Breaking the centralized barrier for cross-device federated learning[*]

Sai Praneeth Karimireddy
EPFL
sai.karimireddy@epfl.ch

Martin Jaggi
EPFL
martin.jaggi@epfl.ch

Satyen Kale
Google Research
satyenkale@google.com

Mehryar Mohri
Google Research
mohri@google.com

Sashank J. Reddi
Google Research
sashank@google.com

Sebastian U. Stich
EPFL
sebastian.stich@epfl.ch

Ananda Theertha Suresh
Google Research
theertha@google.com

Abstract

Federated learning (FL) is a challenging setting for optimization due to the heterogeneity of the data across different clients which can cause a *client drift* phenomenon. In fact, designing an algorithm for FL that is uniformly better than simple centralized training has been a major open problem thus far. In this work, we propose a general algorithmic framework, MIME, which i) mitigates client drift and ii) adapts an arbitrary centralized optimization algorithm such as momentum and Adam to the cross-device federated learning setting. MIME uses a combination of *control-variates* and *server-level optimizer state* (e.g. momentum) at every client-update step to ensure that each local update mimics that of the centralized method run on i.i.d. data. We prove a reduction result showing that MIME can translate the convergence of a generic algorithm in the centralized setting into convergence in the federated setting. Moreover, we show that, when combined with momentum-based variance reduction, MIME is provably *faster than any centralized method*–the first such result. We also perform a thorough experimental exploration of MIME's performance on real world datasets (implemented here).

1 Introduction

Federated learning (FL) is an increasingly important large-scale learning framework where the training data remains distributed over a large number of clients, which may be mobile phones or network sensors [38, 37, 43, 44, 28]. A server then orchestrates the clients to train a single model, here referred to as a *server model*, without ever transmitting client data over the network, thereby providing some basic levels of data privacy and security.

Two important settings are distinguished in FL [28, 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 [25]. Thus, in that setting, we may never make even a single pass over

[*] This work was also appears under the alternative title "Mime: Mimicking Centralized Stochastic Algorithms in Federated Learning" [31].

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

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. In this work, we are interested in the more challenging cross-device setting, for which we will formalize and study stochastic optimization algorithms. Importantly, recent advances in FL optimization, such as SCAFFOLD [32] or FedDyn [1], are *not anymore applicable* since they are designed for the cross-silo setting.

The problem. The de facto standard algorithm for the cross-device setting is FEDAVG [43], which performs multiple SGD updates on the available clients before communicating to the server. While this approach can reduce the *frequency* 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* [32]. This in turn leads to slower convergence and can, somewhat counter-intuitively, require *larger total communication* [69]. Despite significant attention received from the optimization community, the communication complexity of heterogeneous cross-device has not improved upon that of simple centralized methods, which take no local steps (aka SERVER-ONLY methods). Furthermore, algorithmic innovations such as momentum [59, 14], adaptivity [35, 75, 77], and clipping [71, 72, 76] are critical to the success of deep learning applications. The lack of a theoretical understanding of the impact of multiple client steps has also hindered adapting these techniques in a principled manner into the client updates, in order to replace the vanilla SGD update of FEDAVG.

To overcome such deficiencies, we propose a new framework, MIME, that mitigates client drift and can adapt an arbitrary centralized optimization algorithm, e.g. SGD with momentum or Adam, to the federated setting. In each local client update, MIME uses global optimizer state, e.g. momentum or adaptive learning rates, and an SVRG-style correction to mimic the updates of the centralized algorithm run on i.i.d. data. This optimizer state is computed only at the server level and kept fixed throughout the local steps, thereby avoiding overfitting to the atypical local data of any single client.

Contributions. We summarize our main results below.

- MIME **framework.** We formalize the cross-device federated learning problem, and propose a new framework MIME that can adapt arbitrary centralized algorithms to this setting.
- **Convergence result.** We prove a result showing that MIME successfully reduces client drift. We also prove that the convergence of any generic algorithm in the centralized setting translates convergence of its MIME version in the federated setting.
- **Speed-up over centralized methods.** By carefully tracking the bias introduced due to multiple local steps, we prove that MIME with momentum-based variance reduction (MVR) can beat a lower bound for centralized methods, thus breaking a fundamental barrier. This is the first such result in FL, and also the first general result showing asymptotic speed-up due to local steps.
- **Empirical validation.** We 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 indeed converge 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 [78] derived an analysis with asymptotic convergence. Sharper and more refined analyses of the same method, sometimes called local SGD, were provided by [56], and more recently by [57], [47], [34], and [70], for identical functions. Their analysis was extended to heterogeneous clients in [68, 74, 32, 34, 36]. [11] 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 [32] and [69] for general functions, proving that FEDAVG can be slower than even SGD for heterogeneous data, due to the *client-drift*.

Comparison to SCAFFOLD*:* For the cross-silo setting where the number of clients is relatively low, [32] 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. FedDyn [1] reduces the communication requirements, but also requires persistent stateful clients. In contrast, we focus on the cross-device setting where clients may be visited only once during training and where they are *stateless* (and thus SCAFFOLD and FedDyn are inapplicable). This is akin to the difference between the finite-sum (corresponding to cross-silo) and stochastic (cross-device) settings in traditional centralized optimization [39].

Comparison to FedAvg and variants: [26] and [67] observed that using *server momentum* significantly improves over vanilla FEDAVG. This idea was generalized by [49], 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. We henceforth refer to this *meta algorithm* as FedAvg. FedAvgSGD, FedAvgMom, FedAvgAdam denote specific instantiations of the server optimizer in FedAvg with SGD, Momentum or Adam. 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, [40] proposed to add a regularizer to ensure client updates remain close. However, this may slow down convergence (cf. Fig. 5 and [32, 66]). Other orthogonal directions which can be combined with MIME include tackling computation heterogeneity, where some clients perform many more updates than others [66], improving fairness by modifying the objective [44, 41], incorporating differential privacy [20, 2, 61], Byzantine adversaries [48, 65, 30], secure aggregation [8, 24], etc. We defer additional discussion to the extensive survey by [28].

Momentum based variance reduction. Initial optimal methods for stochastic non-convex optimization like SPIDER [17] and SARAH [46] required intermittently computing very large batch gradients. Subsequently, it was shown that momentum based variance reduction (MVR) methods obtained a similar optimal rate without needing such large batch gradient computations [62, 14]. Momentum is an exponential moving average of many stochastic gradients and so it has much smaller variance than the stochastic gradients themselves. However, because these gradients are computed at different parameters it also has a bias. MVR adds a small additional correction term which significantly reduces this bias and provides improved rates.

2 Problem setup

This section formalizes the problem of cross-device federated learning [28]. Cross-device FL is characterized by a large number of client devices like mobile phones which may potentially connect to the server at most once. Due to their transient nature, it is not possible to store any state on the clients, precluding an algorithm like SCAFFOLD. Furthermore, each client has only a few samples, and there is wide heterogeneity in the samples across clients. Finally, communication is a major bottleneck and a key metric for optimization in this setting is the number of communication rounds.

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

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

Here, f_i denotes the loss function of client i and $\{\zeta_{i,1},\dots,\zeta_{i,n_i}\}$ its local data. Since the number of clients is extremely large, while the 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. Due to the transient and heterogeneous nature of the clients, it is easy to see that the problem becomes intractable with arbitrarily dissimilar clients. Thus, it is **necessary** to assume bounded dissimilarity across clients.

(A1) G^2**-BGV** or bounded inter-client gradient variance: there exists $G \geq 0$ such that

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

Next, we also characterize the variance in the Hessians.

(A2) δ**-BHV** or bounded Hessian variance: Almost surely, the loss function of any client i satisfies

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

This is in contrast to the usual smoothness assumption that can be stated as:

(A2*) L**-smooth**: $\|\nabla^2 f_i(\boldsymbol{x};\zeta)\| \leq L\,,\ \forall \boldsymbol{x}$, a.s. for any i.

Note that if $f_i(\boldsymbol{x};\zeta)$ is L-smooth then (A2) is satisfied with $\delta \leq 2L$, and hence (A2) is *weaker* than (A2*). In realistic examples we expect the clients to be similar and hence that $\boldsymbol{\delta \ll L}$. In addition, we assume that $f(\boldsymbol{x})$ is bounded from below by $f^\star$ and is L-smooth, as is standard.

3 Mime framework

In this section we describe how to adapt an arbitrary centralized optimizer (referred to as the "base" optimizer) which may have internal state (e.g. momentum) to the federated learning problem (1) while ensuring there is no client-drift. Algorithm 4 describes our framework. We develop two variants, MIME and MIMELITE, which consist of three components i) a base optimizer we are seeking to mimic, ii) the global (server) optimizer state computation, and iii) the local client updates.

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

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

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

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

As an example, consider SGD with momentum. The state here is the momentum m_t and uses the following update steps:

$$\begin{aligned} x_t &= x_{t-1} - \eta\left((1-\beta)\nabla f_i(x_{t-1}) \right.\\ &\qquad\left. + \beta m_{t-1}\right), \\ m_t &= (1-\beta)\nabla f_i(x_{t-1}) + \beta m_{t-1}\,. \end{aligned}$$

Thus, SGD with momentum can be represented in the above generic form with $\mathcal{U}(g, s) = (1-\beta)g + \beta s$ and $\mathcal{V}(g, s) = (1-\beta)g + \beta s$. Table 5 in Appendix shows how other algorithms like Adam, Adagrad, etc. can be represented in this manner. We keep the update $\mathcal{U}$ to be linear in the gradient g, whereas $\mathcal{V}$ can be more complicated. This implies that while the parameter update step $\mathcal{U}$ is relatively resilient to receiving a biased gradient g while $\mathcal{V}$ can be much more sensitive.

Compute optimizer state globally, apply locally. When updating the optimizer state 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 optimizer state remain unbiased and representative of the global function $f(\cdot)$. At the end of the round, the server performs

$$\begin{aligned} s &\leftarrow \mathcal{V}\Big(\tfrac{1}{|\mathcal{S}|}\textstyle\sum_{i\in\mathcal{S}} \nabla f_i(x),\ s\Big)\,, \\ \nabla f_i(x) &= \tfrac{1}{n_i}\textstyle\sum_{\nu=1}^{n_i} \nabla f_i(x; \zeta_{i,\nu})\,. \end{aligned} \tag{OPTSTATE}$$

Note that we use full-batch gradients computed at the server parameters x, not client parameters 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 $y_i \leftarrow x$, repeat the following K times

$$y_i \leftarrow y_i - \eta\,\mathcal{U}(g_i, s) \tag{CLIENTSTEP}$$

where $g_i \leftarrow \nabla f_i(y_i; \zeta)$ for MIMELITE, and $g_i \leftarrow \nabla f_i(y_i; \zeta) - \nabla f_i(x; \zeta) + \frac{1}{|\mathcal{S}|}\sum_{j\in\mathcal{S}} \nabla f_j(x)$ for MIME. MIMELITE simply uses the local minibatch gradient whereas MIME uses an SVRG style correction [27]. 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 weight all averages across the clients by the number of datapoints n_i [43], and we perform K epochs instead of K steps [66].

4 Theoretical analysis of Mime

Table 1 summarizes the rates of Mime (highlighted in blue) and MimeLite (highlighted in green) and compares them to Server-Only methods when using SGD, Adam and momentum methods as the base algorithms. We will first examine the convergence of Mime and MimeLite with a generic base optimizer and show that its properties are preserved in the federated setting. We then examine a specific momentum based base optimizer, and prove that Mime and MimeLite can be asymptotically faster than the best server-only method. This is the first result to prove the usefulness of local steps and demonstrate asymptotic speed-ups.

4.1 Convergence with a generic base optimizer

We will prove a generic reduction result demonstrating that if the underlying base algorithm converges, and is robust to slight perturbations, then Mime and MimeLite also preserve the convergence of the algorithm when applied to the federated setting with additinoal local steps.

Theorem I. *Suppose that we have G^2 inter-client gradient variance (A1), L-smooth $\{f_i\}$ (A2*), and σ^2 intra-client gradient variance (A3). Further, suppose that the updater $\mathcal{U}$ of our base-optimizer $\mathcal{B} = (\mathcal{U}, \mathcal{V})$ satisfies i) linearity for a fixed state $\boldsymbol{s}$: $\mathcal{U}(\boldsymbol{g}_1 + \boldsymbol{g}_2; \boldsymbol{s}) = \mathcal{U}(\boldsymbol{g}_1; \boldsymbol{s}) + \mathcal{U}(\boldsymbol{g}_2; \boldsymbol{s})$, and ii) Lipschitzness: $\|\mathcal{U}(\boldsymbol{g}; \boldsymbol{s})\| \leq B\|\boldsymbol{g}\|$ for some $B \geq 0$. Then, running* Mime *or* MimeLite *with K local updates and step-size η is equivalent to running a **centralized** algorithm with step-size $\tilde{\eta} := K\eta \leq \frac{1}{2LB}$, and updates*

$$\boldsymbol{x}_t \leftarrow \boldsymbol{x}_{t-1} - \tilde{\eta}\,\mathcal{U}(\boldsymbol{g}_t + \boldsymbol{e}_t, \boldsymbol{s}_{t-1})\,, \text{ and}$$

$$\boldsymbol{s}_t \leftarrow \mathcal{V}(\boldsymbol{g}_t, \boldsymbol{s}_{t-1})\,, \text{ where we have}$$

an unbiased gradient $\mathbb{E}_t[\boldsymbol{g}_t] = \nabla f(\boldsymbol{x}_{t-1})$, with variance bounded as

$$\mathbb{E}_t\|\boldsymbol{g}_t - \nabla f(\boldsymbol{x}_{t-1})\|^2 \leq \begin{cases} \frac{G^2}{S} & \text{Mime}\,, \\ \frac{G^2}{S} + \frac{\sigma^2}{KS} & \text{MimeLite}\,. \end{cases}$$

and finally a small error bounded as

$$\frac{1}{B^2L^2\tilde{\eta}^2}\,\mathbb{E}_t\|\boldsymbol{e}_t\|^2 \leq \begin{cases} \mathbb{E}_t\|\boldsymbol{g}_t\|^2 & \text{Mime}\,, \\ \mathbb{E}_t\|\boldsymbol{g}_t\|^2 + G^2 + \frac{\sigma^2}{K} & \text{MimeLite}\,. \end{cases}$$

Here, we have proven that Mime and MimeLite truly mimic the centralized base algorithm with very small perturbations—the magnitude of $\boldsymbol{e}_t$ is $\mathcal{O}(\tilde{\eta}^2)$. The key to the result is the linearity of the parameter update step $\mathcal{U}(\,\cdot\,; \boldsymbol{s})$. By separating the base optimizer into a very simple parameter step $\mathcal{U}$ and a more complicated optimizer state update step $\mathcal{V}$, we can ensure that commonly used algorithms such as momentum, Adam, Adagrad, and others all satisfy this property. Armed with this general reduction, we can easily obtain specific convergence results.

Corollary II ((Mime/MimeLite) with SGD)**.** *Given that the conditions in Theorem I are satisfied, let us run T rounds with K local steps using SGD as the base optimizer and output $\boldsymbol{x}^{out}$. This output satisfies $\mathbb{E}\|\nabla f(\boldsymbol{x}^{out})\|^2 \leq \epsilon$ for $F := f(\boldsymbol{x}_0) - f^\star$, $\tilde{G}^2 := G^2 + \sigma^2/K$ and*

- ***μ-PL inequality:** $\eta = \tilde{\mathcal{O}}\big(\frac{1}{\mu KT}\big)$, and*

$$T = \begin{cases} \tilde{\mathcal{O}}\Big(\frac{LG^2}{\mu S\epsilon} + \frac{LF}{\mu}\log\big(\frac{1}{\epsilon}\big)\Big) & \text{Mime}\,, \\ \tilde{\mathcal{O}}\Big(\frac{L\tilde{G}^2}{\mu S\epsilon} + \frac{L\tilde{G}}{\mu\sqrt{\epsilon}} + \frac{LF}{\mu}\log\big(\frac{1}{\epsilon}\big)\Big) & \text{MimeLite}\,. \end{cases}$$

- ***Non-convex:** for $\eta = \mathcal{O}\Big(\sqrt{\frac{FS}{L\tilde{G}^2TK^2}}\Big)$, and*

$$T = \begin{cases} \mathcal{O}\Big(\frac{LG^2F}{S\epsilon^2} + \frac{LF}{\epsilon}\Big) & \text{Mime}\,, \\ \mathcal{O}\Big(\frac{L\tilde{G}^2F}{S\epsilon^2} + \frac{L^2\tilde{G}F}{\epsilon^{3/2}} + \frac{LF}{\epsilon}\Big) & \text{MimeLite}\,. \end{cases}$$

Table 1: Number of communication rounds required to reach $\|\nabla f(\boldsymbol{x})\|^2 \leq \epsilon$ (log factors are ignored) with S clients sampled each round. All analyses except SCAFFOLD assume G^2 bounded gradient dissimilarity (A1). All analyses assume L-smooth losses, except MimeLiteMVR and MimeMVR, which only assume δ bounded Hessian dissimilarity (A2). Convergence of SCAFFOLD depends on the total number of clients N which is potentially infinite. FEDAVG and MIMELITE are slightly slower than the server-only methods due to additional drift terms in most cases. MIME is the fastest and either matches or improves upon the optimal statistical rates (first term in the rates). In fact, MimeMVR and MimeLiteMVR beat lower bounds for any server-only method when $\delta \ll L$.

Algorithm	Non-convex	μ-PL inequality
SCAFFOLD[a] [32]	$\left(\frac{N}{S}\right)^{\frac{2}{3}} \frac{L}{\epsilon}$	$\frac{N}{S} + \frac{L}{\mu}$
SGD		
SERVER-ONLY [21]	$\frac{LG^2}{S\epsilon^2} + \frac{L}{\epsilon}$	$\frac{G^2}{\mu S\epsilon} + \frac{L}{\mu}$
MimeLiteSGD≡ FedAvgSGD [c]	$\frac{LG^2}{S\epsilon^2} + \frac{L^2 G}{\epsilon^{3/2}} + \frac{L}{\epsilon}$	$\frac{G^2}{\mu S\epsilon} + \frac{LG}{\mu\sqrt{\epsilon}} + \frac{L}{\mu}$
MimeSGD	$\frac{LG^2}{S\epsilon^2} + \frac{L}{\epsilon}$	$\frac{G^2}{\mu S\epsilon} + \frac{L}{\mu}$
ADAM		
SERVER-ONLY [75][b]	$\frac{L}{\epsilon - G^2/S}$	–
MimeLiteAdam[bc]	$\frac{L\sqrt{S}}{\epsilon - G^2/S}$	–
MimeAdam[b]	$\frac{L}{\epsilon - G^2/S}$	–
Momentum Variance Reduction (MVR)		
SERVER-ONLY [14]	$\frac{LG}{\sqrt{S}\epsilon^{3/2}} + \frac{G^2}{S\epsilon} + \frac{L}{\epsilon}$	–
MimeLiteMVR[d]	$\frac{\delta(G+\sigma)}{\epsilon^{3/2}} + \frac{G^2+\sigma^2}{\epsilon} + \frac{\delta}{\epsilon}$	–
MimeMVR[d]	$\frac{\delta G}{\sqrt{S}\epsilon^{3/2}} + \frac{G^2}{S\epsilon} + \frac{\delta}{\epsilon}$	–
SERVER-ONLY lower bound [5]	$\Omega\left(\frac{LG}{\sqrt{S}\epsilon^{3/2}} + \frac{G^2}{S\epsilon} + \frac{L}{\epsilon}\right)$	$\Omega\left(\frac{G^2}{S\epsilon}\right)$

[a] Num. clients (N) can be same order as num. total rounds or even ∞, making the bounds vacuous.
[b] Adam requires large batch-size $S \geq G^2/\epsilon$ to converge [50, 75]. Convergence of FedAdam with client sampling is unknown ([49] only analyze with full client participation).
[c] Requires $K \geq \sigma^2/G^2$ number of local updates. Typically, intra-client variance is small ($\sigma^2 \lesssim G^2$).
[d] Requires $K \geq L/\delta$ number of local updates. Faster than the lower bound (and hence any SERVER-ONLY algorithm) when $\delta \ll L$ i.e. our methods can take advantage of Hessian similarity, whereas SERVER-ONLY methods cannot. In worst case, $\delta \approx L$ and all methods are comparable.

