Ordered Momentum for Asynchronous SGD

Chang-Wei Shi **Yi-Rui Yang** **Wu-Jun Li***
National Key Laboratory for Novel Software Technology,
School of Computer Science, Nanjing University, Nanjing, China
{shicw, yangyr}@smail.nju.edu.cn, liwujun@nju.edu.cn

Abstract

Distributed learning is essential for training large-scale deep models. Asynchronous SGD (ASGD) and its variants are commonly used distributed learning methods, particularly in scenarios where the computing capabilities of workers in the cluster are heterogeneous. Momentum has been acknowledged for its benefits in both optimization and generalization in deep model training. However, existing works have found that naively incorporating momentum into ASGD can impede the convergence. In this paper, we propose a novel method called ordered momentum (OrMo) for ASGD. In OrMo, momentum is incorporated into ASGD by organizing the gradients in order based on their iteration indexes. We theoretically prove the convergence of OrMo with both constant and delay-adaptive learning rates for non-convex problems. To the best of our knowledge, this is the first work to establish the convergence analysis of ASGD with momentum without dependence on the maximum delay. Empirical results demonstrate that OrMo can achieve better convergence performance compared with ASGD and other asynchronous methods with momentum.

1 Introduction

Many machine learning problems can be formulated as optimization problems of the following form:

$$\min_{\mathbf{w}\in\mathbb{R}^d} F(\mathbf{w}) = \mathbb{E}_{\xi\sim\mathcal{D}}\left[f(\mathbf{w};\xi)\right], \tag{1}$$

where $\mathbf{w}$ denotes the model parameter, d is the dimension of the parameter, $\mathcal{D}$ represents the distribution of the training instances and $f(\mathbf{w};\xi)$ denotes the loss on the training instance ξ.

Stochastic gradient descent (SGD) [28] and its variants [6, 12] are widely employed to solve the problem in (1). At each iteration, SGD uses one stochastic gradient or a mini-batch of stochastic gradients as an estimate of the full gradient to update the model parameter. In practice, momentum [26, 36, 24, 33] is often incorporated into SGD as a crucial technique for faster convergence and better generalization performance. Many popular machine learning libraries, such as TensorFlow [1] and PyTorch [25], include SGD with momentum (SGDm) as one of the optimizers.

Due to the rapid increase in the sizes of both models and datasets in recent years, a single machine is often insufficient to complete the training task of machine learning models within a reasonable time. Distributed learning [42, 37] aims to distribute the computations across multiple machines (workers) to accelerate the training process. Because of its necessity for training large-scale machine learning models, distributed learning has become a hot research topic in recent years. Existing distributed learning methods can be categorized into two main types: synchronous distributed learning (SDL) methods [19, 35, 29, 38, 44, 39] and asynchronous distributed learning (ADL) methods [2, 27, 8, 47, 17, 43]. In SDL methods, faster workers that have completed the computation must wait idly for

*Corresponding author.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

the other slower workers in each communication round. Hence, the speed of SDL methods is often hindered by slow workers. In contrast, faster workers do not necessarily wait idly for the other slower workers in ADL methods, because ADL methods require aggregating information from only one worker or a subset of workers in each communication round. Representative ADL methods include asynchronous SGD (ASGD) and its variants [2, 8, 46, 30, 49, 48, 7, 3, 31, 22, 13]. In ASGD, once a worker finishes its gradient computation, the parameter (typically on the server) is immediately updated using this gradient through an SGD step, without waiting for other workers.

Momentum has been acknowledged for its benefits in both optimization and generalization in deep model training [33]. In SDL methods, momentum is extensively utilized across various domains, including decentralized algorithms [18, 45], communication compression algorithms [19, 29, 38, 40, 34, 41], infrequent communication algorithms [44, 39, 40], and federated learning algorithms [21, 32]. However, in ADL methods, some works [23, 9] have found that naively incorporating momentum into ASGD may decrease the convergence rate or even result in divergence. To tackle this challenge, some more sophisticated methods have been proposed to incorporate momentum into ASGD. The works in [23, 9] recommend tuning the momentum coefficient to enhance convergence performance when naively incorporating momentum into ASGD. The work in [9] proposes shifted momentum, which maintains local momentum on each worker. Inspired by Nesterov's accelerated gradient, the work in [4] proposes SMEGA^2, which leverages the momentum to estimate the future parameter. However, the process of tuning the momentum coefficient in [23, 9] is time-consuming and yields limited improvement in practice. Although shifted momentum and SMEGA^2 can achieve better empirical convergence performance than the method which naively incorporates momentum into ASGD, both of them lack theoretical convergence analysis.

In this paper, we propose a novel method, called ordered momentum (OrMo), for asynchronous SGD. The main contributions of this paper are outlined as follows:

- OrMo incorporates momentum into ASGD by organizing the gradients in order based on their iteration indexes.
- We theoretically prove the convergence of OrMo with both constant and delay-adaptive learning rates for non-convex problems. To the best of our knowledge, this is the first work to establish the convergence analysis for ASGD with momentum without dependence on the maximum delay.
- Empirical results demonstrate that OrMo can achieve better convergence performance compared with ASGD and other asynchronous methods with momentum.

2 Preliminary

In this paper, we use $\|\cdot\|$ to denote the L_2 norm. For a positive integer n, we use $[n]$ to denote the set $\{0, 1, 2, \ldots, n-1\}$. $\nabla f(\mathbf{w}; \xi)$ denotes the stochastic gradient computed over the training instance ξ and model parameter $\mathbf{w}$. In this paper, we focus on the widely used Parameter Server framework [15], where the server is responsible for storing and updating the model parameter and the workers are responsible for sampling training instances and computing stochastic gradients. For simplicity, we assume that each worker samples one training instance for gradient computation each time. The analysis of mini-batch sampling on each worker follows a similar approach.

One of the most representative methods for distributing SGD across multiple workers is Synchronous SGD (SSGD) [20, 10]. Distributed SGD (DSGD), as presented in Algorithm 1, unifies SSGD and ASGD within a single framework [13]. The waiting set $\mathcal{C}$ in Algorithm 1 is a collection of workers (indexes) that are awaiting the server to send the latest parameter. The only difference between SSGD and ASGD is the communication scheduler associated with the waiting set. SSGD corresponds to DSGD with a synchronous communication scheduler, while ASGD corresponds to DSGD with an asynchronous communication scheduler. We use $\mathbf{g}_{ite(k_t,t)}^{k_t}$ to denote the stochastic gradient $\nabla f(\mathbf{w}_{ite(k_t,t)}; \xi^{k_t})$, where k_t is the index of the worker whose gradient participates in the parameter update at iteration t and ξ^{k_t} denotes a training instance sampled on worker k_t. The function $ite(k, t)$ denotes the iteration index of the latest parameter sent to worker k before iteration t, where $k \in [K]$ and $t \in [T]$. The delay of the gradient $\mathbf{g}_{ite(k_t,t)}^{k_t}$ is defined as $\tau_t = t - ite(k_t, t)$. When $K = 1$, DSGD degenerates to vanilla SGD, i.e., $ite(k_t, t) \equiv t$.

Algorithm 1 Distributed SGD

Server:
Input: number of workers K, number of iterations T, learning rate η;
Initialization: initial parameter $\mathbf{w}_0$, waiting set $\mathcal{C} = \emptyset$;
Send the initial parameter $\mathbf{w}_0$ to all workers;
for $t = 0$ **to** $T - 1$ **do**
  Receive a stochastic gradient $\mathbf{g}^{k_t}_{ite(k_t,t)}$ from some worker k_t;
  Update the parameter $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \mathbf{g}^{k_t}_{ite(k_t,t)}$;
  Add the worker k_t to the waiting set $\mathcal{C} = \mathcal{C} \cup \{k_t\}$;
  Execute the communication scheduler:
  Option I: (Synchronous) only when all the workers are in the waiting set, i.e., $\mathcal{C} = [K]$, send the parameter $\mathbf{w}_{t+1}$ to the workers in $\mathcal{C}$ and set $\mathcal{C}$ to $\emptyset$;
  Option II: (Asynchronous) once the waiting set is not empty, i.e., $\mathcal{C} \neq \emptyset$, immediately send the parameter $\mathbf{w}_{t+1}$ to the worker in $\mathcal{C}$ and set $\mathcal{C}$ to $\emptyset$;
end for
Notify all workers to stop;
Worker k : $(k \in [K])$
repeat
  Wait until receiving the parameter $\mathbf{w}$ from the server;
  Randomly sample $\xi^k \sim \mathcal{D}$ and then compute the stochastic gradient $\mathbf{g}^k = \nabla f(\mathbf{w}; \xi^k)$;
  Send the stochastic gradient $\mathbf{g}^k$ to the server;
until receive server's notification to stop

In ASGD, the latest parameter $\mathbf{w}_{t+1}$ will be immediately sent back to the worker after the server updates the parameter at each iteration. The function $ite(k,t)$ in ASGD can be formulated as follows:

$$ite(k,t) = \begin{cases} 0 & t = 0, k \in [K], \\ t & t > 0, k = k_{t-1}, \\ ite(k, t-1) & t > 0, k \neq k_{t-1}, \end{cases}$$

where $k \in [K], t \in [T]$.

In SSGD, there is a barrier in the synchronous communication scheduler since the latest parameter $\mathbf{w}_{t+1}$ will be sent back to the workers only when all the workers are in the waiting set. The function $ite(k,t)$ in SSGD can be formulated as $ite(k,t) = \lfloor \frac{t}{K} \rfloor K$, where $k \in [K], t \in [T]$ and $\lfloor \cdot \rfloor$ is the floor function.

Remark 1. *In existing works [10, 22], SSGD is often presented in the form of mini-batch SGD:*

$$\tilde{\mathbf{w}}_{s+1} = \tilde{\mathbf{w}}_s - \frac{\tilde{\eta}}{K} \sum_{k \in [K]} \nabla f(\tilde{\mathbf{w}}_s; \xi^k), \tag{2}$$

where $s \in [S]$ and S denotes the number of iterations. Here, all workers aggregate their stochastic gradients to obtain the mini-batch gradient $\frac{1}{K} \sum_{k \in [K]} \nabla f(\tilde{\mathbf{w}}_s; \xi^k)$, which is then used to update the parameter. To unify SSGD and ASGD into a single framework in Algorithm 1, we reformulate SSGD in the form of mini-batch SGD in (2). Specifically, one update using a mini-batch gradient computed over K training instances in (2) is split into K updates, each using a stochastic gradient over a single training instance. Letting $\eta = \frac{\tilde{\eta}}{K}, T = KS$ and $\mathbf{w}_0 = \tilde{\mathbf{w}}_0$, the sequence $\{\mathbf{w}_{sK}\}_{s \in [S]}$ in SSGD in Algorithm 1 matches $\{\tilde{\mathbf{w}}_s\}_{s \in [S]}$ in (2).

3 Ordered Momentum

In this section, we first propose a new reformulation of SSGD with momentum, which inspires the design of ordered momentum (OrMo) for ASGD. Then, we present the details of OrMo, including the algorithm and convergence analysis.

3.1 Reformulation of SSGD with Momentum

The widely used SGD with momentum (SGDm) [26] can be expressed as follows:

$$\tilde{\mathbf{w}}_{s+1} = \tilde{\mathbf{w}}_s - \beta \tilde{\mathbf{u}}_s - \frac{\tilde{\eta}}{|\mathcal{B}_s|} \sum_{\xi \in \mathcal{B}_s} \nabla f\left(\tilde{\mathbf{w}}_s; \xi\right), \tag{3}$$

$$\tilde{\mathbf{u}}_{s+1} = \beta \tilde{\mathbf{u}}_s + \frac{\tilde{\eta}}{|\mathcal{B}_s|} \sum_{\xi \in \mathcal{B}_s} \nabla f\left(\tilde{\mathbf{w}}_s; \xi\right), \tag{4}$$

where $\tilde{\mathbf{u}}_0 = \mathbf{0}, \beta \in [0, 1), s \in [S]$ and S denotes the number of iterations. β is the momentum coefficient. $\tilde{\mathbf{u}}_s$ represents the Polyak's momentum. $\frac{1}{|\mathcal{B}_s|} \sum_{\xi \in \mathcal{B}_s} \nabla f\left(\tilde{\mathbf{w}}_s; \xi\right)$ denotes the stochastic gradient computed over the sampled training instance set $\mathcal{B}_s$, which contains either a single training instance or a mini-batch of training instances sampled from $\mathcal{D}$. (3) denotes the parameter update step and (4) denotes the momentum update step. When $\beta = 0$, SGDm degenerates to (mini-batch) SGD.

Since SSGD can be presented in the form of mini-batch SGD as depicted in Remark 1, it's straightforward to implement SSGD with momentum (SSGDm) as follows:

$$\begin{aligned} \tilde{\mathbf{w}}_{s+1} &= \tilde{\mathbf{w}}_s - \beta \tilde{\mathbf{u}}_s - \frac{\tilde{\eta}}{K} \sum_{k \in [K]} \nabla f(\tilde{\mathbf{w}}_s; \xi^k), \\ \tilde{\mathbf{u}}_{s+1} &= \beta \tilde{\mathbf{u}}_s + \frac{\tilde{\eta}}{K} \sum_{k \in [K]} \nabla f(\tilde{\mathbf{w}}_s; \xi^k), \end{aligned} \tag{5}$$

where $\tilde{\mathbf{u}}_0 = \mathbf{0}, \beta \in [0, 1), s \in [S]$ and S denotes the number of iterations. Here, the server aggregates the stochastic gradients from all the workers to obtain the mini-batch gradient $\frac{1}{K} \sum_{k \in [K]} \nabla f(\tilde{\mathbf{w}}_s; \xi^k)$, which is then used to update both the parameter and the momentum in (5).

To gain insights from SSGDm on incorporating momentum into ASGD, we reformulate SSGDm in (5) to fit into the framework of Algorithm 1. Similar to the reformulation in Remark 1, the updates using a mini-batch gradient computed over K training instances in (5) are split into K updates, each using a stochastic gradient over a single training instance. The corresponding implementation details of SSGDm are presented in Algorithm 3 in Appendix B. In this way, the update rules of SSGDm in (5) can be reformulated as follows:

$$\begin{aligned} \mathbf{w}_{t+\frac{1}{2}} &= \begin{cases} \mathbf{w}_t - \beta \mathbf{u}_t & K \mid t, \\ \mathbf{w}_t & K \nmid t, \end{cases} \\ \mathbf{u}_{t+\frac{1}{2}} &= \begin{cases} \beta \mathbf{u}_t & K \mid t, \\ \mathbf{u}_t & K \nmid t, \end{cases} \\ \mathbf{w}_{t+1} &= \mathbf{w}_{t+\frac{1}{2}} - \eta \mathbf{g}^{k_t}_{\lfloor \frac{t}{K} \rfloor K}, \\ \mathbf{u}_{t+1} &= \mathbf{u}_{t+\frac{1}{2}} + \eta \mathbf{g}^{k_t}_{\lfloor \frac{t}{K} \rfloor K}, \end{aligned} \tag{6}$$

where $\mathbf{u}_0 = \mathbf{0}, \mathbf{g}^{k_t}_{\lfloor \frac{t}{K} \rfloor K} = \nabla f(\mathbf{w}_{\lfloor \frac{t}{K} \rfloor K}; \xi^{k_t})$ and $t \in [T]$. We give the following proposition about the relationship between the sequences in (5) and those in (6). The proof details can be found in Appendix C.1.1.

Proposition 1. *Letting $\eta = \frac{\tilde{\eta}}{K}, T = KS$ and $\mathbf{w}_0 = \tilde{\mathbf{w}}_0$, the sequences $\{\mathbf{w}_{sK}\}_{s \in [S]}$ and $\{\mathbf{u}_{sK}\}_{s \in [S]}$ in (6) are equivalent to $\{\tilde{\mathbf{w}}_s\}_{s \in [S]}$ and $\{\tilde{\mathbf{u}}_s\}_{s \in [S]}$ in (5), respectively.*

We investigate how the momentum term $\mathbf{u}_{t+1}$ evolves during the iterations in (6). For $t \geq K$ and $t \in [T]$, $\mathbf{u}_{t+1}$ can be formulated as:

$$\mathbf{u}_{t+1} = \sum_{i=0}^{\lfloor \frac{t}{K} \rfloor - 1} \left(\beta^{\lfloor \frac{t}{K} \rfloor - i} \times \sum_{k \in [K]} \eta \mathbf{g}^k_{iK} \right) + \beta^0 \times \sum_{j=\lfloor \frac{t}{K} \rfloor K}^{t} \eta \mathbf{g}^{k_j}_{\lfloor \frac{t}{K} \rfloor K},$$

where the superscript of the scalar β indicates the exponent. For $t < K$, $\mathbf{u}_{t+1} = \beta^0 \times \sum_{j=0}^{t} \eta \mathbf{g}^{k_j}_0$. Figure 1 shows $\mathbf{u}_{10}$ as an example when $K = 4$. We define $\{\eta \mathbf{g}^0_{iK}, \eta \mathbf{g}^1_{iK}, \cdots, \eta \mathbf{g}^{K-1}_{iK}\}$ as the i-th

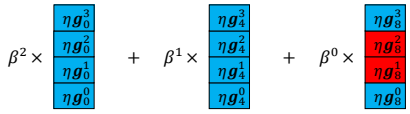


Figure 1: An example of the momentum term $\mathbf{u}_{10}$ in SSGDm when $K = 4$. The gradients shown in red indicate those having not arrived at the server. In this case, $\mathbf{u}_{10} = \beta^2 \times \left(\eta \mathbf{g}_0^0 + \eta \mathbf{g}_0^1 + \eta \mathbf{g}_0^2 + \eta \mathbf{g}_0^3\right) + \beta^1 \times \left(\eta \mathbf{g}_4^0 + \eta \mathbf{g}_4^1 + \eta \mathbf{g}_4^2 + \eta \mathbf{g}_4^3\right) + \beta^0 \times \left(\eta \mathbf{g}_8^0 + \eta \mathbf{g}_8^3\right)$.

(scaled) gradient group, which contains K gradients scaled by the learning rate η. The order of the gradient groups is based on the iteration indexes of their corresponding gradients. Though some gradients may be missing because they have not yet arrived at the server, the momentum is a weighted sum of the gradients from the first several gradient groups. Hence, the momentum in SSGDm is referred to as an *ordered momentum*. Specifically, the gradients in the i-th gradient group are weighted by $\beta^{\lfloor \frac{t}{K} \rfloor - i}$ in the momentum $\mathbf{u}_{t+1}$, where $i \in [\lfloor \frac{t}{K} \rfloor + 1]$. We refer to the gradient group whose gradients are weighted by β^0 as the latest gradient group, which contains the latest gradients. For $\mathbf{u}_{t+1}$ in SSGDm, the latest gradient group corresponds to the $\lfloor \frac{t}{K} \rfloor$-th gradient group.

Due to the barrier in the synchronous communication scheduler in SSGDm as presented in Algorithm 3, the gradients in SSGDm consistently arrive at the server in the order of their iteration indexes. The arriving gradient always belongs to the latest gradient group at each iteration. Thus, maintaining such an ordered momentum in SSGDm is straightforward. As shown in line 13 of Algorithm 3, the scaled gradient $\eta \mathbf{g}_{ite(k_t,t)}^{k_t}$ is always added to the momentum with a weight of β^0 at each iteration. However, for ASGD, since the gradients arrive at the server out of order, it's not trivial to incorporate such an ordered momentum. To address this problem, we propose a solution in the following subsection.

3.2 OrMo for ASGD

In this subsection, we introduce our novel method called ordered momentum (OrMo) for ASGD, and present it in Algorithm 2.

Firstly, we define the (scaled) gradient groups in OrMo for ASGD. Due to the differences in the communication scheduler, the iteration indexes of the parameters used to compute the gradients in ASGD differ from those in SSGD (SSGDm). Specifically, the sequence of gradients computed in SSGD (SSGDm) can be formulated as:

$$\mathbf{g}_0^0, \mathbf{g}_0^1, \cdots, \mathbf{g}_0^{K-1}, \mathbf{g}_K^0, \mathbf{g}_K^1, \cdots, \mathbf{g}_K^{K-1}, \mathbf{g}_{2K}^0, \mathbf{g}_{2K}^1, \cdots, \mathbf{g}_{2K}^{K-1}, \cdots. \tag{7}$$

In contrast, the sequence of gradients computed in ASGD is given by:

$$\mathbf{g}_0^0, \mathbf{g}_0^1, \cdots, \mathbf{g}_0^{K-1}, \mathbf{g}_1^{k_0}, \mathbf{g}_2^{k_1}, \cdots, \mathbf{g}_K^{k_{K-1}}, \mathbf{g}_{K+1}^{k_K}, \mathbf{g}_{K+2}^{k_{K+1}}, \cdots, \mathbf{g}_{2K}^{k_{2K-1}}, \cdots. \tag{8}$$

Thus, the i-th (scaled) gradient group in OrMo for ASGD is defined as:

$$\left\{\eta \mathbf{g}_{(i-1)K+1}^{k_{(i-1)K}}, \eta \mathbf{g}_{(i-1)K+2}^{k_{(i-1)K+1}}, \cdots, \eta \mathbf{g}_{iK}^{k_{iK-1}}\right\},$$

where $i \geq 1$. The 0-th (scaled) gradient group in OrMo is $\{\eta \mathbf{g}_0^0, \eta \mathbf{g}_0^1, \cdots, \eta \mathbf{g}_0^{K-1}\}$. Despite the difference in the gradients' iteration indexes, each gradient group in OrMo for ASGD also contains K gradients scaled by the learning rate η, similar to that in SSGDm as discussed in Subsection 3.1.

