EVENT-DRIVEN ONLINE VERTICAL FEDERATED LEARNING

Ganyu Wang[1] **Boyu Wang**[1,2] **Bin Gu**[3*] **Charles Ling**[1,2*]
[1]Western University [2]Vector Institute [3]Jilin University
gwang382@uwo.ca bwang@csd.uwo.ca jsgubin@gmail.com
charles.ling@uwo.ca

ABSTRACT

Online learning is more adaptable to real-world scenarios in Vertical Federated Learning (VFL) compared to offline learning. However, integrating online learning into VFL presents challenges due to the unique nature of VFL, where clients possess non-intersecting feature sets for the same sample. In real-world scenarios, the clients may not receive data streaming for the disjoint features for the same entity synchronously. Instead, the data are typically generated by an *event* relevant to only a subset of clients. We are the first to identify these challenges in online VFL, which have been overlooked by previous research. To address these challenges, we proposed an event-driven online VFL framework. In this framework, only a subset of clients were activated during each event, while the remaining clients passively collaborated in the learning process. Furthermore, we incorporated *dynamic local regret (DLR)* into VFL to address the challenges posed by online learning problems with non-convex models within a non-stationary environment. We conducted a comprehensive regret analysis of our proposed framework, specifically examining the DLR under non-convex conditions with event-driven online VFL. Extensive experiments demonstrated that our proposed framework was more stable than the existing online VFL framework under non-stationary data conditions while also significantly reducing communication and computation costs.

1 INTRODUCTION

Vertical Federated Learning (VFL) (Vepakomma et al., 2018; Yang et al., 2019; Liu et al., 2019; Chen et al., 2020; Gu et al., 2020; Zhang et al., 2021b;a; Wang et al., 2023; Qi et al., 2022; Wang et al., 2024) is a privacy-preserving machine learning paradigm wherein multiple entities collaborate to construct a model without sharing their raw data. In VFL, each participant possesses non-intersecting features for the same set of samples, which is significantly different from the Horizontal Federated Learning (HFL) (McMahan et al., 2017; Karimireddy et al., 2020; Li et al., 2020; 2021; Marfoq et al., 2022; Mishchenko et al., 2019) where each client possesses the non-overlap samples of the same features.[1]

Current research on VFL primarily focuses on the offline scenario, characterized by a pre-established dataset. However, the limitations of offline learning become obvious when building real-world applications of VFL. First, offline learning is unsuitable for scenarios where the dataset undergoes continual updates, which is typical in real-world applications. For example, in the application scenario of VFL scenarios involving companies as clients (Wei et al., 2022; Vepakomma et al., 2018), new data is constantly generated as new customers engage with the companies, or as existing customers update their records through ongoing activities. Similarly, in VFL scenarios involving edge devices (Wang & Xu, 2023; Liu et al., 2022), the sensors, acting as the clients, continuously receive data streams from the environment rather than maintaining a static dataset. Second, the dynamic nature of real-world environments leads to data distribution drift, which is particularly evident in edge devices. In response, the offline learning paradigm requires retraining the model from scratch to accommodate shifts in data distribution. While this retraining process may not present significant

[*] Corresponding authors.

[1] The term Federated Learning commonly refers to HFL. However, this is not the setting of this study.

challenges in centralized learning, it becomes prohibitively expensive in distributed learning scenarios, as the training process imposes substantial communication and computation costs in distributed learning. Consequently, online learning may provide greater adaptability in VFL by allowing models to update continuously as new data arrives and handle dynamic environments.

However, applying online learning to VFL is not straightforward due to its inherent nature. First, in online VFL, clients receive non-intersecting features of the data from the environment. In real-world scenarios, it is rare for all clients to receive all these features of a sample simultaneously. Instead, it is more common for only a subset of clients to obtain the relevant features in response to a specific *event*. For example, in VFL implementations within large companies (Wei et al., 2022; Hu et al., 2019; Vepakomma et al., 2018), when a customer takes an action such as making a payment or a purchase, this action typically involves only one company of the VFL, while the data from the other companies remain unchanged. Similarly, in VFL involving sensor networks (Wang & Xu, 2023; Liu et al., 2022), only the sensor triggered by an event will be activated, while others remain inactive (Suh, 2007; Heemels et al., 2012; Trimpe & D'Andrea, 2014; Beuchert et al., 2020). The above scenario has brought about the demand for an event-driven online VFL framework, wherein certain participants are activated by events during each round, thereby dominating the learning process, while the rest of the participants remain inactive or passively cooperate with the learning.

The second challenge in online VFL lies in addressing non-convex models and dynamic environments, which are prevalent in practical applications. Current research in online VFL still primarily focuses on online convex optimization, assuming both convex models and stationary data streams (Wang & Xu, 2023). While convex models are easier to optimize, they fail to capture the complex patterns necessary for tackling more challenging tasks. Additionally, the assumption of stationary data streams is unrealistic in dynamic real-world environments, where data distributions can shift over time. For example, the environmental sensor that monitors the air quality may experience dynamic change due to natural phenomena. These are challenges that current online VFL frameworks have yet to resolve.