If we take a sufficient number of local steps $K \geq G^2/\sigma^2$, then we have $\tilde{G} = \mathcal{O}(G)$ in the above rates. On comparing with the rates in Table 1 for SERVER-ONLY SGD, we see that MIME exactly matches its rates. MIMELITE matches the asymptotic term but has a few higher order terms. Note that when using SGD as the base optimizer, MIMELITE becomes exactly the same as FEDAVG and hence has the same rate of convergence.

Corollary III ((Mime/MimeLite) with Adam)**.** *Suppose that the conditions in Theorem I are satisfied, and further $|\nabla_j f_i(\boldsymbol{x})| \leq H$ for any coordinate $j \in [d]$. Then let us run T rounds using Adam as the base optimizer with K local steps, $\beta_1 = 0$, $\varepsilon_0 > 0$, $\eta \leq \varepsilon_0^2/KL(H+\varepsilon_0)$, and any $\beta_2 \in [0,1)$. Output $\boldsymbol{x}^{out}$ chosen randomly from $\{\boldsymbol{x}_1, \ldots \boldsymbol{x}_T\}$ satisfies $\mathbb{E}\|\nabla f(\boldsymbol{x}^{out})\|^2 \leq \epsilon$ for*

$$T = \begin{cases} \mathcal{O}\left(\frac{LF(H+\varepsilon_0)^2}{\varepsilon_0^2(\epsilon - \tilde{G}^2/S)}\right) & \text{MIME Adam}\,, \\ \mathcal{O}\left(\frac{LF(H+\varepsilon_0)^2\sqrt{S}}{\varepsilon_0^2(\epsilon - \tilde{G}^2/S)}\right) & \text{MIMELITE Adam}\,. \end{cases}$$

where $F := f(\boldsymbol{x}_0) - f^\star$, $\tilde{G}^2 := G^2 + \sigma^2/K$.

Note that here ε_0 represents a small positive parameter used in Adam for regularization, and is different from the error ϵ. Similar to the SERVER-ONLY analysis of Adam [75], we assume $\beta_1 = 0$

and that batch size is large enough such that $S \geq G^2/\epsilon$. A similar analysis can also be carried out for AdaGrad, and other novel variants of Adam [42].

4.2 Circumventing server-only lower bounds

The rates obtained above, while providing a safety-check, do not beat those of the SERVER-ONLY approach. The previous best rates for cross-device FL correspond to MimeLiteSGD which is $\mathcal{O}(\frac{LG^2}{S\epsilon^2} + \frac{L^2G}{\epsilon^{3/2}})$ [34, 36, 69]. While, using a separate server-learning rate can remove the effect of the second term [33], this at best matches the rate of SERVER-ONLY SGD $\mathcal{O}(\frac{LG^2}{S\epsilon^2})$. This is significantly slower than simply using momentum based variance reduction (MVR) as in in the FL setting (SERVER-ONLY MVR) which has a communication complexity of $\mathcal{O}(\frac{LG}{\sqrt{S}\epsilon^{3/2}})$ [14]. Thus, even though the main reason for studying local-step methods was to improve the communication complexity, none thus far show such improvement. The above difficulty of beating SERVER-ONLY may not be surprising given the two sets of strong lower bounds known.

Necessity of local steps. Firstly, [5] show a gradient oracle lower bound of $\Omega(\frac{LG}{\sqrt{S}\epsilon^{3/2}})$. This matches the complexity of MVR, and hence at first glance it seems that SERVER-ONLY MVR is optimal. However, the lower bound is really only on the number of gradients computed and not on the number of clients sampled (sample complexity) [18], or number of rounds of communication required. In particular, multiple local updates increases number of gradients computed *without needing additional communication* offers us a potential way to side-step such lower bounds. A careful analysis of the bias introduced as a result of such local steps is a key part of our analysis.

Necessity of δ-BHD. A second set of lower bounds directly study the number of communication rounds required in heterogeneous optimization [6, 69]. These results prove that there exist settings where local steps provide no advantage and SERVER-ONLY methods are optimal. This however contradicts real world experimental evidence [43]. As before, the disparity arises due to the contrived settings considered by the lower bounds. For distributed optimization (with full client participation) and convex quadratic objectives, δ-BHD (A2) was shown to be a sufficient [54, 51] and *necessary* [6] condition to circumvent these lower bounds and yield highly performant methods. We similarly leverage δ-BHD (A2) to design novel methods which significantly extend prior results to i) all smooth non-convex functions (not just quadratics), and ii) cross-device FL with client sampling.

We now state our convergence results with momentum based variance reduction (MVR) as the base-algorithm since it is known to be optimal in the SERVER-ONLY setting.

Theorem IV. *For L-smooth f with G^2 gradient dissimilarity* (A1), *δ Hessian dissimilarity* (A2) *and $F := (f(\boldsymbol{x}^0) - f^\star)$, let us run MVR as the base algorithm for T rounds with $K \geq L/\delta$ local steps and generate an output $\boldsymbol{x}^{out}$. This output satisfies $\mathbb{E}\|\nabla f(\boldsymbol{x}^{out})\|^2 \leq \epsilon$ for*

- ***MimeMVR*** *: $\eta = \mathcal{O}\Big(\min\big(\frac{1}{\delta K}, (\frac{SF}{G^2TK^3})^{1/3}\big)\Big)$, momentum $\beta = 1 - \mathcal{O}(\frac{\delta^2 S^{2/3}}{(TG^2)^{2/3}})$, and*
$$T = \mathcal{O}(\frac{\delta G F}{\sqrt{S}\epsilon^{3/2}} + \frac{G^2}{S\epsilon} + \frac{\delta F}{\epsilon}) .$$
- ***MimeLiteMVR*** *: $\eta = \mathcal{O}\Big(\min\big(\frac{1}{\delta K}, (\frac{F}{\hat{G}^2TK^3})^{1/3}\big)\Big)$, momentum $\beta = 1 - \mathcal{O}(\frac{\delta^2}{(T\hat{G}^2)^{2/3}})$, and*
$$T = \mathcal{O}(\frac{\delta \hat{G} F}{\epsilon^{3/2}} + \frac{\hat{G}^2}{\epsilon} + \frac{\delta F}{\epsilon}) .$$

Here, we define $\hat{G}^2 := G^2 + \sigma^2$ and the expectation in $\mathbb{E}\|\nabla f(\boldsymbol{x}^{\text{out}})\|^2 \leq \epsilon$ is taken both over the sampling of the clients during the running of the algorithm, the sampling of the mini-batches in local updates, and the choice of $\boldsymbol{x}^{\text{out}}$ (which is chosen randomly from the client iterates $\boldsymbol{y}_i$).

Remarkably, the rates of our methods are independent of L and only depend on δ. Thus, when $\delta \leq L$ and $\delta \leq {}^{L}/_{S}$ for MimeMVR and MimeLiteMVR, the rates beat the server only lower bound of $\Omega(\frac{LG}{\sqrt{S}\epsilon^{3/2}})$. In fact, if the Hessian variance is small and $\delta \approx 0$, our methods only need $\mathcal{O}({}^{1}/_{\epsilon})$ rounds to communicate. Intuitively, our results show that local steps are very useful when heterogeneity (represented by δ) is smaller than optimization difficulty (captured by smoothness constant L).

MimeMVR uses a momentum parameter β of the order of $(1 - \mathcal{O}(TG^2)^{-2/3})$ i.e. as T increases, β asymptotically approaches 1. In contrast, previous analyses of distributed momentum (e.g. [73]) prove rates of the form $\frac{G^2}{S(1-\beta)\epsilon^2}$, which are worse than that of standard SGD by a factor of $\frac{1}{1-\beta}$.

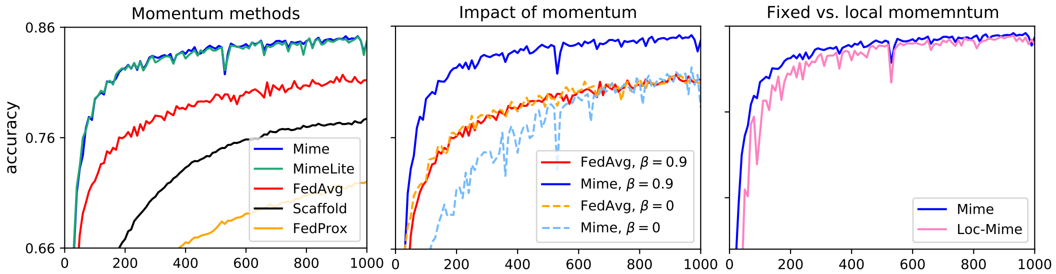

Figure 1: **Mime**, **MimeLite**, **FedAvg**, **Scaffold**, **FedProx**, and **Loc-Mime** with SGD+momentum using 10 local epochs, run on EMNIST62 and a 2 hidden layer (300u-100) MLP. (Left) Mime and MimeLite are nearly identical and outperform the rest (7× faster). (Center) Mime makes better use of momentum than FedAvg, with a large increase in performance. (Right) Locally adapting momentum slows down convergence and makes it more unstable.

Thus, ours is also the first result which theoretically showcases the usefulness of using large momentum in distributed and federated learning. While we only prove the utility of local steps for MimeMVR, we believe our theory can be extended to other local update methods as well.

Our analysis is highly non-trivial and involves two 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, and ii) carefully keeping track of the bias introduced via additional local steps. Our experiments (Sec. 5) verify our theoretical insights are indeed applicable in deep learning settings as well. See App. B for a proof sketch and App. G–H detailed proofs.

5 Experimental analysis on real world datasets

We run experiments on *natively* federated datasets to confirm our theory and accurately measure real world performance. Our main findings are i) MIME and MIMELITE consistently outperform FEDAVG, and ii) momentum and adaptivity significantly improves performance.

5.1 Setup

Algorithms. We consider three (meta) algorithms: FEDAVG, MIME, and MIMELITE. Each of these adapt four base optimizers: SGD, momentum, Adam, and Adagrad.
FEDAVG follows [49] 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 [67]. This is done to ensure all algorithms have the same number of hyper-parameters.
MIME and MIMELITE follow Algorithm 4 and also run a fixed number of epochs on the client. However, note that this requires communicating both the full local-batch gradient as well as the parameter updates doubling the communication required to be sent by the client. For a fairer comparison, we split the sampled clients in MIME and MIMELITE into two groups–the first communicates only full local-batch gradient and the latter communicates only parameter updates. Thus, all methods have **equal client communication** to the server. This variant retains the convergence guarantees up to constants (details in the Appendix). We also run Loc-MIME where instead of keeping the global optimizer state fixed, we update it locally within the client. The optimizer state is reset after the round finishes. In all methods, aggregation is weighted by the number of samples on the clients.

Datasets and models. We run five simulations on three real-world federated datasets: EMNIST62 with i) a linear classifier, ii) an MLP, and iii) a CNN, iv) a charRNN on Shakespeare, and v) an LSTM for next word prediction on StackOverflow, all accessed through Tensorflow Federated [60]. The learning rates were individually tuned and other optimizer hyper-parameters such as β for momentum, β_1, β_2, ε_0 for Adam and AdaGrad were left to their default values, unless explicitly stated otherwise. We refer to Appendix C for additional setup details and discussion.

5.2 Ablation and comparative study

In order to study the different algorithms, we train a 2 hidden layer (300μ-100) MLP on EMNIST62 with 10 local epochs for 1k rounds and use SGD+momentum (with tuned β) as the base optimizer.

Mime $\approx$ MimeLite $>$ FedAvg $>$ SCAFFOLD $>$ FedProx. Fig. 1 (left) shows MIME and MIMELITE have nearly identical performance, and are about 7× faster than FedAvg. This implies

Table 2: Validation % accuracies after training for 1000 rounds. Best results for each dataset is underlined and the best within each base optimizer is bolded. The number of clients sampled per round has been reduced for Mime and MimeLite to ensure all methods have **equal client and server communication**. Final accuracies obtained by Mime and MimeLite are competitive with FedAvg, especially with adaptive base optimizers. FedAvg seems unstable with Adam.

		EMNIST logistic	EMNIST CNN	Shakespeare	StackOverflow
SGD	FedAvgSGD	66.8	**85.8**	**56.7**	**23.8**
	MimeLiteSGD	66.8	**85.8**	**56.7**	**23.8**
	MimeSGD	**67.4**	85.3	56.1	12.5
Momentum	FedAvgMom	67.4	85.7	**55.4**	**22.2**
	MimeLiteMom	67.4	**86.0**	49.8	19.9
	MimeMom	**67.5**	85.9	53.6	19.3
Adam	FedAvgAdam	67.3	85.9	18.5	3.2
	MimeLiteAdam	<u>**68.0**</u>	86.4	54.0	21.5
	MimeAdam	<u>**68.0**</u>	<u>**86.6**</u>	**54.1**	**22.8**
Adagrad	FedAvgAdagrad	**67.6**	86.3	55.5	<u>**24.2**</u>
	MimeLiteAdagrad	66.6	85.5	56.8	23.8
	MimeAdagrad	67.4	**86.3**	<u>**57.1**</u>	14.7

our strategy of applying momentum to client updates is faster than simply using server momentum. FedProx [40] uses an additional regularizer μ tuned over $[0.1, 0.5, 1]$ ($\mu = 0$ is the same as FedAvg). Regularization does not seem to reduce client drift but still slows down convergence [66]. SCAFFOLD [32] is also slower than Mime and FedAvg in this setup. This is because in cross-device setting with a large number of clients ($N = 3.4k$) means that each client is visited less than 6 times during the entire training (20 clients per round for 1k rounds). This means that the correction term utilized by SCAFFOLD uses control-variates which are quite stale (computed about 200 rounds ago) which slows down the convergence. In contrast, the SVRG correction term in Mime is computed using clients sampled in the current or previous rounds, and so is much more accurate.

With momentum $>$ without momentum. Fig. 1 (center) 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 $>$ locally updated optimizer state. Finally, we check how the performance of Mime changes if instead of keeping the momentum fixed throughout a round, we let it change. The latter is a way to combine global and local momentum. The momentum is reset at the end of the round ignoring the changes the clients make to it. Fig. 1 (right) shows that this *worsens* the performance, confirming that it is better to keep the global optimizer state fixed as predicted by our theory.

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.

5.3 Large scale comparison with equal server and client communication

We perform a larger scale study closely matching the setup of [49]. For both Mime and MimeLite, only half the clients compute and transmit the updated parameters, and other half transmit the full local-batch gradients. Hence, client to server communication cost is the same for all methods for all clients. However, Mime and MimeLite require sending additional optimization state to the clients. Hence, we also reduce the number of clients sampled in each round to ensure *sum total* of communication at each round is $40\times$ model size for EMNIST and Shakespeare experiments, and $100\times$ model size for the StackOverflow next word prediction experiment.

Since we only perform 1 local epoch, the hyper-parameters (e.g. epsilon for adaptive methods) are more carefully chosen following [49], and Mime and MimeLite use significantly fewer clients per round, the difference between FedAvg and Mime is smaller here. Table 2 summarizes the results.

For the image classification tasks of EMNIST62 logistic and EMNIST62 CNN, Mime and MimeLite with Adam achieve the best performance. Using momentum (both with SGD and in Adam) significantly improves their performance. In contrast, FedAvgAdam is more unstable with worse performance. This is because FedAvg is excessively sensitive to hyperparameters (cf. App. E).

We next consider the character prediction task on Shakespeare dataset, and next word prediction on StackOverflow. Here, the momentum based methods (SGD+momentum and Adam) are slower than their non-momentum counterparts (vanilla SGD and AdaGrad). This is because the mini-batch gradients in these tasks are *sparse*, with the gradients corresponding to tokens not in the mini-batch being zero. This sparsity structure is however destroyed when using momentum or Adam. For the same reason, Mime which uses an SVRG correction also significantly increases the gradient density.

Discussion. For traditional tasks such as image classification, we observe that Mime (especially with Adam) usually outperforms MimeLite which in turn outperforms FedAvg. These methods are able to successfully leverage momentum and adaptivity to improve performance. For tasks where the client gradients are sparse, the SVRG correction used by Mime hinders performance. Adapting our techniques to work with sparse gradients (à la Yogi [75]) could lead to further improvements. Also, note that we reduce communication by naïvely reducing the number of participating clients per round. More sophisticated approaches to save on client communication including quantization or sparsification [58, 3], or even novel algorithmic innovations [1] could be explored. Further, server communication could be reduced using memory efficient optimizers e.g. AdaFactor [55] or SM3 [4].

6 Conclusion

Our work initiated a formal study of the cross-device federated learning problem and provided theoretically justified algorithms. We introduced a new framework MIME which overcomes the natural client-heterogeneity in such a setting, and can adapt arbitrary centralized algorithms such as Adam without additional hyper-parameters. We demonstrated the superiority of MIME via strong convergence guarantees and empirical evaluations. Further, we proved that a particular instance of our method, MimeMVR, beat centralized lower-bounds, demonstrating that additional local steps can yield asymptotic improvements for the first time. We believe our analysis will be of independent interest beyond the federated setting for understanding the sample complexity of non-convex optimization, and for yielding improved analysis of decentralized optimization algorithms.

References

[1] Durmus Alp Emre Acar, Yue Zhao, Ramon Matas Navarro, Matthew Mattina, Paul N Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. In *International Conference on Learning Representations*, 2021.

[2] 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.

[3] Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. Qsgd: Communication-efficient sgd via gradient quantization and encoding. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[4] Rohan Anil, Vineet Gupta, Tomer Koren, and Yoram Singer. Memory-efficient adaptive optimization. *arXiv preprint arXiv:1901.11150*, 2019.

[5] 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.

[6] Yossi Arjevani and Ohad Shamir. Communication complexity of distributed convex learning and optimization. In *Advances in neural information processing systems*, pages 1756–1764, 2015.

[7] 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.

[8] 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.

[9] Sebastian Caldas, Jakub Konečny, H Brendan McMahan, and Ameet Talwalkar. Expanding the reach of federated learning by reducing client resource requirements. *arXiv preprint arXiv:1812.07210*, 2018.

[10] 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.

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

[12] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. Emnist: Extending mnist to handwritten letters. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 2921–2926. IEEE, 2017.

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

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

[15] 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.

[16] Stack Exchange. Stack exchange data dump. https://archive.org/details/stackexchange, 2021.

[17] Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. SPIDER: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. In *Advances in Neural Information Processing Systems*, pages 689–699, 2018.

[18] Dylan J Foster, Ayush Sekhari, Ohad Shamir, Nathan Srebro, Karthik Sridharan, and Blake Woodworth. The complexity of making the gradient small in stochastic convex optimization. In *Conference on Learning Theory*, pages 1319–1345. PMLR, 2019.

[19] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *International Conference on Learning Representations (ICLR)*, 2019.

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

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

[22] Jenny Hamer, Mehryar Mohri, and Ananda Theertha Suresh. FedBoost: Communication-efficient algorithms for federated learning. In *37th International Conference on Machine Learning (ICML)*, 2020.

[23] 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.

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

[25] Arne Holst. Smartphone users worldwide 2016-2021. *Statista* https://web.archive.org/web/20210608080335/https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/, 2019.

[26] 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.