We use I_{t+1} to denote the index of the latest gradient group of $\mathbf{u}_{t+1}$ in OrMo. The iteration index of the latest gradient in $\mathbf{u}_{t+1}$ can be t at most. Since the gradient with iteration index t belongs to the $\lceil \frac{t}{K} \rceil$-th gradient group, the latest gradient group for $\mathbf{u}_{t+1}$ should be the $\lceil \frac{t}{K} \rceil$-th gradient group, i.e., $I_{t+1} \equiv \lceil \frac{t}{K} \rceil, \forall t \in [T]$. $\mathbf{u}_{t+1}$ is the weighted sum of the gradients from the first $I_{t+1} + 1$ gradient groups, where some gradients may be missing because they have not yet arrived at the server. The gradients in the i-th gradient group are weighted by $\beta^{\lceil \frac{t}{K} \rceil - i}$ in the momentum $\mathbf{u}_{t+1}$, where $i \in [I_{t+1} + 1]$ and the superscript of the scalar β indicates the exponent. Figure 2 shows an example of $\mathbf{u}_{10}$ in OrMo when $K = 4$.

For the t-th iteration in OrMo, the server performs the following operations:

Algorithm 2 OrMo

Server:
Input: number of workers K, number of iterations T, learning rate η, momentum coefficient $\beta \in [0, 1)$;
Initialization: initial parameter $\mathbf{w}_0$, momentum $\mathbf{u}_0 = \mathbf{0}$, index of the latest gradient group $I_0 = 0$, waiting set $\mathcal{C} = \emptyset$;
Send the initial parameter $\mathbf{w}_0$ and its iteration index 0 to all workers;
for $t = 0$ **to** $T - 1$ **do**
  if the waiting set $\mathcal{C}$ is empty and $\lceil \frac{t}{K} \rceil > I_t$ **then**
    $\mathbf{w}_{t+\frac{1}{2}} = \mathbf{w}_t - \beta \mathbf{u}_t$, $\mathbf{u}_{t+\frac{1}{2}} = \beta \mathbf{u}_t$, $I_{t+1} = I_t + 1$;
  else
    $\mathbf{w}_{t+\frac{1}{2}} = \mathbf{w}_t$, $\mathbf{u}_{t+\frac{1}{2}} = \mathbf{u}_t$, $I_{t+1} = I_t$;
  end if
  Receive a stochastic gradient $\mathbf{g}^{k_t}_{ite(k_t,t)}$ and its iteration index $ite(k_t, t)$ from some worker k_t and then calculate $\lceil \frac{ite(k_t,t)}{K} \rceil$ (i.e., the index of the gradient group that $\mathbf{g}^{k_t}_{ite(k_t,t)}$ belongs to);
  Update the momentum $\mathbf{u}_{t+1} = \mathbf{u}_{t+\frac{1}{2}} + \beta^{I_{t+1} - \lceil \frac{ite(k_t,t)}{K} \rceil} \times \left(\eta \mathbf{g}^{k_t}_{ite(k_t,t)} \right)$;
  Update the parameter $\mathbf{w}_{t+1} = \mathbf{w}_{t+\frac{1}{2}} - \frac{1 - \beta^{I_{t+1} - \lceil \frac{ite(k_t,t)}{K} \rceil + 1}}{1 - \beta} \times \left(\eta \mathbf{g}^{k_t}_{ite(k_t,t)} \right)$;
  Add the worker k_t to the waiting set $\mathcal{C} = \mathcal{C} \cup \{k_t\}$;
  Execute the asynchronous communication scheduler: once the waiting set is not empty, i.e., $\mathcal{C} \neq \emptyset$, immediately send the parameter $\mathbf{w}_{t+1}$ and its iteration index $t + 1$ to the worker in $\mathcal{C}$ and set $\mathcal{C}$ to $\emptyset$;
end for
Notify all workers to stop;
Worker k : $(k \in [K])$
repeat
  Wait until receiving the parameter $\mathbf{w}_{t'}$ and its iteration index t' from the server;
  Randomly sample $\xi^k \sim \mathcal{D}$ and then compute the stochastic gradient $\mathbf{g}^k_{t'} = \nabla f(\mathbf{w}_{t'}; \xi^k)$;
  Send the stochastic gradient $\mathbf{g}^k_{t'}$ and its iteration index t' to the server;
until receive server's notification to stop

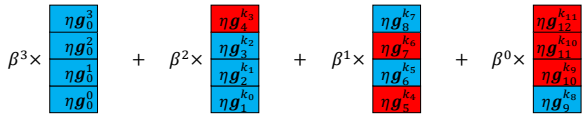


Figure 2: An example of the momentum term $\mathbf{u}_{10}$ in OrMo when $K = 4$. The gradients shown in red indicate those having not arrived at the server. In this case, $\mathbf{u}_{10} = \beta^3 \times \left(\eta \mathbf{g}^0_0 + \eta \mathbf{g}^1_0 + \eta \mathbf{g}^2_0 + \eta \mathbf{g}^3_0 \right) + \beta^2 \times \left(\eta \mathbf{g}^{k_0}_1 + \eta \mathbf{g}^{k_1}_2 + \eta \mathbf{g}^{k_2}_3 \right) + \beta^1 \times \left(\eta \mathbf{g}^{k_5}_6 + \eta \mathbf{g}^{k_7}_8 \right) + \beta^0 \times \left(\eta \mathbf{g}^{k_8}_9 \right)$.

- If the parameter with iteration index t that satisfies $\lceil \frac{t}{K} \rceil > I_t$ has been sent to some worker, update the parameter using the momentum and multiply the momentum with β: $\mathbf{w}_{t+\frac{1}{2}} = \mathbf{w}_t - \beta \mathbf{u}_t, \mathbf{u}_{t+\frac{1}{2}} = \beta \mathbf{u}_t, I_{t+1} = I_t + 1$.

  In this way, the momentum changes the index of its latest gradient group to $\lceil \frac{t}{K} \rceil$ and gets ready to accommodate the new gradient with iteration index t.

- Receive a stochastic gradient $\mathbf{g}^{k_t}_{ite(k_t,t)}$ and its iteration index $ite(k_t, t)$ from some worker k_t and calculate $\lceil \frac{ite(k_t,t)}{K} \rceil$, which is the index of the gradient group that $\mathbf{g}^{k_t}_{ite(k_t,t)}$ belongs to.

- Update the momentum: $\mathbf{u}_{t+1} = \mathbf{u}_{t+\frac{1}{2}} + \beta^{I_{t+1} - \lceil \frac{ite(k_t,t)}{K} \rceil} \times \left(\eta \mathbf{g}^{k_t}_{ite(k_t,t)} \right)$.

  Since the weight of the scaled gradients from the latest gradient group in the momentum is β^0, the weight of the gradients from the $\lceil \frac{ite(k_t,t)}{K} \rceil$-th gradient group should be $\beta^{I_{t+1} - \lceil \frac{ite(k_t,t)}{K} \rceil}$.

OrMo updates the momentum by adding the scaled gradient $\eta \mathbf{g}_{ite(k_t,t)}^{k_t}$ into the momentum with a weight of $\beta^{I_{t+1}-\lceil \frac{ite(k_t,t)}{K} \rceil}$.

- Update the parameter: $\mathbf{w}_{t+1} = \mathbf{w}_{t+\frac{1}{2}} - \frac{1-\beta^{I_{t+1}-\lceil \frac{ite(k_t,t)}{K} \rceil+1}}{1-\beta} \times \left(\eta \mathbf{g}_{ite(k_t,t)}^{k_t}\right)$.

  The update rule of the parameter in OrMo is motivated by that in SSGDm, as presented in Algorithm 3. In SSGDm, the scaled gradient $\eta \mathbf{g}_{ite(k_t,t)}^{k_t}$ is always added to the momentum with a weight of β^0. At the current iteration, this scaled gradient updates the parameter with a coefficient $-\beta^0$. In subsequent iterations, this scaled gradient in the momentum updates the parameter with the coefficients $-\beta, -\beta^2, -\beta^3, \cdots$. By the time this scaled gradient is weighted by $\beta^{I_{t+1}-\lceil \frac{ite(k_t,t)}{K} \rceil}$ in the momentum of SSGDm, it has already updated the parameter for $I_{t+1} - \lceil \frac{ite(k_t,t)}{K} \rceil + 1$ steps, with a total coefficient $-\sum_{j=0}^{I_{t+1}-\lceil \frac{ite(k_t,t)}{K} \rceil} \beta^j$. In OrMo, for the scaled gradient $\eta \mathbf{g}_{ite(k_t,t)}^{k_t}$ which is added to the momentum with a weight of $\beta^{I_{t+1}-\lceil \frac{ite(k_t,t)}{K} \rceil}$, we compensate for the missed $I_{t+1} - \lceil \frac{ite(k_t,t)}{K} \rceil + 1$ steps compared with SSGDm and update the parameter with the coefficient $-\frac{1-\beta^{I_{t+1}-\lceil \frac{ite(k_t,t)}{K} \rceil+1}}{1-\beta}$ at the current iteration. The design of the parameter update rule is crucial for the derivation of Lemma 2, which is further supported by the ablation study in Appendix A.3.
- Add the worker k_t to the waiting set $\mathcal{C} = \mathcal{C} \cup \{k_t\}$ and execute the asynchronous communication scheduler.

Remark 2. *Compared to ASGD, the additional communication overhead introduced by the iteration index in OrMo is negligible since the iteration index is only a scalar.*

Remark 3. *When the momentum coefficient β is set to 0, OrMo degenerates to ASGD in Algorithm 1. If the asynchronous communication scheduler in line 15 of Algorithm 2 is replaced by a synchronous communication scheduler: only when all the workers are in the waiting set, i.e., $\mathcal{C} = [K]$, send the parameter $\mathbf{w}_{t+1}$ and the iteration index $t+1$ to the workers in $\mathcal{C}$ and set $\mathcal{C}$ to $\emptyset$, OrMo degenerates to SSGDm in Algorithm 3.*

3.3 Convergence Analysis

In this section, we prove the convergence of OrMo in Algorithm 2 for non-convex problems. We only present the main results here. The proof details can be found in Appendix C.

We make the following assumptions, which are widely used in distributed learning [47, 43, 41, 22].

Assumption 1. *For any stochastic gradient $\nabla f(\mathbf{w};\xi)$, we assume that it satisfies:*

$$\mathbb{E}_{\xi\sim\mathcal{D}}\left[\nabla f(\mathbf{w};\xi)\right] = \nabla F(\mathbf{w}), \mathbb{E}_{\xi\sim\mathcal{D}}\|\nabla f(\mathbf{w};\xi) - \nabla F(\mathbf{w})\|^2 \leq \sigma^2, \forall \mathbf{w} \in \mathbb{R}^d.$$

Assumption 2. *For any stochastic gradient $\nabla f(\mathbf{w};\xi)$, we assume that it satisfies:*

$$\mathbb{E}_{\xi\sim\mathcal{D}}\|\nabla f(\mathbf{w};\xi)\|^2 \leq G^2, \forall \mathbf{w} \in \mathbb{R}^d.$$

Assumption 3. *$F(\mathbf{w})$ is L-smooth ($L > 0$):*

$$F(\mathbf{w}) \leq F(\mathbf{w}') + \nabla F(\mathbf{w}')^T(\mathbf{w} - \mathbf{w}') + \frac{L}{2}\|\mathbf{w} - \mathbf{w}'\|^2, \forall \mathbf{w}, \mathbf{w}' \in \mathbb{R}^d.$$

Assumption 4. *The objective function $F(\mathbf{w})$ is lower bounded by F^*: $F(\mathbf{w}) \geq F^*, \forall \mathbf{w} \in \mathbb{R}^d$.*

Firstly, we define the auxiliary sequence $\{\hat{\mathbf{u}}_t\}_{t\geq 1}$ for the momentum: $\hat{\mathbf{u}}_1 = \sum_{k\in[K]} \eta \mathbf{g}_0^k$, and

$$\hat{\mathbf{u}}_{t+1} = \begin{cases} \beta \hat{\mathbf{u}}_t + \eta \mathbf{g}_t^{k_{t-1}} & K \mid (t-1), \\ \hat{\mathbf{u}}_t + \eta \mathbf{g}_t^{k_{t-1}} & K \nmid (t-1), \end{cases}$$

for $t \geq 1$.

Lemma 1. *For any $t \geq 0$, the gap between $\mathbf{u}_{t+1}$ and $\hat{\mathbf{u}}_{t+1}$ can be formulated as follows:*

$$\hat{\mathbf{u}}_{t+1} - \mathbf{u}_{t+1} = \sum_{k\in[K], k\neq k_t} \beta^{\lceil \frac{t}{K} \rceil - \lceil \frac{ite(k,t)}{K} \rceil} \eta \mathbf{g}_{ite(k,t)}^k. \tag{9}$$

Then, we define the auxiliary sequence $\{\hat{\mathbf{w}}_t\}_{t\geq 1}$ for the parameter: $\hat{\mathbf{w}}_1 = \mathbf{w}_0 - \sum_{k\in[K]} \eta \mathbf{g}_0^k$, and

$$\hat{\mathbf{w}}_{t+1} = \begin{cases} \hat{\mathbf{w}}_t - \beta \hat{\mathbf{u}}_t - \eta \mathbf{g}_t^{k_{t-1}} & K \mid (t-1), \\ \hat{\mathbf{w}}_t - \eta \mathbf{g}_t^{k_{t-1}} & K \nmid (t-1), \end{cases}$$

for $t \geq 1$.

Lemma 2. *For any $t \geq 0$, the gap between $\mathbf{w}_{t+1}$ and $\hat{\mathbf{w}}_{t+1}$ can be formulated as follows:*

$$\hat{\mathbf{w}}_{t+1} - \mathbf{w}_{t+1} = -\sum_{k\in[K], k\neq k_t} \frac{1-\beta^{\lceil \frac{t}{K} \rceil - \lceil \frac{ite(k,t)}{K} \rceil + 1}}{1-\beta} \eta \mathbf{g}_{ite(k,t)}^k. \tag{10}$$

Then, we define another auxiliary sequence $\{\hat{\mathbf{y}}_t\}_{t\geq 1}$: $\hat{\mathbf{y}}_1 = \frac{\hat{\mathbf{w}}_1 - \beta \mathbf{w}_0}{1-\beta}$, and $\hat{\mathbf{y}}_{t+1} = \hat{\mathbf{y}}_t - \frac{\eta}{1-\beta} \mathbf{g}_t^{k_{t-1}}$, for $t \geq 1$.

Lemma 3. *For any $t \geq 1$, the gap between $\hat{\mathbf{y}}_t$ and $\hat{\mathbf{w}}_t$ can be formulated as follows:*

$$\hat{\mathbf{y}}_t - \hat{\mathbf{w}}_t = -\frac{\beta}{1-\beta} \hat{\mathbf{u}}_t. \tag{11}$$

Theorem 1. *With Assumptions 1, 2, 3 and 4, letting $\eta = \min\{\frac{1-\beta}{2KL}, \frac{(1-\beta)\Delta^{\frac{1}{2}}}{(LT)^{\frac{1}{2}}\sigma}, \frac{(1-\beta)^{\frac{5}{3}}\Delta^{\frac{1}{3}}}{(LKG)^{\frac{2}{3}}T^{\frac{1}{3}}}\}$, Algorithm 2 has the following convergence rate:*

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\|\nabla F(\mathbf{w}_t)\|^2 \leq \mathcal{O}\left(\sqrt{\frac{L\sigma^2}{T}} + \left(\frac{KLG}{T}\right)^{\frac{2}{3}} + \frac{KL}{T}\right),$$

where $\Delta = F(\mathbf{w}_0) - F^*$ and $T \geq K$.

Many works [46, 30, 7, 13, 22] consider delay-adaptive methods for ASGD. The key insight of these methods is to penalize the gradients with large delays and reduce their contribution to the parameter update. OrMo is orthogonal to these delay-adaptive methods. Concretely, we can replace the constant learning rate η in Algorithm 2 with a delay-adaptive learning rate η_t, which is dependent on the delay of the gradient τ_t. Inspired by [13], we adopt the following delay-adaptive learning rate η_t:

$$\eta_t = \begin{cases} \eta & \tau_t \leq 2K, \\ \min\{\eta, \frac{1}{4L\tau_t}\} & \tau_t > 2K. \end{cases}$$

The convergence of OrMo with the above delay-adaptive learning rate (called OrMo-DA) is guaranteed by Theorem 2.

Theorem 2. *With Assumptions 1, 3 and 4, letting $\eta = \min\{\frac{(1-\beta)^2}{8KL}, \sqrt{\frac{(1-\beta)^3\Delta}{TL\sigma^2}}\}$, OrMo-DA has the following convergence rate:*

$$\mathbb{E}\,\|\nabla F(\bar{\mathbf{w}}_T)\|^2 \leq \mathcal{O}\left(\sqrt{\frac{L\sigma^2}{T}} + \frac{KL}{T}\right),$$

where $\Delta = F(\mathbf{w}_0) - F^*$ and $\bar{\mathbf{w}}_T$ is randomly chosen from $\{\mathbf{w}_0, \mathbf{w}_1, \cdots, \mathbf{w}_{T-1}\}$ based on a probability distribution which is related to the delay-adaptive learning rates.

The proof details can be found in Appendix C.3. Compared with Theorem 1, Theorem 2 removes the dependence on the bounded gradient in Assumption 2 and provides a better convergence bound.

Remark 4. *We focus on the scenario where the training instances across all workers are independent and identically distributed (i.i.d.) from $\mathcal{D}$. This scenario commonly appears in the data-center setup for distributed training [5], where all workers have access to the full training dataset. Our analysis for the i.i.d. scenario can also provide insights into the analysis in a non-i.i.d. scenario [22], which will be studied in future work.*

Remark 5. *Most existing theoretical analyses of ASGD [16, 49, 3, 20] rely on the maximum delay τ_{max} (e.g., $\mathcal{O}(\sqrt{\frac{L\sigma^2}{T}} + \frac{\tau_{max}L}{T})$ in [20]), where $\tau_{max} = \max_{t\in[T]} \tau_t$. However, since ASGD can still perform well even when the maximum delay is extremely large ($\tau_{max} \gg K$) in practice, these theoretical analyses don't accurately reflect the true behavior of ASGD. The most closely related works to this work are [13, 22], which analyze ASGD without relying on the maximum delay. But the works in [13, 22] do not consider momentum. To the best of our knowledge, this is the first work to establish the convergence guarantee of ASGD with momentum without relying on the maximum delay.*

Table 1: Empirical results of different methods on CIFAR10 dataset.

Number of Workers	16 (hom.)		64 (hom.)		16 (het.)		64 (het.)	
Methods	Training Loss	Test Accuracy	Training Loss	Test Accuracy	Training Loss	Test Accuracy	Training Loss	Test Accuracy
ASGD	0.06 ± 0.00	89.77 ± 0.11	0.40 ± 0.02	83.14 ± 0.55	0.06 ± 0.00	89.73 ± 0.19	0.38 ± 0.01	83.94 ± 0.21
naive ASGDm	0.20 ± 0.07	88.15 ± 1.70	0.44 ± 0.06	82.39 ± 1.79	0.58 ± 0.86	73.23 ± 31.61	0.78 ± 0.77	68.75 ± 29.51
shifted momentum	0.08 ± 0.01	90.23 ± 0.27	0.38 ± 0.00	83.72 ± 0.29	0.10 ± 0.02	89.95 ± 0.32	0.37 ± 0.01	83.99 ± 0.23
SMEGA^2	0.05 ± 0.01	90.60 ± 0.42	0.23 ± 0.04	86.82 ± 0.69	0.04 ± 0.01	90.88 ± 0.25	0.22 ± 0.07	86.89 ± 1.42
OrMo	0.04 ± 0.01	90.95 ± 0.27	$\mathbf{0.15 \pm 0.02}$	$\mathbf{88.03 \pm 0.28}$	0.04 ± 0.00	91.01 ± 0.10	0.16 ± 0.03	87.76 ± 0.57
OrMo-DA	$\mathbf{0.03 \pm 0.01}$	$\mathbf{91.17 \pm 0.18}$	0.16 ± 0.02	88.03 ± 0.33	$\mathbf{0.03 \pm 0.01}$	$\mathbf{91.28 \pm 0.37}$	$\mathbf{0.15 \pm 0.02}$	$\mathbf{88.08 \pm 0.38}$

Table 2: Empirical results of different methods on CIFAR100 dataset.

Number of Workers	16 (hom.)		64 (hom.)		16 (het.)		64 (het.)	
Methods	Training Loss	Test Accuracy	Training Loss	Test Accuracy	Training Loss	Test Accuracy	Training Loss	Test Accuracy
ASGD	0.51 ± 0.01	66.16 ± 0.36	0.96 ± 0.03	61.61 ± 0.59	0.51 ± 0.01	65.94 ± 0.39	0.95 ± 0.03	61.74 ± 0.30
naive ASGDm	0.54 ± 0.01	65.46 ± 0.20	1.03 ± 0.05	59.96 ± 0.90	0.53 ± 0.00	65.69 ± 0.42	0.97 ± 0.06	61.13 ± 1.02
shifted momentum	0.47 ± 0.01	66.37 ± 0.14	0.82 ± 0.01	63.55 ± 0.32	0.47 ± 0.00	66.28 ± 0.14	0.82 ± 0.04	63.28 ± 0.66
SMEGA^2	0.41 ± 0.00	67.32 ± 0.22	0.69 ± 0.00	64.16 ± 0.12	0.40 ± 0.01	67.29 ± 0.16	0.68 ± 0.02	64.12 ± 0.53
OrMo	0.41 ± 0.01	67.56 ± 0.34	0.56 ± 0.00	65.48 ± 0.17	0.40 ± 0.01	67.71 ± 0.33	0.58 ± 0.02	65.43 ± 0.35
OrMo-DA	$\mathbf{0.40 \pm 0.00}$	$\mathbf{67.72 \pm 0.21}$	$\mathbf{0.56 \pm 0.01}$	$\mathbf{65.79 \pm 0.12}$	$\mathbf{0.04 \pm 0.00}$	$\mathbf{67.82 \pm 0.20}$	$\mathbf{0.57 \pm 0.01}$	$\mathbf{65.82 \pm 0.30}$

4 Experiments

In this section, we evaluate the performance of OrMo, OrMo-DA and other baseline methods. All the experiments are implemented based on the Parameter Server framework [15]. Our distributed platform is conducted with Docker. Each Docker container corresponds to either a server or a worker. All the methods are implemented with PyTorch 1.3.

