbb{E}\|\boldsymbol{y}_{i,k}^t-\boldsymbol{x}^{t-2}\|^2\,, \mathbb{E}\|\boldsymbol{y}_{i,k}^t-\boldsymbol{x}^{t-1}\|^2, \mathbb{E}\|\boldsymbol{x}^{t-1}-\boldsymbol{x}^{t-2}\|^2\Big)$. Combining the three inequalities together, we get

$$\begin{aligned}\mathbb{E}\,f(\boldsymbol{y}_{i,k}^t) + \frac{3\eta}{a}\,\mathbb{E}\|V^t\|^2 + \frac{8\eta\delta^2K}{a}\left(1+\frac{2}{K}\right)^{K-k}\Delta_{i,k}^t \\ \le{}& \mathbb{E}\,f(\boldsymbol{y}_{i,k-1}^t) + \frac{3\eta}{a}\,\mathbb{E}\|V^{t-1}\|^2 + \frac{8\eta\delta^2K}{a}\left(1+\frac{2}{K}\right)^{K-(k-1)}\Delta_{i,k-1}^t \\ &+ \left(\frac{384\eta^3\delta^2K^2}{a} - \frac{\eta}{2}\right)\mathbb{E}\|\nabla f(\boldsymbol{y}_{i,k-1}^t)\|^2 \\ &+ \left(1152\eta^2\delta^2K^2 + 6 + \frac{3a}{2}\right)\frac{a\eta G^2}{S} \\ &+ \left(\frac{3\eta}{2} + \frac{1152\eta^3\delta^2K^2}{a} - 3\eta\right)\mathbb{E}\|V^{t-1}\|^2 \\ &+ \left(\frac{3}{2} + \frac{6}{a} - \frac{8}{a}\right)\eta\delta^2\Delta_{i,k-1}^t\,.\end{aligned}$$

Note that $\frac{1152\eta^3\delta^2K^2}{a} = \frac{3\eta}{4}$ since we defined $a = 1536\eta^2\delta^2K^2$. Further, $a \le 1$ when defined this way since we assumed $\eta \le \frac{1}{40\delta K}$. Similarly, the definition of a implies that $\frac{384\eta^3\delta^2K^2}{a} = \frac{\eta}{4}$. Thus, we can simplify the above expression as

$$\begin{aligned}\mathbb{E}\,f(\boldsymbol{y}_{i,k}^t) + \frac{3\eta}{a}\,\mathbb{E}\|V^t\|^2 + \frac{8\eta\delta^2K}{a}\left(1+\frac{2}{K}\right)^{K-k}\Delta_{i,k}^t \\ \le{}& \mathbb{E}\,f(\boldsymbol{y}_{i,k-1}^t) + \frac{3\eta}{a}\,\mathbb{E}\|V^{t-1}\|^2 + \frac{8\eta\delta^2K}{a}\left(1+\frac{2}{K}\right)^{K-(k-1)}\Delta_{i,k-1}^t \\ &- \frac{\eta}{4}\,\mathbb{E}\|\nabla f(\boldsymbol{y}_{i,k-1}^t)\|^2 + \frac{11136\eta^3\delta^2K^2G^2}{S}\,.\end{aligned}$$

This proves the lemma. □

Progress in one round. Let us sum over all the steps within a round to compute the progress made in a full round.

Lemma 12. *For any round of MimeMVR with step size $\eta \le \min\big(\frac{1}{L}, \frac{1}{40\delta K}\big)$ and momentum parameter $a = 1536\eta^2\delta^2K^2$. Then, given that (A1)–(A3) hold, we have*

$$\frac{\eta}{4KS}\sum_{k\in[K], j\in\mathcal{S}^t}\mathbb{E}\|\nabla f(\boldsymbol{y}_{i,k-1}^t)\|^2 \le \Phi^{t-1} - \Phi^t + \frac{11136\eta^3\delta^2K^2G^2}{S}\,,$$

where we define the sequence

$$\Phi^t := \tfrac{1}{K}\,\mathbb{E}[f(\boldsymbol{x}^t)] + \frac{3\eta K}{a}\,\mathbb{E}\|V^t\|^2 + \frac{8\eta\delta^2}{a}\,\mathbb{E}\|\boldsymbol{x}^t-\boldsymbol{x}^{t-1}\|^2\,.$$

Proof. We start by summing Lemma 11 over the client updates

$$\frac{\eta}{4}\sum_{k\in[K]}\mathbb{E}\|\nabla f(\boldsymbol{y}_{i,k-1}^t)\|^2 \le \mathbb{E}[f(\boldsymbol{y}_{i,0}^t)] + \frac{3\eta K}{a}\,\mathbb{E}\|V^{t-1}\|^2 + \frac{8\eta\delta^2 K}{a}\left(1+\frac{2}{K}\right)^K \Delta_{i,0}^t$$
$$- \mathbb{E}[f(\boldsymbol{y}_{i,K}^t)] - \frac{3\eta K}{a}\,\mathbb{E}\|V^t\|^2 + \frac{8\eta\delta^2 K}{a}\Delta_{i,K}^t$$
$$+ \frac{11136\eta^3\delta^2K^3G^2}{S}\,.$$