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

[28] 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.

[29] 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.

[30] Sai Praneeth Karimireddy, Lie He, and Martin Jaggi. Learning from history for byzantine robust optimization. In *38th International Conference on Machine Learning (ICML)*, 2021.

[31] Sai Praneeth Karimireddy, Martin Jaggi, Satyen Kale, Mehryar Mohri, Sashank J Reddi, Sebastian U Stich, and Ananda Theertha Suresh. Mime: Mimicking centralized stochastic algorithms in federated learning. *arXiv preprint arXiv:2008.03606*, 2020.

[32] 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.

[33] 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.

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

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

[36] 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.

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

[38] 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*, 2016.

[39] Lihua Lei and Michael Jordan. Less than a single pass: Stochastically controlled stochastic gradient. In *AISTATS*, pages 148–156, 2017.

[40] 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.

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

[42] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. *arXiv preprint arXiv:1908.03265*, 2019.

[43] 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.

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

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

[46] Lam M. Nguyen, Jie Liu, Katya Scheinberg, and Martin Takáč. Stochastic recursive gradient algorithm for nonconvex optimization. *arXiv preprint arXiv:1705.07261*, 2017.

[47] 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.

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

[49] 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.

[50] Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. *International Conference on Learning Representations (ICLR)*, 2018.

[51] Sashank J. Reddi, Jakub Konečnỳ, Peter Richtárik, Barnabás Póczós, and Alex Smola. Aide: Fast and communication efficient distributed optimization. *arXiv preprint arXiv:1608.06879*, 2016.

[52] Jae Hun Ro, Ananda Theertha Suresh, and Ke Wu. FedJAX: Federated learning simulation with JAX, 2020.

[53] Jae Hun Ro, Ananda Theertha Suresh, and Ke Wu. Fedjax: Federated learning simulation with jax. *arXiv preprint arXiv:2108.02117*, 2021.

[54] Ohad Shamir, Nati Srebro, and Tong Zhang. Communication-efficient distributed optimization using an approximate newton-type method. In *International conference on machine learning*, pages 1000–1008, 2014.

[55] Noam Shazeer and Mitchell Stern. Adafactor: Adaptive learning rates with sublinear memory cost. In *International Conference on Machine Learning*, pages 4596–4604. PMLR, 2018.

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

[57] 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.

[58] Ananda Theertha Suresh, Felix X. Yu, Sanjiv Kumar, and H. Brendan McMahan. Distributed mean estimation with limited communication. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3329–3337. JMLR. org, 2017.

[59] 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.

[60] TFF. Tensorflow federated datasets. `https://www.tensorflow.org/federated/api_docs/python/tff/simulation/datasets`, 2020.

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

[62] 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.

[63] 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.

[64] Thijs Vogels, Sai Praneeth Karimireddy, and Martin Jaggi. Powersgd: Practical low-rank gradient compression for distributed optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[65] 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*, 2020.

[66] 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*, 2020.

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

[68] 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.

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

[70] 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)*, 2020.

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

[72] 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.

[73] Hao Yu, Rong Jin, and Sen Yang. On the linear speedup analysis of communication efficient momentum sgd for distributed non-convex optimization. *arXiv preprint arXiv:1905.03817*, 2019.

[74] 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.

[75] 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.

[76] 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.

[77] 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.

[78] 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.

Supplementary material for MIME

Contents of Appendix

A How momentum can help reduce client drift

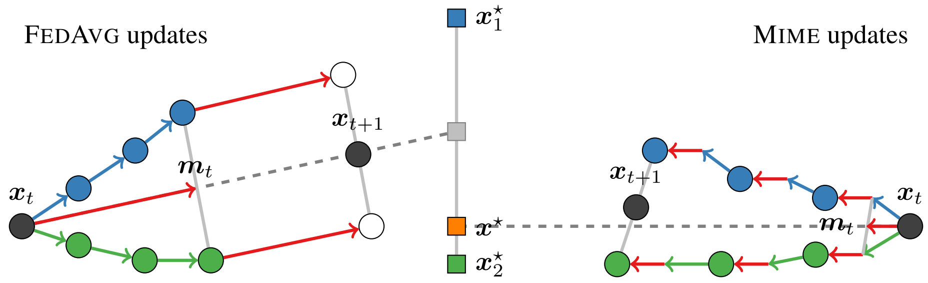

Figure 2: Client-drift in FEDAVG (left) and MIME (right) is illustrated for 2 clients with 3 local steps and momentum parameter $\beta = 0.5$. The local SGD updates of FEDAVG (shown using arrows for client 1 and client2) move towards the average of client optima $\frac{\boldsymbol{x}_1^\star + \boldsymbol{x}_2^\star}{2}$ which can be quite different from the true global optimum $\boldsymbol{x}^\star$. Server momentum $\boldsymbol{m}_t$ only speeds up the convergence to the wrong point in this case. In contrast, MIME uses unbiased momentum computed at the server parameter x_t and applies it locally at every update. This keeps the updates of MIME closer to the true optimum $\boldsymbol{x}^\star$.

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 [32]. 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 $i \sim \mathcal{C}$ and run SGD with momentum (Mom) 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. The FedAvg meta algorithm utilizes a base optimizer, a client learning rate and a server learning rate. Each client performs K local update steps of SGD using the client learning rate and communicates the net update (difference between final and initial parameters) to the server. This difference is then treated as a 'pseudo-gradient' and is input into the optimizer (say momentum or Adam) to update the server parameters using the server learning rate. When the base optimizer uses momentum, this momentum is computed at the server level using the pseudo-gradients and is referred to as server momentum.

Starting from $\boldsymbol{y}_0 = \boldsymbol{x}_{t-1}$, FEDAVG [43] runs multiple SGD steps on the sampled client $i \sim \mathcal{C}$

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

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 [26]. However, such updates give rise to *client-drift* resulting in performance worse than the naïve 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 effect is that FEDAVG moves towards an incorrect point (see Fig 2, left). If K is sufficiently large, approximately

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

Further, the server momentum is based on $\tilde{\boldsymbol{g}}_t$ 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 optimizer state as in (2) using the sampled client $i \sim \mathcal{C}$:

$$\boldsymbol{m}_t = (1-\beta)\nabla f_i(\boldsymbol{x}_{t-1}) + \beta \boldsymbol{m}_{t-1}\,. \tag{4}$$

To reduce the bias in the local updates, we will apply this unbiased momentum every step $k \in [K]$:

$$\boldsymbol{y}_k = \boldsymbol{y}_{k-1} - \eta((1-\beta)\nabla f_i(\boldsymbol{y}_{k-1}) + \beta \boldsymbol{m}_{t-1})\,. \tag{5}$$

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 $\boldsymbol{m}_{t-1}$ 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 $\boldsymbol{m}_{t-1}$, 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 2, right). Appendix B makes this intuition precise.

B Proof sketch

In this section, we provide an intuition behind our proof of convergence of MimeMVR. There are three main components: 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. Intuitively, using momentum locally at every client update reduces client drift by incorporating information about other clients from past rounds. 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. Of course, this also introduces additional bias into the update. To reduce this bias requires slightly modifying the momentum algorithm similar to [14]. The full analysis is carried out in Appendix H.

Improving the optimization term via local steps. The optimization (second) term in Theorem IV is $\frac{\delta K + L}{\epsilon K}$. In contrast, the optimization term of the server-only methods is L/ϵ. 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 [32] 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. Note that this improved analysis is potentially applicable to any local update methods and is not specific to Mime.

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. In practical deep learning, this SVRG correction may not very effective [15] and so can be dropped, though it is useful to derive the optimal theoretical rates.

C Experimental setup

C.1 Description of ablation study

We train a 2 hidden layer MLP with 300u-100 neurons on the EMNIST62 (extended MNIST) dataset [12]. The clients' data is separated according to the original authors of the characters [10]. All methods are augmented with momentum–Mime and MimeLite use momentum in the client updates, and the others use server momentum. The momentum parameter is searched over $\beta \in [0, 0.9, 0.99]$. For Adam, we fix $\beta_1 = 0.9$, $\beta_2 = 0.99$, and $\epsilon = 10^{-3}$. For both FedProx and SCAFFOLD, $\beta = 0$ (no server momentum) yielded the best performance. For FedAvg, Mime, and MimeLite $\beta = 0.9$ was the fastest. For FedProx, the regularization parameter μ was searched over $[0.1, 0.5, 1]$ and $\mu = 0.1$ had highest test accuracy.

C.2 Description of large scale experiments

We perform 4 tasks over 3 datasets: i) On the EMNIST62 dataset [12] we run a convex multi-class (62 classes) logistic regression model, and ii) a convolution model with two CNN layers and two dense layers and dropout. iii) On the SHAKESPEARE dataset, we train a single layer LSTM model with state size of 256 and embedding size of 8 to predict the next character [43]. iv) Finally, on the STACKOVERFLOW dataset [16], we train a next word prediction language model with embedding size of 96, a LSTM layer of size 670, and a vocabulary size of 1000. In all cases we report the top-1 test accuracy in our experiments.

All datasets use the metadata indicating the original authors to separate them into multiple clients yielding naturally partitioned datasets. Table 3 summarizes the statistics about the different datasets. Note that the average number of rounds a client participates in (computed as sampled clients×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 3: Details about the datasets used and experiment setting.

	EMNIST62	SHAKESPEARE	STACKOVERFLOW
Clients	3,400	715	342,477
Examples	671,585	16,068	135,818,730
Batch size	10	10	10
Number of local epochs	1	1	1
Total number of rounds	1000	1000	1000
Avg. rounds each client participates	5.9	28	0.15

We use Tensorflow federated datasets [60] to generate the datasets. Our federated learning simulation code is written in FedJAX [52, 53] and is open-sourced at `github.cm/google/fedjax` (see documentation). Black and white was reversed in EMNIST62 (i.e. subtracted from 1) to make them similar to MNIST. The preprocessing for SHAKESPEARE and STACKOVERFLOW datasets exactly matches that of [49].

Table 4: Effective number of sampled clients.

	Total Comm.	EMNIST62	SHAKESPEARE	STACKOVERFLOW
FedAvg	2×	20	20	50
MimeLiteMom	5×	8	8	20
MimeLiteAdagrad	5×	8	8	20
MimeLiteAdam	6×	6	6	16
MimeMom	6×	6	6	16
MimeAdagrad	6×	6	6	16
MimeAdam	7×	5	5	14

C.3 Practicality of experiments

In the experiments we only cared about the number of communication rounds, ignoring that MIME actually needs twice the number of bits per round and that the SERVER-ONLY methods have a much smaller computational requirement. This is standard in the federated learning setting as introduced by [43] and is justified because most of the time in cross-device FL is spent in establishing connections with devices rather than performing useful work such as communication or computation. In other words, *latency* and not bandwidth or computation are critical in cross device FL. However, one can certainly envision cases where this is not true. Incorporating communication compression strategies [58, 3, 33, 64] or client-model compression strategies [9, 19, 22] into our MIME framework can potentially address such issues and are important future research directions.

Regarding the algorithms evaluated, we chose not to include MVR as a base optimizer. This is because it is not a popular choice is practice even in the centralized setting, and serves more as a theoretical stand in to explain the benefit of the simpler SGD with momentum algorithm. Hence, we wouldn't expect MimeMVR to perform better than MimeMom. In general, our goal was to "mimic" centralized methods – methods which have better empirical performance (momentum and Adam) we showed also perform well in the federated setting when combined with Mime, and similarly methods which have better theoretical rates (MVR) have good rates with Mime as well.

Further, as we noted previously, we believe both the datasets and the tasks being studied here are close to real world settings since they contain natural heterogeneity. 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 3 mins in the real world and is relatively independent of the size of communication [7] implying that training 1000 rounds takes **2 days** even for small models. In contrast, running a centralized simulation takes about 15 mins. 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 centralized dataset on the server (as in [23]). Thus, it is crucial for federated frameworks to be able to translate algorithms which work well in centralized settings directly to the federated setting without additinal hyper-parameter tuning. 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 20–50 workers per round due to computational constraints, in real world FL thousands of devices are often available for training simultaneously each round [7]. 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.

C.4 Hyperparameter search

We run two hyper-parameter sweeps in our experiments: first a *light* setup which is reported in the main paper, and one we believe reflects the real world performance, and second a *heavy* tuning setting to showcase the performance of the methods as we vary the hyper-parameters.

Light-sweep setting (9×). For all Momentum methods, we pick momentum $\beta = 0.9$. For Adam methods, we fix $\beta_1 = 0.9$ and $\beta_2 = 0.99$, and $\varepsilon_0 = 1 \times 10^{-7}$. For Adagrad we use the default initialization value of 0.1 and use $\varepsilon_0 = 1 \times 10^{-7}$. 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 (9×):

$$\eta \in [1, 1 \times 10^{-0.5}, 1 \times 10^{-1}, 1 \times 10^{-1.5}, 1 \times 10^{-2}, 1 \times 10^{-2.5}, 1 \times 10^{-3}, 1 \times 10^{-3.5}, 1 \times 10^{-4}]\,.$$

The server learning rate for all methods is kept at its default value of 1.

Heavy-sweep setting (567×). For all Momentum methods, we pick momentum $\beta = 0.9$. For Adam methods, we fix $\beta_1 = 0.9$ and $\beta_2 = 0.99$. For Adagrad we use the default initialization value of 0.1. 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 **client** learning rate is searched over a grid (9×):

$$\eta_{\text{client}} \in [1, 1 \times 10^{-0.5}, 1 \times 10^{-1}, 1 \times 10^{-1.5}, 1 \times 10^{-2}, 1 \times 10^{-2.5}, 1 \times 10^{-3}, 1 \times 10^{-3.5}, 1 \times 10^{-4}]\,.$$

Further, we also search for the **server** learning rate is searched over a grid (9×):

$$\eta_{\text{server}} \in [1 \times 10^{1}, 1 \times 10^{0.5}, 1, 1 \times 10^{-0.5}, 1 \times 10^{-1}, 1 \times 10^{-1.5}, 1 \times 10^{-2}, 1 \times 10^{-2.5}, 1 \times 10^{-3}]\,.$$

Finally, for the **adaptive** methods such as Adam and Adagrad, we also tune the ε_0 parameter over a grid (7×):

$$\varepsilon_0 \in [1, 1 \times 10^{-1}, 1 \times 10^{-2}, 1 \times 10^{-3}, 1 \times 10^{-4}, 1 \times 10^{-5}, 1 \times 10^{-6}, 1 \times 10^{-7}]\,.$$

C.5 Comparison with previous results

As far as we are aware, [49] 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. FedAvgSGD, FedAvgMom, and FedAvgAdam) have qualitatively similar performance except with one exception: FedAvgAdam consistently underperforms FedAvgMom. This difference, as we show later, is because FedAvgAdam does not work with the default choices of hyper-parameters such as ϵ and requires additional tuning. As we explain in Section C.3, we chose to keep these parameters to the default values of their centralized counterparts to compare methods in a 'low-tuning' setting. We also point that while FedAvgAdam struggles to perform in this setup, MimeAdam and MimeLiteAdam are very stable and even often outperform their SGD counterparts.

C.6 Additional algorithmic details

Table 5: 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/Mom	$\boldsymbol{m}$	$\boldsymbol{x} - \eta((1-\beta)\boldsymbol{g} + \beta\boldsymbol{m})$	$\boldsymbol{m} = (1-\beta)\boldsymbol{g} + \beta\boldsymbol{m}$
AdaGrad	$\boldsymbol{v}$	$\boldsymbol{x} - \frac{\eta}{\epsilon + \sqrt{\boldsymbol{v}}}\boldsymbol{g}$	$\boldsymbol{v} = \boldsymbol{g}^2 + \boldsymbol{v}$
Adam	$\boldsymbol{m}, \boldsymbol{v}$	$\boldsymbol{x} - \frac{\eta}{\epsilon + \sqrt{\boldsymbol{v}}}((1-\beta_1)\boldsymbol{g} + \beta_1\boldsymbol{m})$	$\boldsymbol{m} = (1-\beta_1)\boldsymbol{g} + \beta_1\boldsymbol{m}$ $\boldsymbol{v} = (1-\beta_2)\boldsymbol{g}^2 + \beta_2\boldsymbol{v}$

D Additional Adam experiments

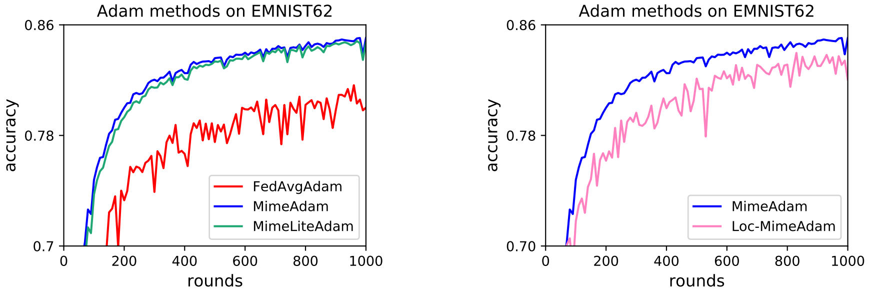

Figure 3: **Mime**, **MimeLite**, **FedAvg**, and **Loc-Mime** with Adam using 10 local epochs, run on EMNIST62 and a 2 hidden layer (300u-100) MLP. (Left) Mime and MimeLite are nearly identical and outperform FedAvg. (Right) Locally adapting Adam state slows down convergence and makes it more unstable. Both these results are consistent with the earlier momentum results.

E Stability of methods to hyper-parameters

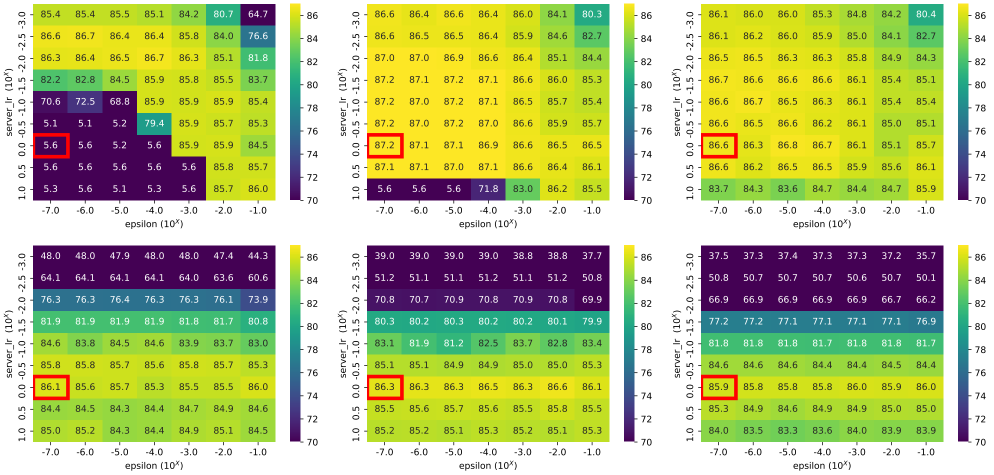

Figure 4: Stability of adaptive methods with varying server learning: FedAvg (left), Mime (middle) and MimeLite (right) with Adam (top) and Adagrad (bottom) as base algorithms are run on EMNIST62 with CNN. For each value of server learning rate (y-axis) and ε_0 (x-axis), the client learning rate was tuned over the $9\times$ grid and the accuracy reported. The red box highlights the default configuration in a centralized setting. We see that FedAvgAdam is very sensitive to the server learning rate and ε_0, performing poorly in the default centralized parameter regimes. Mime and MimeLite acheive their best performance with the centralized parameters. This justifies our claim that Mime and MimeLite can **adapt** any centralized method with the same hyper-parameters and only require tuning of a single learning rate. This, we believe, is crucial for real world deployment.