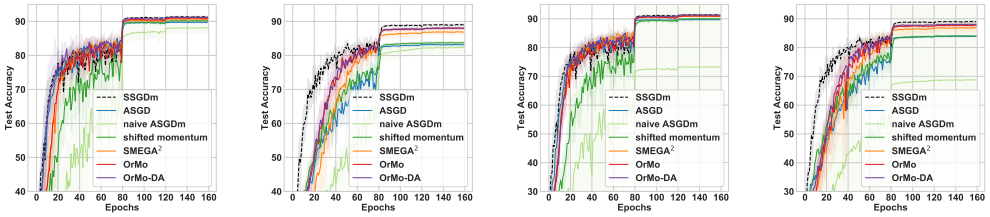
The baseline methods include ASGD, *naive ASGDm* which naively incorporates momentum into ASGD [23], shifted momentum [9] and SMEGA^2 [4]. The details of naive ASGDm are shown in Algorithm 4. In OrMo-DA, for a gradient with a large delay satisfying $\tau_t > 2K$, its corresponding learning rate will be multiplied by $\frac{1}{\tau_t}$. We evaluate these methods by training ResNet20 model [11] on CIFAR10 and CIFAR100 datasets [14]. The number of workers is set to 16 and 64. The batch size on each worker is set to 64. The momentum coefficient is set to 0.9. Each experiment is repeated 5 times. The experiments are conducted under two settings:

- Setting I [homogeneous (hom.)]: each worker has similar computing capabilities, which ensures comparable average time for gradient computations.
- Setting II [heterogeneous (het.)]: some workers ($\frac{1}{16}$ of all) are designated as slow workers, with an average computation time that is 10 times longer than that of the others.

For the CIFAR10 dataset, the weight decay is set to 0.0001 and the model is trained with 160 epochs. The learning rate is multiplied by 0.1 at the 80-th and 120-th epoch, as suggested in [11]. The experiments for the CIFAR10 dataset are conducted on NVIDIA RTX 2080 Ti GPUs. For the CIFAR100 dataset, the weight decay is set to 0.0005 and the model is trained with 200 epochs. The learning rate is multiplied by 0.2 at the 60-th, 120-th and 160-th epoch, as suggested in [40]. The experiments for the CIFAR100 dataset are conducted on NVIDIA V100 GPUs.

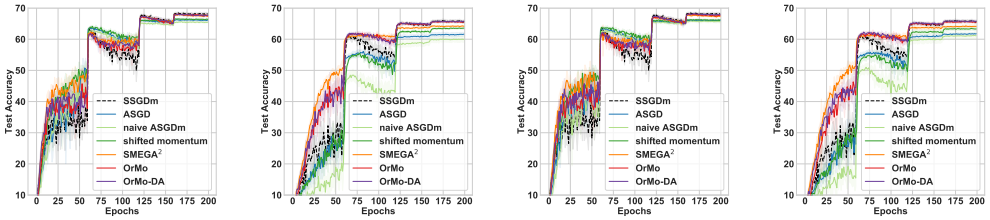
Table 1 and Table 2 show the empirical results of different methods. Figure 3 and Figure 4 show the test accuracy curves of different methods. We also present the test accuracy curves of SSGDm for reference in Figure 3 and Figure 4. Training loss curves can be found in Appendix A.1. Compared with other asynchronous methods, OrMo and OrMo-DA can achieve better training loss and test accuracy. As shown in Figure 3, naive ASGDm occasionally fails to converge under Setting II. We can find that naively incorporating momentum into ASGD will impede its convergence. Due to the existence of slow workers, the maximum delay under Setting II is far greater than that under Setting I. For example, when training on the CIFAR10 dataset with $K = 64$, the maximum delay under Setting II is about 30000, while it's around 300 under Setting I. OrMo and OrMo-DA perform well under both settings, which aligns with our theoretical results without dependence on the maximum delay.

Figure 5 presents the training curves of OrMo and SSGDm with respect to wall-clock time. As shown in Figure 5(b), OrMo can be 8 times faster than SSGDm under Setting II since the training speed of SSGDm is hindered by slow workers. In contrast, slow workers have a limited impact on OrMo's training speed. Even under Setting I where each worker possesses similar computing



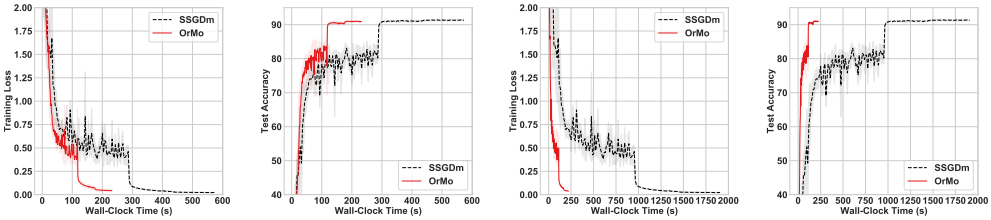
(a) $K = 16$ (hom.) (b) $K = 64$ (hom.) (c) $K = 16$ (het.) (d) $K = 64$ (het.)

Figure 3: Test accuracy curves on CIFAR10 with different numbers of workers.



(a) $K = 16$ (hom.) (b) $K = 64$ (hom.) (c) $K = 16$ (het.) (d) $K = 64$ (het.)

Figure 4: Test accuracy curves on CIFAR100 with different numbers of workers.



(a) homogeneous (b) heterogeneous

Figure 5: Training curves with respect to wall-clock time on CIFAR10 when $K = 16$.

capability, OrMo can still be more than twice as fast as SSGDm, as shown in Figure 5(a). This advantage arises because the computation time of each worker varies within a certain range even under the homogeneous setting and some workers must wait for others to finish gradient computations in SSGDm.

5 Conclusion

In this paper, we propose a novel method named ordered momentum (OrMo) for asynchronous SGD. We theoretically prove the convergence of OrMo with both constant and delay-adaptive learning rates for non-convex problems. To the best of our knowledge, this is the first work to establish the convergence analysis of ASGD with momentum without dependence on the maximum delay. Empirical results demonstrate that OrMo can achieve state-of-the-art performance.

Acknowledgment

This work is supported by National Key R&D Program of China (No. 2020YFA0713900), NSFC Project (No. 12326615), Major Key Project of Pengcheng Laboratory (No. PCL2024A06) and Key R&D Project of Jiangsu Province (No. BE2023652).

References

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek Gordon Murray, Benoit Steiner, Paul A. Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. Tensorflow: A system for large-scale machine learning. In *Proceedings of the USENIX Symposium on Operating Systems Design and Implementation*, pages 265–283, 2016.

[2] Alekh Agarwal and John C. Duchi. Distributed delayed stochastic optimization. In *Advances in Neural Information Processing Systems*, pages 873–881, 2011.

[3] Yossi Arjevani, Ohad Shamir, and Nathan Srebro. A tight convergence analysis for stochastic gradient descent with delayed updates. In *Proceedings of the International Conference on Algorithmic Learning Theory*, pages 111–132, 2020.

[4] Refael Cohen, Ido Hakimi, and Assaf Schuster. SMEGA2: distributed asynchronous deep neural network training with a single momentum buffer. In *Proceedings of the International Conference on Parallel Processing*, pages 1–10, 2022.

[5] Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Quoc V. Le, Mark Z. Mao, Marc'Aurelio Ranzato, Andrew W. Senior, Paul A. Tucker, Ke Yang, and Andrew Y. Ng. Large scale distributed deep networks. In *Advances in Neural Information Processing Systems*, pages 1232–1240, 2012.

[6] Aaron Defazio, Francis R. Bach, and Simon Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems*, pages 1646–1654, 2014.

[7] Sanghamitra Dutta, Gauri Joshi, Soumyadip Ghosh, Parijat Dube, and Priya Nagpurkar. Slow and stale gradients can win the race: Error-runtime trade-offs in distributed SGD. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pages 803–812, 2018.

[8] Hamid Reza Feyzmahdavian, Arda Aytekin, and Mikael Johansson. An asynchronous mini-batch algorithm for regularized stochastic optimization. *IEEE Transactions on Automatic Control*, 61(12):3740–3754, 2016.

[9] Niv Giladi, Mor Shpigel Nacson, Elad Hoffer, and Daniel Soudry. At stability's edge: How to adjust hyperparameters to preserve minima selection in asynchronous training of neural networks? In *Proceedings of the International Conference on Learning Representations*, 2020.

[10] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[12] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, pages 315–323, 2013.

[13] Anastasia Koloskova, Sebastian U. Stich, and Martin Jaggi. Sharper convergence guarantees for asynchronous SGD for distributed and federated learning. In *Advances in Neural Information Processing Systems*, pages 17202–17215, 2022.

[14] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Technical report*, 2009.

[15] Mu Li, David G. Andersen, Alexander J. Smola, and Kai Yu. Communication efficient distributed machine learning with the parameter server. In *Advances in Neural Information Processing Systems*, pages 19–27, 2014.

[16] Xiangru Lian, Yijun Huang, Yuncheng Li, and Ji Liu. Asynchronous parallel stochastic gradient for nonconvex optimization. In *Advances in Neural Information Processing Systems*, pages 2737–2745, 2015.

[17] Xiangru Lian, Wei Zhang, Ce Zhang, and Ji Liu. Asynchronous decentralized parallel stochastic gradient descent. In *Proceedings of the International Conference on Machine Learning*, pages 3043–3052, 2018.

[18] Tao Lin, Sai Praneeth Karimireddy, Sebastian U. Stich, and Martin Jaggi. Quasi-global momentum: Accelerating decentralized deep learning on heterogeneous data. In *Proceedings of the International Conference on Machine Learning*, pages 6654–6665, 2021.

[19] Yujun Lin, Song Han, Huizi Mao, Yu Wang, and Bill Dally. Deep gradient compression: Reducing the communication bandwidth for distributed training. In *Proceedings of the International Conference on Learning Representations*, 2018.

[20] Ji Liu and Ce Zhang. Distributed learning systems with first-order methods. *Foundations and Trends® in Databases*, 9(1):1–100, 2020.

[21] Wei Liu, Li Chen, Yunfei Chen, and Wenyi Zhang. Accelerating federated learning via momentum gradient descent. *IEEE Transactions on Parallel and Distributed Systems*, 31(8):1754–1766, 2020.

[22] Konstantin Mishchenko, Francis R. Bach, Mathieu Even, and Blake E. Woodworth. Asynchronous SGD beats minibatch SGD under arbitrary delays. In *Advances in Neural Information Processing Systems*, pages 420–433, 2022.

[23] Ioannis Mitliagkas, Ce Zhang, Stefan Hadjis, and Christopher Ré. Asynchrony begets momentum, with an application to deep learning. In *Proceedings of the Annual Allerton Conference on Communication, Control, and Computing*, pages 997–1004, 2016.

[24] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*. Springer Science & Business Media, 2013.

[25] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019.

[26] Boris Polyak. Some methods of speeding up the convergence of iteration methods. *Ussr Computational Mathematics and Mathematical Physics*, 4:1–17, 12 1964.

[27] Benjamin Recht, Christopher Ré, Stephen J. Wright, and Feng Niu. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *Advances in Neural Information Processing Systems*, pages 693–701, 2011.

[28] Herbert Robbins and Sutton Monro. A stochastic approximation method. *Annals of Mathematical Statistics*, 22:400–407, 1951.

[29] Chang-Wei Shi, Shen-Yi Zhao, Yin-Peng Xie, Hao Gao, and Wu-Jun Li. Global momentum compression for sparse communication in distributed learning. *arXiv preprint arXiv:1905.12948*, 2019.

[30] Suvrit Sra, Adams Wei Yu, Mu Li, and Alex Smola. Adadelay: Delay adaptive distributed stochastic optimization. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pages 957–965, 2016.

[31] Sebastian U. Stich, Amirkeivan Mohtashami, and Martin Jaggi. Critical parameters for scalable distributed learning with large batches and asynchronous updates. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pages 4042–4050, 2021.

[32] Jianhui Sun, Xidong Wu, Heng Huang, and Aidong Zhang. On the role of server momentum in federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 15164–15172, 2024.

[33] Ilya Sutskever, James Martens, George E. Dahl, and Geoffrey E. Hinton. On the importance of initialization and momentum in deep learning. In *Proceedings of the International Conference on Machine Learning*, pages 1139–1147, 2013.

[34] Hanlin Tang, Shaoduo Gan, Ammar Ahmad Awan, Samyam Rajbhandari, Conglong Li, Xiangru Lian, Ji Liu, Ce Zhang, and Yuxiong He. 1-bit adam: Communication efficient large-scale training with adam's convergence speed. In *Proceedings of the International Conference on Machine Learning*, pages 10118–10129, 2021.

[35] Hanlin Tang, Shaoduo Gan, Ce Zhang, Tong Zhang, and Ji Liu. Communication compression for decentralized training. In *Advances in Neural Information Processing Systems*, pages 7663–7673, 2018.

[36] Paul Tseng. An incremental gradient(-projection) method with momentum term and adaptive stepsize rule. *SIAM Journal on Optimization*, 8(2):506–531, 1998.

[37] Joost Verbraeken, Matthijs Wolting, Jonathan Katzy, Jeroen Kloppenburg, Tim Verbelen, and Jan S Rellermeyer. A survey on distributed machine learning. *ACM Computing Surveys*, 53(2):1–33, 2020.

[38] Thijs Vogels, Sai Praneeth Karimireddy, and Martin Jaggi. Powersgd: Practical low-rank gradient compression for distributed optimization. In *Advances in Neural Information Processing Systems*, pages 14236–14245, 2019.

[39] Jianyu Wang, Vinayak Tantia, Nicolas Ballas, and Michael G. Rabbat. Slowmo: Improving communication-efficient distributed SGD with slow momentum. In *Proceedings of the International Conference on Learning Representations*, 2020.

[40] Cong Xie, Shuai Zheng, Oluwasanmi Koyejo, Indranil Gupta, Mu Li, and Haibin Lin. CSER: Communication-efficient SGD with error reset. In *Advances in Neural Information Processing Systems*, pages 12593–12603, 2020.

[41] An Xu and Heng Huang. Detached error feedback for distributed SGD with random sparsification. In *Proceedings of the International Conference on Machine Learning*, pages 24550–24575, 2022.

[42] Tao Yang, Xinlei Yi, Junfeng Wu, Ye Yuan, Di Wu, Ziyang Meng, Yiguang Hong, Hong Wang, Zongli Lin, and Karl H Johansson. A survey of distributed optimization. *Annual Reviews in Control*, 47:278–305, 2019.

[43] Yi-Rui Yang and Wu-Jun Li. BASGD: buffered asynchronous SGD for byzantine learning. In *Proceedings of the International Conference on Machine Learning*, pages 11751–11761, 2021.

[44] Hao Yu, Rong Jin, and Sen Yang. On the linear speedup analysis of communication efficient momentum SGD for distributed non-convex optimization. In *Proceedings of the International Conference on Machine Learning*, pages 7184–7193, 2019.

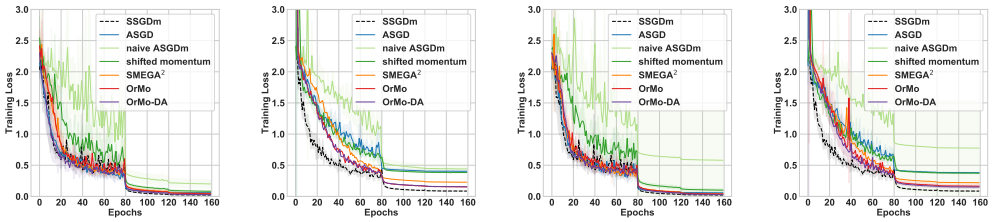
[45] Kun Yuan, Yiming Chen, Xinmeng Huang, Yingya Zhang, Pan Pan, Yinghui Xu, and Wotao Yin. Decentlam: Decentralized momentum sgd for large-batch deep training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3029–3039, 2021.

[46] Wei Zhang, Suyog Gupta, Xiangru Lian, and Ji Liu. Staleness-aware async-sgd for distributed deep learning. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 2350–2356, 2016.

[47] Shen-Yi Zhao and Wu-Jun Li. Fast asynchronous parallel stochastic gradient descent: A lock-free approach with convergence guarantee. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2379–2385, 2016.

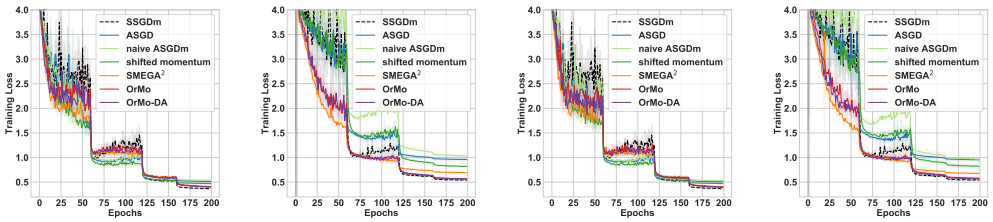
[48] Shen-Yi Zhao, Gong-Duo Zhang, and Wu-Jun Li. Lock-free optimization for non-convex problems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2935–2941, 2017.

[49] Shuxin Zheng, Qi Meng, Taifeng Wang, Wei Chen, Nenghai Yu, Zhiming Ma, and Tie-Yan Liu. Asynchronous stochastic gradient descent with delay compensation. In *Proceedings of the International Conference on Machine Learning*, pages 4120–4129, 2017.



(a) $K = 16$ (hom.) (b) $K = 64$ (hom.) (c) $K = 16$ (het.) (d) $K = 64$ (het.)

Figure 6: Training loss curves of different methods on CIFAR10 dataset with different numbers of workers.



(a) $K = 16$ (hom.) (b) $K = 64$ (hom.) (c) $K = 16$ (het.) (d) $K = 64$ (het.)

Figure 7: Training loss curves of different methods on CIFAR100 dataset with different numbers of workers.

Table 3: Empirical results of naive ASGDm with different β when training ResNet20 on CIFAR10 dataset.

Number of workers	16 (hom.)		64 (hom.)		16 (het.)		64 (het.)	
Algorithm (β)	Training Loss	Test Accuracy	Training Loss	Test Accuracy	Training Loss	Test Accuracy	Training Loss	Test Accuracy
naive ASGDm (0.1)	0.06$\pm$ 0.01	89.85 $\pm$ 0.24	0.38 $\pm$ 0.01	83.93 $\pm$ 0.25	0.06 $\pm$0.00	89.95 $\pm$ 0.19	0.38 $\pm$ 0.01	83.76 $\pm$ 0.34
naive ASGDm (0.3)	0.06 $\pm$0.00	89.91 $\pm$ 0.20	0.36 $\pm$ 0.02	84.23 $\pm$ 0.49	0.05 $\pm$ 0.01	90.26 $\pm$ 0.05	0.35 $\pm$ 0.02	84.43 $\pm$ 0.22
naive ASGDm (0.6)	0.07 $\pm$0.00	90.39 $\pm$ 0.24	0.38 $\pm$0.02	83.87 $\pm$ 0.28	0.06 $\pm$ 0.01	90.56 $\pm$ 0.13	0.37 $\pm$ 0.02	84.07 $\pm$ 0.38
naive ASGDm (0.9)	0.20 $\pm$ 0.07	88.15 $\pm$ 1.70	0.44 $\pm$ 0.06	82.39 $\pm$ 1.79	0.58 $\pm$ 0.86	73.23 $\pm$ 31.61	0.78 $\pm$ 0.77	68.75 $\pm$ 29.51
OrMo (0.9)	**0.04 $\pm$ 0.01**	**90.95 $\pm$ 0.27**	**0.15 $\pm$ 0.02**	**88.03 $\pm$ 0.28**	**0.04 $\pm$ 0.00**	**91.01 $\pm$ 0.10**	**0.16 $\pm$ 0.03**	**87.76 $\pm$ 0.57**

A More Experimental Results

A.1 Loss Curves

Figure 6 and Figure 7 show the training loss curves of ResNet20 model.

A.2 Tuning β for Naive ASGDm

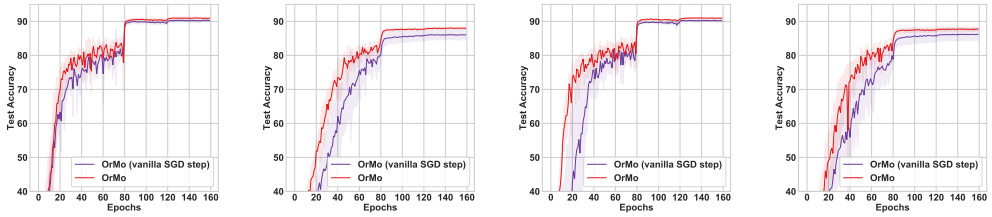
Following the suggestion in [23, 9], we conduct experiments to tune the momentum coefficient β for naive ASGDm and present the results when training ResNet20 on CIFAR10 in Table 3. While tuning the momentum coefficient can enhance the performance of naive ASGDm, hyperparameter tuning is quite time-consuming and costly. In contrast, OrMo achieves better performance using the commonly used momentum value of 0.9, without requiring extensive tuning.

A.3 Ablation Study

An ablation study is also conducted to justify the parameter update rule in line 13 of Algorithm 2. We replace the update rule in line 13 of Algorithm 2 with a vanilla SGD step, $\mathbf{w}_{t+1} = \mathbf{w}_{t+\frac{1}{2}} - \eta \mathbf{g}^{k_t}_{ite(k_t,t)}$, and name it OrMo (vanilla SGD step). The comparison between the experimental results of OrMo and OrMo (vanilla SGD step) are presented in Table 4 and Figure 8.