To address the aforementioned challenges, we propose a novel event-driven online VFL framework, which is well-suited for the online learning scenario in non-convex cases and non-stationary environments. Figure 1 depicts a schematic graph of our framework. In our framework, a subset of the clients are activated by the event at each round, while others passively contribute to the training. This approach substantially reduces communication-computation costs and facilitates the client model in learning relevant content. Moreover, we adapt the dynamic local regret approach proposed by Aydore et al. (2019) to our event-driven online VFL framework to effectively handle online learning in non-convex cases and non-stationary environments.

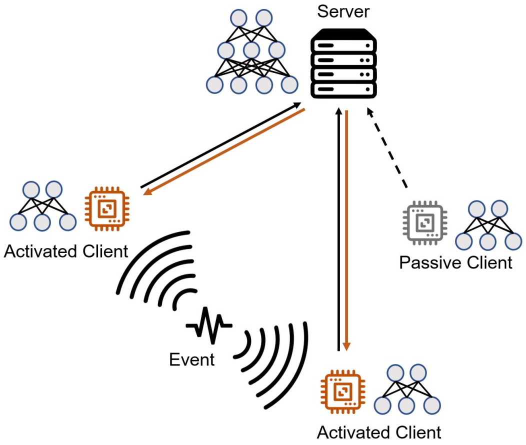

Figure 1: Event-driven online VFL

In summary, the contributions of our paper are:

- We identify the unrealistic assumption of synchronous data reception in online VFL research and propose a novel event-driven online VFL paradigm that is better suited to real-world scenarios.
- We adapt the dynamic local regret approach to our event-driven online VFL to effectively handle non-convex models in non-stationary data streaming scenarios, and we theoretically prove the dynamic local regret bound for this framework, which incorporates partial activation of the client.
- Our experiments demonstrate that our event-driven online VFL exhibits greater stability compared to existing methods when confronted with non-stationary data conditions. Additionally, it significantly reduces communication and computation costs.

Notation We use a square bracket with multiple items to denote concatenation for convenience. For instance, given $w_A \in \mathbb{R}^{d_A}$, $w_B \in \mathbb{R}^{d_B}$, we define $[w_A, w_B] \triangleq [w_A^\top, w_B^\top]^\top$. The superscript t

attached to the parameter w indicates the number of rounds (time step). A square bracket enclosing a single integer represents the set of natural numbers from 1 to that particular number. For instance, $[M] = \{1, 2, \ldots, M\}$.

2 Related work: online HFL and online VFL[2]

Online HFL Most existing online federated learning research focuses on HFL because extending the HFL framework to online learning is relatively straightforward. In HFL, each participant possesses the complete feature set of the local sample. Therefore, online HFL can be easily achieved by assigning each client a unique data stream containing non-overlapping samples. The current research on online HFL is focused on speeding up optimization (Mitra et al., 2021; Eshraghi & Liang, 2020), reducing communication cost (Hong & Chae, 2021) and dealing with concept drift (Ganguly & Aggarwal, 2023). Mitra et al. (2021) applied online mirror descent within the Federated Learning framework, demonstrating sub-linear regret in convex scenarios. Hong & Chae (2021) introduced a randomized multi-kernel algorithm for online federated learning, which maintains the performance of the multi-kernel algorithm while mitigating the linearly increasing communication cost. Kwon et al. (2023) incorporated client sampling and quantization into online federated learning. Ganguly & Aggarwal (2023) proposed a non-stationary detection and restart algorithm for online federated learning, addressing the concept of drift during online learning.

Online VFL In VFL, each client possesses a non-overlapping feature set, which presents challenges when integrating online learning into this framework. The existing approach to Online VFL is proposed by Wang & Xu (2023), which applies online convex optimization to synchronous VFL. However, they naïvely assume that all clients receive a synchronous data stream, which does not align with real-world applications. Apart from this study, no other research has explored online VFL and the characteristics of dataset streaming in this context. Through the exploration of event-driven mechanisms, we open up new possibilities for real-time data streaming processing across distributed nodes within the VFL framework.

3 Method

3.1 Problem definition

In the VFL framework, there is a single server and M clients. The server produces the label y^t at round t, while each client may receive non-intersecting features x_m^t from the environment during the same round. The model for client m, denoted as $h_m(w_m; x_m^t)$, is parameterized by $w_m \in \mathbb{R}^{d_m}$ and takes the local feature x_m^t as input to produce an embedding. We denote $\mathbf{w}$ as the concatenation of the parameter from all clients, i.e. $\mathbf{w} = [w_1, \cdots, w_M]$. The server, parameterized by w_0, recieves embeddings $h_m(\cdot)$ from all clients and then calculates the losses with the label y^t. We define the VFL framework in composite form.

$$f^t(w_0, \mathbf{w}, x^t; y^t) = f\left(w_0, h_1(w_1; x_1^t), \cdots, h_M(w_M; x_M^t); y^t\right) \tag{1}$$

where $f(\cdot)$ denotes the model on the server. For brevity, we denote $f^t(w_0, \mathbf{w}, x^t; y^t)$ by $f^t(w_0, \mathbf{w})$ throughout all following sections.