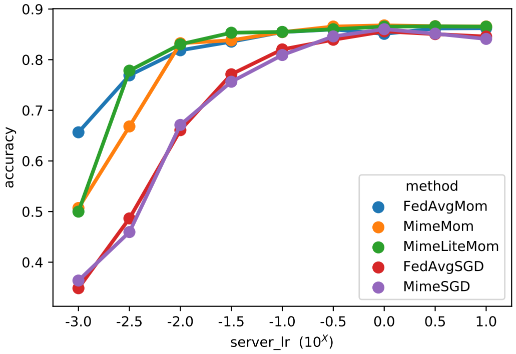

Figure 5: Stability of non-adaptive methods with varying server learning: FedAvg, Mime and MimeLite with SGD and momentum ($\beta = 0.9$) as base algorithms are run on EMNIST62 with CNN. For each value of server learning rate, the client learning rate was tuned over the $9\times$ grid. The momentum methods are more insensitive to the server learning rate than the SGD methods. Server learning rate of 1 (default value) seems to work well for all methods.

F Technicalities

We examine some additional definitions and introduce some technical lemmas.

F.1 Assumptions and definitions

We make precise a few definitions and explain some of their implications. We first discuss the two assumptions on the dissimilarity between the gradients (A1) and the Hessians (A2). Loosely, these two quantities are an extension of the concepts of **variance** and **smoothness** which occur in centralized SGD analysis to the federated learning setting. Just as the variance and smoothness are completely orthogonal concepts, we can have settings where G^2 (gradient dissimilarity) is large while δ (Hessian dissimilarity) is small, or vice-versa.

Our assumption about the bound on the G gradient dissimilarity can easily be extended to (G, B) gradient dissimilarity used by [33]:

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

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 [63]. 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).

The second crucial definition we use in this work is that of δ bounded *Hessian* dissimilarity (A2). This has been used previously in the analyses of distributed [54, 6, 51] and federated learning [32], but has been restricted to quadratics. Here, we show how to extend both the notion as well as the analysis to general smooth functions. The main manner we will use this assumption is in Lemma 3 to claim that for any $\boldsymbol{x}$ and $\boldsymbol{y}$ the following holds:

$$\mathbb{E}\|\nabla f_i(\boldsymbol{y};\zeta) - \nabla f_i(\boldsymbol{x};\zeta) + \nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\|^2 \leq \delta^2 \|\boldsymbol{y} - \boldsymbol{x}\|^2 \,. \tag{8}$$

Here the expectation is over the choice of client i. To understand what the above condition means, it is illuminating to define $\Psi_i(\boldsymbol{z}) = f_i(\boldsymbol{z};\zeta) - f(\boldsymbol{z})$. Then, we can rewrite (A2) and (8) respectively as

$$\|\nabla^2 \Psi_i(\boldsymbol{z})\| \leq \delta \quad \text{and} \quad \mathbb{E}\|\nabla \Psi_i(\boldsymbol{y}) - \nabla \Psi_i(\boldsymbol{x})\|^2 \leq \delta^2 \|\boldsymbol{y} - \boldsymbol{x}\|^2 \,.$$

Thus (8) and (A2) are both different notions of smoothness of $\Psi_i(\boldsymbol{x})$ (formal definition of smoothness will follow soon). The latter definition closely matches the notion of *squared-smoothness* used by [5] and is a promising relaxation of (A2). However, we run into some technical issues since in our case the variable $\boldsymbol{y}$ can also be a random variable and depend on the choice of the client i. Extending our results to this weaker notion of Hessian-similarity and proving tight non-convex lower bounds is an exciting theoretical challenge.

Finally note that if the functions $f_i(\boldsymbol{x};\zeta)$ are assumed to be smooth as in [54, 6, 32], then $\Psi_i((\boldsymbol{x})$ is $2L$-smooth. Thus, we *always* have that $\delta \leq 2L$. But, as shown in [54], it is possible to have $\delta \ll L$ if the data distribution amongst the clients is similar. Further, the lower bound from [6] proves that Hessian-similarity is the crucial quantity capturing the number of rounds of communication required for distributed/federated optimization.

We next define the terms smoothness and strong-convexity which we repeatedly use in the paper.

(A2*) f_i is almost surely **L-smooth** and satisfies:

$$\|\nabla f_i(\boldsymbol{x};\zeta) - \nabla f_i(\boldsymbol{y};\zeta)\| \leq L\|\boldsymbol{x}-\boldsymbol{y}\|\,, \text{ for any } \boldsymbol{x},\boldsymbol{y}\,. \tag{9}$$

The assumption (A2*) also implies the following quadratic upper bound on f_i

$$f_i(\boldsymbol{y}) \leq f_i(\boldsymbol{x}) + \langle \nabla f_i(\boldsymbol{x}), \boldsymbol{y}-\boldsymbol{x}\rangle + \frac{L}{2}\|\boldsymbol{y}-\boldsymbol{x}\|^2\,. \tag{10}$$

Further, if f_i is twice-differentiable, (A2*) implies that $\|\nabla^2 f_i(\boldsymbol{x};\zeta)\| \leq \beta$ for any $\boldsymbol{x}$.

(A3) We assume that the **intra-client gradient variance** is bounded by σ^2. For any client i, the following holds almost surely at any fixed $\boldsymbol{x}$:

$$\mathbb{E}_{\zeta_i}[\nabla f_i(\boldsymbol{x};\zeta)] = \nabla f_i(\boldsymbol{x})\,, \quad \text{and} \quad \mathbb{E}_{\zeta_i}\|\nabla f_i(\boldsymbol{x};\zeta) - \nabla f_i(\boldsymbol{x})\|^2 \leq \sigma^2\,.$$

Note that we expect the intra-client variance to be smaller than inter-client variance and so typically $\sigma^2 \leq G^2$.

(A4) f satisfies the **μ-PL inequality** [29] for $\mu > 0$ if:

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

Note that PL-inequality is much weaker than the standard notion of strong-convexity, and in fact is even satisfied by some non-convex functions [29].

F.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 $\{\boldsymbol{v}_1,\dots,\boldsymbol{v}_\tau\}$ 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

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

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

Lemma 2 (separating mean and variance). *Let $\{\Xi_1,\dots,\Xi_\tau\}$ 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. □

F.3 Properties of functions with bounded Hessian dissimilarity

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

Lemma 3 (similarity)**.** *The following holds for any two functions* $f_i(\cdot)$ *and* $f(\cdot)$ *satisfying* (A2) *and* (A3)*, 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 [45], 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". This is used to get a handle on the effect of averaging of parameters in FedAvg.

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* $\boldsymbol{x}$*:*

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

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 $\frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}\Phi(\boldsymbol{y}_i)\geq\Phi(\bar{\boldsymbol{y}})$ by convexity. □

G Convergence with a generic base optimizer

Let us rewrite the Mime and MimeLite updates 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 $\boldsymbol{c}^t$ defined as

$$\boldsymbol{c}^t = \frac{1}{S}\sum_{i \in \mathcal{S}^t} \nabla f_i(\boldsymbol{x}^{t-1})\,. \tag{11}$$

Note that computing $\boldsymbol{c}^t$ (required only by Mime but not by MimeLite) itself requires additional communication. In this proof, we do not make any assumption on how $\boldsymbol{c}^t$ is computed as long as it is unbiased and is computed over S clients. In particular, it can either be computed on the sampled $\mathcal{S}^t$ or a different set of an independent sampled clients $\tilde{\mathcal{S}}^t$.

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

$$\boldsymbol{y}^t_{i,k} = \boldsymbol{y}^t_{i,k-1} - \eta\mathcal{U}(\nabla f_i(\boldsymbol{y}^t_{i,k-1}; \zeta^t_{i,k}) - \nabla f_i(\boldsymbol{x}^{t-1}; \zeta^t_{i,k}) + \boldsymbol{c}^t; \boldsymbol{s}^{t-1}) \qquad \text{(Mime client update)}$$

$$= \boldsymbol{y}^t_{i,k-1} - \eta\mathcal{U}(\nabla f_i(\boldsymbol{y}^t_{i,k-1}; \zeta^t_{i,k}); \boldsymbol{s}^{t-1})\,. \qquad \text{(MimeLite client update)}$$

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

$$\boldsymbol{x}^t = \frac{1}{S}\sum_{i \in \mathcal{S}^t} \boldsymbol{y}^t_{i,K} \qquad \text{(Update server parameters)}$$

$$\boldsymbol{s}^t = \mathcal{V}(\boldsymbol{c}^t, \boldsymbol{s}^{t-1})\,. \qquad \text{(Update server statistics)}$$

G.1 Proof of Theorem I (generic reduction)

Computing server update.

Lemma 5 (Deviation from central update.)**.** *For a linear updater $\mathcal{U}$ the server update for Mime can be written as*

$$\boldsymbol{x}^t = \boldsymbol{x}^{t-1} - \tilde{\eta}\mathcal{U}\left(\frac{1}{S}\sum_i \nabla f_i(\boldsymbol{x}) + \boldsymbol{e}^t; \boldsymbol{s}^{t-1}\right),$$

and for MimeLite is becomes

$$\boldsymbol{x}^t = \boldsymbol{x}^{t-1} - \tilde{\eta}\mathcal{U}\left(\frac{1}{KS}\sum_{i,k} \nabla f_i(\boldsymbol{x}; \zeta_{i,k}) + \boldsymbol{e}^t; \boldsymbol{s}^{t-1}\right),$$

for $\tilde{\eta} := K\eta$. The error is defined as $\boldsymbol{e}^t = \frac{1}{KS}\sum_{i,k}(\nabla f_i(\boldsymbol{y}_{i,k-1}; \zeta_{i,k}) - \nabla f_i(\boldsymbol{x}; \zeta_{i,k}))$

Proof. Because the updater $\mathcal{U}$ is linear in its first parameter, we can rewrite the update to the server for MimeLite as

$$\boldsymbol{x}^t - \boldsymbol{x}^{t-1} = \frac{1}{S}\sum_{i \in \mathcal{S}^t}\sum_{k=1}^{K} -\eta\mathcal{U}(\nabla f_i(\boldsymbol{y}^t_{i,k-1}; \zeta^t_{i,k}); \boldsymbol{s}^{t-1})$$

$$= \eta K\mathcal{U}\left(\frac{1}{KS}\sum_{i,k} \nabla f_i(\boldsymbol{y}^t_{i,k-1}; \zeta^t_{i,k}); \boldsymbol{s}^{t-1}\right)$$

We drop the dependence on t when obvious from context and i by default sums over $\mathcal{S}^t$ and k over $[K]$ by default. Using our definition of $\boldsymbol{e}^t$ we have

$$\boldsymbol{x}^t - \boldsymbol{x}^{t-1} = \eta K\mathcal{U}\left(\frac{1}{KS}\sum_{i,k} \nabla f_i(\boldsymbol{y}^t_{i,k-1}; \zeta^t_{i,k}); \boldsymbol{s}^{t-1}\right)$$

$$= \tilde{\eta}\mathcal{U}\left(\frac{1}{KS}\sum_{i,k} \nabla f_i(\boldsymbol{x}; \zeta_{i,k}) + \boldsymbol{e}^t; \boldsymbol{s}^{t-1}\right).$$

Now let us examine the update of Mime. Again assuming K is a multiple of epoch, we have $\sum_{i,k} \nabla f_i(\boldsymbol{x}; \zeta^t_{i,k}) = K \sum_i \nabla f_i(\boldsymbol{x}) = KS\boldsymbol{x}$. Hence,

$$\begin{aligned}
\boldsymbol{x}^t - \boldsymbol{x}^{t-1} &= \frac{1}{S}\sum_{i\in\mathcal{S}^t}\sum_{k=1}^{K} -\eta\mathcal{U}(\nabla f_i(\boldsymbol{y}_{i,k-1}; \zeta^t_{i,k}) - \nabla f_i(\boldsymbol{x}; \zeta^t_{i,k}) + \boldsymbol{c}; \boldsymbol{s}^{t-1})\\
&= \eta K\mathcal{U}\big(\boldsymbol{c} + \boldsymbol{e}^t; \boldsymbol{s}^{t-1}\big)\\
&= \eta K\mathcal{U}\left(\frac{1}{S}\sum_i \nabla f_i(\boldsymbol{x}) + \boldsymbol{e}^t; \boldsymbol{s}^{t-1}\right).
\end{aligned}$$

Thus we showed the lemma for both Mime and MimeLite. □

Lemma 6 (Defining error)**.** *For $\boldsymbol{e}^t$ defined in Lemma 5, assuming all functions $f_i(\,\cdot\,, \zeta)$ are L-smooth, we have*

$$\mathbb{E}\|\boldsymbol{e}^t\|^2 \le L^2\mathcal{E}^t_K\,, \quad \textit{where} \;\; \mathcal{E}^t_K := \frac{1}{KS}\sum_{i,k}\mathbb{E}\|\boldsymbol{y}_{i,k-1} - \boldsymbol{x}\|^2\,.$$

Proof. Using the smoothness of the individual functions and the definition of $\boldsymbol{e}^t$,

$$\begin{aligned}
\mathbb{E}\|\boldsymbol{e}^t\|^2 &= \mathbb{E}\|\frac{1}{KS}\sum_{i,k}(\nabla f_i(\boldsymbol{y}_{i,k-1}; \zeta_{i,k}) - \nabla f_i(\boldsymbol{x}; \zeta_{i,k}))\|^2\\
&\le \frac{1}{KS}\sum_{i,k}\mathbb{E}\|\nabla f_i(\boldsymbol{y}_{i,k-1}) - \nabla f_i(\boldsymbol{x}; \zeta_{i,k})\|^2 \le L^2\mathcal{E}^t_K\,.
\end{aligned}$$

□

Henceforth, we will call $\mathcal{E}^t_K$ as the error, or as the client-drift following [32].

Bounding error in MimeLite. Now we will try bound the client drift $\mathcal{E}^t$ for MimeLite.

Lemma 7 (MimeLite error)**.** *Suppose that all functions $f_i(\,\cdot\,, \zeta)$ are L-smooth (A2*), σ^2 variance (A3), and (A1) is satisfied, and the updater $\mathcal{U}$ has B-Lipschitz updates. Then using step-size $\tilde{\eta} \le \frac{1}{2BL}$,*

$$\frac{1}{18B^2\tilde{\eta}^2}\mathcal{E}^K \le \mathbb{E}\|\nabla f(\boldsymbol{x})\|^2 + G^2 + \frac{\sigma^2}{2K}\,.$$

Proof. For $K = 1$, we have $\mathbb{E}\|\boldsymbol{y}_{i,1} - \boldsymbol{x}\|^2 \le B^2\eta^2(G^2 + \sigma^2) + B^2\eta^2\,\mathbb{E}\|\nabla f(\boldsymbol{x})\|^2$. The lemma is easily shown to be true. Assuming $K \ge 2$ henceforth, and starting from the client update of MimeLite we have

$$\begin{aligned}
\mathbb{E}\|\boldsymbol{y}_{i,k} - \boldsymbol{x}\|^2 &= \mathbb{E}\|\boldsymbol{y}_{i,k-1} - \eta\mathcal{U}(\nabla f_i(\boldsymbol{y}^t_{i,k-1}; \zeta^t_{i,k}); \boldsymbol{s}^{t-1}) - \boldsymbol{x}\|^2\\
&\le \mathbb{E}\|\boldsymbol{y}_{i,k-1} - \eta\mathcal{U}(\nabla f_i(\boldsymbol{y}^t_{i,k-1}; \boldsymbol{s}^{t-1}) - \boldsymbol{x}\|^2 + B^2\eta^2\sigma^2\\
&\le \left(1 + \frac{1}{K-1}\right)\mathbb{E}\|\boldsymbol{y}_{i,k-1} - \boldsymbol{x}\|^2 + K\eta^2\,\mathbb{E}\|\mathcal{U}(\nabla f_i(\boldsymbol{y}^t_{i,k-1}; \boldsymbol{s}^{t-1})\|^2 + B^2\eta^2\sigma^2\\
&\le \left(1 + \frac{1}{K-1}\right)\mathbb{E}\|\boldsymbol{y}_{i,k-1} - \boldsymbol{x}\|^2 + KB^2\eta^2\,\mathbb{E}\|\nabla f_i(\boldsymbol{y}_{i,k-1}) \pm \nabla f_i(\boldsymbol{x})\|^2 + B^2\eta^2\sigma^2\\
&\le \left(1 + \frac{1}{K-1}\right)\mathbb{E}\|\boldsymbol{y}_{i,k-1} - \boldsymbol{x}\|^2\\
&\qquad\qquad + 2KB^2\eta^2\,\mathbb{E}\|\nabla f_i(\boldsymbol{x})\|^2 + 2KB^2L^2\eta^2\,\mathbb{E}\|\boldsymbol{y}_{i,k-1} - \boldsymbol{x}\|^2 + B^2\eta^2\sigma^2\\
&\le \left(1 + \frac{2}{K-1}\right)\mathbb{E}\|\boldsymbol{y}_{i,k-1} - \boldsymbol{x}\|^2 + 2KB^2\eta^2\,\mathbb{E}\|\nabla f(\boldsymbol{x})\|^2 + 2KB^2\eta^2G^2 + B^2\eta^2\sigma^2\,.
\end{aligned}$$

Here, we used the condition on our step size that $\tilde{\eta} = K\eta \leq \frac{1}{2LB}$, which implies that $2KB^2L^2\eta^2 \leq \frac{1}{K-1}$. Unrolling this recursion, we have

$$\mathbb{E}\|\boldsymbol{y}_{i,k} - \boldsymbol{x}\|^2 \leq \left(2KB^2\eta^2\,\mathbb{E}\|\nabla f(\boldsymbol{x})\|^2 + 2KB^2\eta^2G^2 + B^2\eta^2\sigma^2\right)\sum_{k=1}^{K}\left(1 + \frac{2}{K-1}\right)^k .$$

Note that $\left(1 + \frac{2}{K-1}\right)^k \leq 9$. Averaging then over k and i, we get

$$\mathcal{E}_K^t \leq 18K^2B^2\eta^2\,\mathbb{E}\|\nabla f(\boldsymbol{x})\|^2 + 18K^2B^2\eta^2G^2 + 9KB^2\eta^2\sigma^2 .$$

Finally, recalling that $\tilde{\eta} = K\eta$ finishes the lemma. □

Bounding error in Mime. Next we will try bound the client drift $\mathcal{E}^t$ for Mime. The additional SVRG correction term used in Mime improves the bound on the error.