Table 4: Empirical results of OrMo and OrMo (vanilla SGD step) when training ResNet20 on CIFAR10 dataset.

Number of Workers	16 (hom.)		64 (hom.)		16 (het.)		64 (het.)	
Methods	Training Loss	Test Accuracy	Training Loss	Test Accuracy	Training Loss	Test Accuracy	Training Loss	Test Accuracy
OrMo (vanilla SGD step)	0.07 ± 0.02	90.32 ± 0.45	0.27 ± 0.07	86.08 ± 1.33	0.07 ± 0.01	90.23 ± 0.32	0.27 ± 0.07	86.10 ± 1.71
OrMo	$\mathbf{0.04 \pm 0.01}$	$\mathbf{90.95 \pm 0.27}$	$\mathbf{0.15 \pm 0.02}$	$\mathbf{88.03 \pm 0.28}$	$\mathbf{0.04 \pm 0.00}$	$\mathbf{91.01 \pm 0.10}$	$\mathbf{0.16 \pm 0.03}$	$\mathbf{87.76 \pm 0.57}$

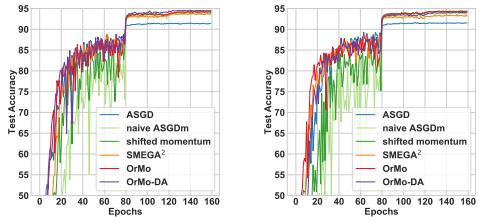


(a) $K = 16$ (hom.) (b) $K = 64$ (hom.) (c) $K = 16$ (het.) (d) $K = 64$ (het.)

Figure 8: Test accuracy curves when training ResNet20 model on CIFAR10 dataset with different numbers of worker number.

Table 5: Test accuracy of different methods when training ResNet18 on CIFAR10 dataset.

	homogeneous	heterogeneous
ASGD	91.45	91.52
naive ASGDm	93.74	93.10
shifted momentum	94.02	94.20
SMEGA^2	93.72	93.36
OrMo	94.32	**94.26**
OrMo-DA	**94.50**	94.03



(a) homogeneous (b) heterogeneous

Figure 9: Test accuracy curves when training ResNet18 model on CIFAR10 dataset.

A.4 Experimental Results on ResNet18 Model

Table 5 and Figure 9 show the performance of different methods when training the ResNet18 model on the CIFAR10 dataset. The number of workers is set to 8. In the homogeneous setting, each worker has similar computing capabilities. In the heterogeneous setting, one worker is designated as the slow worker, whose average computation time is 10 times longer than that of the others.

B Algorithm Details

Algorithm 3 and Algorithm 4 show the details of SSGDm and naive ASGDm.

Algorithm 3 SSGDm

Server:
Input: number of workers K, number of iterations T, learning rate η, momentum coefficient $\beta \in [0, 1)$;
Initialization: initial parameter $\mathbf{w}_0$, momentum $\mathbf{u}_0 = \mathbf{0}$, waiting set $\mathcal{C} = \emptyset$;
Send the initial parameter $\mathbf{w}_0$ to all workers;
for $t = 0$ **to** $T - 1$ **do**
  if the waiting set $\mathcal{C}$ is empty **then**
    $\mathbf{w}_{t+\frac{1}{2}} = \mathbf{w}_t - \beta \mathbf{u}_t$, $\mathbf{u}_{t+\frac{1}{2}} = \beta \mathbf{u}_t$;
  else
    $\mathbf{w}_{t+\frac{1}{2}} = \mathbf{w}_t$, $\mathbf{u}_{t+\frac{1}{2}} = \mathbf{u}_t$;
  end if
  Receive a stochastic gradient $\mathbf{g}^{k_t}_{ite(k_t,t)}$ from some worker k_t;
  Update the parameter $\mathbf{w}_{t+1} = \mathbf{w}_{t+\frac{1}{2}} - \eta \mathbf{g}^{k_t}_{ite(k_t,t)}$;
  Update the momentum $\mathbf{u}_{t+1} = \mathbf{u}_{t+\frac{1}{2}} + \eta \mathbf{g}^{k_t}_{ite(k_t,t)}$;
  Add the worker k_t to the waiting set $\mathcal{C} = \mathcal{C} \cup \{k_t\}$;
  Execute the synchronous communication scheduler: only when all the workers are in the waiting set, i.e., $\mathcal{C} = [K]$, send the parameter $\mathbf{w}_{t+1}$ to the workers in $\mathcal{C}$ and set $\mathcal{C}$ to $\emptyset$;
end for
Notify all workers to stop;
Worker k **:** $(k \in [K])$
repeat
  Wait until receiving the parameter $\mathbf{w}$ from the server;
  Randomly sample $\xi^k \sim \mathcal{D}$ and then compute the stochastic gradient $\mathbf{g}^k = \nabla f(\mathbf{w}; \xi^k)$;
  Send the stochastic gradient $\mathbf{g}^k$ to the server;
until receive server's notification to stop

Algorithm 4 naive ASGDm

Server:
Input: number of workers K, number of iterations T, learning rate η, momentum coefficient $\beta \in [0, 1)$;
Initialization: initial parameter $\mathbf{w}_0$, momentum $\mathbf{u}_0 = \mathbf{0}$, waiting set $\mathcal{C} = \emptyset$;
Send the initial parameter $\mathbf{w}_0$ to all workers;
for $t = 0$ **to** $T - 1$ **do**
  Receive a stochastic gradient $\mathbf{g}^{k_t}_{ite(k_t,t)}$ from some worker k_t;
  Update the momentum $\mathbf{u}_{t+1} = \beta \mathbf{u}_t + \eta \mathbf{g}^{k_t}_{ite(k_t,t)}$
  Update the parameter $\mathbf{w}_{t+1} = \mathbf{w}_t - \mathbf{u}_{t+1}$
  Add the worker k_t to the waiting set $\mathcal{C} = \mathcal{C} \cup \{k_t\}$;
  Execute the asynchronous communication scheduler: once the waiting set is not empty, i.e., $\mathcal{C} \neq \emptyset$, immediately send the parameter $\mathbf{w}_{t+1}$ to the worker in $\mathcal{C}$ and set $\mathcal{C}$ to $\emptyset$;
end for
Notify all workers to stop;
Worker k **:** $(k \in [K])$
repeat
  Wait until receiving the parameter $\mathbf{w}$ from the server;
  Randomly sample $\xi^k \sim \mathcal{D}$ and then compute the stochastic gradient $\mathbf{g}^k = \nabla f(\mathbf{w}; \xi^k)$;
  Send the stochastic gradient $\mathbf{g}^k$ to the server;
until receive server's notification to stop

C Proof Details

C.1 Reformulation of SSGDm

C.1.1 Proof of Proposition 1

Proof. It's easy to verify that $\{k_t, k_{t+1}, \cdots, k_{t+K-1}\} = [K]$ in (6), where $K \mid t$.

Base case: for s = 0, $\tilde{\mathbf{w}}_0 = \mathbf{w}_0$ and $\tilde{\mathbf{u}}_0 = \mathbf{u}_0$.

Inductive hypothesis: for some arbitrary integer $s' \geq 0$, assume that $\tilde{\mathbf{w}}_{s'} = \mathbf{w}_{s'K}, \tilde{\mathbf{u}}_{s'} = \mathbf{u}_{s'K}$.

Inductive step:

$$\tilde{\mathbf{u}}_{s'+1} = \beta\tilde{\mathbf{u}}_{s'} + \frac{\tilde{\eta}}{K}\sum_{k\in[K]}\nabla f(\tilde{\mathbf{w}}_{s'};\xi^k) = \beta\mathbf{u}_{s'K} + \eta\sum_{k\in[K]}\nabla f(\mathbf{w}_{s'K};\xi^k) = \mathbf{u}_{(s'+1)K},$$

$$\begin{aligned}\tilde{\mathbf{w}}_{s'+1} &= \tilde{\mathbf{w}}_{s'} - \beta\tilde{\mathbf{u}}_{s'} - \frac{\tilde{\eta}}{K}\sum_{k\in[K]}\nabla f(\tilde{\mathbf{w}}_{s'};\xi^k) = \mathbf{w}_{s'K} - \beta\mathbf{u}_{s'K} - \eta\sum_{k\in[K]}\nabla f(\mathbf{w}_{s'K};\xi^k)\\ &= \mathbf{w}_{(s'+1)K}.\end{aligned}$$

We can conclude that $\tilde{\mathbf{w}}_s = \mathbf{w}_{sK}$ and $\tilde{\mathbf{u}}_s = \mathbf{u}_{sK}$ for any $s \in [S]$. □

C.2 OrMo

C.2.1 Proof of Lemma 1

Lemma 1 can be viewed as a special case of Lemma 5, and its proof is completed by substituting each delay-adaptive learning rate $\hat{\eta}_{k,t}$ in the proof of Lemma 5 with η for all $k \in [K]$ and $t \in [T]$.

C.2.2 Proof of Lemma 2

Lemma 2 can be viewed as a special case of Lemma 6, and its proof is completed by substituting each delay-adaptive learning rate $\hat{\eta}_{k,t}$ in the proof of Lemma 6 with η for all $k \in [K]$ and $t \in [T]$.

C.2.3 Proof of Lemma 3

Lemma 3 can be viewed as a special case of Lemma 7, and its proof is completed by substituting each delay-adaptive learning rate $\hat{\eta}_{k,t}$ in the proof of Lemma 7 with η for all $k \in [K]$ and $t \in [T]$.

Lemma 4. *With Assumption 2, the gap between $\hat{\mathbf{y}}_t$ and $\hat{\mathbf{w}}_t$ can be bounded:*

$$\mathbb{E}\|\hat{\mathbf{y}}_t - \hat{\mathbf{w}}_t\|^2 \leq \frac{\beta^2\eta^2K^2G^2}{(1-\beta)^4}, \forall t \geq 1. \tag{12}$$

Proof. For any $t \geq 1$, $\hat{\mathbf{u}}_t$ can be formulated as follows:

$$\hat{\mathbf{u}}_t = \beta^{\lfloor\frac{t+K-2}{K}\rfloor}\left(\sum_{k\in[K]}\eta\mathbf{g}_0^k\right) + \sum_{s=1}^{\lfloor\frac{t+K-2}{K}\rfloor}\beta^{\lfloor\frac{t+K-2}{K}\rfloor-s}\left(\sum_{j=(s-1)K+1}^{\min\{sK,t-1\}}\eta\mathbf{g}_j^{k_{j-1}}\right).$$

$$\begin{aligned}\|\hat{\mathbf{y}}_t - \hat{\mathbf{w}}_t\|^2 &= \frac{\beta^2}{(1-\beta)^2}\|\hat{\mathbf{u}}_t\|^2\\ &= \frac{\beta^2}{(1-\beta)^2}\left\|\beta^{\lfloor\frac{t+K-2}{K}\rfloor}\left(\sum_{k\in[K]}\eta\mathbf{g}_0^k\right) + \sum_{s=1}^{\lfloor\frac{t+K-2}{K}\rfloor}\beta^{\lfloor\frac{t+K-2}{K}\rfloor-s}\left(\sum_{j=(s-1)K+1}^{\min\{sK,t-1\}}\eta\mathbf{g}_j^{k_{j-1}}\right)\right\|^2\end{aligned}$$

Let $q_t = \sum_{k\in[K]} \beta^{\lfloor \frac{t+K-2}{K} \rfloor} + \sum_{s=1}^{\lfloor \frac{t+K-2}{K} \rfloor} \sum_{j=(s-1)K+1}^{\min\{sK,t-1\}} \beta^{\lfloor \frac{t+K-2}{K} \rfloor - s}$, then we have

$$
\begin{aligned}
\|\hat{\mathbf{y}}_t - \hat{\mathbf{w}}_t\|^2 &= \frac{\beta^2}{(1-\beta)^2} \left\| \beta^{\lfloor \frac{t+K-2}{K} \rfloor} \left(\sum_{k\in[K]} \eta \mathbf{g}_0^k \right) + \sum_{s=1}^{\lfloor \frac{t+K-2}{K} \rfloor} \beta^{\lfloor \frac{t+K-2}{K} \rfloor - s} \left(\sum_{j=(s-1)K+1}^{\min\{sK,t-1\}} \eta \mathbf{g}_j^{k_{j-1}} \right) \right\|^2 \\
&= \frac{\beta^2 q_t^2}{(1-\beta)^2} \left\| \sum_{k\in[K]} \frac{\beta^{\lfloor \frac{t+K-2}{K} \rfloor}}{q_t} \eta \mathbf{g}_0^k + \sum_{s=1}^{\lfloor \frac{t+K-2}{K} \rfloor} \sum_{j=(s-1)K+1}^{\min\{sK,t-1\}} \frac{\beta^{\lfloor \frac{t+K-2}{K} \rfloor - s}}{q_t} \eta \mathbf{g}_j^{k_{j-1}} \right\|^2 \\
&\le \frac{\beta^2 q_t}{(1-\beta)^2} \left(\sum_{k\in[K]} \beta^{\lfloor \frac{t+K-2}{K} \rfloor} \left\| \eta \mathbf{g}_0^k \right\|^2 + \sum_{s=1}^{\lfloor \frac{t+K-2}{K} \rfloor} \sum_{j=(s-1)K+1}^{\min\{sK,t-1\}} \beta^{\lfloor \frac{t+K-2}{K} \rfloor - s} \left\| \eta \mathbf{g}_j^{k_{j-1}} \right\|^2 \right).
\end{aligned}
$$

$$
\begin{aligned}
\mathbb{E}\|\hat{\mathbf{y}}_t - \hat{\mathbf{w}}_t\|^2 &\le \frac{\beta^2 q_t}{(1-\beta)^2} \left(\sum_{k\in[K]} \beta^{\lfloor \frac{t+K-2}{K} \rfloor} \mathbb{E}\|\eta \mathbf{g}_0^k\|^2 + \sum_{s=1}^{\lfloor \frac{t+K-2}{K} \rfloor} \sum_{j=(s-1)K+1}^{\min\{sK,t-1\}} \beta^{\lfloor \frac{t+K-2}{K} \rfloor - s} \mathbb{E}\|\eta \mathbf{g}_j^{k_{j-1}}\|^2 \right) \\
&\le \frac{\beta^2 \eta^2 G^2 q_t}{(1-\beta)^2} \left(\sum_{k\in[K]} \beta^{\lfloor \frac{t+K-2}{K} \rfloor} + \sum_{s=1}^{\lfloor \frac{t+K-2}{K} \rfloor} \sum_{j=(s-1)K+1}^{\min\{sK,t-1\}} \beta^{\lfloor \frac{t+K-2}{K} \rfloor - s} \right) \\
&= \frac{\beta^2 \eta^2 G^2 q_t^2}{(1-\beta)^2} \le \frac{\beta^2 \eta^2 K^2 G^2}{(1-\beta)^4}
\end{aligned}
$$

□

C.2.4 Proof of Theorem 1

Proof.

$$
\begin{aligned}
\mathbb{E}F(\hat{\mathbf{y}}_{t+1}) &\le F(\hat{\mathbf{y}}_t) - \frac{\eta}{1-\beta} \langle \nabla F(\hat{\mathbf{y}}_t), \mathbb{E}\mathbf{g}_t^{k_{t-1}} \rangle + \frac{L\eta^2}{2(1-\beta)^2} \mathbb{E}\|\mathbf{g}_t^{k_{t-1}}\|^2 \\
&\le F(\hat{\mathbf{y}}_t) - \frac{\eta}{1-\beta} \langle \nabla F(\hat{\mathbf{y}}_t), \nabla F(\mathbf{w}_t) \rangle + \frac{L\eta^2}{2(1-\beta)^2} \mathbb{E}\|\mathbf{g}_t^{k_{t-1}} - \nabla F(\mathbf{w}_t)\|^2 \\
&\quad + \frac{L\eta^2}{2(1-\beta)^2} \|\nabla F(\mathbf{w}_t)\|^2 \\
&\le F(\hat{\mathbf{y}}_t) - \frac{\eta}{1-\beta} \langle \nabla F(\hat{\mathbf{y}}_t), \nabla F(\mathbf{w}_t) \rangle + \frac{L\eta^2 \sigma^2}{2(1-\beta)^2} + \frac{L\eta^2}{2(1-\beta)^2} \|\nabla F(\mathbf{w}_t)\|^2
\end{aligned}
$$

$$
\begin{aligned}
-\langle \nabla F(\hat{\mathbf{y}}_t), \nabla F(\mathbf{w}_t) \rangle &= -\langle \nabla F(\hat{\mathbf{y}}_t) - \nabla F(\mathbf{w}_t) + \nabla F(\mathbf{w}_t), \nabla F(\mathbf{w}_t) \rangle \\
&= -\langle \nabla F(\hat{\mathbf{y}}_t) - \nabla F(\mathbf{w}_t), \nabla F(\mathbf{w}_t) \rangle - \|\nabla F(\mathbf{w}_t)\|^2 \\
&\le \frac{1}{2} \|\nabla F(\hat{\mathbf{y}}_t) - \nabla F(\mathbf{w}_t)\|^2 + \frac{1}{2} \|\nabla F(\mathbf{w}_t)\|^2 - \|\nabla F(\mathbf{w}_t)\|^2 \\
&\le \frac{L^2}{2} \|\hat{\mathbf{y}}_t - \mathbf{w}_t\|^2 - \frac{1}{2} \|\nabla F(\mathbf{w}_t)\|^2
\end{aligned}
$$

$$
\mathbb{E}\|\hat{\mathbf{y}}_t - \mathbf{w}_t\|^2 \le 2\mathbb{E}\|\hat{\mathbf{y}}_t - \hat{\mathbf{w}}_t\|^2 + 2\mathbb{E}\|\hat{\mathbf{w}}_t - \mathbf{w}_t\|^2 \le \frac{2\beta^2 \eta^2 K^2 G^2}{(1-\beta)^4} + \frac{2\eta^2 K^2 G^2}{(1-\beta)^2} \le \frac{2\eta^2 K^2 G^2}{(1-\beta)^4}
$$

$$
\begin{aligned}
\mathbb{E}F(\hat{\mathbf{y}}_{t+1}) &\le \mathbb{E}F(\hat{\mathbf{y}}_t) - \frac{\eta}{1-\beta} \mathbb{E}\langle \nabla F(\hat{\mathbf{y}}_t), \nabla F(\mathbf{w}_t) \rangle + \frac{L\eta^2 \sigma^2}{2(1-\beta)^2} + \frac{L\eta^2}{2(1-\beta)^2} \mathbb{E}\|\nabla F(\mathbf{w}_t)\|^2 \\
&\le \mathbb{E}F(\hat{\mathbf{y}}_t) + \frac{\eta L^2}{2(1-\beta)} \mathbb{E}\|\hat{\mathbf{y}}_t - \mathbf{w}_t\|^2 + \left(\frac{L\eta^2}{2(1-\beta)^2} - \frac{\eta}{2(1-\beta)} \right) \mathbb{E}\|\nabla F(\mathbf{w}_t)\|^2 + \frac{L\eta^2 \sigma^2}{2(1-\beta)^2} \\
&\overset{\eta \le \frac{1-\beta}{2KL}}{\le} \mathbb{E}F(\hat{\mathbf{y}}_t) + \frac{\eta L^2}{2(1-\beta)} \mathbb{E}\|\hat{\mathbf{y}}_t - \mathbf{w}_t\|^2 - \frac{\eta}{4(1-\beta)} \mathbb{E}\|\nabla F(\mathbf{w}_t)\|^2 + \frac{L\eta^2 \sigma^2}{2(1-\beta)^2} \\
&\le \mathbb{E}F(\hat{\mathbf{y}}_t) - \frac{\eta}{4(1-\beta)} \mathbb{E}\|\nabla F(\mathbf{w}_t)\|^2 + \frac{L\eta^2 \sigma^2}{2(1-\beta)^2} + \frac{\eta^3 K^2 G^2 L^2}{(1-\beta)^5}
\end{aligned}
$$

Summing up the above equation from $t = 1$ to $t = T$, we can get that

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla F(\mathbf{w}_t)\|^2 \leq \frac{4(1-\beta)\left[\mathbb{E}F(\hat{\mathbf{y}}_1) - F^*\right]}{T\eta} + \frac{2L\eta\sigma^2}{1-\beta} + \frac{4\eta^2K^2G^2L^2}{(1-\beta)^4}.$$

Since $\hat{\mathbf{y}}_1 - \mathbf{w}_0 = \frac{1}{1-\beta}(\hat{\mathbf{w}}_1 - \mathbf{w}_0) = -\frac{\eta}{1-\beta}\sum_{k\in[K]}\mathbf{g}_0^k$, we have

$$\begin{aligned}
\mathbb{E}F(\hat{\mathbf{y}}_1) &\leq F(\mathbf{w}_0) - \frac{\eta}{1-\beta}\mathbb{E}\langle\nabla F(\mathbf{w}_0), \sum_{k\in[K]}\mathbf{g}_0^k\rangle + \frac{L\eta^2}{2(1-\beta)^2}\mathbb{E}\left\|\sum_{k\in[K]}\mathbf{g}_0^k\right\|^2 \\
&\leq F(\mathbf{w}_0) - \frac{K\eta}{1-\beta}\left\|\nabla F(\mathbf{w}_0)\right\|^2 + \frac{L\eta^2}{2(1-\beta)^2}\mathbb{E}\left\|\sum_{k\in[K]}\mathbf{g}_0^k\right\|^2 \\
&\leq F(\mathbf{w}_0) - \frac{K\eta}{1-\beta}\left\|\nabla F(\mathbf{w}_0)\right\|^2 + \frac{L\eta^2}{2(1-\beta)^2}\mathbb{E}\left\|\sum_{k\in[K]}\mathbf{g}_0^k - K\nabla F(\mathbf{w}_0) + K\nabla F(\mathbf{w}_0)\right\|^2 \\
&\leq F(\mathbf{w}_0) + \left(\frac{LK^2\eta^2}{2(1-\beta)^2} - \frac{K\eta}{1-\beta}\right)\left\|\nabla F(\mathbf{w}_0)\right\|^2 + \frac{KL\sigma^2\eta^2}{2(1-\beta)^2} \\
&\leq F(\mathbf{w}_0) + \frac{KL\sigma^2\eta^2}{2(1-\beta)^2}.
\end{aligned}$$

The last inequality above holds because $\eta \leq \frac{1-\beta}{2KL}$.