Recall that we defined $\Delta_{i,k}^t := \max\Big(\mathbb{E}\|\boldsymbol{y}_{i,k}^t - \boldsymbol{x}^{t-2}\|^2\,, \mathbb{E}\|\boldsymbol{y}_{i,k}^t - \boldsymbol{x}^{t-1}\|^2, \mathbb{E}\|\boldsymbol{x}^{t-1} - \boldsymbol{x}^{t-2}\|^2\Big)$. Because $\boldsymbol{y}_{i,0}^t = \boldsymbol{x}^{t-1}$, we can simplify

$$\mathbb{E}[f(\boldsymbol{y}_{i,0}^t)] + \frac{8\eta\delta^2K}{a}\Delta_{i,0}^t \le \mathbb{E}[f(\boldsymbol{x}^{t-1})] + \frac{8\eta\delta^2K}{a}\,\mathbb{E}\|\boldsymbol{x}^{t-1}-\boldsymbol{x}^{t-2}\|^2$$

Then by the averaging Lemma 4, we have

$$\frac{1}{S}\sum_{j\in\mathcal{S}^t}\mathbb{E}[f(\boldsymbol{y}_{j,K}^t)] + \frac{8\eta\delta^2K}{a}\Delta_{j,K}^t \ge \frac{1}{S}\sum_{j\in\mathcal{S}}\mathbb{E}[f(\boldsymbol{y}_{j,K}^t)] + \mathbb{E}\|\boldsymbol{x}^{t-1}-\boldsymbol{y}_{j,K}^t\|^2$$
$$\ge \mathbb{E}[f(\boldsymbol{x}^t)] + \frac{8\eta\delta^2K}{a}\,\mathbb{E}\|\boldsymbol{x}^{t-1}-\boldsymbol{x}^t\|^2\,.$$

So by averaging our recursion over the sampled clients, and diving our summation over the updates by K, we get

$$\frac{\eta}{4KS}\sum_{k\in[K],j\in\mathcal{S}^t}\mathbb{E}\|\nabla f(\boldsymbol{y}_{i,k-1}^t)\|^2 \le \underbrace{\tfrac{1}{K}\,\mathbb{E}[f(\boldsymbol{x}^{t-1})] + \frac{3\eta}{a}\,\mathbb{E}\|V^{t-1}\|^2 + \frac{8\eta\delta^2}{a}\,\mathbb{E}\|\boldsymbol{x}^{t-1}-\boldsymbol{x}^{t-2}\|^2}_{=\Phi^{t-1}}$$
$$-\underbrace{\tfrac{1}{K}\,\mathbb{E}[f(\boldsymbol{x}^t)] + \frac{3\eta K}{a}\,\mathbb{E}\|V^t\|^2 + \frac{8\eta\delta^2}{a}\,\mathbb{E}\|\boldsymbol{x}^t-\boldsymbol{x}^{t-1}\|^2}_{=\Phi^t}$$
$$+\frac{11136\eta^3\delta^2K^2G^2}{S}\,.$$

□

Theorem IV (non-convex convergence of MimeMVR)**.** *Let us run MimeMVR with step size* $\eta \le \min\Big(\frac{1}{15K}\big(\frac{FS}{T\delta^2G^2}\big)^{1/3}, \frac{1}{L}, \frac{1}{40\delta K}\Big)$ *and momentum parameter* $a = 1536\eta^2\delta^2K^2$*. Then, given that* (A1)–(A3) *hold, we have*

$$\frac{1}{KST}\sum_{t\in[T]}\sum_{k\in[K]}\sum_{j\in\mathcal{S}^t}\mathbb{E}\|\nabla f(\boldsymbol{y}_{i,k-1}^t)\|^2 \le \mathcal{O}\Big(\Big(\frac{(1+\delta)G^2F}{ST}\Big)^{2/3} + \frac{(L+\delta K)F}{KT}\Big)\,,$$

where we define $F := f(\boldsymbol{x}^0) - f^\star$.

Proof. Unroll the one round progress Lemma 12 and average over T rounds to get

$$\frac{1}{KST}\sum_{t\in[T]}\sum_{k\in[K]}\sum_{j\in\mathcal{S}^t}\mathbb{E}\|f(\boldsymbol{y}_{i,k-1}^t)\|^2 \le \frac{4(\Phi^0-\Phi^T)}{\eta KT} + \frac{11136\eta^2\delta^2K^2G^2}{S}$$
$$\le \frac{4(f(\boldsymbol{x}^0)-f^\star)}{\eta KT} + \frac{11136\eta^2\delta^2K^2G^2}{S}\,.$$

Our choice of step size now yields the desired rate. □