Lemma 8 (Mime Error)**.** *Suppose that all functions $f_i(\,\cdot\,,\zeta)$ are L-smooth (A2*), σ^2 variance (A3), and (A1) is satisfied, and the updater $\mathcal{U}$ has B-Lipschitz updates. Then using step-size $\tilde{\eta} \leq \frac{1}{2BL}$,*

$$\mathcal{E}^K \leq 18B^2\tilde{\eta}^2\,\mathbb{E}\left\|\frac{1}{S}\sum_i \nabla_i f(\boldsymbol{x})\right\|^2 .$$

Proof. For $K = 1$, the Mime update loos like

$$\begin{aligned}\mathbb{E}\|\boldsymbol{y}_{i,1} - \boldsymbol{x}\|^2 &= \eta^2\,\mathbb{E}\|\mathcal{U}(\boldsymbol{c}; \boldsymbol{s}^{t-1})\|^2 \\ &\leq \eta^2B^2\,\mathbb{E}\|\boldsymbol{c}\|^2 .\end{aligned}$$

Assuming $K \geq 2$ henceforth, and starting from the client update of Mime we have

$$\begin{aligned}\mathbb{E}\|\boldsymbol{y}_{i,k} - \boldsymbol{x}\|^2 &= \mathbb{E}\|\boldsymbol{y}_{i,k-1} - \eta\mathcal{U}(\nabla f_i(\boldsymbol{y}_{i,k-1}; \zeta_{i,k}^t) - \nabla f_i(\boldsymbol{x}; \zeta_{i,k}^t) + \boldsymbol{c}^t; \boldsymbol{s}^{t-1}) - \boldsymbol{x}\|^2 \\ &\leq \left(1 + \frac{1}{K-1}\right)\mathbb{E}\|\boldsymbol{y}_{i,k-1} - \boldsymbol{x}\|^2 \\ &\qquad + K\eta^2\,\mathbb{E}\|\mathcal{U}(\nabla f_i(\boldsymbol{y}_{i,k-1}; \zeta_{i,k}^t) - \nabla f_i(\boldsymbol{x}; \zeta_{i,k}^t) + \boldsymbol{c}^t; \boldsymbol{s}^{t-1})\|^2 \\ &\leq \left(1 + \frac{1}{K-1}\right)\mathbb{E}\|\boldsymbol{y}_{i,k-1} - \boldsymbol{x}\|^2 + K\eta^2B^2\,\mathbb{E}\|\nabla f_i(\boldsymbol{y}_{i,k-1}; \zeta_{i,k}^t) - \nabla f_i(\boldsymbol{x}; \zeta_{i,k}^t) + \boldsymbol{c}^t\|^2 \\ &\leq \left(1 + \frac{1}{K-1}\right)\mathbb{E}\|\boldsymbol{y}_{i,k-1} - \boldsymbol{x}\|^2 \\ &\qquad + 2K\eta^2B^2\,\mathbb{E}\|\nabla f_i(\boldsymbol{y}_{i,k-1}; \zeta_{i,k}^t) - \nabla f_i(\boldsymbol{x}; \zeta_{i,k}^t)\|^2 + 2K\eta^2B^2\,\mathbb{E}\|\boldsymbol{c}^t\|^2 \\ &\leq \left(1 + \frac{1}{K-1} + 2K\eta^2B^2L^2\right)\mathbb{E}\|\boldsymbol{y}_{i,k-1} - \boldsymbol{x}\|^2 + 2K\eta^2B^2\,\mathbb{E}\|\boldsymbol{c}^t\|^2 \\ &\leq \left(1 + \frac{2}{K-1}\right)\mathbb{E}\|\boldsymbol{y}_{i,k-1} - \boldsymbol{x}\|^2 + 2K\eta^2B^2\,\mathbb{E}\|\boldsymbol{c}^t\|^2 .\end{aligned}$$

Here, we used the condition on our step size that $\tilde{\eta} = K\eta \leq \frac{1}{2LB}$, which implies that $2KB^2L^2\eta^2 \leq \frac{1}{K-1}$. Unrolling this recursion, we have

$$\mathbb{E}\|\boldsymbol{y}_{i,k} - \boldsymbol{x}\|^2 \leq 2KB^2\eta^2\,\mathbb{E}\|\boldsymbol{c}^t\|^2\sum_{k=1}^{K}\left(1 + \frac{2}{K-1}\right)^k \leq 18K^2B^2\eta^2\,\mathbb{E}\|\boldsymbol{c}^t\|^2 .$$

Note that $\left(1 + \frac{2}{K-1}\right)^k \leq 9$. Averaging then over k and i, recalling that $\tilde{\eta} = K\eta$ get

$$\mathcal{E}_K^t \leq 18B^2\tilde{\eta}^2\,\mathbb{E}\|\boldsymbol{c}^t\|^2 .$$

□

Putting it together (Theorem I).

Lemma 9. *The updates of Mime and MimeLite for $\boldsymbol{g}^t$ satisfying $\mathbb{E}[\boldsymbol{g}^t] = \nabla f(\boldsymbol{x}^{t-1})$, and we have for $\tilde\eta \le \frac{1}{2BL}$*

$$\begin{aligned} \boldsymbol{x}^t &= \boldsymbol{x}^{t-1} - \tilde\eta\,\mathcal{U}(\boldsymbol{c}^t + \boldsymbol{e}^t; \boldsymbol{s}^{t-1}) \\ \boldsymbol{s}^t &= \mathcal{V}(\boldsymbol{c}^t; \boldsymbol{s}^{t-1})\,. \end{aligned}$$

Where, we have

$$\tfrac{1}{18B^2L^2\tilde\eta^2}\,\mathbb{E}_t\|\,\boldsymbol{e}_t\,\|^2 \le \begin{cases} \mathbb{E}\|\boldsymbol{c}_t\|^2 & \textsc{Mime}\,, \\ \mathbb{E}\|\nabla f(\boldsymbol{x}^t)\|^2 + G^2 + \frac{\sigma^2}{2K} & \textsc{MimeLite}\,. \end{cases}$$

Proof. Now, combining Lemmas 5, 6, shows that running Mime or MimeLite is equivalent to

$$\begin{aligned} \boldsymbol{x}^t &= \boldsymbol{x}^{t-1} - \tilde\eta\,\mathcal{U}(\boldsymbol{g}^t + \boldsymbol{e}^t; \boldsymbol{s}^{t-1}) \\ \boldsymbol{s}^t &= \mathcal{V}(\boldsymbol{g}^t; \boldsymbol{s}^{t-1})\,, \end{aligned}$$

where for Mime we use

$$\boldsymbol{g}^t_{\text{Mime}} = \frac{1}{S}\sum_i \nabla f_i(\boldsymbol{x}) \text{ with } \mathbb{E}[\boldsymbol{g}^t_{\text{Mime}}] = \nabla f(\boldsymbol{x}^{t-1}) \text{ and } \mathbb{E}\|\boldsymbol{g}^t_{\text{Mime}} - \nabla f(\boldsymbol{x}^{t-1})\|^2 \le \frac{G^2}{S}\,.$$

and for MimeLite we use

$$\boldsymbol{g}^t_{\text{MimeLite}} = \frac{1}{KS}\sum_{i,k} \nabla f_i(\boldsymbol{x}; \zeta_{i,k}) \text{ with } \mathbb{E}[\boldsymbol{g}^t_{\text{Mime}}] = \nabla f(\boldsymbol{x}^{t-1}) \text{ and } \mathbb{E}\|\boldsymbol{g}^t_{\text{Mime}} - \nabla f(\boldsymbol{x}^{t-1})\|^2 \le \frac{G^2}{S} + \frac{\sigma^2}{KS}\,.$$

This shows the first part of the theorem. For the second part of the theorem, using the bound from Lemma 8 for Mime,

$$\mathbb{E}\|\boldsymbol{e}^t\| \le L^2\mathcal{E}^t_K \le 18L^2B^2\tilde\eta^2\,\mathbb{E}\|\boldsymbol{c}^t\|^2\,.$$

For MimeLite, we will instead use the bound from Lemma 7,

$$\mathbb{E}\|\boldsymbol{e}^t_{\text{MimeLite}}\| \le L^2\mathcal{E}^t_K + \frac{\sigma^2}{KS} \le 18L^2B^2\tilde\eta^2\,\mathbb{E}\|\nabla f(\boldsymbol{x}^t)\|^2 + 18L^2B^2\tilde\eta^2G^2 + \frac{9L^2B^2\tilde\eta^2\sigma^2}{K} + \frac{\sigma^2}{KS}\,.$$

□

Note that the Lemma we proved here is slightly stronger than the theorem in the main section (up to constants which were suppressed).

G.2 Convergence of MimeSGD and MimeLiteSGD (Corollary II)

Theorem I shows that Mime and MimeLite mimic a centralized algorithm quite closely up to error $\mathcal{O}(\tilde\eta^2)$. Then, analyzing the sensitivity of the base algorithm to such perturbation yields specific rates of convergence. We perform such an analysis using SGD as our base optimizer.

Properties of SGD as the base optimizer:

- $\boldsymbol{s}^t$ is empty i.e. there are no global statistics used.
- $\mathcal{U}(\boldsymbol{g}; \boldsymbol{s}^{t-1}) = \boldsymbol{g}$ for any $\boldsymbol{g}$ and $B = 1$.

With this in mind, we proceed.

Lemma 10 (Progress in one round)**.** *Given that f is L-smooth, and for any step-size $\tilde\eta \le \frac{1}{2(B+2)L}$ for $B \ge 1$ we have*

$$f(\boldsymbol{x}^t) \le f(\boldsymbol{x}^{t-1}) - \frac{\tilde\eta}{4}\,\mathbb{E}\|\nabla f(\boldsymbol{x}^{t-1})\|^2 + \tilde\eta\,\mathbb{E}\|\boldsymbol{e}^t\|^2 + \frac{L\tilde\eta^2G^2}{S}\,.$$

Proof. Starting from the update equation and the smoothness of f, we have

$$\begin{aligned}
\mathbb{E}\, f(\boldsymbol{x}^t) &\le \mathbb{E}\, f(\boldsymbol{x}^{t-1}) + \mathbb{E}\langle \nabla f(\boldsymbol{x}^{t-1}), \boldsymbol{x}^t - \boldsymbol{x}^{t-1}\rangle + \frac{L}{2}\,\mathbb{E}\|\boldsymbol{x}^t - \boldsymbol{x}^{t-1}\|^2 \\
&= \mathbb{E}\, f(\boldsymbol{x}^{t-1}) - \tilde\eta\,\mathbb{E}\|\nabla f(\boldsymbol{x}^{t-1})\|^2 + \tilde\eta\langle \nabla f(\boldsymbol{x}^{t-1}), \boldsymbol{e}^t\rangle + \frac{L\tilde\eta^2}{2}\,\mathbb{E}\|\boldsymbol{c}^t + \boldsymbol{e}^t\|^2 \\
&\le \mathbb{E}\, f(\boldsymbol{x}^{t-1}) - \frac{\tilde\eta}{2}\,\mathbb{E}\|\nabla f(\boldsymbol{x}^{t-1})\|^2 + \frac{\tilde\eta}{2}\|\boldsymbol{e}^t\|^2 + \frac{2L\tilde\eta^2}{2}\,\mathbb{E}\|\boldsymbol{c}^t\|^2 + \frac{2L\tilde\eta^2}{2}\,\mathbb{E}\|\boldsymbol{e}^t\|^2 \\
&\le \mathbb{E}\, f(\boldsymbol{x}^{t-1}) - \left(\frac{\tilde\eta}{2} - \frac{2L\tilde\eta^2}{2}\right)\mathbb{E}\|\nabla f(\boldsymbol{x}^{t-1})\|^2 + \left(L\tilde\eta^2 + \frac{\tilde\eta}{2}\right)\mathbb{E}\|\boldsymbol{e}^t\|^2 + \frac{2L\tilde\eta^2 G^2}{2S}\,.
\end{aligned}$$

Using the bound on the step size that $\tilde\eta \le \frac{1}{4L}$ yields the lemma. □

One round progress for MimeSGD. Next, we specialize the convergence rate for Mime.

Lemma 11. *Suppose f is a L-smooth function satisfying PL-inequality for $\mu \ge 0$ ($\mu = 0$ corresponds to the general case). Running MimeSGD for $\tilde\eta \le \frac{1}{12BL}$ satisfies*

$$\frac{\tilde\eta}{16}\,\mathbb{E}\|\nabla f(\boldsymbol{x}^{t-1})\|^2 \le (1 - \tfrac{\mu\tilde\eta}{8})(f(\boldsymbol{x}^{t-1}) - f^\star) - (f(\boldsymbol{x}^t) - f^\star) + \frac{3L\tilde\eta^2 G^2}{S}\,.$$

Proof. Recall from Lemma 9 that for Mime,

$$\mathbb{E}\|\boldsymbol{e}^t\|^2 \le 18L^2B^2\tilde\eta^2\,\mathbb{E}\|\boldsymbol{c}^t\|^2 \le 18L^2B^2\tilde\eta^2\,\mathbb{E}\|\nabla f(\boldsymbol{x}^{t-1})\|^2 + \frac{18L^2B^2\tilde\eta^2G^2}{S}\,.$$

Combining this with Lemma 10 yields the following progress for Mime

$$\begin{aligned}
f(\boldsymbol{x}^t) &\le f(\boldsymbol{x}^{t-1}) - \left(\frac{\tilde\eta}{4} - 18L^2B^2\tilde\eta^3\right)\mathbb{E}\|\nabla f(\boldsymbol{x}^{t-1})\|^2 + \frac{(L\tilde\eta^2 + 18L^2B^2\tilde\eta^3)G^2}{S} \\
&\le f(\boldsymbol{x}^{t-1}) - \frac{\tilde\eta}{8}\,\mathbb{E}\|\nabla f(\boldsymbol{x}^{t-1})\|^2 + \frac{3L\tilde\eta^2G^2}{S}\,.
\end{aligned}$$

Here, we used the bound on the step size that $\tilde\eta \le \frac{1}{12LB}$ implies $18L^2B^2\tilde\eta^2 \le \frac{1}{8}$. Now using PL-inequality, we can write

$$f(\boldsymbol{x}^t) - f^\star \le f(\boldsymbol{x}^{t-1}) - f^\star - \frac{\mu\tilde\eta}{8}(f(\boldsymbol{x}^{t-1}) - f^\star) - \frac{\tilde\eta}{16}\,\mathbb{E}\|\nabla f(\boldsymbol{x}^{t-1})\|^2 + \frac{3L\tilde\eta^2G^2}{S}\,.$$

This yields the lemma. □

We are now ready to derive the convergence rate.

Convergence rate of MimeSGD on general non-convex functions. Set $\mu = 0$ in Lemma 11 and sum over t

$$\begin{aligned}
\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla f(\boldsymbol{x}^{t-1})\|^2 &\le \frac{16(f(\boldsymbol{x}^0) - f^\star)}{\tilde\eta T} + \frac{48L\tilde\eta G^2}{S} \\
&\le 16\sqrt{\frac{3LG^2(f(\boldsymbol{x}^0) - f^\star)}{ST}} + \frac{192BL(f(\boldsymbol{x}^0) - f^\star)}{T}\,.
\end{aligned}$$

The final step used a step-size of $\tilde\eta = \min\left(\frac{1}{12BL}, \frac{1}{4L}, \sqrt{\frac{S(f(\boldsymbol{x}^0) - f^\star)}{3LTG^2}}\right)$. Here, we used $\boldsymbol{x}^{\text{out}} = \boldsymbol{x}^\tau$ where τ is uniformly at random chosen in $[T]$.

Convergence rate of MimeSGD on PL-inequality. Multiply Lemma 11 by $(1-\frac{\mu\tilde\eta}{8})^{T-t}$ and sum over t

$$\begin{aligned}\sum_{t=1}^{T}(1-\tfrac{\mu\tilde\eta}{8})^{T-t}\,\mathbb{E}\|\nabla f(\boldsymbol{x}^{t-1})\|^2 &\leq \sum_{t=1}^{T}(1-\tfrac{\mu\tilde\eta}{8})^{T-(t-1)}\frac{16(f(\boldsymbol{x}^{t-1})-f^\star)}{\tilde\eta}\\ &\qquad -(1-\tfrac{\mu\tilde\eta}{8})^{T-t}\frac{16(f(\boldsymbol{x}^{t})-f^\star)}{\tilde\eta}+(1-\tfrac{\mu\tilde\eta}{8})^{T-t}\frac{48L\tilde\eta G^2}{S}\\ &\leq (1-\tfrac{\mu\tilde\eta}{8})^{T}\frac{16(f(\boldsymbol{x}^{0})-f^\star)}{\tilde\eta}+\sum_{t=1}^{T}(1-\tfrac{\mu\tilde\eta}{8})^{T-t}\frac{48L\tilde\eta G^2}{S}\,.\end{aligned}$$

Output $\boldsymbol{x}^{\text{out}}=\boldsymbol{x}^{\tau}$ where τ is chosen with probability proportional to $(1-\frac{\mu\tilde\eta}{8})^{T-t}$. Then, this yields

$$\mathbb{E}\|\nabla f(\boldsymbol{x}^{\text{out}})\|^2\leq(1-\tfrac{\mu\tilde\eta}{8})^{T}\frac{16(f(\boldsymbol{x}^{0})-f^\star)}{\tilde\eta}+\frac{48L\tilde\eta G^2}{S}\leq\tilde{\mathcal{O}}\Big(\frac{\sigma^2}{\mu T}+L(f(\boldsymbol{x}^{0})-f^\star)\exp\Big(-\frac{\mu T}{12BL}\Big)\Big)\,.$$

Using an appropriate step-size $\tilde\eta$ yields the final rate (see Lemma 1 of [32]).

One round progress for MimeLiteSGD. Next, we specialize the convergence rate for MimeLite.

Lemma 12. *Suppose f is a L-smooth function satisfying PL-inequality for $\mu\geq 0$ ($\mu=0$ corresponds to the general case). Running MimeLiteSGD for $\tilde\eta\leq\frac{1}{12BL}$ satisfies*

$$\frac{\tilde\eta}{16}\,\mathbb{E}\|\nabla f(\boldsymbol{x}^{t-1})\|^2\leq(1-\tfrac{\mu\tilde\eta}{8})(f(\boldsymbol{x}^{t-1})-f^\star)-(f(\boldsymbol{x}^{t})-f^\star)+\frac{L\tilde\eta^2G^2}{S}+18L^2B^2\tilde\eta^3\big(G^2+\sigma^2/K\big)\,.$$

Proof. Recall from Lemma 9 that,

$$\mathbb{E}\|\boldsymbol{e}^t\|^2\leq 18L^2B^2\tilde\eta^2\,\mathbb{E}\|\boldsymbol{c}^t\|^2\leq 18L^2B^2\tilde\eta^2\,\mathbb{E}\|\nabla f(\boldsymbol{x}^{t-1})\|^2+18L^2B^2\tilde\eta^2G^2+\frac{9L^2B^2\tilde\eta^2\sigma^2}{K}\,.$$

Combining this with Lemma 10 yields the following progress for Mime

$$\begin{aligned}f(\boldsymbol{x}^t)&\leq f(\boldsymbol{x}^{t-1})-\Big(\frac{\tilde\eta}{4}-18L^2B^2\tilde\eta^3\Big)\,\mathbb{E}\|\nabla f(\boldsymbol{x}^{t-1})\|^2+\frac{L\tilde\eta^2G^2}{S}+18L^2B^2\tilde\eta^3\big(G^2+\sigma^2/K\big)\\ &\leq f(\boldsymbol{x}^{t-1})-\frac{\tilde\eta}{8}\,\mathbb{E}\|\nabla f(\boldsymbol{x}^{t-1})\|^2+\frac{L\tilde\eta^2G^2}{S}+18L^2B^2\tilde\eta^3\big(G^2+\sigma^2/K\big)\,.\end{aligned}$$

Here, we used the bound on the step size that $\tilde\eta\leq\frac{1}{12LB}$ implies $18L^2B^2\tilde\eta^2\leq\frac{1}{8}$. Now using PL-inequality, we can write

$$\begin{aligned}&f(\boldsymbol{x}^t)-f^\star-(f(\boldsymbol{x}^{t-1})-f^\star)\leq\\ &\qquad-\frac{\mu\tilde\eta}{8}(f(\boldsymbol{x}^{t-1})-f^\star)-\frac{\tilde\eta}{16}\,\mathbb{E}\|\nabla f(\boldsymbol{x}^{t-1})\|^2+\frac{L\tilde\eta^2G^2}{S}+18L^2B^2\tilde\eta^3\big(G^2+\sigma^2/K\big)\,.\end{aligned}$$

This yields the lemma. □

We are now ready to derive the convergence rate.