Combining the above equations, we can get that

$$\begin{aligned}
\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left\|\nabla F(\mathbf{w}_t)\right\|^2 &\leq \frac{4(1-\beta)\left[F(\mathbf{w}_0) - F^*\right]}{T\eta} + \frac{2L\eta\sigma^2}{1-\beta} + \frac{2LK\sigma^2\eta}{(1-\beta)T} + \frac{4\eta^2L^2K^2G^2}{(1-\beta)^4} \\
&\overset{T\geq K}{\leq} \frac{4(1-\beta)\left[F(\mathbf{w}_0) - F^*\right]}{T\eta} + \frac{4L\eta\sigma^2}{1-\beta} + \frac{4\eta^2L^2K^2G^2}{(1-\beta)^4}.
\end{aligned}$$

Let $\eta = \min\{\frac{1-\beta}{2KL}, \frac{(1-\beta)[F(\mathbf{w}_0)-F^*]^{\frac{1}{2}}}{(LT)^{\frac{1}{2}}\sigma}, \frac{(1-\beta)^{\frac{5}{3}}[F(\mathbf{w}_0)-F^*]^{\frac{1}{3}}}{(LKG)^{\frac{2}{3}}T^{\frac{1}{3}}}\}$, then we can get that

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla F(\mathbf{w}_t)\|^2 \leq \mathcal{O}\left(\sqrt{\frac{L\sigma^2}{T}} + \left(\frac{KLG}{T}\right)^{\frac{2}{3}} + \frac{KL}{T}\right).$$

□

C.3 OrMo with Delay-Adaptive Learning Rate

C.3.1 Algorithm

The details of OrMo with delay-adaptive learning rate (OrMo-DA) are presented in Algorithm 5.

C.3.2 Notation

For a positive integer n, $[n] = \{0, 1, 2, \cdots, n-1\}$. $[0]$ is defined as $\emptyset$.

The function $ite(k, t)$ denotes the iteration index of the latest parameter sent to worker k before iteration t, which can be formulated as

$$ite(k,t) = \begin{cases} 0 & t = 0, k \in [K], \\ t & t > 0, k = k_{t-1}, \\ ite(k, t-1) & t > 0, k \neq k_{t-1}, \end{cases}$$

where $k \in [K], t \in [T+1]$.

Algorithm 5 OrMo-DA

Server:
Input: number of workers K, number of iterations T, momentum coefficient $\beta \in [0, 1)$;
Initialization: initial parameter $\mathbf{w}_0$, momentum $\mathbf{u}_0 = \mathbf{0}$, index of the latest gradient group $I_0 = 0$, waiting set $\mathcal{C} = \emptyset$;
Send the initial parameter $\mathbf{w}_0$ and its iteration index 0 to all workers;
for $t = 0$ **to** $T - 1$ **do**
  if the waiting set $\mathcal{C}$ is empty and $\lceil \frac{t}{K} \rceil > I_t$ **then**
    $\mathbf{w}_{t+\frac{1}{2}} = \mathbf{w}_t - \beta \mathbf{u}_t$, $\mathbf{u}_{t+\frac{1}{2}} = \beta \mathbf{u}_t$, $I_{t+1} = I_t + 1$;
  else
    $\mathbf{w}_{t+\frac{1}{2}} = \mathbf{w}_t$, $\mathbf{u}_{t+\frac{1}{2}} = \mathbf{u}_t$, $I_{t+1} = I_t$;
  end if
  Receive a stochastic gradient $\mathbf{g}^{k_t}_{ite(k_t,t)}$ and its iteration index $ite(k_t, t)$ from some worker k_t and then calculate $\lceil \frac{ite(k_t,t)}{K} \rceil$ (i.e., the index of the gradient group that $\mathbf{g}^{k_t}_{ite(k_t,t)}$ belongs to);
  Update the momentum $\mathbf{u}_{t+1} = \mathbf{u}_{t+\frac{1}{2}} + \beta^{I_{t+1} - \lceil \frac{ite(k_t,t)}{K} \rceil} \times \left(\eta_t \mathbf{g}^{k_t}_{ite(k_t,t)} \right)$;
  Update the parameter $\mathbf{w}_{t+1} = \mathbf{w}_{t+\frac{1}{2}} - \frac{1 - \beta^{I_{t+1} - \lceil \frac{ite(k_t,t)}{K} \rceil + 1}}{1 - \beta} \times \left(\eta_t \mathbf{g}^{k_t}_{ite(k_t,t)} \right)$;
  Add the worker k_t to the waiting set $\mathcal{C} = \mathcal{C} \cup \{k_t\}$;
  Execute the asynchronous communication scheduler: once the waiting set is not empty, i.e., $\mathcal{C} \neq \emptyset$, immediately send the parameter $\mathbf{w}_{t+1}$ and its iteration index $t + 1$ to the worker in $\mathcal{C}$ and set $\mathcal{C}$ to $\emptyset$;
end for
Notify all workers to stop;
Worker k **:** $(k \in [K])$
repeat
  Wait until receiving the parameter $\mathbf{w}_{t'}$ and its iteration index t' from the server;
  Randomly sample $\xi^k \sim \mathcal{D}$ and then compute the stochastic gradient $\mathbf{g}^k_{t'} = \nabla f(\mathbf{w}_{t'}; \xi^k)$;
  Send the stochastic gradient $\mathbf{g}^k_{t'}$ and its iteration index t' to the server;
until receive server's notification to stop

η_t is the learning rate at iteration t and satisfies that

$$\eta_t = \begin{cases} \eta & \tau_t \leq 2K, \\ \min\{\eta, \dfrac{1}{4L\tau_t}\} & \tau_t > 2K, \end{cases}$$

where $t \in [T]$.

The function $next(k, t)$ denotes the index of the next iteration that the gradient from worker k will participate in the parameter update after iteration t (including iteration t), which can be formulated as

$$next(k, t) = \begin{cases} \min\{j \geq t : k_j = k\} & \exists j \in [T] \setminus [t], k_j = k, \\ T & \forall j \in [T] \setminus [t], k_j \neq k, \end{cases}$$

where $k \in [K]$, $t \in [T]$. It's easy to verify that $ite(k, next(k, t)) = ite(k, t), next(k, ite(k, t)) = next(k, t)$, where $k \in [K], t \in [T]$.

We define $\hat{\tau}_{k,t} = t - ite(k, t)$, where $k \in [K]$ and $t \in [T + 1]$. $\hat{\tau}_{k,t}$ denotes the current delay of the gradient $\mathbf{g}^k_{ite(k,t)}$ at iteration t, which is the number of iterations that have happened since $ite(k, t)$. It's easy to verify that $\hat{\tau}_{k_t,t} = t - ite(k_t, t) = \tau_t$, where $t \in [T]$.

We also define an auxiliary sequence $\hat{\eta}_{k,t}$ for the adaptive learning rates. $\hat{\eta}_{k,t}$ denotes the learning rate corresponding to the gradient $\mathbf{g}^k_{ite(k,t)}$.

$$\hat{\eta}_{k,t} = \begin{cases} \eta & \hat{\tau}_{k,next(k,t)} \leq 2K, \\ \min\{\eta, \dfrac{1}{4L\hat{\tau}_{k,next(k,t)}}\} & \hat{\tau}_{k,next(k,t)} > 2K, \end{cases}$$

where $k \in [K]$ and $t \in [T]$. Since $next(k,t) = next(k, ite(k,t))$, we can have that $\hat{\eta}_{k,t} = \hat{\eta}_{k,ite(k,t)}$, where $k \in [K], t \in [T]$.

If $next(k,t) \in [T]$, $\hat{\tau}_{k,next(k,t)} = \hat{\tau}_{k_{next(k,t)},next(k,t)} = \tau_{next(k,t)}$ and then we have that

$$\hat{\eta}_{k,t} = \eta_{next(k,t)} = \begin{cases} \eta & \tau_{next(k,t)} \leq 2K, \\ \min\{\eta, \dfrac{1}{4L\tau_{next(k,t)}}\} & \tau_{next(k,t)} > 2K. \end{cases}$$

It's easy to verify that $\hat{\eta}_{k_t,t} = \eta_{next(k_t,t)} = \eta_t$, where $t \in [T]$.

If $next(k,t) = T$,

$$\hat{\eta}_{k,t} = \begin{cases} \eta & \hat{\tau}_{k,T} \leq 2K, \\ \min\{\eta, \dfrac{1}{4L\hat{\tau}_{k,T}}\} & \hat{\tau}_{k,T} > 2K. \end{cases}$$

C.3.3 Convergence Analysis for OrMo-DA

Firstly, we define one auxiliary sequence $\{\hat{\mathbf{u}}_t\}_{t\geq 1}$ for the momentum: $\hat{\mathbf{u}}_1 = \sum_{k\in[K]} \hat{\eta}_{k,0}\mathbf{g}_0^k$, and

$$\hat{\mathbf{u}}_{t+1} = \begin{cases} \beta\hat{\mathbf{u}}_t + \hat{\eta}_{k_{t-1},t}\mathbf{g}_t^{k_{t-1}} & K \mid (t-1), \\ \hat{\mathbf{u}}_t + \hat{\eta}_{k_{t-1},t}\mathbf{g}_t^{k_{t-1}} & K \nmid (t-1), \end{cases} \tag{13}$$

for $t \geq 1$.

Lemma 5. *For any $t \geq 0$, the gap between $\mathbf{u}_{t+1}$ and $\hat{\mathbf{u}}_{t+1}$ can be formulated as follows:*

$$\hat{\mathbf{u}}_{t+1} - \mathbf{u}_{t+1} = \sum_{k\in[K],k\neq k_t} \beta^{\lceil \frac{t}{K} \rceil - \lceil \frac{ite(k,t)}{K} \rceil} \hat{\eta}_{k,ite(k,t)}\mathbf{g}_{ite(k,t)}^k. \tag{14}$$

Proof. Base case: for $t = 0$, $\hat{\mathbf{u}}_1 = \sum_{k\in[K]} \hat{\eta}_{k,0}\mathbf{g}_0^k$, $\mathbf{u}_1 = \hat{\eta}_{k_0,0}\mathbf{g}_0^{k_0}$, then we have

$$\begin{aligned} \hat{\mathbf{u}}_1 - \mathbf{u}_1 &= \sum_{k\in[K],k\neq k_0} \beta^{\lceil \frac{0}{K} \rceil - \lceil \frac{0}{K} \rceil} \hat{\eta}_{k,0}\mathbf{g}_0^k \\ &= \sum_{k\in[K],k\neq k_0} \beta^{\lceil \frac{0}{K} \rceil - \lceil \frac{ite(k,0)}{K} \rceil} \hat{\eta}_{k,ite(k,0)}\mathbf{g}_{ite(k,0)}^k. \end{aligned}$$

Inductive hypothesis: for some arbitrary integer $t' - 1 \geq 0$, assume that (14) is true for $t = t' - 1$.

Inductive step: We will prove that (14) is true for $t = t'$. Firstly, we divide our discussion into two cases based on whether $t' - 1$ is divisible by K and prove that

$$\begin{aligned} \hat{\mathbf{u}}_{t'+1} - \mathbf{u}_{t'+1} = &\sum_{k\in[K],k\neq k_{t'-1}} \beta^{\lceil \frac{t'}{K} \rceil - \lceil \frac{ite(k,t')}{K} \rceil} \hat{\eta}_{k,ite(k,t')}\mathbf{g}_{ite(k,t')}^k + \hat{\eta}_{k_{t'-1},t'}\mathbf{g}_{t'}^{k_{t'-1}} \\ &- \beta^{\lceil \frac{t'}{K} \rceil - \lceil \frac{ite(k_{t'},t')}{K} \rceil} \hat{\eta}_{k_{t'},ite(k_{t'},t')}\mathbf{g}_{ite(k_{t'},t')}^{k_{t'}}. \end{aligned}$$

Case 1: $K \mid (t'-1)$

$$
\begin{aligned}
\hat{\mathbf{u}}_{t'+1} &= \beta \hat{\mathbf{u}}_{t'} + \hat{\eta}_{k_{t'-1},t'} \mathbf{g}_{t'}^{k_{t'-1}}, \\
\mathbf{u}_{t'+1} &= \beta \mathbf{u}_{t'} + \beta^{\lceil \frac{t'}{K} \rceil - \lceil \frac{ite(k_{t'},t')}{K} \rceil} \hat{\eta}_{k_{t'},ite(k_{t'},t')} \mathbf{g}_{ite(k_{t'},t')}^{k_{t'}}, \\
\hat{\mathbf{u}}_{t'+1} - \mathbf{u}_{t'+1} &= \beta(\hat{\mathbf{u}}_{t'} - \mathbf{u}_{t'}) + \hat{\eta}_{k_{t'-1},t'} \mathbf{g}_{t'}^{k_{t'-1}} - \beta^{\lceil \frac{t'}{K} \rceil - \lceil \frac{ite(k_{t'},t')}{K} \rceil} \hat{\eta}_{k_{t'},ite(k_{t'},t')} \mathbf{g}_{ite(k_{t'},t')}^{k_{t'}} \\
&= \sum_{k \in [K], k \neq k_{t'-1}} \beta^{\lceil \frac{t'-1}{K} \rceil + 1 - \lceil \frac{ite(k,t'-1)}{K} \rceil} \hat{\eta}_{k,ite(k,t'-1)} \mathbf{g}_{ite(k,t'-1)}^{k} \\
&\quad + \hat{\eta}_{k_{t'-1},t'} \mathbf{g}_{t'}^{k_{t'-1}} - \beta^{\lceil \frac{t'}{K} \rceil - \lceil \frac{ite(k_{t'},t')}{K} \rceil} \hat{\eta}_{k_{t'},ite(k_{t'},t')} \mathbf{g}_{ite(k_{t'},t')}^{k_{t'}} \\
&= \sum_{k \in [K], k \neq k_{t'-1}} \beta^{\lceil \frac{t'}{K} \rceil - \lceil \frac{ite(k,t'-1)}{K} \rceil} \hat{\eta}_{k,ite(k,t'-1)} \mathbf{g}_{ite(k,t'-1)}^{k} \\
&\quad + \hat{\eta}_{k_{t'-1},t'} \mathbf{g}_{t'}^{k_{t'-1}} - \beta^{\lceil \frac{t'}{K} \rceil - \lceil \frac{ite(k_{t'},t')}{K} \rceil} \hat{\eta}_{k_{t'},ite(k_{t'},t')} \mathbf{g}_{ite(k_{t'},t')}^{k_{t'}}.
\end{aligned}
$$

The second equation above holds because $\lceil \frac{t'}{K} \rceil > \lceil \frac{t'-1}{K} \rceil = I_{t'}$ when $K \mid (t'-1)$, $I_{t'+1} = \lceil \frac{t'}{K} \rceil$ and $\hat{\eta}_{k_{t'},ite(k_{t'},t')} = \eta_{next(k_{t'},ite(k_{t'},t'))} = \eta_{t'}$. The last equation above holds because $\lceil \frac{t'}{K} \rceil = \lceil \frac{t'-1}{K} \rceil + 1$.

Case 2: $K \nmid (t'-1)$

$$
\begin{aligned}
\hat{\mathbf{u}}_{t'+1} &= \hat{\mathbf{u}}_{t'} + \hat{\eta}_{k_{t'-1},t'} \mathbf{g}_{t'}^{k_{t'-1}}, \\
\mathbf{u}_{t'+1} &= \mathbf{u}_{t'} + \beta^{\lceil \frac{t'}{K} \rceil - \lceil \frac{ite(k_{t'},t')}{K} \rceil} \hat{\eta}_{k_{t'},ite(k_{t'},t')} \mathbf{g}_{ite(k_{t'},t')}^{k_{t'}}, \\
\hat{\mathbf{u}}_{t'+1} - \mathbf{u}_{t'+1} &= (\hat{\mathbf{u}}_{t'} - \mathbf{u}_{t'}) + \hat{\eta}_{k_{t'-1},t'} \mathbf{g}_{t'}^{k_{t'-1}} - \beta^{\lceil \frac{t'}{K} \rceil - \lceil \frac{ite(k_{t'},t')}{K} \rceil} \hat{\eta}_{k_{t'},ite(k_{t'},t')} \mathbf{g}_{ite(k_{t'},t')}^{k_{t'}} \\
&= \sum_{k \in [K], k \neq k_{t'-1}} \beta^{\lceil \frac{t'-1}{K} \rceil - \lceil \frac{ite(k,t'-1)}{K} \rceil} \hat{\eta}_{k,ite(k,t'-1)} \mathbf{g}_{ite(k,t'-1)}^{k} \\
&\quad + \hat{\eta}_{k_{t'-1},t'} \mathbf{g}_{t'}^{k_{t'-1}} - \beta^{\lceil \frac{t'}{K} \rceil - \lceil \frac{ite(k_{t'},t')}{K} \rceil} \hat{\eta}_{k_{t'},ite(k_{t'},t')} \mathbf{g}_{ite(k_{t'},t')}^{k_{t'}} \\
&= \sum_{k \in [K], k \neq k_{t'-1}} \beta^{\lceil \frac{t'}{K} \rceil - \lceil \frac{ite(k,t'-1)}{K} \rceil} \hat{\eta}_{k,ite(k,t'-1)} \mathbf{g}_{ite(k,t'-1)}^{k} \\
&\quad + \hat{\eta}_{k_{t'-1},t'} \mathbf{g}_{t'}^{k_{t'-1}} - \beta^{\lceil \frac{t'}{K} \rceil - \lceil \frac{ite(k_{t'},t')}{K} \rceil} \hat{\eta}_{k_{t'},ite(k_{t'},t')} \mathbf{g}_{ite(k_{t'},t')}^{k_{t'}}.
\end{aligned}
$$

The last equation above holds because $\lceil \frac{t'}{K} \rceil = \lceil \frac{t'-1}{K} \rceil$.

Since $ite(k,t') = ite(k,t'-1), \forall k \neq k_{t'-1}$, we can get the following equation for both cases above:

$$
\begin{aligned}
\hat{\mathbf{u}}_{t'+1} - \mathbf{u}_{t'+1} = &\sum_{k \in [K], k \neq k_{t'-1}} \beta^{\lceil \frac{t'}{K} \rceil - \lceil \frac{ite(k,t')}{K} \rceil} \hat{\eta}_{k,ite(k,t')} \mathbf{g}_{ite(k,t')}^{k} + \hat{\eta}_{k_{t'-1},t'} \mathbf{g}_{t'}^{k_{t'-1}} \\
&- \beta^{\lceil \frac{t'}{K} \rceil - \lceil \frac{ite(k_{t'},t')}{K} \rceil} \hat{\eta}_{k_{t'},ite(k_{t'},t')} \mathbf{g}_{ite(k_{t'},t')}^{k_{t'}}.
\end{aligned}
$$

If $k_{t'} = k_{t'-1}$, then we have $ite(k_{t'},t') = t'$ and

$$
\hat{\mathbf{u}}_{t'+1} - \mathbf{u}_{t'+1} = \sum_{k \in [K], k \neq k_{t'}} \beta^{\lceil \frac{t'}{K} \rceil - \lceil \frac{ite(k,t')}{K} \rceil} \hat{\eta}_{k,ite(k,t')} \mathbf{g}_{ite(k,t')}^{k}.
$$

If $k_{t'} \neq k_{t'-1}$, then we have

$$\begin{aligned}
\hat{\mathbf{u}}_{t'+1} - \mathbf{u}_{t'+1} &= \sum_{k\in[K], k\neq k_{t'-1}} \beta^{\lceil \frac{t'}{K} \rceil - \lceil \frac{ite(k,t')}{K} \rceil} \hat{\eta}_{k,ite(k,t')} \mathbf{g}^{k}_{ite(k,t')} + \hat{\eta}_{k_{t'-1},t'} \mathbf{g}^{k_{t'-1}}_{t'} \\
&\quad - \beta^{\lceil \frac{t'}{K} \rceil - \lceil \frac{ite(k_{t'},t')}{K} \rceil} \hat{\eta}_{k_{t'},ite(k_{t'},t')} \mathbf{g}^{k_{t'}}_{ite(k_{t'},t')} \\
&= \sum_{k\in[K], k\neq k_{t'-1}, k\neq k_{t'}} \beta^{\lceil \frac{t'}{K} \rceil - \lceil \frac{ite(k,t')}{K} \rceil} \hat{\eta}_{k,ite(k,t')} \mathbf{g}^{k}_{ite(k,t')} + \hat{\eta}_{k_{t'-1},t'} \mathbf{g}^{k_{t'-1}}_{t'} \\
&= \sum_{k\in[K], k\neq k_{t'-1}, k\neq k_{t'}} \beta^{\lceil \frac{t'}{K} \rceil - \lceil \frac{ite(k,t')}{K} \rceil} \hat{\eta}_{k,ite(k,t')} \mathbf{g}^{k}_{ite(k,t')} \\
&\quad + \beta^{\lceil \frac{t'}{K} \rceil - \lceil \frac{ite(k_{t'-1},t')}{K} \rceil} \hat{\eta}_{k_{t'-1},ite(k_{t'-1},t')} \mathbf{g}^{k_{t'-1}}_{ite(k_{t'-1},t')} \\
&= \sum_{k\in[K], k\neq k_{t'}} \beta^{\lceil \frac{t'}{K} \rceil - \lceil \frac{ite(k,t')}{K} \rceil} \hat{\eta}_{k,ite(k,t')} \mathbf{g}^{k}_{ite(k,t')}.
\end{aligned}$$

We can conclude that $\hat{\mathbf{u}}_{t+1} - \mathbf{u}_{t+1} = \sum_{k\in[K], k\neq k_t} \beta^{\lceil \frac{t}{K} \rceil - \lceil \frac{ite(k,t)}{K} \rceil} \hat{\eta}_{k,ite(k,t)} \mathbf{g}^{k}_{ite(k,t)}$ is true for any $t \geq 0$. □

Then, we define one auxiliary sequence $\{\hat{\mathbf{w}}_t\}_{t\geq 1}$ for the parameter: $\hat{\mathbf{w}}_1 = \mathbf{w}_0 - \sum_{k\in[K]} \hat{\eta}_{k,0} \mathbf{g}^k_0$, and

$$\hat{\mathbf{w}}_{t+1} = \begin{cases} \hat{\mathbf{w}}_t - \beta \hat{\mathbf{u}}_t - \hat{\eta}_{k_{t-1},t} \mathbf{g}^{k_{t-1}}_t & K \mid (t-1), \\ \hat{\mathbf{w}}_t - \hat{\eta}_{k_{t-1},t} \mathbf{g}^{k_{t-1}}_t & K \nmid (t-1), \end{cases}$$

for $t \geq 1$.