Convergence rate of MimeLiteSGD on general non-convex functions. Define $\tilde G^2=G^2+\sigma^2/K$. Set $\mu=0$ in Lemma 12 and sum over t

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla f(\boldsymbol{x}^{t-1})\|^2&\leq\frac{16(f(\boldsymbol{x}^0)-f^\star)}{\tilde\eta T}+\frac{16L\tilde\eta G^2}{S}+288L^2B^2\tilde\eta^2\tilde G^2\\ &\leq 16\sqrt{\frac{LG^2(f(\boldsymbol{x}^0)-f^\star)}{ST}}+84\bigg(\frac{L\tilde G(f(\boldsymbol{x}^0)-f^\star)}{T}\bigg)^{2/3}+\frac{192BL(f(\boldsymbol{x}^0)-f^\star)}{T}\,.\end{aligned}$$

The final step used an appropriate step-size of $\tilde\eta$, see Lemma 2 of [32]. Here, we used $\boldsymbol{x}^{\text{out}}=\boldsymbol{x}^{\tau}$ where τ is uniformly at random chosen in $[T]$. Finally note that if $K\geq\frac{\sigma^2}{G^2}$, then $\tilde G^2\leq 2G^2$.

Convergence rate of MimeLiteSGD on PL-inequality. Multiply Lemma 12 by $(1-\frac{\mu\tilde\eta}{8})^{T-t}$ and sum over t

$$
\begin{aligned}
\sum_{t=1}^{T}(1-\tfrac{\mu\tilde\eta}{8})^{T-t}\,\mathbb{E}\|\nabla f(\boldsymbol{x}^{t-1})\|^2 \leq \sum_{t=1}^{T}(1-\tfrac{\mu\tilde\eta}{8})^{T-(t-1)}&\frac{16(f(\boldsymbol{x}^{t-1})-f^\star)}{\tilde\eta}\\
&-(1-\tfrac{\mu\tilde\eta}{8})^{T-t}\frac{16(f(\boldsymbol{x}^{t})-f^\star)}{\tilde\eta}\\
&+\sum_{t=1}^{T}(1-\tfrac{\mu\tilde\eta}{8})^{T-t}\Big(\frac{16L\tilde\eta G^2}{S}+288L^2B^2\tilde\eta^2\tilde G^2\Big)\\
\leq (1-\tfrac{\mu\tilde\eta}{8})^{T}&\frac{16(f(\boldsymbol{x}^{0})-f^\star)}{\tilde\eta}\\
&+\sum_{t=1}^{T}(1-\tfrac{\mu\tilde\eta}{8})^{T-t}\Big(\frac{16L\tilde\eta G^2}{S}+288L^2B^2\tilde\eta^2\tilde G^2\Big)\,.
\end{aligned}
$$

Output $\boldsymbol{x}^{\text{out}}=\boldsymbol{x}^{\tau}$ where τ is chosen with probability proportional to $(1-\frac{\mu\tilde\eta}{8})^{T-t}$. Then, this yields with appropriate step-size $\tilde\eta$ yields the final rate (see Lemma 1 of [32]).

$$
\mathbb{E}\|\nabla f(\boldsymbol{x}^{\text{out}})\|^2 \leq \tilde{\mathcal{O}}\Bigg(\frac{\sigma^2}{\mu T}+\frac{L^2\tilde G^2}{\mu^2T^2}+L(f(\boldsymbol{x}^0)-f^\star)\exp\Big(-\frac{\mu T}{12BL}\Big)\Bigg)\,.
$$

G.3 Convergence of MimeAdam and MimeLiteAdam (Corollary III)

We will largely follow the convergence analysis of [75] for the analysis of Adam. A crucial difference between their setting and ours is that in our algorithm we use the global statistics (second order moment) corresponding to $t-1$ i.e. $\sqrt{\boldsymbol{v}^{t-1}}$ instead of $\sqrt{\boldsymbol{v}^{t}}$ where the $\sqrt{\cdot}$ operator is applied element wise. Practically, this does not make a significant difference since the discount (momentum) factor for the second momentum is very large. Theoretically however, this difference simplifies our proof significantly removing otherwise hard to handle stochastic dependencies.

In this section, we will use Adam as our base optimizer with $\varepsilon_0>0$ parameter for stability and $\beta_1=0$ (i.e. RMSProp). This is identical to the setting in the centralized algorithm analyzed by [75]. The properties of our base optimizer are then:

- $\boldsymbol{s}^t=\boldsymbol{v}^t$ which is a running average estimate of the second moment and satisfies $\boldsymbol{v}^t>0$.
- $\mathcal{U}(\boldsymbol{g};\boldsymbol{v}^{t-1})=\frac{\boldsymbol{g}}{\sqrt{\boldsymbol{v}^{t-1}}+\varepsilon_0}$ for any $\boldsymbol{g}$. This update for any $\boldsymbol{v}^{t-1}$ is B-Lipschitz for $B=\frac{1}{\varepsilon_0}$.

In this sub-section, all operations on vectors (multiplication, division, addition, comparison) are applied element-wise with appropriate broad-casting.

One round progress of Adam.

Lemma 13 (Effective step-sizes)**.** *Suppose that $|\nabla_j f_i(\boldsymbol{x})|\leq H$. Then Adam has effective step-sizes*

$$
\frac{1}{H+\varepsilon_0}\boldsymbol{g}\leq\mathcal{U}(\boldsymbol{g};\boldsymbol{v}^{t-1})\leq\frac{1}{\varepsilon_0}\boldsymbol{g}\,.
$$

Proof. Recall that $\boldsymbol{v}^t=\beta_2\boldsymbol{v}^{t-1}+(1-\beta_2)(\boldsymbol{c}^t)^2$ starting from $\boldsymbol{v}^0=0$. Thus for any $t\geq 0$, we have $\boldsymbol{v}^t\geq 0$ and hence $\sqrt{\boldsymbol{v}^{t-1}}+\varepsilon_0\geq\varepsilon_0$. For the other side, recall that $\boldsymbol{v}^t$ is updated with centralized stochastic gradients $\boldsymbol{c}^t=\frac{1}{S}\sum_i\nabla f_i(\boldsymbol{x})$.

$$
[\boldsymbol{c}^t]_j=\frac{1}{S}\sum_i[\nabla f_i(\boldsymbol{x})]_j\leq H\,.
$$

Further,

$$
[\boldsymbol{v}^t]_j=\beta_2[\boldsymbol{v}^{t-1}]_j+(1-\beta_2)[\boldsymbol{c}^t]_j^2\leq\beta_2[\boldsymbol{v}^{t-1}]_j+(1-\beta_2)H^2\leq H^2\,.
$$

Hence $\sqrt{\boldsymbol{v}^{t-1}+\varepsilon_0}\leq H+\varepsilon_0$. □

Lemma 14 (One round progress)**.** *For one round of Adam with error $\boldsymbol{e}^t$ in the update $\mathcal{U}$ and using $\boldsymbol{c}^t$ for update $\mathcal{V}$, we have*

$$\mathbb{E}\, f(\boldsymbol{x}^t) \leq \mathbb{E}\, f(\boldsymbol{x}^{t-1}) - \frac{\tilde\eta}{4(H+\varepsilon_0)}\|\nabla f(\boldsymbol{x}^{t-1})\|^2 + \frac{\tilde\eta((H+\varepsilon_0)+\varepsilon_0/(H+\varepsilon_0))}{2\varepsilon_0^2}\,\mathbb{E}\|\boldsymbol{e}^t\|^2 + \frac{L\tilde\eta^2 G^2}{S\varepsilon_0^2}\,.$$

Proof. Starting from Lemma 13 and the smoothness of f, we have

$$\begin{aligned}
\mathbb{E}\, f(\boldsymbol{x}^t) &\leq \mathbb{E}\, f(\boldsymbol{x}^{t-1}) - \tilde\eta\,\mathbb{E}\langle \nabla f(\boldsymbol{x}^{t-1}), \mathbb{E}_t[\mathcal{U}(\boldsymbol{c}^t + \boldsymbol{e}^t)]\rangle + \frac{L\tilde\eta^2}{2}\,\mathbb{E}\|\mathcal{U}(\boldsymbol{c}^t + \boldsymbol{e}^t; \boldsymbol{v}^{t-1})\|^2 \\
&\leq \mathbb{E}\, f(\boldsymbol{x}^{t-1}) - \tilde\eta\,\mathbb{E}\langle \nabla f(\boldsymbol{x}^{t-1}), \mathbb{E}_t\left[\frac{\boldsymbol{c}^t + \boldsymbol{e}^t}{\sqrt{\boldsymbol{v}^{t-1}} + \varepsilon_0}\right]\rangle + \frac{L\tilde\eta^2}{2}\,\mathbb{E}\|\mathcal{U}(\boldsymbol{c}^t + \boldsymbol{e}^t; \boldsymbol{v}^{t-1})\|^2 \\
&\leq \mathbb{E}\, f(\boldsymbol{x}^{t-1}) - \tilde\eta\,\mathbb{E}\langle \nabla f(\boldsymbol{x}^{t-1}), \left[\frac{\nabla f(\boldsymbol{x}^{t-1}) + \boldsymbol{e}^t}{\sqrt{\boldsymbol{v}^{t-1}} + \varepsilon_0}\right]\rangle + \frac{L\tilde\eta^2}{2\varepsilon_0^2}\,\mathbb{E}\|\boldsymbol{c}^t + \boldsymbol{e}^t\|^2 \\
&\leq \mathbb{E}\, f(\boldsymbol{x}^{t-1}) - \frac{\tilde\eta}{H+\varepsilon_0}\|\nabla f(\boldsymbol{x}^{t-1})\|^2 - \tilde\eta\,\mathbb{E}\langle \nabla f(\boldsymbol{x}^{t-1}), \frac{\boldsymbol{e}^t}{\sqrt{\boldsymbol{v}^{t-1}} + \varepsilon_0}\rangle + \frac{L\tilde\eta^2}{2\varepsilon_0^2}\,\mathbb{E}\|\boldsymbol{c}^t + \boldsymbol{e}^t\|^2 \\
&\leq \mathbb{E}\, f(\boldsymbol{x}^{t-1}) - \frac{\tilde\eta}{2(H+\varepsilon_0)}\|\nabla f(\boldsymbol{x}^{t-1})\|^2 + \frac{\tilde\eta(H+\varepsilon_0)}{2}\,\mathbb{E}\|\frac{\boldsymbol{e}^t}{\sqrt{\boldsymbol{v}^{t-1}} + \varepsilon_0}\|^2 + \frac{L\tilde\eta^2}{2\varepsilon_0^2}\,\mathbb{E}\|\boldsymbol{c}^t + \boldsymbol{e}^t\|^2 \\
&\leq \mathbb{E}\, f(\boldsymbol{x}^{t-1}) - \left(\frac{\tilde\eta}{2(H+\varepsilon_0)} - \frac{L\tilde\eta^2}{\varepsilon_0^2}\right)\|\nabla f(\boldsymbol{x}^{t-1})\|^2 + \frac{\tilde\eta(H+\varepsilon_0) + 2L\tilde\eta^2}{2\varepsilon_0^2}\,\mathbb{E}\|\boldsymbol{e}^t\|^2 + \frac{L\tilde\eta^2 G^2}{S\varepsilon_0^2} \\
&\leq \mathbb{E}\, f(\boldsymbol{x}^{t-1}) - \frac{\tilde\eta}{4(H+\varepsilon_0)}\|\nabla f(\boldsymbol{x}^{t-1})\|^2 + \frac{\tilde\eta((H+\varepsilon_0)+\varepsilon_0/(H+\varepsilon_0))}{2\varepsilon_0^2}\,\mathbb{E}\|\boldsymbol{e}^t\|^2 + \frac{L\tilde\eta^2 G^2}{S\varepsilon_0^2}
\end{aligned}$$

Here we used our bound on the step-size that $\tilde\eta \leq \frac{\varepsilon_0}{4L(H+\varepsilon_0)}$. □

Convergence of MimeAdam.

Lemma 15. *Suppose that assumptions A1–(A3) hold and further $|\nabla_j f_i(\boldsymbol{x})| \leq H$. Then, running MimeAdam with step-size $\tilde\eta \leq \frac{\varepsilon_0^2}{12L(H+\varepsilon_0)}$, we have*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla f(\boldsymbol{x}^{t-1})\|^2 \leq \frac{96L(H+\varepsilon_0)^2(f(\boldsymbol{x}_0) - f^\star)}{\varepsilon_0^2 T} + \frac{2G^2}{S}\,.$$

Combining Lemma 14 with the bound on $\boldsymbol{e}^t$ from Lemma 9 we get,

$$\begin{aligned}
\mathbb{E}\, f(\boldsymbol{x}^t) &\leq \mathbb{E}\, f(\boldsymbol{x}^{t-1}) - \frac{\tilde\eta}{4(H+\varepsilon_0)}\|\nabla f(\boldsymbol{x}^{t-1})\|^2 + \frac{\tilde\eta((H+\varepsilon_0)+\varepsilon_0/(H+\varepsilon_0))}{2\varepsilon_0^2}\,\mathbb{E}\|\boldsymbol{e}^t\|^2 + \frac{L\tilde\eta^2 G^2}{S\varepsilon_0^2} \\
&\leq \mathbb{E}\, f(\boldsymbol{x}^{t-1}) - \frac{\tilde\eta}{4(H+\varepsilon_0)}\|\nabla f(\boldsymbol{x}^{t-1})\|^2 + \frac{9L^2\tilde\eta^3((H+\varepsilon_0)+\varepsilon_0/(H+\varepsilon_0))}{\varepsilon_0^4}\,\mathbb{E}\|\boldsymbol{c}^t\|^2 \\
&\qquad + \frac{L\tilde\eta^2 G^2}{S\varepsilon_0^2} \\
&\leq \mathbb{E}\, f(\boldsymbol{x}^{t-1}) - \left(\frac{\tilde\eta}{4(H+\varepsilon_0)} - \frac{9L^2\tilde\eta^3((H+\varepsilon_0)+\varepsilon_0/(H+\varepsilon_0))}{\varepsilon_0^4}\right)\|\nabla f(\boldsymbol{x}^{t-1})\|^2 \\
&\qquad + \frac{L\tilde\eta^2 G^2}{S\varepsilon_0^2} + \frac{9L^2\tilde\eta^3((H+\varepsilon_0)+\varepsilon_0/(H+\varepsilon_0))G^2}{S\varepsilon_0^4} \\
&\leq \mathbb{E}\, f(\boldsymbol{x}^{t-1}) - \left(\frac{\tilde\eta}{4(H+\varepsilon_0)} - \frac{18L^2\tilde\eta^3(H+\varepsilon_0)}{\varepsilon_0^4}\right)\|\nabla f(\boldsymbol{x}^{t-1})\|^2 \\
&\qquad + \frac{L\tilde\eta^2 G^2}{S\varepsilon_0^2} + \frac{18L^2\tilde\eta^3(H+\varepsilon_0)G^2}{S\varepsilon_0^4} \\
&\leq \mathbb{E}\, f(\boldsymbol{x}^{t-1}) - \frac{\tilde\eta}{8(H+\varepsilon_0)}\|\nabla f(\boldsymbol{x}^{t-1})\|^2 + \frac{\tilde\eta G^2}{4S(H+\varepsilon_0)}\,.
\end{aligned}$$

To simplify computations, here we assumed we assumed $(H+\varepsilon_0)^2 \geq \varepsilon_0$ without loss of generality. If this is not true, we can replace H by $\max(H, \sqrt{\varepsilon_0} - \varepsilon_0)$. Assuming $\tilde\eta \leq \frac{\varepsilon_0^2}{12L(H+\varepsilon_0)}$, we have $\frac{18L^2\tilde\eta^2(H+\varepsilon_0)}{\varepsilon_0^4} \leq \frac{1}{8(H+\varepsilon_0)}$. Rearranging the terms and substituting the bounds on the step-size yields the lemma.

Convergence of MimeLiteAdam.
Lemma 16. *Suppose that assumptions A1–(A3) hold and further $|\nabla_j f_i(\boldsymbol{x})| \leq H$. Then, running MimeLiteAdam with step-size $\tilde\eta \leq \frac{\varepsilon_0^2}{12L\sqrt{S}(H+\varepsilon_0)}$, we have for $\tilde G^2 := G^2 + \sigma^2/K$,*

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\|\nabla f(\boldsymbol{x}^{t-1})\|^2 \leq \frac{96L\sqrt{S}(H+\varepsilon_0)^2(f(\boldsymbol{x}_0)-f^\star)}{\varepsilon_0^2 T} + \frac{2\tilde G^2}{S}\,.$$

Combining Lemma 14 with the bound on $\boldsymbol{e}^t$ from Lemma 9 we get for $\tilde G^2 := G^2 + \sigma^2/K$,

$$\begin{aligned}
\mathbb{E}\, f(\boldsymbol{x}^t) &\leq \mathbb{E}\, f(\boldsymbol{x}^{t-1}) - \frac{\tilde\eta}{4(H+\varepsilon_0)}\|\nabla f(\boldsymbol{x}^{t-1})\|^2 + \frac{\tilde\eta(H+\varepsilon_0)}{\varepsilon_0^2}\,\mathbb{E}\|\boldsymbol{e}^t\|^2 + \frac{L\tilde\eta^2 G^2}{S\varepsilon_0^2} \\
&\leq \mathbb{E}\, f(\boldsymbol{x}^{t-1}) - \frac{\tilde\eta}{4(H+\varepsilon_0)}\|\nabla f(\boldsymbol{x}^{t-1})\|^2 \\
&\qquad + \frac{18L^2\tilde\eta^3(H+\varepsilon_0)}{\varepsilon_0^4}\,\mathbb{E}\|\nabla f(\boldsymbol{x}^{t-1})\|^2 + \frac{18L^2\tilde\eta^3(H+\varepsilon_0)(\tilde G^2)}{\varepsilon_0^4} + \frac{L\tilde\eta^2 G^2}{S\varepsilon_0^2} \\
&\leq \mathbb{E}\, f(\boldsymbol{x}^{t-1}) - \frac{\tilde\eta}{8(H+\varepsilon_0)}\|\nabla f(\boldsymbol{x}^{t-1})\|^2 + \frac{\tilde\eta\tilde G^2}{4S(H+\varepsilon_0)}
\end{aligned}$$

Again as before to simplify computations, here we assumed $(H+\varepsilon_0)^2 \geq \varepsilon_0$ without loss of generality. If this is not true, we can replace H by $\max(H, \sqrt{\varepsilon_0} - \varepsilon_0)$. Assuming $\tilde\eta \leq \frac{\varepsilon_0^2}{12L(H+\varepsilon_0)\sqrt{S}}$, we have $\frac{18L^2\tilde\eta^2(H+\varepsilon_0)}{\varepsilon_0^4} \leq \frac{1}{8S(H+\varepsilon_0)}$. Rearranging the terms and substituting the bounds on the step-size yields the lemma.

H Circumventing server-only lower bounds

In this section we see how to use momentum based variance reduction [14, 62] to reduce the variance of the updates and improve convergence. It should be noted that MVR does not exactly fit the MIME framework (BASEOPT) 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 [13], which does fit the framework of MIME.

For the sake of convenience, we summarize the notation used in the proof in a table.