Lemma 6. *For any $t \geq 0$, the gap between $\mathbf{w}_{t+1}$ and $\hat{\mathbf{w}}_{t+1}$ can be formulated as follows:*

$$\hat{\mathbf{w}}_{t+1} - \mathbf{w}_{t+1} = - \sum_{k\in[K], k\neq k_t} \frac{1 - \beta^{\lceil \frac{t}{K} \rceil - \lceil \frac{ite(k,t)}{K} \rceil + 1}}{1-\beta} \hat{\eta}_{k,ite(k,t)} \mathbf{g}^{k}_{ite(k,t)}. \tag{15}$$

Proof. Base case: for $t = 0$, $\hat{\mathbf{w}}_1 = \mathbf{w}_0 - \sum_{k\in[K]} \hat{\eta}_{k,0} \mathbf{g}^k_0$, $\mathbf{w}_1 = \mathbf{w}_0 - \hat{\eta}_{k_0,0} \mathbf{g}^{k_0}_0$, then we have

$$\hat{\mathbf{w}}_1 - \mathbf{w}_1 = - \sum_{k\in[K], k\neq k_0} \frac{1 - \beta^{\lceil \frac{0}{K} \rceil - \lceil \frac{ite(k,0)}{K} \rceil + 1}}{1-\beta} \hat{\eta}_{k,ite(k,0)} \mathbf{g}^{k}_{ite(k,0)}.$$

Inductive hypothesis: for some arbitrary integer $t' - 1 \geq 0$, assume that (15) is true for $t = t' - 1$.

Inductive step: We will prove that (15) is true for $t = t'$. Firstly, we divide our discussion into two cases based on whether $t' - 1$ is divisible by K and prove that

$$\begin{aligned}
\hat{\mathbf{w}}_{t'+1} - \mathbf{w}_{t'+1} = &- \sum_{k\in[K], k\neq k_{t'-1}} \frac{1 - \beta^{\lceil \frac{t'}{K} \rceil - \lceil \frac{ite(k,t')}{K} \rceil + 1}}{1-\beta} \hat{\eta}_{k,ite(k,t')} \mathbf{g}^{k}_{ite(k,t')} - \hat{\eta}_{k_{t'-1},t'} \mathbf{g}^{k_{t'-1}}_{t'} \\
&+ \frac{1 - \beta^{\lceil \frac{t'}{K} \rceil - \lceil \frac{ite(k_{t'},t')}{K} \rceil + 1}}{1-\beta} \hat{\eta}_{k_{t'},ite(k_{t'},t')} \mathbf{g}^{k_{t'}}_{ite(k_{t'},t')}.
\end{aligned}$$

Case 1: $K \mid (t'-1)$

$$\hat{\mathbf{w}}_{t'+1} = \hat{\mathbf{w}}_{t'} - \beta \hat{\mathbf{u}}_{t'} - \hat{\eta}_{k_{t'-1},t'} \mathbf{g}_{t'}^{k_{t'-1}},$$

$$\mathbf{w}_{t'+1} = \mathbf{w}_{t'} - \beta \mathbf{u}_{t'} - \frac{1-\beta^{\lceil \frac{t'}{K} \rceil - \lceil \frac{ite(k_{t'},t')}{K} \rceil + 1}}{1-\beta} \hat{\eta}_{k_{t'},ite(k_{t'},t')} \mathbf{g}_{ite(k_{t'},t')}^{k_{t'}},$$

$$\begin{aligned}
\hat{\mathbf{w}}_{t'+1} - \mathbf{w}_{t'+1} &= \hat{\mathbf{w}}_{t'} - \mathbf{w}_{t'} - \beta \left(\hat{\mathbf{u}}_{t'} - \mathbf{u}_{t'} \right) - \hat{\eta}_{k_{t'-1},t'} \mathbf{g}_{t'}^{k_{t'-1}} \\
&\quad + \frac{1-\beta^{\lceil \frac{t'}{K} \rceil - \lceil \frac{ite(k_{t'},t')}{K} \rceil + 1}}{1-\beta} \hat{\eta}_{k_{t'},ite(k_{t'},t')} \mathbf{g}_{ite(k_{t'},t')}^{k_{t'}} \\
&= - \sum_{k \in [K], k \neq k_{t'-1}} \frac{1-\beta^{\lceil \frac{t'-1}{K} \rceil - \lceil \frac{ite(k,t'-1)}{K} \rceil + 1}}{1-\beta} \hat{\eta}_{k,ite(k,t'-1)} \mathbf{g}_{ite(k,t'-1)}^{k} \\
&\quad - \sum_{k \in [K], k \neq k_{t'-1}} \beta^{\lceil \frac{t'-1}{K} \rceil - \lceil \frac{ite(k,t'-1)}{K} \rceil + 1} \hat{\eta}_{k,ite(k,t'-1)} \mathbf{g}_{ite(k,t'-1)}^{k} \\
&\quad - \left(\hat{\eta}_{k_{t'-1},t'} \mathbf{g}_{t'}^{k_{t'-1}} - \frac{1-\beta^{\lceil \frac{t'}{K} \rceil - \lceil \frac{ite(k_{t'},t')}{K} \rceil + 1}}{1-\beta} \hat{\eta}_{k_{t'},ite(k_{t'},t')} \mathbf{g}_{ite(k_{t'},t')}^{k_{t'}} \right) \\
&= - \sum_{k \in [K], k \neq k_{t'-1}} \frac{1-\beta^{\lceil \frac{t'-1}{K} \rceil - \lceil \frac{ite(k,t'-1)}{K} \rceil + 2}}{1-\beta} \hat{\eta}_{k,ite(k,t'-1)} \mathbf{g}_{ite(k,t'-1)}^{k} \\
&\quad - \hat{\eta}_{k_{t'-1},t'} \mathbf{g}_{t'}^{k_{t'-1}} + \frac{1-\beta^{\lceil \frac{t'}{K} \rceil - \lceil \frac{ite(k_{t'},t')}{K} \rceil + 1}}{1-\beta} \hat{\eta}_{k_{t'},ite(k_{t'},t')} \mathbf{g}_{ite(k_{t'},t')}^{k_{t'}} \\
&= - \sum_{k \in [K], k \neq k_{t'-1}} \frac{1-\beta^{\lceil \frac{t'}{K} \rceil - \lceil \frac{ite(k,t'-1)}{K} \rceil + 1}}{1-\beta} \hat{\eta}_{k,ite(k,t'-1)} \mathbf{g}_{ite(k,t'-1)}^{k} \\
&\quad - \hat{\eta}_{k_{t'-1},t'} \mathbf{g}_{t'}^{k_{t'-1}} + \frac{1-\beta^{\lceil \frac{t'}{K} \rceil - \lceil \frac{ite(k_{t'},t')}{K} \rceil + 1}}{1-\beta} \hat{\eta}_{k_{t'},ite(k_{t'},t')} \mathbf{g}_{ite(k_{t'},t')}^{k_{t'}}.
\end{aligned}$$

The last equation above holds because $\lceil \frac{t'}{K} \rceil = \lceil \frac{t'-1}{K} \rceil + 1$.

Case 2: $K \nmid (t'-1)$

$$\hat{\mathbf{w}}_{t'+1} = \hat{\mathbf{w}}_{t'} - \hat{\eta}_{k_{t'-1},t'} \mathbf{g}_{t'}^{k_{t'-1}},$$

$$\mathbf{w}_{t'+1} = \mathbf{w}_{t'} - \frac{1-\beta^{\lceil \frac{t'}{K} \rceil - \lceil \frac{ite(k_{t'},t')}{K} \rceil + 1}}{1-\beta} \hat{\eta}_{k_{t'},ite(k_{t'},t')} \mathbf{g}_{ite(k_{t'},t')}^{k_{t'}},$$

$$\begin{aligned}
\hat{\mathbf{w}}_{t'+1} - \mathbf{w}_{t'+1} &= \hat{\mathbf{w}}_{t'} - \mathbf{w}_{t'} - \left(\hat{\eta}_{k_{t'-1},t'} \mathbf{g}_{t'}^{k_{t'-1}} - \frac{1-\beta^{\lceil \frac{t'}{K} \rceil - \lceil \frac{ite(k_{t'},t')}{K} \rceil + 1}}{1-\beta} \hat{\eta}_{k_{t'},ite(k_{t'},t')} \mathbf{g}_{ite(k_{t'},t')}^{k_{t'}} \right) \\
&= - \sum_{k \in [K], k \neq k_{t'-1}} \frac{1-\beta^{\lceil \frac{t'-1}{K} \rceil - \lceil \frac{ite(k,t'-1)}{K} \rceil + 1}}{1-\beta} \hat{\eta}_{k,ite(k,t'-1)} \mathbf{g}_{ite(k,t'-1)}^{k} \\
&\quad - \hat{\eta}_{k_{t'-1},t'} \mathbf{g}_{t'}^{k_{t'-1}} + \frac{1-\beta^{\lceil \frac{t'}{K} \rceil - \lceil \frac{ite(k_{t'},t')}{K} \rceil + 1}}{1-\beta} \hat{\eta}_{k_{t'},ite(k_{t'},t')} \mathbf{g}_{ite(k_{t'},t')}^{k_{t'}} \\
&= - \sum_{k \in [K], k \neq k_{t'-1}} \frac{1-\beta^{\lceil \frac{t'}{K} \rceil - \lceil \frac{ite(k,t'-1)}{K} \rceil + 1}}{1-\beta} \hat{\eta}_{k,ite(k,t'-1)} \mathbf{g}_{ite(k,t'-1)}^{k} \\
&\quad - \hat{\eta}_{k_{t'-1},t'} \mathbf{g}_{t'}^{k_{t'-1}} + \frac{1-\beta^{\lceil \frac{t'}{K} \rceil - \lceil \frac{ite(k_{t'},t')}{K} \rceil + 1}}{1-\beta} \hat{\eta}_{k_{t'},ite(k_{t'},t')} \mathbf{g}_{ite(k_{t'},t')}^{k_{t'}}.
\end{aligned}$$

The last equation above holds because $\lceil \frac{t'}{K} \rceil = \lceil \frac{t'-1}{K} \rceil$.

Since $ite(k,t') = ite(k,t'-1), \forall k \neq k_{t'-1}$, we can get the following equation for both cases above:

$$\hat{\mathbf{w}}_{t'+1} - \mathbf{w}_{t'+1} = -\sum_{k\in[K],k\neq k_{t'-1}} \frac{1-\beta^{\lceil \frac{t'}{K}\rceil - \lceil \frac{ite(k,t')}{K}\rceil + 1}}{1-\beta} \hat{\eta}_{k,ite(k,t')} \mathbf{g}^k_{ite(k,t')} - \hat{\eta}_{k_{t'-1},t'} \mathbf{g}_{t'}^{k_{t'-1}}$$
$$+ \frac{1-\beta^{\lceil \frac{t'}{K}\rceil - \lceil \frac{ite(k_{t'},t')}{K}\rceil + 1}}{1-\beta} \hat{\eta}_{k_{t'},ite(k_{t'},t')} \mathbf{g}^{k_{t'}}_{ite(k_{t'},t')}.$$

If $k_{t'} = k_{t'-1}$, then we have $ite(k_{t'},t') = t'$ and

$$\hat{\mathbf{w}}_{t'+1} - \mathbf{w}_{t'+1} = -\sum_{k\in[K],k\neq k_{t'}} \frac{1-\beta^{\lceil \frac{t'}{K}\rceil - \lceil \frac{ite(k,t')}{K}\rceil + 1}}{1-\beta} \hat{\eta}_{k,ite(k,t')} \mathbf{g}^k_{ite(k,t')}.$$

If $k_{t'} \neq k_{t'-1}$, then we have

$$\begin{aligned}
\hat{\mathbf{w}}_{t'+1} - \mathbf{w}_{t'+1} &= -\sum_{k\in[K],k\neq k_{t'-1}} \frac{1-\beta^{\lceil \frac{t'}{K}\rceil - \lceil \frac{ite(k,t')}{K}\rceil + 1}}{1-\beta} \hat{\eta}_{k,ite(k,t')} \mathbf{g}^k_{ite(k,t')} - \hat{\eta}_{k_{t'-1},t'} \mathbf{g}_{t'}^{k_{t'-1}} \\
&\quad + \frac{1-\beta^{\lceil \frac{t'}{K}\rceil - \lceil \frac{ite(k_{t'},t')}{K}\rceil + 1}}{1-\beta} \hat{\eta}_{k_{t'},ite(k_{t'},t')} \mathbf{g}^{k_{t'}}_{ite(k_{t'},t')} \\
&= -\sum_{k\in[K],k\neq k_{t'-1},k\neq k_{t'}} \frac{1-\beta^{\lceil \frac{t'}{K}\rceil - \lceil \frac{ite(k,t')}{K}\rceil + 1}}{1-\beta} \hat{\eta}_{k,ite(k,t')} \mathbf{g}^k_{ite(k,t')} - \hat{\eta}_{k_{t'-1},t'} \mathbf{g}_{t'}^{k_{t'-1}} \\
&= -\sum_{k\in[K],k\neq k_{t'}} \frac{1-\beta^{\lceil \frac{t'}{K}\rceil - \lceil \frac{ite(k,t')}{K}\rceil + 1}}{1-\beta} \hat{\eta}_{k,ite(k,t')} \mathbf{g}^k_{ite(k,t')}
\end{aligned}$$

We can conclude that $\hat{\mathbf{w}}_{t+1} - \mathbf{w}_{t+1} = -\sum_{k\in[K],k\neq k_t} \frac{1-\beta^{\lceil \frac{t}{K}\rceil - \lceil \frac{ite(k,t)}{K}\rceil + 1}}{1-\beta} \hat{\eta}_{k,ite(k,t)} \mathbf{g}^k_{ite(k,t)}$ is true for any $t \geq 0$. □

Then, we define another auxiliary sequence $\{\hat{\mathbf{y}}_t\}_{t\geq 1}$: $\hat{\mathbf{y}}_1 = \mathbf{w}_0 - \frac{1}{1-\beta}\sum_{k\in[K]} \hat{\eta}_{k,0}\mathbf{g}^k_0$, and

$$\hat{\mathbf{y}}_{t+1} = \hat{\mathbf{y}}_t - \frac{1}{1-\beta}\hat{\eta}_{k_{t-1},t}\mathbf{g}_t^{k_{t-1}},$$

for $t \geq 1$.

Lemma 7. *For any $t \geq 1$, the gap between $\hat{\mathbf{y}}_t$ and $\hat{\mathbf{w}}_t$ can be formulated as follows:*

$$\hat{\mathbf{y}}_t - \hat{\mathbf{w}}_t = -\frac{\beta}{1-\beta}\hat{\mathbf{u}}_t. \tag{16}$$

Proof. Base case: For $t = 1$, we have that

$$\begin{aligned}
\hat{\mathbf{y}}_1 - \hat{\mathbf{w}}_1 &= \left(\mathbf{w}_0 - \frac{1}{1-\beta}\sum_{k\in[K]} \hat{\eta}_{k,0}\mathbf{g}^k_0\right) - \left(\mathbf{w}_0 - \sum_{k\in[K]} \hat{\eta}_{k,0}\mathbf{g}^k_0\right) \\
&= -\frac{\beta}{1-\beta}\sum_{k\in[K]} \hat{\eta}_{k,0}\mathbf{g}^k_0 = -\frac{\beta}{1-\beta}\hat{\mathbf{u}}_1.
\end{aligned}$$

Inductive hypothesis: for some arbitrary integer $t' \geq 1$, assume that $\hat{\mathbf{y}}_t - \hat{\mathbf{w}}_t = -\frac{\beta}{1-\beta}\hat{\mathbf{u}}_t$ is true for $t = t'$.

Inductive step: We will prove that $\hat{\mathbf{y}}_t - \hat{\mathbf{w}}_t = -\frac{\beta}{1-\beta}\hat{\mathbf{u}}_t$ is true for $t = t'+1$. We divide our discussion into two cases based on whether $t'-1$ is divisible by K.

Case 1: $K \mid (t'-1)$

$$\begin{aligned}
\hat{\mathbf{y}}_{t'+1} &= \hat{\mathbf{y}}_{t'} - \frac{1}{1-\beta}\hat{\eta}_{k_{t'-1},t'}\mathbf{g}_{t'}^{k_{t'-1}} \\
\hat{\mathbf{w}}_{t'+1} &= \hat{\mathbf{w}}_{t'} - \beta\hat{\mathbf{u}}_{t'} - \hat{\eta}_{k_{t'-1},t'}\mathbf{g}_{t'}^{k_{t'-1}} \\
\hat{\mathbf{y}}_{t'+1} - \hat{\mathbf{w}}_{t'+1} &= \left(\hat{\mathbf{y}}_{t'} - \frac{1}{1-\beta}\hat{\eta}_{k_{t'-1},t'}\mathbf{g}_{t'}^{k_{t'-1}}\right) - \left(\hat{\mathbf{w}}_{t'} - \beta\hat{\mathbf{u}}_{t'} - \hat{\eta}_{k_{t'-1},t'}\mathbf{g}_{t'}^{k_{t'-1}}\right) \\
&= \hat{\mathbf{y}}_{t'} - \hat{\mathbf{w}}_{t'} + \beta\hat{\mathbf{u}}_{t'} - \frac{\beta}{1-\beta}\hat{\eta}_{k_{t'-1},t'}\mathbf{g}_{t'}^{k_{t'-1}} \\
&= -\frac{\beta^2}{1-\beta}\hat{\mathbf{u}}_{t'} - \frac{\beta}{1-\beta}\hat{\eta}_{k_{t'-1},t'}\mathbf{g}_{t'}^{k_{t'-1}} \\
&= -\frac{\beta}{1-\beta}\hat{\mathbf{u}}_{t'+1}
\end{aligned}$$

Case 2: $K \nmid (t'-1)$

$$\begin{aligned}
\hat{\mathbf{y}}_{t'+1} &= \hat{\mathbf{y}}_{t'} - \frac{1}{1-\beta}\hat{\eta}_{k_{t'-1},t'}\mathbf{g}_{t'}^{k_{t'-1}} \\
\hat{\mathbf{w}}_{t'+1} &= \hat{\mathbf{w}}_{t'} - \hat{\eta}_{k_{t'-1},t'}\mathbf{g}_{t'}^{k_{t'-1}} \\
\hat{\mathbf{y}}_{t'+1} - \hat{\mathbf{w}}_{t'+1} &= \left(\hat{\mathbf{y}}_{t'} - \frac{1}{1-\beta}\hat{\eta}_{k_{t'-1},t'}\mathbf{g}_{t'}^{k_{t'-1}}\right) - \left(\hat{\mathbf{w}}_{t'} - \hat{\eta}_{k_{t'-1},t'}\mathbf{g}_{t'}^{k_{t'-1}}\right) \\
&= \hat{\mathbf{y}}_{t'} - \hat{\mathbf{w}}_{t'} - \frac{\beta}{1-\beta}\hat{\eta}_{k_{t'-1},t'}\mathbf{g}_{t'}^{k_{t'-1}} \\
&= -\frac{\beta}{1-\beta}\hat{\mathbf{u}}_{t'} - \frac{\beta}{1-\beta}\hat{\eta}_{k_{t'-1},t'}\mathbf{g}_{t'}^{k_{t'-1}} \\
&= -\frac{\beta}{1-\beta}\hat{\mathbf{u}}_{t'+1}
\end{aligned}$$

We can conclude that $\hat{\mathbf{y}}_t - \hat{\mathbf{w}}_t = -\frac{\beta}{1-\beta}\hat{\mathbf{u}}_t$ is true for any $t \geq 1$. □

Lemma 8. *For any $t \geq 1$, $\hat{\mathbf{u}}_t$ can be formulated as follows:*

$$\hat{\mathbf{u}}_t = \beta^{\lfloor \frac{t+K-2}{K} \rfloor}\left(\sum_{k\in[K]}\hat{\eta}_{k,0}\mathbf{g}_0^k\right) + \sum_{s=1}^{\lfloor \frac{t+K-2}{K} \rfloor}\beta^{\lfloor \frac{t+K-2}{K} \rfloor - s}\left(\sum_{j=(s-1)K+1}^{\min\{sK,t-1\}}\hat{\eta}_{k_{j-1},j}\mathbf{g}_j^{k_{j-1}}\right).$$

Proof. It's straightforward to get this conclusion from the definition of the sequence $\hat{\mathbf{u}}_t$ in (13). □