Table 6: Summary of all notation used in the MVR proofs

σ^2, G^2, and δ	intra-client gradient, inter-client gradient, and inter-client Hessian variance
η, a	step-size, $(1-\beta)$ momentum parameters
T, t	total number, index of communication rounds
K, k	total number, index of client local update steps
$\mathcal{S}^t$, S, and i	sampled set, size, and index of clients in round t
$\boldsymbol{x}^t$	aggregated server model *after* round t
$\boldsymbol{m}^t$	server momentum computed *after* round t
$\boldsymbol{c}^t$	control variate of server *after* round t (only MIME)
$\boldsymbol{y}_{i,k}^t$	model parameters of ith client in round t *after* step k
$\zeta_{i,k}^t$	mini-batch data used by ith client in round t and step k
$\boldsymbol{d}_{i,k}^t$	parameter update by ith client in round t, step k
$\boldsymbol{e}^t$	error in momentum $\boldsymbol{m}^t - \nabla f(\boldsymbol{x}^{t-1})$
$\Delta_{i,k}^t$, Δ^{t-1}	$\mathbb{E}\|\boldsymbol{y}_{i,k}^t - \boldsymbol{x}^{t-2}\|^2$, $\mathbb{E}\|\boldsymbol{x}^{t-1} - \boldsymbol{x}^{t-2}\|^2 = \Delta_{i,0}^t$

H.1 Algorithm descriptions

Now, we formally describe the MIME MVR and MIMELITE MVR algorithms. 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 past parameters $\boldsymbol{x}^{t-2}$, and the momentum $\boldsymbol{m}^{t-1}$ term. MIME additionally uses a control variate $\boldsymbol{c}^{t-1}$ as we describe next.

Control variate in Mime. MIME uses an additional control variate $\boldsymbol{c}^{t-1}$ to reduce the variance.

$$\boldsymbol{c}^{t-1} = \frac{1}{S}\sum_{i\in\mathcal{S}^t} \nabla f_i(\boldsymbol{x}^{t-2})\,. \tag{12}$$

Note that both $\boldsymbol{c}^{t-1}$ and $\boldsymbol{m}^{t-1}$ use gradients and parameters from previous rounds (different from the previous section). A naive implementation of this method requires two steps of communication per round to implement this algorithm. Alternatively, we can reserve some clients in the previous round for computing $\boldsymbol{c}^{t-1}$ which can then be used in the current round, removing the need for two steps of communication. In particular, it can be computed on a different set of an independent sampled clients $\tilde{\mathcal{S}}^{t-1}$. In fact, all our theoretical results hold even if we use *a single client* to perform the local updates and the rest of clients are used only to compute $\boldsymbol{c}^{t-1}$ each round.

Local client updates. 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 $\zeta^t_{i,k}$. MIME performs the following update:

$$\begin{aligned}\boldsymbol{y}^t_{i,k} &= \boldsymbol{y}^t_{i,k-1} - \eta \boldsymbol{d}^t_{i,k}\,, \text{ where}\\ \boldsymbol{d}^t_{i,k} &= a(\nabla f_i(\boldsymbol{y}^t_{i,k-1};\zeta^t_{i,k}) - \nabla f_i(\boldsymbol{x}^{t-1};\zeta^t_{i,k}) + \boldsymbol{c}^{t-1}) + (1-a)\boldsymbol{m}^{t-1}\\ &\qquad + (1-a)(\nabla f_i(\boldsymbol{y}^t_{i,k-1};\zeta^t_{i,k}) - \nabla f_i(\boldsymbol{x}^{t-1};\zeta^t_{i,k}))\,.\end{aligned} \tag{13}$$

MIMELITE on the other hand uses a very similar but simpler update scheme which does not rely on $\boldsymbol{c}^{t-1}$:

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

Server updates. 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

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

As we can see, the momentum update of MVR can be broken down into the usual Mom 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 [14] for more detailed explanation of MVR.

H.2 Bias in updates

The main difference in MimeMVR from the centralized versions of [62, 14] is the additional local steps which are biased. In particular, for $k \geq 1$ the expected gradient $\mathbb{E}[\nabla f_i(\boldsymbol{y}^t_{i,k})] \neq \nabla f(\boldsymbol{y}^t_{i,k})$ because $\boldsymbol{y}^t_{i,k}$ also depends on the sample i. This bias is in fact the underlying cause of client drift and controlling it is a crucial step for our analysis.

Lemma 17 (Mime bias)**.** *For any values of $\boldsymbol{x}$ and $\boldsymbol{y}_i$ where $\boldsymbol{y}_i$ may depend on i, the following holds for any client i almost surely given that* (A1) *and* (A2) *hold:*

$$\mathbb{E}_{\mathcal{S},\zeta}\left\|\nabla f_i(\boldsymbol{y}_i;\zeta) + \frac{1}{|\mathcal{S}|}\sum_{j\in\mathcal{S}}\nabla f_j(\boldsymbol{x}) - \nabla f_i(\boldsymbol{x};\zeta) \quad - \quad \nabla f(\boldsymbol{y}_i)\right\|^2 \le 2\delta^2\,\mathbb{E}_{\mathcal{S}}\|\boldsymbol{y}_i - \boldsymbol{x}\|^2 + \frac{2G^2}{S}\,.$$

Proof. We can separate the noise from the rest of the terms and expand as

$$\begin{aligned}
&\mathbb{E}_{\zeta,\mathcal{S}}\left\|\nabla f_i(\boldsymbol{y}_i;\zeta) + \frac{1}{|\mathcal{S}|}\sum_{j\in\mathcal{S}}\nabla f_j(\boldsymbol{x}) - \nabla f_i(\boldsymbol{x};\zeta) - \nabla f(\boldsymbol{y}_i)\right\|^2 \\
&\quad\le 2\,\mathbb{E}_{\mathcal{S}}\|\nabla f_i(\boldsymbol{y}_i;\zeta) + \nabla f(\boldsymbol{x}) - \nabla f_i(\boldsymbol{x};\zeta) - \nabla f(\boldsymbol{y}_i)\|^2 + 2\,\mathbb{E}_{\mathcal{S}}\left\|\frac{1}{|\mathcal{S}|}\sum_{j\in\mathcal{S}}\nabla f_j(\boldsymbol{x}) - \nabla f(\boldsymbol{x})\right\|^2 \\
&\quad\le 2\,\mathbb{E}_{\mathcal{S}}\|\nabla f_i(\boldsymbol{y}_i;\zeta) + \nabla f(\boldsymbol{x}) - \nabla f_i(\boldsymbol{x};\zeta) - \nabla f(\boldsymbol{y}_i)\|^2 + \frac{2G^2}{S} \\
&\quad\le 2\,\mathbb{E}_{\mathcal{S}}\,\delta^2\|\boldsymbol{y}_i - \boldsymbol{x}\|^2 + \frac{2G^2}{S}\,.
\end{aligned}$$

The first inequality used Young's inequality, the second used (A1), and the last used (A2) in the form of Lemma 3. □

We can perform a similar analysis of the bias of local updates encountered by MIMELITE.

Lemma 18 (MimeLite bias)**.** *For any values of $\boldsymbol{x}$ and $\boldsymbol{y}_i$ where $\boldsymbol{y}_i$ may depend on i, the following holds for any client i randomly chosen from $\mathcal{C}$ given that* (A1), (A2) *and* (A3) *hold:*

$$\mathbb{E}_{i,\zeta}\|\nabla f_i(\boldsymbol{y}_i;\zeta) \quad - \quad \nabla f(\boldsymbol{y}_i)\|^2 \le 2\delta^2\,\mathbb{E}_i\|\boldsymbol{y}_i - \boldsymbol{x}\|^2 + 2G^2 + \sigma^2\,.$$

Proof. We can separate the noise from the rest of the terms and expand as

$$\begin{aligned}
\mathbb{E}_{\zeta,i}\|\nabla f_i(\boldsymbol{y}_i;\zeta) - \nabla f(\boldsymbol{y}_i)\|^2 &= \mathbb{E}_{\zeta,i}\|\nabla f_i(\boldsymbol{y}_i;\zeta) \pm \nabla f_i(\boldsymbol{x}) \pm \nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y}_i)\|^2 \\
&\le \mathbb{E}_i\|\nabla f_i(\boldsymbol{y}_i) \pm \nabla f_i(\boldsymbol{x}) \pm \nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y}_i)\|^2 + \sigma^2 \\
&\le 2\,\mathbb{E}_i\|\nabla f_i(\boldsymbol{y}_i) + \nabla f(\boldsymbol{x}) - \nabla f_i(\boldsymbol{x}) - \nabla f(\boldsymbol{y}_i)\|^2 \\
&\qquad\qquad + 2\,\mathbb{E}_i\|\nabla f_i(\boldsymbol{x}) - \nabla f(\boldsymbol{x})\|^2 + \sigma^2 \\
&\le 2\,\mathbb{E}_i\|\nabla f_i(\boldsymbol{y}_i) + \nabla f(\boldsymbol{x}) - \nabla f_i(\boldsymbol{x}) - \nabla f(\boldsymbol{y}_i)\|^2 + 2G^2 + \sigma^2 \\
&\le 2\delta^2\,\mathbb{E}_i\|\boldsymbol{y}_i - \boldsymbol{x}\|^2 + 2G^2 + \sigma^2\,.
\end{aligned}$$

The first inequality used (A3), the second used Young's inequality, the third used (A1), and the last used (A2) in the form of Lemma 3. □

Note that the bias for MimeLite is very similar to that of Mime, except that Mime has dependence of $\frac{G^2}{S}$, whereas MimeLite has $G^2 + \sigma^2$. Hence, the rate of convergence of MimeLite will depend on G^2 wheras Mime will have the optimal dependency of G^2/S. Hence, in the rest of the proof, we will consider **only Mime** and simply replace G^2/S with $(G^2 + \sigma^2)$ to obtain the corresponding results for MimeLite.

H.3 Change in each client update

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

Lemma 19. *For the client update* (13)*, given* (A1) *and* (A2)*, the following holds for any $a \in [0,1]$ where $\boldsymbol{e}^t := \boldsymbol{m}^t - \nabla f(\boldsymbol{x}^{t-1})$ and $\Delta_{i,k}^t := \mathbb{E}\|\boldsymbol{y}_{i,k}^t - \boldsymbol{x}^{t-2}\|^2$:*

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

Proof. Starting from the client update (13), we can rewrite it as

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

We can use the relaxed triangle inequality Lemma 1 to claim

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

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 20. *For MimeMVR updates* (13) *with* $\eta \le \frac{1}{6K\delta}$ *and given* (A1) *and* (A2)*, the following holds*

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

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

Proof. Starting from the MimeMVR update (13) and the relaxed triangle inequality with $c = 2K$,

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

The last inequality used the update variance bound Lemma 19. We can simplify the expression further since $\eta \le \frac{1}{6K\delta}$ implies $18K\eta^2\delta^2 \le \frac{1}{2K}$. □

Progress in one step. Now we can compute the progress made in each step.

Lemma 21. *For any client update step with step size* $\eta \le \min\left(\frac{1}{L}, \frac{1}{192\delta K}\right)$ *and given that* (A1), (A2) *hold, we have*

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

Proof. The assumption that f is L-smooth implies a quadratic upper bound (10).

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

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 $\eta \le \frac{1}{L}$. Taking expectation on both sides and using the update variance bound Lemma 19,

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

Multiplying the distance bound Lemma 20 by $\delta\big(1+\frac{2}{K}\big)^{K-k}$. 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

$$\begin{aligned} \delta\left(1+\frac{2}{K}\right)^{K-k}\Delta_{i,k}^t &\le \delta\left(1+\frac{2}{K}\right)^{K-k}\Bigg(\left(1+\frac{1}{K}\right)\Delta_{i,k-1}^t + 18\eta^2 K a^2\frac{G^2}{S} \\ &\qquad + 18\eta^2 K\,\mathbb{E}\|\boldsymbol{e}^{t-1}\|^2 + 6\eta^2 K\|\nabla f(\boldsymbol{y}_{i,k-1}^t)\|^2\Bigg) \\ &\le \delta\left(1+\frac{2}{K}\right)^{K-(k-1)}\Delta_{i,k-1}^t - \frac{\delta}{K}\left(1+\frac{2}{K}\right)^{K-k}\Delta_{i,k-1}^t \\ &\qquad + 48\eta^2\delta K\,\mathbb{E}\|\nabla f(\boldsymbol{y}_{i,k-1}^t)\|^2 + \frac{144\eta^2\delta K a^2 G^2}{S} + 144\eta^2\delta K\,\mathbb{E}\|\boldsymbol{e}^{t-1}\|^2 \\ &\le \delta\left(1+\frac{2}{K}\right)^{K-(k-1)}\Delta_{i,k-1}^t - \frac{\delta}{K}\Delta_{i,k-1}^t + 48\eta^2\delta K\,\mathbb{E}\|\nabla f(\boldsymbol{y}_{i,k-1}^t)\|^2 \\ &\qquad + \frac{144\eta^2\delta K a^2 G^2}{S} + 144\eta^2\delta K\,\mathbb{E}\|\boldsymbol{e}^{t-1}\|^2 \,. \end{aligned}$$

Adding these two inequalities together yields

$$\begin{aligned} \mathbb{E}\, f(\boldsymbol{y}_{i,k}^t) + \delta\left(1+\frac{2}{K}\right)^{K-k}\Delta_{i,k}^t &\le \mathbb{E}\, f(\boldsymbol{y}_{i,k-1}^t) + \delta\left(1+\frac{2}{K}\right)^{K-(k-1)}\Delta_{i,k-1}^t \\ &\qquad - \left(\frac{\eta}{2} - 48\eta^2\delta K\right)\mathbb{E}\|\nabla f(\boldsymbol{y}_{i,k-1}^t)\|^2 \\ &\qquad + \left(\frac{3\eta}{2} + 144\eta^2\delta K\right)\mathbb{E}\|\boldsymbol{e}^{t-1}\|^2 \\ &\qquad + \left(\frac{3\eta}{2} + +144\eta^2\delta K\right)\frac{a^2G^2}{S} \,. \end{aligned}$$

Using our bound on the step-size that $\eta \le \frac{1}{192\delta K}$ implies that $\eta\delta K \le \frac{1}{48*4}$. □

H.4 Change in each round

We now see how the quantities we defined change across rounds.

Distance moved in a round.

Lemma 22. *For MimeMVR updates* (13) *with $\eta \leq \frac{1}{6K\delta}$ and given* (A1) *and* (A2)*, the following holds*

$$\Delta^t \leq 54K^2\eta^2\,\mathbb{E}\|\boldsymbol{e}^{t-1}\|^2 + \frac{54K^2\eta^2a^2G^2}{S} + \frac{1}{KS}\sum_{i,k} 18K^2\eta^2\,\mathbb{E}\|\nabla f(\boldsymbol{y}_{i,k-1}^t)\|^2\,,$$

where we define $\Delta^t := \mathbb{E}\|\boldsymbol{x}^t - \boldsymbol{x}^{t-1}\|^2$.

Proof. Starting from the MimeMVR update (13) and following the proof of Lemma 20,

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

Note that $\boldsymbol{x}^t = \frac{1}{S}\sum_{i\in\mathcal{S}}\boldsymbol{y}_{i,K}^t$ and so,

$$\begin{aligned}
&\mathbb{E}\|\boldsymbol{x}^t - \boldsymbol{x}^{t-1}\|^2 \\
&\quad\leq \frac{1}{S}\sum_{i\in\mathcal{S}}\mathbb{E}\|\boldsymbol{y}_{i,K}^t - \boldsymbol{x}^{t-1}\|^2 \\
&\quad\leq \frac{1}{S}\sum_{i\in\mathcal{S}}\sum_{k}\left(18K\eta^2\,\mathbb{E}\|e^{t-1}\|^2 + \frac{18K\eta^2a^2G^2}{S} + 6K\eta^2\,\mathbb{E}\|\nabla f(\boldsymbol{y}_{i,k-1}^t)\|^2\right)\left(1 + \frac{1}{K}\right)^{K-k} \\
&\quad\leq 54K^2\eta^2\,\mathbb{E}\|\boldsymbol{e}^{t-1}\|^2 + \frac{54K^2\eta^2a^2G^2}{S} + \frac{1}{KS}\sum_{i,k}18K^2\eta^2\,\mathbb{E}\|\nabla f(\boldsymbol{y}_{i,k-1}^t)\|^2\,.
\end{aligned}$$

Here we used the inequality that for all k, $\left(1 + \frac{1}{K}\right)^{K-k} \leq 3$. □

Server momentum variance. We compute the error of the server momentum $\boldsymbol{m}^{t-1}$ defined as $\boldsymbol{e}^t = \boldsymbol{m}^t - \nabla f(\boldsymbol{x}^{t-1})$. Its expected norm can be bounded as follows.

Lemma 23. *For the momentum update* (16)*, given* (A1) *and* (A2)*, the following holds for any $\eta \leq \frac{1}{51\delta K}$ and $1 \geq a \geq 2592K^2\delta^2\eta^2$,*

$$\mathbb{E}\|\boldsymbol{e}^t\|^2 \leq (1 - \tfrac{23a}{24})\,\mathbb{E}\|\boldsymbol{e}^{t-1}\|^2 + \frac{3a^2G^2}{S} + \frac{1}{KS}\sum_{i,k}36K^2\delta^2\eta^2\,\mathbb{E}\|\nabla f(\boldsymbol{y}_{i,k-1}^t)\|^2\,.$$

Proof. Starting from the momentum update (16),

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

Now, the term $\boldsymbol{e}^{t-1}$ 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 $\boldsymbol{e}^{t-1}$ following Lemma 2 and then the relaxed triangle inequality Lemma 1 to claim

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

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 $(1-a)^2 \leq (1-a) \leq 1$ for $a \in [0,1]$. We can continue by bounding Δ^{t-1} using Lemma 22.

$$\begin{aligned}
\mathbb{E}\|\boldsymbol{e}^t\|^2 &\leq (1-a)\,\mathbb{E}\|\boldsymbol{e}^{t-1}\|^2 + 2\delta^2\Delta^{t-1} + \frac{2a^2G^2}{S} \\
&\leq (1-a)\,\mathbb{E}\|\boldsymbol{e}^{t-1}\|^2 + \frac{2a^2G^2}{S} \\
&\quad + 108K^2\delta^2\eta^2\,\mathbb{E}\|\boldsymbol{e}^{t-1}\|^2 + \frac{108K^2\delta^2\eta^2a^2G^2}{S} + \frac{1}{KS}\sum_{i,k}36K^2\delta^2\eta^2\,\mathbb{E}\|\nabla f(\boldsymbol{y}_{i,k-1}^t)\|^2 \\
&\leq (1-\tfrac{23a}{24})\,\mathbb{E}\|\boldsymbol{e}^{t-1}\|^2 + \frac{3a^2G^2}{S} + \frac{1}{KS}\sum_{i,k}36K^2\delta^2\eta^2\,\mathbb{E}\|\nabla f(\boldsymbol{y}_{i,k-1}^t)\|^2\,.
\end{aligned}$$

The last step used our bound on the momentum parameter that $1 \geq a \geq 2592\eta^2\delta^2K^2$. Note that $\eta \leq \frac{1}{51\delta K}$ ensures that this set is non-empty. □

Progress in one round. Finally, we can compute the progress made in a round. Note that we need a technical condition that f is δ-weakly convex. However, this is only needed because we insist on running the algorithm on S clients in parallel and then averaging their weights—the averaging requires weak convexity to ensure that the loss doesn't blow up. It has been experimentally observed in [43] that with the right initialization, averaging of the parameters does not increase the loss value and so weak convexity within this region might be vaalid. Finally note that if we instead simply run the local updates on a single chosen client with all the rest only being used to compute $\boldsymbol{c}^{t-1}$, we will retain all convergence rates without needing weak-convexity.

Lemma 24. *For any round of MimeMVR with step size $\eta \leq \min\left(\frac{1}{L}, \frac{1}{864\delta K}\right)$ and momentum parameter $a \geq 912\eta^2\delta^2K^2$. Then, given that (A1)–(A2) hold and f is δ-weakly convex, we have*

$$\frac{\eta}{24KS}\sum_{k\in[K],j\in\mathcal{S}^t}\mathbb{E}\|\nabla f(\boldsymbol{y}_{i,k-1}^t)\|^2 \leq \Phi^{t-1} - \Phi^t + \frac{17\eta a\delta^2K^2G^2}{S}\,,$$

where we define the sequence

$$\Phi^t := \tfrac{1}{K}\,\mathbb{E}[f(\boldsymbol{x}^t) - f^\star] + \frac{96\eta}{23a}\,\mathbb{E}\|\boldsymbol{e}^t\|^2 + \frac{8\delta}{K}\Delta^t\,.$$

Proof. We start by summing over the progress in single client updates as in Lemma 21

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

Recall that $\Delta_{i,k}^t = \mathbb{E}\|\boldsymbol{y}_{i,k}^t - \boldsymbol{x}^{t-2}\|^2$ and $\boldsymbol{y}_{i,0}^t = \boldsymbol{x}^{t-1}$. This gives the last step above, making $\Delta_{i,0}^t = \Delta^{t-1}$. Then by the averaging Lemma 4, we have

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

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

$$\begin{aligned}
&\frac{\eta}{4KS}\sum_{k\in[K],j\in\mathcal{S}^t} \mathbb{E}\|\nabla f(\boldsymbol{y}_{i,k-1}^t)\|^2 \\
&\qquad\le \tfrac{1}{K}\,\mathbb{E}[f(\boldsymbol{x}^{t-1})] + 3\eta\,\mathbb{E}\|\boldsymbol{e}^{t-1}\|^2 + \frac{8\delta}{K}\Delta^{t-1} - \tfrac{1}{K}\,\mathbb{E}[f(\boldsymbol{x}^t)] + \frac{3\eta a^2 G^2}{S}\,.
\end{aligned}$$

We can use the bound on Δ_t from Lemma 22 to proceed as

$$\begin{aligned}
&\frac{\eta}{4KS}\sum_{k\in[K],j\in\mathcal{S}^t} \mathbb{E}\|\nabla f(\boldsymbol{y}_{i,k-1}^t)\|^2 \\
&\quad\le \tfrac{1}{K}\,\mathbb{E}[f(\boldsymbol{x}^{t-1})] - \tfrac{1}{K}\,\mathbb{E}[f(\boldsymbol{x}^t)] + 3\eta\,\mathbb{E}\|\boldsymbol{e}^{t-1}\|^2 + \frac{3\eta a^2 G^2}{S} \\
&\qquad + \frac{8\delta}{K}\Delta^{t-1} - \frac{8\delta}{K}\Delta^t \\
&\qquad + 432K\delta\eta^2\,\mathbb{E}\|\boldsymbol{e}^{t-1}\|^2 + \frac{432K\delta\eta^2 a^2 G^2}{S} + \frac{1}{KS}\sum_{i,k} 144K\delta\eta^2\,\mathbb{E}\|\nabla f(\boldsymbol{y}_{i,k-1}^t)\|^2 \\
&\quad\le \tfrac{1}{K}\,\mathbb{E}[f(\boldsymbol{x}^{t-1})] - \tfrac{1}{K}\,\mathbb{E}[f(\boldsymbol{x}^t)] + 4\eta\,\mathbb{E}\|\boldsymbol{e}^{t-1}\|^2 + \frac{4\eta a^2 G^2}{S} \\
&\qquad + \frac{8\delta}{K}\Delta^{t-1} - \frac{8\delta}{K}\Delta^t + \frac{\eta}{6KS}\sum_{i,k}\mathbb{E}\|\nabla f(\boldsymbol{y}_{i,k-1}^t)\|^2
\end{aligned}$$

The last step used the bound on the step size that $\eta \le \frac{1}{864\delta K}$. Now, multiplying the error bound Lemma 23 by $\frac{96\eta}{23a}$ gives

$$\frac{96\eta}{23a}\,\mathbb{E}\|\boldsymbol{e}^t\|^2 \le \frac{4*24\eta}{23a}(1-\tfrac{23a}{24})\,\mathbb{E}\|\boldsymbol{e}^{t-1}\|^2 + \frac{13\eta a G^2}{S} + \frac{1}{KS}\sum_{i,k}\frac{38K^2\delta^2\eta^3}{a}\,\mathbb{E}\|\nabla f(\boldsymbol{y}_{i,k-1}^t)\|^2\,.$$

Adding this to the previously obtained bound yields

$$\begin{aligned}\frac{\eta}{4KS}\sum_{k\in[K],j\in\mathcal{S}^t}\mathbb{E}\|\nabla f(\boldsymbol{y}_{i,k-1}^t)\|^2 \le \Big(\frac{1}{6}+\frac{38K^2\delta^2\eta^2}{a}\Big)&\frac{\eta}{KS}\sum_{k\in[K],j\in\mathcal{S}^t}\mathbb{E}\|\nabla f(\boldsymbol{y}_{i,k-1}^t)\|^2\\ &+\tfrac{1}{K}\,\mathbb{E}[f(\boldsymbol{x}^{t-1})]-\tfrac{1}{K}\,\mathbb{E}[f(\boldsymbol{x}^{t})]\\ &+\frac{96\eta}{23a}\,\mathbb{E}\|\boldsymbol{e}^{t-1}\|^2-\frac{96\eta}{23a}\,\mathbb{E}\|\boldsymbol{e}^{t}\|^2\\ &+\frac{8\delta}{K}\Delta^{t-1}-\frac{8\delta}{K}\Delta^{t}\\ &-\tfrac{1}{K}\,\mathbb{E}[f(\boldsymbol{x}^{t})]-\frac{4\eta}{a}\,\mathbb{E}\|\boldsymbol{e}^{t}\|^2\\ &+\big(13\eta a+3\eta a^2\big)\frac{G^2}{S}\,.\end{aligned}$$

Since $a\ge 912\eta^2K^2\delta^2$, we have $\frac{1}{4}-\Big(\frac{1}{6}-\frac{38K^2\delta^2\eta^2}{a}\Big)\ge\frac{1}{24}$. Using this proves the lemma. □

H.5 Final convergence rates

Theorem V (Convergence of MimeMVR)**.** *Let us run MimeMVR with step size* $\eta=\min\Big(\frac{1}{L},\frac{1}{864\delta K},\Big(\frac{S(f(\boldsymbol{x}^0)-f^\star)}{6936K^3T\delta^2G^2}\Big)^{1/3}\Big)$ *and momentum parameter* $a=\max\big(1536\eta^2\delta^2K^2,\frac{1}{T}\big)$. *Then, given that* (A1) *and* (A2) *hold, we have*

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

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

Proof. Unroll the one round progress Lemma 24 and average over T rounds to get

$$\frac{1}{KST}\sum_{t\in[T]}\sum_{k\in[K]}\sum_{j\in\mathcal{S}^t}\mathbb{E}\|f(\boldsymbol{y}_{i,k-1}^t)\|^2\le\frac{24(\Phi^0-\Phi^T)}{\eta T}+\frac{408aG^2}{S}\,.$$

Recall that we defined

$$\Phi^t:=\tfrac{1}{K}\,\mathbb{E}[f(\boldsymbol{x}^t)-f^\star]+\frac{96\eta}{23a}\,\mathbb{E}\|\boldsymbol{e}^t\|^2+\frac{8\delta}{K}\Delta^t\,.$$

Hence, $\Phi^T\ge 0$. Further, note that by definition $\Delta^0=0$ and $\mathbb{E}\|\boldsymbol{e}_0\|^2:=\mathbb{E}\|\boldsymbol{m}^0-\nabla f(\boldsymbol{x}^0)\|^2$. [14] show that by using time-varying step sizes, it is possible to directly control the error $\boldsymbol{e}_0$. Alternatively, [62] use a large initial accumulation for the momentum term. For the sake of simplicity, we will follow the latter approach. It is straightforward to extend our techniques to the time-varying step-size case as well but with additional proof complexity. Note that either way, the total complexity only changes by a factor of 2. Suppose that we run the algorithm for $2T$ rounds wherein for the first T rounds, we simply compute $\boldsymbol{m}^0=\frac{1}{T_0S}\sum_{t=1}^{T_0}\sum_{j\in\mathcal{S}^t}\nabla f_j(\boldsymbol{x}^0)$. With this, we have $\boldsymbol{e}_0=\mathbb{E}\|\boldsymbol{m}^0-\nabla f(\boldsymbol{x}^0)\|^2\le\frac{G^2}{ST}$. Thus, we have for the first round $t=1$

$$\Phi^0=\tfrac{1}{K}\,\mathbb{E}[f(\boldsymbol{x}^0)-f^\star]+\frac{96\eta}{23a}\,\mathbb{E}\|\boldsymbol{e}^0\|^2\le\tfrac{1}{K}\,\mathbb{E}[f(\boldsymbol{x}^0)-f^\star]+\frac{96\eta G^2}{23aTS}\,.$$

Together, this gives

$$\frac{1}{KST}\sum_{t\in[T]}\sum_{k\in[K]}\sum_{i\in\mathcal{S}^t}\mathbb{E}\|f(\boldsymbol{y}_{i,k-1}^t)\|^2\le\frac{24(f(\boldsymbol{x}^0)-f^\star)}{\eta KT}+\frac{96G^2}{aT^2S}+\frac{408aG^2}{S}\,.$$

The above equation holds for any choice of $\eta\le\min\big(\frac{1}{L},\frac{1}{864\delta K}\big)$ and momentum parameter $a\ge 912\eta^2\delta^2K^2$. Set the momentum parameter as

$$a=\max\Big(912\eta^2\delta^2K^2,\frac{1}{T}\Big)$$

With this choice, we can simplify the rate of convergence as

$$\frac{24(f(\boldsymbol{x}^0) - f^\star)}{\eta KT} + \frac{96G^2}{TS} + \frac{166464\eta^2\delta^2K^2G^2}{S} + \frac{408G^2}{ST}\,.$$

Now let us pick

$$\eta = \min\bigg(\frac{1}{L}, \frac{1}{864\delta K}, \Big(\frac{S(f(\boldsymbol{x}^0) - f^\star)}{6936K^3T\delta^2G^2}\Big)^{1/3}\bigg)\,.$$

For this combination of step size η and a, the rate simplifies to

$$\frac{504G^2}{TS} + 916\Big(\frac{(f(\boldsymbol{x}^0) - f^\star)\delta^2G^2}{ST^2}\Big)^{1/3} + \frac{24(L + 864\delta K)(f(\boldsymbol{x}^0) - f^\star)}{KT}\,.$$

This finishes the proof of the theorem. □

Theorem VI (Convergence of MimeLiteMVR)**.** *Let us run MimeLiteMVR with step size* $\eta = \min\Big(\frac{1}{L}, \frac{1}{864\delta K}, \Big(\frac{(f(\boldsymbol{x}^0)-f^\star)}{6936K^3T\delta^2(G^2+\sigma^2)}\Big)^{1/3}\Big)$ *and momentum parameter* $a = \max\big(1536\eta^2\delta^2K^2, \frac{1}{T}\big)$. *Then, given that* (A1) *and* (A2*) *hold, we have*

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

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

Proof. The proof for MimeLiteMVR is identical to that of MimeMVR, except that as noted in Lemma 18, the $\frac{G^2}{S}$ term in Mime gets replaced by $(G^2 + \sigma^2)$ everywhere. Note that MimeLiteMVR (Lemma 18) requires a weaker Hessian variance condition of $\|\nabla^2 f_i(\boldsymbol{x}) - \nabla^2 f(\boldsymbol{x})\| \leq \delta$ as opposed to MimeMVR which needs $\|\nabla^2 f_i(\boldsymbol{x};\zeta) - \nabla^2 f(\boldsymbol{x})\| \leq \delta$. □

Note that the final convergence rates of MimeMVR and MimeLiteMVR both include the intermediate client parameters. To implement this algorithm would require additional communication where in each round a random client parameter in $\{\boldsymbol{y}_{i,1}^t, \ldots, \boldsymbol{y}_{i,K}^t\}$ is communicated to the server. However, as is common in non-convex stochastic analysis, we expect the last iterate to converge at a similar rate as well in practice.

I Algorithm pseudocodes

Algorithm 2 FedAvg framework

```
input: initial $\boldsymbol{x}$ and $\boldsymbol{s}$, server learning rate $\eta_g$, client learning rate $\eta_l$, and base optimizer $\mathcal{B} = (\mathcal{U}, \mathcal{V})$
for each round $t = 1, \cdots, T$ do
    sample subset $\mathcal{S}$ of clients
    communicate $\boldsymbol{x}$ to all clients $i \in \mathcal{S}$
    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 $\zeta$ from local data
            update $\boldsymbol{y}_i \leftarrow \boldsymbol{y}_i - \eta_l \nabla f_i(\boldsymbol{y}_i; \zeta)$
        end for
        communicate $\boldsymbol{y}_i$
    end on client
    compute aggregate pseudo-gradient $\boldsymbol{g} \leftarrow \frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}(\boldsymbol{x} - \boldsymbol{y}_i)$
    $\boldsymbol{x} \leftarrow \eta_g \mathcal{U}(\boldsymbol{g}, \boldsymbol{s})$  (update server parameters)
    $\boldsymbol{s} \leftarrow \mathcal{V}(\boldsymbol{g}, \boldsymbol{s})$  (update optimizer state)
end for
```

Algorithm 3 **MimeMom** and **MimeLiteMom**

```
input: initial $\boldsymbol{x}$, and hyperparameters $\eta$, $\beta$. optional $\eta_g$ (default $= 1$)
initialize $\boldsymbol{m} \leftarrow 0$, $\boldsymbol{c} \leftarrow 0$
for each round $t = 1, \cdots, T$ do
    sample subset $\mathcal{S}$ of clients
    communicate $(\boldsymbol{x}, \boldsymbol{s} = \boldsymbol{m})$ and $\boldsymbol{c}$ (only Mime) to all clients $i \in \mathcal{S}$
    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 $\zeta$ from local data
            $\boldsymbol{g}_i \leftarrow \nabla f_i(\boldsymbol{y}_i; \zeta) - \nabla f_i(\boldsymbol{x}; \zeta) + \boldsymbol{c}$ (Mime)
            $\boldsymbol{g}_i \leftarrow \nabla f_i(\boldsymbol{y}_i; \zeta)$ (MimeLite)
            update using server momentum $\boldsymbol{y}_i \leftarrow \boldsymbol{y}_i - \eta((1 - \beta_1)\boldsymbol{g}_i + \beta_1 \boldsymbol{m})$
        end for
        compute full local-batch gradient $\nabla f_i(\boldsymbol{x})$
        communicate $(\boldsymbol{y}_i, \nabla f_i(\boldsymbol{x}))$
    end on client
    compute $\boldsymbol{c} \leftarrow \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \nabla f_i(\boldsymbol{x})$
    $\boldsymbol{m} \leftarrow ((1 - \beta_1)\boldsymbol{c} + \beta_1 \boldsymbol{m})$ (update server momentum)
    $\boldsymbol{x} \leftarrow \boldsymbol{x} - \eta_g \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} (\boldsymbol{x} - \boldsymbol{y}_i)$ (update server parameters)
end for
```

Algorithm 4 **MimeAdam** and **MimeLiteAdam**

```
input: initial $\boldsymbol{x}$, and hyperparameters $\eta$, $\beta_1, \beta_2, \varepsilon_0$. optional $\eta_g$ (default $= 1$)
initialize $\boldsymbol{m} \leftarrow 0$, $\boldsymbol{v} \leftarrow 0$, $\boldsymbol{c} \leftarrow 0$
for each round $t = 1, \cdots, T$ do
    sample subset $\mathcal{S}$ of clients
    communicate $(\boldsymbol{x}, \boldsymbol{s} = (\boldsymbol{m}, \boldsymbol{v}))$ and $\boldsymbol{c}$ (only Mime) to all clients $i \in \mathcal{S}$
    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 $\zeta$ from local data
            $\boldsymbol{g}_i \leftarrow \nabla f_i(\boldsymbol{y}_i; \zeta) - \nabla f_i(\boldsymbol{x}; \zeta) + \boldsymbol{c}$ (Mime)
            $\boldsymbol{g}_i \leftarrow \nabla f_i(\boldsymbol{y}_i; \zeta)$ (MimeLite)
            update $\boldsymbol{y}_i \leftarrow \boldsymbol{y}_i - \eta((1 - \beta_1)\boldsymbol{g}_i + \beta_1 \boldsymbol{m}) / \big((\sqrt{\boldsymbol{v}} + \varepsilon_0)(1 - \beta_1^t)\big)$
        end for
        compute full local-batch gradient $\nabla f_i(\boldsymbol{x})$
        communicate $(\boldsymbol{y}_i, \nabla f_i(\boldsymbol{x}))$
    end on client
    compute $\boldsymbol{c} \leftarrow \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \nabla f_i(\boldsymbol{x})$
    $\boldsymbol{m} \leftarrow ((1 - \beta_1)\boldsymbol{c} + \beta_1 \boldsymbol{m}) / (1 - \beta_1^t)$
    $\boldsymbol{v} \leftarrow ((1 - \beta_2)\boldsymbol{c}^2 + \beta_2 \boldsymbol{v}) / (1 - \beta_2^t)$
    $\boldsymbol{x} \leftarrow \boldsymbol{x} - \eta_g \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} (\boldsymbol{x} - \boldsymbol{y}_i)$ (update server parameters)
end for
```

Algorithm 5 MimeMVR pseudocode

```
input: initial $\boldsymbol{x}^0$, learning rate $\eta$
initialize $\boldsymbol{c}^0 \leftarrow 0$, $\boldsymbol{m}^0 \leftarrow 0$
for each round $t = 1, \cdots, T$ do
    sample subset $\mathcal{S}$ of clients
    communicate $\boldsymbol{x}^{t-1}, \boldsymbol{x}^{t-2}, m^{t-1}, c^{t-1}$ to all clients $i \in \mathcal{S}$
    on client $i \in \mathcal{S}$ in parallel do
        initialize local model $\boldsymbol{y}_{i,0}^t \leftarrow \boldsymbol{x}^{t-1}$
        for $k = 1, \cdots, K$ do
            sample mini-batch $\zeta_{i,k}^t$ from local data
            compute SVRG gradient $\boldsymbol{g}_{i,k}^t \leftarrow \nabla f_i(\boldsymbol{y}_{i,k-1}^t; \zeta_{i,k}^t) - \nabla f_i(\boldsymbol{x}^{t-1}; \zeta_{i,k}^t) + \boldsymbol{c}^{t-1}$
            compute corrected momentum $\boldsymbol{d}_{i,k}^t \leftarrow a\boldsymbol{g}_{i,k}^t + (1-a)\boldsymbol{m}^{t-1} + (1-a)(\nabla f_i(\boldsymbol{y}_{i,k-1}^t; \zeta_{i,k}^t) - \nabla f_i(\boldsymbol{x}^{t-1}; \zeta_{i,k}^t))$
            update $\boldsymbol{y}_{i,k}^t = \boldsymbol{y}_{i,k-1}^t - \eta \boldsymbol{d}_{i,k}^t$
        end for
        compute full local-batch gradients $\nabla f_i(\boldsymbol{x}^{t-1}), \nabla f_i(\boldsymbol{x}^{t-2})$
        communicate $(\boldsymbol{y}_{i,K}^t, \nabla f_i(\boldsymbol{x}^{t-1}), \nabla f_i(\boldsymbol{x}^{t-2}))$
    end on client
    compute new aggregate pseudo-gradient $\boldsymbol{c}^t \leftarrow \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \nabla f_i(\boldsymbol{x}^{t-1})$
    compute old aggregate pseudo-gradient $\tilde{\boldsymbol{c}}^t \leftarrow \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \nabla f_i(\boldsymbol{x}^{t-2})$
    update server momentum $\boldsymbol{m}^t \leftarrow a\boldsymbol{c}^t + (1-a)\boldsymbol{m}^{t-1} + (1-a)(\boldsymbol{c}^t - \tilde{\boldsymbol{c}}^t)$
    update server parameters $\boldsymbol{x}^t \leftarrow \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \boldsymbol{y}_{i,K}^t$
end for
```