Lemma 9. *(descent lemma) With Assumptions 1 and 3, we have the following descent lemma for $t \geq 1$,:*

$$\begin{aligned}
\mathbb{E}F(\hat{\mathbf{y}}_{t+1}) \leq F(\hat{\mathbf{y}}_t) &+ \left(\frac{L(\hat{\eta}_{k_{t-1},t})^2}{2(1-\beta)^2} - \frac{\hat{\eta}_{k_{t-1},t}}{2(1-\beta)}\right)\|\nabla F(\mathbf{w}_t)\|^2 + \frac{(\hat{\eta}_{k_{t-1},t})^2\sigma^2 L}{2(1-\beta)^2} \\
&+ \frac{L^2\hat{\eta}_{k_{t-1},t}}{1-\beta}\|\hat{\mathbf{y}}_t - \hat{\mathbf{w}}_t\|^2 + \frac{L^2\hat{\eta}_{k_{t-1},t}}{1-\beta}\|\hat{\mathbf{w}}_t - \mathbf{w}_t\|^2.
\end{aligned}$$

Proof.

$$\hat{\mathbf{y}}_{t+1} = \hat{\mathbf{y}}_t - \frac{1}{1-\beta}\hat{\eta}_{k_{t-1},t}\mathbf{g}_t^{k_{t-1}}$$

$$\begin{aligned}
\mathbb{E}F(\hat{\mathbf{y}}_{t+1}) &\leq F(\hat{\mathbf{y}}_t) + \mathbb{E}\langle \nabla F(\hat{\mathbf{y}}_t), \hat{\mathbf{y}}_{t+1} - \hat{\mathbf{y}}_t\rangle + \frac{L}{2}\mathbb{E}\left\|\hat{\mathbf{y}}_{t+1} - \hat{\mathbf{y}}_t\right\|^2 \\
&= F(\hat{\mathbf{y}}_t) - \frac{1}{1-\beta}\mathbb{E}\langle \nabla F(\hat{\mathbf{y}}_t), \hat{\eta}_{k_{t-1},t}\mathbf{g}_t^{k_{t-1}}\rangle + \frac{L}{2(1-\beta)^2}\mathbb{E}\left[(\hat{\eta}_{k_{t-1},t})^2\left\|\mathbf{g}_t^{k_{t-1}}\right\|^2\right] \\
&= F(\hat{\mathbf{y}}_t) - \frac{1}{1-\beta}\langle \nabla F(\hat{\mathbf{y}}_t), \hat{\eta}_{k_{t-1},t}\nabla F(\mathbf{w}_t)\rangle + \frac{L}{2(1-\beta)^2}\mathbb{E}\left[(\hat{\eta}_{k_{t-1},t})^2\left\|\mathbf{g}_t^{k_{t-1}}\right\|^2\right].
\end{aligned}$$

$$\begin{aligned}
-\frac{\hat{\eta}_{k_{t-1},t}}{1-\beta}\langle \nabla F(\hat{\mathbf{y}}_t), \nabla F(\mathbf{w}_t)\rangle &= -\frac{\hat{\eta}_{k_{t-1},t}}{1-\beta}\langle \nabla F(\hat{\mathbf{y}}_t) - \nabla F(\mathbf{w}_t) + \nabla F(\mathbf{w}_t), \nabla F(\mathbf{w}_t)\rangle \\
&= -\frac{\hat{\eta}_{k_{t-1},t}}{1-\beta}\langle \nabla F(\hat{\mathbf{y}}_t) - \nabla F(\mathbf{w}_t), \nabla F(\mathbf{w}_t)\rangle - \frac{\hat{\eta}_{k_{t-1},t}}{1-\beta}\left\|\nabla F(\mathbf{w}_t)\right\|^2 \\
&\leq \frac{\hat{\eta}_{k_{t-1},t}}{2(1-\beta)}\left\|\nabla F(\hat{\mathbf{y}}_t) - \nabla F(\mathbf{w}_t)\right\|^2 - \frac{\hat{\eta}_{k_{t-1},t}}{2(1-\beta)}\left\|\nabla F(\mathbf{w}_t)\right\|^2 \\
&\leq \frac{\hat{\eta}_{k_{t-1},t}L^2}{2(1-\beta)}\left\|\hat{\mathbf{y}}_t - \mathbf{w}_t\right\|^2 - \frac{\hat{\eta}_{k_{t-1},t}}{2(1-\beta)}\left\|\nabla F(\mathbf{w}_t)\right\|^2
\end{aligned}$$

$$\begin{aligned}
\frac{L}{2(1-\beta)^2}\mathbb{E}\left[(\hat{\eta}_{k_{t-1},t})^2\left\|\mathbf{g}_t^{k_{t-1}}\right\|^2\right] &= \frac{L(\hat{\eta}_{k_{t-1},t})^2}{2(1-\beta)^2}\mathbb{E}\left\|\mathbf{g}_t^{k_{t-1}} - \nabla F(\mathbf{w}_t) + \nabla F(\mathbf{w}_t)\right\|^2 \\
&\leq \frac{L(\hat{\eta}_{k_{t-1},t})^2}{2(1-\beta)^2}\left(\mathbb{E}\left\|\mathbf{g}_t^{k_{t-1}} - \nabla F(\mathbf{w}_t)\right\|^2 + \left\|\nabla F(\mathbf{w}_t)\right\|^2\right) \\
&\leq \frac{(\hat{\eta}_{k_{t-1},t})^2 L}{2(1-\beta)^2}\left(\sigma^2 + \left\|\nabla F(\mathbf{w}_t)\right\|^2\right)
\end{aligned}$$

$$\begin{aligned}
\mathbb{E}F(\hat{\mathbf{y}}_{t+1}) &\leq F(\hat{\mathbf{y}}_t) + \left(\frac{L(\hat{\eta}_{k_{t-1},t})^2}{2(1-\beta)^2} - \frac{\hat{\eta}_{k_{t-1},t}}{2(1-\beta)}\right)\left\|\nabla F(\mathbf{w}_t)\right\|^2 + \frac{(\hat{\eta}_{k_{t-1},t})^2\sigma^2 L}{2(1-\beta)^2} \\
&\quad + \frac{L^2\hat{\eta}_{k_{t-1},t}}{2(1-\beta)}\left\|\hat{\mathbf{y}}_t - \mathbf{w}_t\right\|^2 \\
&\leq F(\hat{\mathbf{y}}_t) + \left(\frac{L(\hat{\eta}_{k_{t-1},t})^2}{2(1-\beta)^2} - \frac{\hat{\eta}_{k_{t-1},t}}{2(1-\beta)}\right)\left\|\nabla F(\mathbf{w}_t)\right\|^2 + \frac{(\hat{\eta}_{k_{t-1},t})^2\sigma^2 L}{2(1-\beta)^2} \\
&\quad + \frac{L^2\hat{\eta}_{k_{t-1},t}}{1-\beta}\left\|\hat{\mathbf{y}}_t - \hat{\mathbf{w}}_t\right\|^2 + \frac{L^2\hat{\eta}_{k_{t-1},t}}{1-\beta}\left\|\hat{\mathbf{w}}_t - \mathbf{w}_t\right\|^2.
\end{aligned}$$

□

Lemma 10. *With Assumption 1, the gap between $\hat{\mathbf{y}}_t$ and $\hat{\mathbf{w}}_t$ in OrMo-DA can be bounded as follows:*

$$\begin{aligned}
\sum_{t=1}^{T-1}\mathbb{E}\left(\hat{\eta}_{k_{t-1},t}\left\|\hat{\mathbf{y}}_t - \hat{\mathbf{w}}_t\right\|^2\right) &\leq \frac{2\beta^2\eta K^2}{(1-\beta)^4}\left[\sum_{k\in[K]}(\hat{\eta}_{k,0})^2\left\|\nabla F(\mathbf{w}_0)\right\|^2 + \sum_{t=1}^{T-1}\left(\hat{\eta}_{k_{t-1},t}\right)^2\mathbb{E}\left\|\nabla F(\mathbf{w}_t)\right\|^2\right] \\
&\quad + \frac{2\beta^2\eta K\sigma^2}{(1-\beta)^3}\left[\sum_{k\in[K]}(\hat{\eta}_{k,0})^2 + \sum_{t=1}^{T-1}\left(\hat{\eta}_{k_{t-1},t}\right)^2\right].
\end{aligned}$$

Proof.

$$
\begin{aligned}
&\mathbb{E}\left\|\hat{\mathbf{u}}_t\right\|^2 = \mathbb{E}\left\|\beta^{\lfloor\frac{t+K-2}{K}\rfloor}\left(\sum_{k\in[K]}\hat{\eta}_{k,0}\mathbf{g}_0^k\right) + \sum_{s=1}^{\lfloor\frac{t+K-2}{K}\rfloor}\beta^{\lfloor\frac{t+K-2}{K}\rfloor-s}\left(\sum_{j=(s-1)K+1}^{\min\{sK,t-1\}}\hat{\eta}_{k_{j-1},j}\mathbf{g}_j^{k_{j-1}}\right)\right\|^2 \\
&\leq 2\mathbb{E}\left\|\beta^{\lfloor\frac{t+K-2}{K}\rfloor}\left(\sum_{k\in[K]}\hat{\eta}_{k,0}\left(\mathbf{g}_0^k - \nabla F(\mathbf{w}_0)\right)\right)\right. \\
&\quad\left. + \sum_{s=1}^{\lfloor\frac{t+K-2}{K}\rfloor}\beta^{\lfloor\frac{t+K-2}{K}\rfloor-s}\left(\sum_{j=(s-1)K+1}^{\min\{sK,t-1\}}\hat{\eta}_{k_{j-1},j}\left(\mathbf{g}_j^{k_{j-1}} - \nabla F(\mathbf{w}_j)\right)\right)\right\|^2 \\
&\quad + 2\mathbb{E}\left\|\beta^{\lfloor\frac{t+K-2}{K}\rfloor}\left(\sum_{k\in[K]}\hat{\eta}_{k,0}\nabla F(\mathbf{w}_0)\right) + \sum_{s=1}^{\lfloor\frac{t+K-2}{K}\rfloor}\beta^{\lfloor\frac{t+K-2}{K}\rfloor-s}\left(\sum_{j=(s-1)K+1}^{\min\{sK,t-1\}}\hat{\eta}_{k_{j-1},j}\nabla F(\mathbf{w}_j)\right)\right\|^2 \\
&\leq 2\sum_{k\in[K]}\beta^{2\lfloor\frac{t+K-2}{K}\rfloor}(\hat{\eta}_{k,0})^2\sigma^2 + 2\sum_{s=1}^{\lfloor\frac{t+K-2}{K}\rfloor}\sum_{j=(s-1)K+1}^{\min\{sK,t-1\}}\beta^{2\lfloor\frac{t+K-2}{K}\rfloor-2s}\left(\hat{\eta}_{k_{j-1},j}\right)^2\sigma^2 \\
&\quad + 2\mathbb{E}\left\|\beta^{\lfloor\frac{t+K-2}{K}\rfloor}\left(\sum_{k\in[K]}\hat{\eta}_{k,0}\nabla F(\mathbf{w}_0)\right) + \sum_{s=1}^{\lfloor\frac{t+K-2}{K}\rfloor}\beta^{\lfloor\frac{t+K-2}{K}\rfloor-s}\left(\sum_{j=(s-1)K+1}^{\min\{sK,t-1\}}\hat{\eta}_{k_{j-1},j}\nabla F(\mathbf{w}_j)\right)\right\|^2
\end{aligned}
$$

Let $q_t = \sum_{k\in[K]}\beta^{\lfloor\frac{t+K-2}{K}\rfloor} + \sum_{s=1}^{\lfloor\frac{t+K-2}{K}\rfloor}\sum_{j=(s-1)K+1}^{\min\{sK,t-1\}}\beta^{\lfloor\frac{t+K-2}{K}\rfloor-s}$, then we have

$$
\begin{aligned}
&\mathbb{E}\left\|\beta^{\lfloor\frac{t+K-2}{K}\rfloor}\left(\sum_{k\in[K]}\hat{\eta}_{k,0}\nabla F(\mathbf{w}_0)\right) + \sum_{s=1}^{\lfloor\frac{t+K-2}{K}\rfloor}\beta^{\lfloor\frac{t+K-2}{K}\rfloor-s}\left(\sum_{j=(s-1)K+1}^{\min\{sK,t-1\}}\hat{\eta}_{k_{j-1},j}\nabla F(\mathbf{w}_j)\right)\right\|^2 \\
&= q_t^2\mathbb{E}\left\|\left(\sum_{k\in[K]}\frac{\beta^{\lfloor\frac{t+K-2}{K}\rfloor}}{q_t}\hat{\eta}_{k,0}\nabla F(\mathbf{w}_0)\right) + \sum_{s=1}^{\lfloor\frac{t+K-2}{K}\rfloor}\left(\sum_{j=(s-1)K+1}^{\min\{sK,t-1\}}\frac{\beta^{\lfloor\frac{t+K-2}{K}\rfloor-s}}{q_t}\hat{\eta}_{k_{j-1},j}\nabla F(\mathbf{w}_j)\right)\right\|^2 \\
&\leq q_t\left[\sum_{k\in[K]}\beta^{\lfloor\frac{t+K-2}{K}\rfloor}(\hat{\eta}_{k,0})^2\left\|\nabla F(\mathbf{w}_0)\right\|^2\right. \\
&\quad\left. + \sum_{s=1}^{\lfloor\frac{t+K-2}{K}\rfloor}\sum_{j=(s-1)K+1}^{\min\{sK,t-1\}}\beta^{\lfloor\frac{t+K-2}{K}\rfloor-s}\left(\hat{\eta}_{k_{j-1},j}\right)^2\mathbb{E}\left\|\nabla F(\mathbf{w}_j)\right\|^2\right] \\
&\leq \frac{K}{1-\beta}\left[\sum_{k\in[K]}\beta^{\lfloor\frac{t+K-2}{K}\rfloor}(\hat{\eta}_{k,0})^2\left\|\nabla F(\mathbf{w}_0)\right\|^2\right. \\
&\quad\left. + \sum_{s=1}^{\lfloor\frac{t+K-2}{K}\rfloor}\sum_{j=(s-1)K+1}^{\min\{sK,t-1\}}\beta^{\lfloor\frac{t+K-2}{K}\rfloor-s}\left(\hat{\eta}_{k_{j-1},j}\right)^2\mathbb{E}\left\|\nabla F(\mathbf{w}_j)\right\|^2\right].
\end{aligned}
$$

Thus, we get that

$$
\begin{aligned}
&\mathbb{E}\left\|\hat{\mathbf{u}}_t\right\|^2 \\
&\leq 2\mathbb{E}\left\|\beta^{\lfloor\frac{t+K-2}{K}\rfloor}\left(\sum_{k\in[K]}\hat{\eta}_{k,0}\nabla F(\mathbf{w}_0)\right)+\sum_{s=1}^{\lfloor\frac{t+K-2}{K}\rfloor}\beta^{\lfloor\frac{t+K-2}{K}\rfloor-s}\left(\sum_{j=(s-1)K+1}^{\min\{sK,t-1\}}\hat{\eta}_{k_{j-1},j}\nabla F(\mathbf{w}_j)\right)\right\|^2 \\
&\quad+2\sum_{k\in[K]}\beta^{2\lfloor\frac{t+K-2}{K}\rfloor}(\hat{\eta}_{k,0})^2\sigma^2+2\sum_{s=1}^{\lfloor\frac{t+K-2}{K}\rfloor}\sum_{j=(s-1)K+1}^{\min\{sK,t-1\}}\beta^{2\lfloor\frac{t+K-2}{K}\rfloor-2s}\left(\hat{\eta}_{k_{j-1},j}\right)^2\sigma^2 \\
&\leq\frac{2K}{1-\beta}\left[\sum_{k\in[K]}\beta^{\lfloor\frac{t+K-2}{K}\rfloor}(\hat{\eta}_{k,0})^2\left\|\nabla F(\mathbf{w}_0)\right\|^2+\sum_{s=1}^{\lfloor\frac{t+K-2}{K}\rfloor}\sum_{j=(s-1)K+1}^{\min\{sK,t-1\}}\beta^{\lfloor\frac{t+K-2}{K}\rfloor-s}\left(\hat{\eta}_{k_{j-1},j}\right)^2\mathbb{E}\left\|\nabla F(\mathbf{w}_j)\right\|^2\right] \\
&\quad+2\sum_{k\in[K]}\beta^{2\lfloor\frac{t+K-2}{K}\rfloor}(\hat{\eta}_{k,0})^2\sigma^2+2\sum_{s=1}^{\lfloor\frac{t+K-2}{K}\rfloor}\sum_{j=(s-1)K+1}^{\min\{sK,t-1\}}\beta^{2\lfloor\frac{t+K-2}{K}\rfloor-2s}\left(\hat{\eta}_{k_{j-1},j}\right)^2\sigma^2.
\end{aligned}
$$

$$
\begin{aligned}
&\sum_{t=1}^{T-1}\mathbb{E}\left(\hat{\eta}_{k_{t-1},t}\left\|\hat{\mathbf{y}}_t-\hat{\mathbf{w}}_t\right\|^2\right)\leq\frac{\beta^2\eta}{(1-\beta)^2}\sum_{t=1}^{T-1}\mathbb{E}\left\|\hat{\mathbf{u}}_t\right\|^2 \\
&\leq\frac{2\beta^2K\eta}{(1-\beta)^3}\sum_{t=1}^{T-1}\sum_{k\in[K]}\beta^{\lfloor\frac{t+K-2}{K}\rfloor}(\hat{\eta}_{k,0})^2\left\|\nabla F(\mathbf{w}_0)\right\|^2 \\
&\quad+\frac{2\beta^2K\eta}{(1-\beta)^3}\sum_{t=1}^{T-1}\sum_{s=1}^{\lfloor\frac{t+K-2}{K}\rfloor}\sum_{j=(s-1)K+1}^{\min\{sK,t-1\}}\beta^{\lfloor\frac{t+K-2}{K}\rfloor-s}\left(\hat{\eta}_{k_{j-1},j}\right)^2\mathbb{E}\left\|\nabla F(\mathbf{w}_j)\right\|^2 \\
&\quad+\frac{2\beta^2\eta\sigma^2}{(1-\beta)^2}\sum_{t=1}^{T-1}\left[\sum_{k\in[K]}\beta^{2\lfloor\frac{t+K-2}{K}\rfloor}(\hat{\eta}_{k,0})^2+\sum_{s=1}^{\lfloor\frac{t+K-2}{K}\rfloor}\sum_{j=(s-1)K+1}^{\min\{sK,t-1\}}\beta^{2\lfloor\frac{t+K-2}{K}\rfloor-2s}\left(\hat{\eta}_{k_{j-1},j}\right)^2\right] \\
&\leq\frac{2\beta^2\eta K^2}{(1-\beta)^4}\left[\sum_{k\in[K]}(\hat{\eta}_{k,0})^2\left\|\nabla F(\mathbf{w}_0)\right\|^2+\sum_{t=1}^{T-1}\left(\hat{\eta}_{k_{t-1},t}\right)^2\mathbb{E}\left\|\nabla F(\mathbf{w}_t)\right\|^2\right] \\
&\quad+\frac{2\beta^2\eta K\sigma^2}{(1-\beta)^3}\left[\sum_{k\in[K]}(\hat{\eta}_{k,0})^2+\sum_{t=1}^{T-1}\left(\hat{\eta}_{k_{t-1},t}\right)^2\right].
\end{aligned}
$$

□

Lemma 11. *With Assumption 1, letting $\eta\leq\frac{1}{8KL}$, the gap between $\hat{\mathbf{w}}_t$ and $\mathbf{w}_t$ in OrMo-DA can be bounded as follows:*

$$
\begin{aligned}
\sum_{t=1}^{T-1}\mathbb{E}\left(\hat{\eta}_{k_{t-1},t}\left\|\hat{\mathbf{w}}_t-\mathbf{w}_t\right\|^2\right)\leq\frac{\eta K}{2L(1-\beta)^2}&\left[\sum_{t=1}^{T-1}\hat{\eta}_{k_{t-1},t}\mathbb{E}\left\|\nabla F(\mathbf{w}_t)\right\|^2+\left(\sum_{k\in[K]}\hat{\eta}_{k,0}\right)\left\|\nabla F(\mathbf{w}_0)\right\|^2\right] \\
&+\frac{\eta\sigma^2}{2L(1-\beta)^2}\left(\sum_{t=1}^{T-1}\hat{\eta}_{k_{t-1},t}+\sum_{k\in[K]}\hat{\eta}_{k,0}\right).
\end{aligned}
$$

Proof.

$$
\begin{aligned}
&\sum_{t=1}^{T-1}\mathbb{E}\left(\hat{\eta}_{k_{t-1},t}\left\|\hat{\mathbf{w}}_t-\mathbf{w}_t\right\|^2\right)\leq\eta\sum_{t=1}^{T-1}\mathbb{E}\left\|\hat{\mathbf{w}}_t-\mathbf{w}_t\right\|^2\\
&=\eta\sum_{t=1}^{T-1}\mathbb{E}\left\|\sum_{k\in[K],k\neq k_{t-1}}\frac{1-\beta^{\lceil\frac{t-1}{K}\rceil-\lceil\frac{ite(k,t-1)}{K}\rceil+1}}{1-\beta}\hat{\eta}_{k,ite(k,t-1)}\mathbf{g}^k_{ite(k,t-1)}\right\|^2\\
&\leq 2\eta\sum_{t=1}^{T-1}\mathbb{E}\left\|\sum_{k\in[K],k\neq k_{t-1}}\frac{1-\beta^{\lceil\frac{t-1}{K}\rceil-\lceil\frac{ite(k,t-1)}{K}\rceil+1}}{1-\beta}\hat{\eta}_{k,ite(k,t-1)}\left(\mathbf{g}^k_{ite(k,t-1)}-\nabla F(\mathbf{w}_{ite(k,t-1)})\right)\right\|^2\\
&\quad+2\eta\sum_{t=1}^{T-1}\mathbb{E}\left\|\sum_{k\in[K],k\neq k_{t-1}}\frac{1-\beta^{\lceil\frac{t-1}{K}\rceil-\lceil\frac{ite(k,t-1)}{K}\rceil+1}}{1-\beta}\hat{\eta}_{k,ite(k,t-1)}\nabla F(\mathbf{w}_{ite(k,t-1)})\right\|^2\\
&\leq\frac{2\eta}{(1-\beta)^2}\sum_{t=0}^{T-2}\sum_{k\in[K],k\neq k_t}\left(\hat{\eta}_{k,ite(k,t)}\right)^2\left(\sigma^2+K\mathbb{E}\left\|\nabla F(\mathbf{w}_{ite(k,t)})\right\|^2\right)\\
&=\frac{2\eta}{(1-\beta)^2}\sum_{j=0}^{T-2}\sum_{t=0}^{T-2}\sum_{k\in[K]}\left(\hat{\eta}_{k,ite(k,t)}\right)^2\left(\sigma^2+K\mathbb{E}\left\|\nabla F(\mathbf{w}_{ite(k,t)})\right\|^2\right)\mathbb{1}\left(k\neq k_t\right)\mathbb{1}\left(j=ite(k,t)\right)\\
&=\frac{2\eta}{(1-\beta)^2}\sum_{j=1}^{T-2}\sum_{t=0}^{T-2}\sum_{k\in[K]}\left(\hat{\eta}_{k,ite(k,t)}\right)^2\left(\sigma^2+K\mathbb{E}\left\|\nabla F(\mathbf{w}_{ite(k,t)})\right\|^2\right)\mathbb{1}\left(k\neq k_t\right)\mathbb{1}\left(j=ite(k,t)\right)\\
&\quad+\frac{2\eta}{(1-\beta)^2}\sum_{t=0}^{T-2}\sum_{k\in[K]}\left(\hat{\eta}_{k,ite(k,t)}\right)^2\left(\sigma^2+K\mathbb{E}\left\|\nabla F(\mathbf{w}_{ite(k,t)})\right\|^2\right)\mathbb{1}\left(k\neq k_t\right)\mathbb{1}\left(0=ite(k,t)\right)\\
&\leq\frac{2\eta}{(1-\beta)^2}\sum_{j=1}^{T-2}\left(\hat{\eta}_{k_{j-1},j}\right)^2\left(\sigma^2+K\mathbb{E}\left\|\nabla F(\mathbf{w}_j)\right\|^2\right)(next(k_{j-1},j)-j)\\
&\quad+\frac{2\eta}{(1-\beta)^2}\sum_{k\in[K]}\left(\hat{\eta}_{k,0}\right)^2\left(\sigma^2+K\left\|\nabla F(\mathbf{w}_0)\right\|^2\right)(next(k,0)-0)
\end{aligned}
$$

$$
\begin{aligned}
\sum_{t=1}^{T-1}\mathbb{E}\left(\hat{\eta}_{k_{t-1},t}\left\|\hat{\mathbf{w}}_t-\mathbf{w}_t\right\|^2\right)\leq&\frac{2\eta}{(1-\beta)^2}\sum_{t=1}^{T-1}\left(\hat{\eta}_{k_{t-1},t}\right)^2\left(\sigma^2+K\mathbb{E}\left\|\nabla F(\mathbf{w}_t)\right\|^2\right)\hat{\tau}_{k_{t-1},next(k_{t-1},t)}\\
&+\frac{2\eta}{(1-\beta)^2}\sum_{k\in[K]}\left(\hat{\eta}_{k,0}\right)^2\left(\sigma^2+K\left\|\nabla F(\mathbf{w}_0)\right\|^2\right)\hat{\tau}_{k,next(k,0)}
\end{aligned}
$$

$$
\begin{aligned}
&\sum_{t=1}^{T-1}\mathbb{E}\left(\hat{\eta}_{k_{t-1},t}\left\|\hat{\mathbf{w}}_t-\mathbf{w}_t\right\|^2\right)\\
&\leq\frac{\eta}{2L(1-\beta)^2}\left[\sum_{t=1}^{T-1}\hat{\eta}_{k_{t-1},t}\left[\sigma^2+K\mathbb{E}\left\|\nabla F(\mathbf{w}_t)\right\|^2\right]+\sum_{k\in[K]}\hat{\eta}_{k,0}\left[\sigma^2+K\left\|\nabla F(\mathbf{w}_0)\right\|^2\right]\right]\\
&\leq\frac{\eta K}{2L(1-\beta)^2}\left[\sum_{t=1}^{T-1}\hat{\eta}_{k_{t-1},t}\mathbb{E}\left\|\nabla F(\mathbf{w}_t)\right\|^2+\left(\sum_{k\in[K]}\hat{\eta}_{k,0}\right)\left\|\nabla F(\mathbf{w}_0)\right\|^2\right]\\
&\quad+\frac{\eta\sigma^2}{2L(1-\beta)^2}\left(\sum_{t=1}^{T-1}\hat{\eta}_{k_{t-1},t}+\sum_{k\in[K]}\hat{\eta}_{k,0}\right).
\end{aligned}
$$

□

Theorem 3. *With Assumptions 1, 3 and 4, letting*

$$\eta_t = \begin{cases} \eta & \tau_t \leq 2K \\ \min\{\eta, \dfrac{1}{4L\tau_t}\} & \tau_t > 2K, \end{cases}$$

and $\eta = \min\{\frac{(1-\beta)^2}{8KL}, \sqrt{\frac{(1-\beta)^3\Delta}{TL\sigma^2}}\}$*, Algorithm 5 has the following convergence rate:*

$$\mathbb{E}\left\|\nabla F(\bar{\mathbf{w}}_T)\right\|^2 \leq \mathcal{O}(\sqrt{\frac{L\sigma^2}{T}} + \frac{KL}{T}).$$

where $\bar{\mathbf{w}}_T$ *is randomly chosen from* $\{\mathbf{w}_0, \mathbf{w}_1, \cdots, \mathbf{w}_{T-1}\}$ *according to (17).*

Proof. According to Lemma 9, we can get that

$$\begin{aligned}
&\mathbb{E}F(\hat{\mathbf{y}}_T) - \mathbb{E}F(\hat{\mathbf{y}}_1) \\
&\leq \sum_{t=1}^{T-1}\left[\left(\frac{L(\hat{\eta}_{k_{t-1},t})^2}{2(1-\beta)^2} - \frac{\hat{\eta}_{k_{t-1},t}}{2(1-\beta)}\right)\mathbb{E}\left\|\nabla F(\mathbf{w}_t)\right\|^2\right] + \frac{L\sigma^2}{2(1-\beta)^2}\sum_{t=1}^{T-1}(\hat{\eta}_{k_{t-1},t})^2 \\
&\quad + \mathbb{E}\left[\frac{L^2}{1-\beta}\sum_{t=1}^{T-1}\left(\hat{\eta}_{k_{t-1},t}\left\|\hat{\mathbf{w}}_t - \mathbf{w}_t\right\|^2 + \hat{\eta}_{k_{t-1},t}\left\|\hat{\mathbf{y}}_t - \hat{\mathbf{w}}_t\right\|^2\right)\right] \\
&\leq \sum_{t=1}^{T-1}\left[\left(\frac{L(\hat{\eta}_{k_{t-1},t})^2}{2(1-\beta)^2} - \frac{\hat{\eta}_{k_{t-1},t}}{2(1-\beta)}\right)\mathbb{E}\left\|\nabla F(\mathbf{w}_t)\right\|^2\right] + \frac{L\sigma^2}{2(1-\beta)^2}\sum_{t=1}^{T-1}(\hat{\eta}_{k_{t-1},t})^2 \\
&\quad + \frac{\eta L\sigma^2}{2(1-\beta)^3}\left(\sum_{t=1}^{T-1}\hat{\eta}_{k_{t-1},t} + \sum_{k\in[K]}\hat{\eta}_{k,0}\right) \\
&\quad + \frac{\eta KL}{2(1-\beta)^3}\left[\sum_{t=1}^{T-1}\hat{\eta}_{k_{t-1},t}\mathbb{E}\left\|\nabla F(\mathbf{w}_t)\right\|^2 + \left(\sum_{k\in[K]}\hat{\eta}_{k,0}\right)\left\|\nabla F(\mathbf{w}_0)\right\|^2\right] \\
&\quad + \frac{2\beta^2\eta K^2L^2}{(1-\beta)^5}\left[\sum_{k\in[K]}(\hat{\eta}_{k,0})^2\left\|\nabla F(\mathbf{w}_0)\right\|^2 + \sum_{t=1}^{T-1}\left(\hat{\eta}_{k_{t-1},t}\right)^2\mathbb{E}\left\|\nabla F(\mathbf{w}_t)\right\|^2\right] \\
&\quad + \frac{2\beta^2\eta KL^2\sigma^2}{(1-\beta)^4}\left[\sum_{k\in[K]}(\hat{\eta}_{k,0})^2 + \sum_{t=1}^{T-1}\left(\hat{\eta}_{k_{t-1},t}\right)^2\right] \\
&\leq \sum_{t=1}^{T-1}\left[\left(\frac{L(\hat{\eta}_{k_{t-1},t})^2}{2(1-\beta)^2} - \frac{\hat{\eta}_{k_{t-1},t}}{2(1-\beta)}\right)\mathbb{E}\left\|\nabla F(\mathbf{w}_t)\right\|^2\right] + \frac{L\sigma^2}{2(1-\beta)^2}\sum_{t=1}^{T-1}(\hat{\eta}_{k_{t-1},t})^2 \\
&\quad + \left(\frac{\eta L\sigma^2}{2(1-\beta)^3} + \frac{\eta\beta^2L\sigma^2}{4(1-\beta)^2}\right)\left(\sum_{t=1}^{T-1}\hat{\eta}_{k_{t-1},t} + \sum_{k\in[K]}\hat{\eta}_{k,0}\right) \\
&\quad + \frac{\eta KL}{2(1-\beta)^3}\left[\sum_{t=1}^{T-1}\hat{\eta}_{k_{t-1},t}\mathbb{E}\left\|\nabla F(\mathbf{w}_t)\right\|^2 + \sum_{k\in[K]}\hat{\eta}_{k,0}\left\|\nabla F(\mathbf{w}_0)\right\|^2\right] \\
&\quad + \frac{\beta^2\eta KL}{4(1-\beta)^3}\left[\sum_{k\in[K]}\hat{\eta}_{k,0}\left\|\nabla F(\mathbf{w}_0)\right\|^2 + \sum_{t=1}^{T-1}\hat{\eta}_{k_{t-1},t}\mathbb{E}\left\|\nabla F(\mathbf{w}_t)\right\|^2\right]
\end{aligned}$$

$$
\begin{aligned}
&\leq \sum_{t=1}^{T-1}\left[\left(\frac{L(\hat{\eta}_{k_{t-1},t})^2}{2(1-\beta)^2}-\frac{\hat{\eta}_{k_{t-1},t}}{2(1-\beta)}\right)\mathbb{E}\left\|\nabla F(\mathbf{w}_t)\right\|^2\right]+\frac{L\sigma^2}{2(1-\beta)^2}\sum_{t=1}^{T-1}(\hat{\eta}_{k_{t-1},t})^2\\
&\quad+\frac{\eta L\sigma^2}{(1-\beta)^3}\left(\sum_{t=1}^{T-1}\hat{\eta}_{k_{t-1},t}+\sum_{k\in[K]}\hat{\eta}_{k,0}\right)\\
&\quad+\frac{\eta KL}{(1-\beta)^3}\left[\sum_{t=1}^{T-1}\hat{\eta}_{k_{t-1},t}\mathbb{E}\left\|\nabla F(\mathbf{w}_t)\right\|^2+\sum_{k\in[K]}\hat{\eta}_{k,0}\left\|\nabla F(\mathbf{w}_0)\right\|^2\right]\\
&\leq \sum_{t=1}^{T-1}\left[\left(\frac{L(\hat{\eta}_{k_{t-1},t})^2}{2(1-\beta)^2}-\frac{\hat{\eta}_{k_{t-1},t}}{2(1-\beta)}\right)\mathbb{E}\left\|\nabla F(\mathbf{w}_t)\right\|^2\right]+\frac{L\sigma^2}{2(1-\beta)^2}\sum_{t=1}^{T-1}(\hat{\eta}_{k_{t-1},t})^2\\
&\quad+\frac{\eta L\sigma^2}{(1-\beta)^3}\left(\sum_{t=1}^{T-1}\hat{\eta}_{k_{t-1},t}+\sum_{k\in[K]}\hat{\eta}_{k,0}\right)\\
&\quad+\frac{1}{4(1-\beta)}\left[\sum_{t=1}^{T-1}\hat{\eta}_{k_{t-1},t}\mathbb{E}\left\|\nabla F(\mathbf{w}_t)\right\|^2+\sum_{k\in[K]}\hat{\eta}_{k,0}\left\|\nabla F(\mathbf{w}_0)\right\|^2\right]
\end{aligned}
$$

$$
\begin{aligned}
\mathbb{E}F(\hat{\mathbf{y}}_1)&\leq F(\mathbf{w}_0)-\frac{1}{1-\beta}\mathbb{E}\langle\nabla F(\mathbf{w}_0),\left(\sum_{k\in[K]}\hat{\eta}_{k,0}\mathbf{g}_0^k\right)\rangle+\frac{L}{2(1-\beta)^2}\mathbb{E}\left\|\sum_{k\in[K]}\hat{\eta}_{k,0}\mathbf{g}_0^k\right\|^2\\
&=F(\mathbf{w}_0)-\frac{\sum_{k\in[K]}\hat{\eta}_{k,0}}{1-\beta}\left\|\nabla F(\mathbf{w}_0)\right\|^2\\
&\quad+\frac{L}{2(1-\beta)^2}\mathbb{E}\left\|\sum_{k\in[K]}\hat{\eta}_{k,0}\left(\mathbf{g}_0^k-\nabla F(\mathbf{w}_0)+\nabla F(\mathbf{w}_0)\right)\right\|^2\\
&\leq F(\mathbf{w}_0)+\left[\frac{L\left(\sum_{k\in[K]}\hat{\eta}_{k,0}\right)^2}{2(1-\beta)^2}-\frac{\sum_{k\in[K]}\hat{\eta}_{k,0}}{1-\beta}\right]\left\|\nabla F(\mathbf{w}_0)\right\|^2+\frac{L\sigma^2}{2(1-\beta)^2}\sum_{k\in[K]}(\hat{\eta}_{k,0})^2
\end{aligned}
$$

$$
\begin{aligned}
&\mathbb{E}F(\hat{\mathbf{y}}_T)-F(\mathbf{w}_0)\\
&\leq \sum_{t=1}^{T-1}\left[\left(\frac{L(\hat{\eta}_{k_{t-1},t})^2}{2(1-\beta)^2}-\frac{\hat{\eta}_{k_{t-1},t}}{2(1-\beta)}\right)\mathbb{E}\left\|\nabla F(\mathbf{w}_t)\right\|^2\right]+\frac{L\sigma^2}{2(1-\beta)^2}\sum_{t=1}^{T-1}(\hat{\eta}_{k_{t-1},t})^2\\
&\quad+\frac{\eta L\sigma^2}{(1-\beta)^3}\left(\sum_{t=1}^{T-1}\hat{\eta}_{k_{t-1},t}+\sum_{k\in[K]}\hat{\eta}_{k,0}\right)\\
&\quad+\frac{1}{4(1-\beta)}\left[\sum_{t=1}^{T-1}\hat{\eta}_{k_{t-1},t}\mathbb{E}\left\|\nabla F(\mathbf{w}_t)\right\|^2+\sum_{k\in[K]}\hat{\eta}_{k,0}\left\|\nabla F(\mathbf{w}_0)\right\|^2\right]\\
&\quad+\left[\frac{L\left(\sum_{k\in[K]}\hat{\eta}_{k,0}\right)^2}{2(1-\beta)^2}-\frac{\sum_{k\in[K]}\hat{\eta}_{k,0}}{1-\beta}\right]\left\|\nabla F(\mathbf{w}_0)\right\|^2+\frac{L\sigma^2}{2(1-\beta)^2}\sum_{k\in[K]}(\hat{\eta}_{k,0})^2\\
&\leq \sum_{t=1}^{T-1}\left[\left(\frac{L(\hat{\eta}_{k_{t-1},t})^2}{2(1-\beta)^2}-\frac{\hat{\eta}_{k_{t-1},t}}{4(1-\beta)}\right)\mathbb{E}\left\|\nabla F(\mathbf{w}_t)\right\|^2\right]+\frac{2\eta L\sigma^2}{(1-\beta)^3}\left(\sum_{t=1}^{T-1}\hat{\eta}_{k_{t-1},t}+\sum_{k\in[K]}\hat{\eta}_{k,0}\right)\\
&\quad+\left(\frac{L\left(\sum_{k\in[K]}\hat{\eta}_{k,0}\right)^2}{2(1-\beta)^2}-\frac{\sum_{k\in[K]}\hat{\eta}_{k,0}}{4\left(1-\beta\right)}\right)\left\|\nabla F(\mathbf{w}_0)\right\|^2
\end{aligned}
$$

$$\mathbb{E}F(\hat{\mathbf{y}}_T) - F(\mathbf{w}_0) \leq -\frac{1}{8\left(1-\beta\right)}\left[\sum_{t=1}^{T-1}\hat{\eta}_{k_{t-1},t}\mathbb{E}\left\|\nabla F(\mathbf{w}_t)\right\|^2 + \sum_{k\in[K]}\hat{\eta}_{k,0}\left\|\nabla F(\mathbf{w}_0)\right\|^2\right] + \frac{2\eta L\sigma^2}{\left(1-\beta\right)^3}\left(\sum_{t=1}^{T-1}\hat{\eta}_{k_{t-1},t} + \sum_{k\in[K]}\hat{\eta}_{k,0}\right)$$

Thus, we can get that

$$\frac{1}{\sum_{t=0}^{T-1}\bar{\eta}_t}\mathbb{E}\left(\sum_{t=0}^{T-1}\bar{\eta}_t\left\|\nabla F(\mathbf{w}_t)\right\|^2\right) \leq \frac{8\left(1-\beta\right)\left(F(\mathbf{w}_0)-F^*\right)}{\sum_{t=0}^{T-1}\bar{\eta}_t} + \frac{16\eta L\sigma^2}{\left(1-\beta\right)^2},$$

where $\bar{\eta}_0 = \sum_{k\in[K]}\hat{\eta}_{k,0}$ and $\bar{\eta}_t = \hat{\eta}_{k_{t-1},t}(t \geq 1)$.

Then, we analyse the lower bound of $\sum_{t=0}^{T-1}\bar{\eta}_t$.

It's easy to verify that

$$\hat{\tau}_{k,t+1} = t+1-ite(k,t+1) = \begin{cases} t-ite(k,t)+1 = \hat{\tau}_{k,t}+1 & k \neq k_t, \\ 0 = \hat{\tau}_{k,t} - \tau_t & k = k_t. \end{cases}$$

Since $\sum_{k\in[K]}\hat{\tau}_{k,0} = 0$, we have $\sum_{k\in[K]}\hat{\tau}_{k,T} + \sum_{t=0}^{T-1}\tau_t = (K-1)T$. Moreover, $\hat{\tau}_{k_{T-1},T} = T - ite(k_{T-1},T) = 0$. We get that $\sum_{k\in[K],k\neq k_{T-1}}\hat{\tau}_{k,T} + \sum_{t=0}^{T-1}\tau_t = (K-1)T$. Thus, at least $\frac{T}{2}$ delays are smaller than $2K$. $\sum_{t=0}^{T}\bar{\eta}_t = \sum_{t=1}^{T-1}\hat{\eta}_{k_{t-1},t} + \sum_{k\in[K]}\hat{\eta}_{k,0} \geq \frac{T\eta}{2}$.

$$\frac{1}{\sum_{t=0}^{T-1}\bar{\eta}_t}\sum_{t=0}^{T-1}\bar{\eta}_t\mathbb{E}\left\|\nabla F(\mathbf{w}_t)\right\|^2 \leq \frac{16(1-\beta)}{T\eta}\left(F(\mathbf{w}_0)-F^*\right) + \frac{16\eta L\sigma^2}{\left(1-\beta\right)^2}.$$

If we choose an output $\bar{\mathbf{w}}_T$ from $\{\mathbf{w}_0, \mathbf{w}_1, \cdots, \mathbf{w}_{T-1}\}$ according to

$$\mathbb{P}(\bar{\mathbf{w}}_T = \mathbf{w}_t) \propto \bar{\eta}_t, \tag{17}$$

where $t \in [T]$ and let $\eta = \min\{\frac{(1-\beta)^2}{8KL}, \sqrt{\frac{(1-\beta)^3(F(\mathbf{w}_0)-F^*)}{TL\sigma^2}}\}$, we have that

$$\mathbb{E}\left\|\nabla F(\bar{\mathbf{w}}_T)\right\|^2 \leq \mathcal{O}\left(\sqrt{\frac{L\sigma^2}{T}} + \frac{KL}{T}\right).$$

□

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope. We propose OrMo for ASGD in Subsection 3.2. We prove the convergence of OrMo theoretically in Subsection 3.3 and verify its superiority empirically in Section 4.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of the assumptions of OrMo in Subsection 3.3.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The full set of assumptions and the proof sketch can be found in Section 3.3. Please refer to Appendix C for the complete proof details.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: This paper fully discloses all the information needed to reproduce the main experimental results. The implementation details can be found in Section 4.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide the code. The dataset is public.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The implementation details can be found in Section 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Each experiment is repeated 5 times as reported in Section 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The type of compute workers and the time of execution are presented in Section 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research conforms with the NeurIPS Code of Ethics in every respect.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: There is no societal impact of the work performed.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have cited the original papers.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: This paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.