Following the work from Aydore et al. (2019), we employ dynamic local regret analysis in online non-convex optimization. To reformulate the dynamic local regret under the context of the VFL framework, we begin by introducing the concept of exponentially weighted sliding-window average as the basis for its computation.

Definition 1. ***Exponential weighted sliding-window average:*** *Let $w_0^t \in \mathbb{R}^{d_0}$ denote the server's parameter at time t, $w_m^t \in \mathbb{R}^{d_m}$ be the client m's parameter at time t. $\mathbf{w} = [w_1, w_2 \cdots w_M]$. Let l denote the length of the sliding window. Then the exponential weighted sliding-window average can be defined as follows:*

$$S_{t,l,\alpha}(w_0^t, \mathbf{w}^t) \triangleq \frac{1}{W} \sum_{i=0}^{l-1} \alpha^i f^{t-i}(w_0^{t-i}, \mathbf{w}^{t-i}) \tag{2}$$

[2] Extra discussion on the related work of VFL and online learning is in Appendix E.

where $0 < \alpha < 1$ and the superscript i of the α^i indicates the exponent. $W = \sum_{i=0}^{l-1} \alpha^i$ serves as the normalization parameter for the exponential average, ensuring that $\frac{1}{W}\sum_{i=0}^{l-1} \alpha^i = 1$. It is worth noting that this window gives more weight to recent values, with the weight decaying exponentially, and the loss for $f^{t-i}(w_0^{t-i}, \mathbf{w}^{t-i})$ is computed on the past parameter at round $t - i$.

Then, the DLR can be formally defined based on the accumulated square norm of the gradient of the exponentially weighted sliding-window average.

Definition 2. ***Dynamic l-local regret:*** *Let $S_{t,l,\alpha}(w_0^t, \mathbf{w}^t)$ be the sliding-window defined above, w_0 be the server's parameter and $\mathbf{w}$ be the aggregated clients' parameter. The Dynamic l-Local Regret can be defined as:*

$$DLR_l(T) \triangleq \sum_{t=1}^{T} \left\| \nabla S_{t,l,\alpha}(w_0^t, \mathbf{w}^t) \right\|^2 \tag{3}$$

3.2 Adapt DLR to online VFL

We integrate the dynamic exponentially time-smoothed online gradient descent method introduced by Aydore et al. (2019) into the VFL framework by incorporating a buffer to store the past gradients.

Server update Based on the special characteristic of the dynamic local regret, the server is required to maintain a buffer of the past intermediate derivative values of length l. At each time step, the server computes the gradient and updates the buffer by enqueuing the latest gradient and dequeuing the oldest. Subsequently, the server utilizes the buffer to compute the dynamic exponentially time-smoothed gradient. Specifically, the partial derivative w.r.t. the server is shown in Eq. 4, and the buffer stores l past gradients.

$$\nabla_{w_0} S_{t,l,\alpha}(w_0^t, \mathbf{w}^t) = \frac{1}{W} \sum_{i=0}^{l-1} \underbrace{\alpha^i \nabla_{w_0} f^{t-i}(w_0^{t-i}, \mathbf{w}^{t-i})}_{\text{Server Buffer, } i = 0, 1, \cdots l-1} \tag{4}$$

Finally, the server updated its model with stochastic gradient descent, i.e. $w_0^{t+1} \leftarrow w_0^t - \eta_0 \cdot \nabla_{w_0} S_{t,l,\alpha}(w_0^t, \mathbf{w}^t)$, where η_0 is the learning rate for the server.

Client update The client cannot calculate the partial derivative w.r.t. its model by themselves because they do not hold the label. Consequently, they depend on the server to transmit the partial derivative $v_m^t = \frac{\partial f^t(w_0^t, \mathbf{w}^t)}{\partial h_m(w_m^t; x_m^t)}$ w.r.t. the client's model output h_m to facilitate model updates. The partial derivative w.r.t. the client m's model is computed through chain rules:

$$\begin{aligned}
\nabla_{w_m} \tilde{S}_{t,l,\alpha}(w_0^t, \mathbf{w}^t) =& \frac{1}{W} \sum_{i=0}^{l-1} \alpha^i \nabla_{w_m} f^{t-i}(w_0^{t-i}, \mathbf{w}^{t-i}) \\
=& \frac{1}{W} \sum_{i=0}^{l-1} \underbrace{\alpha^i \frac{\partial f^{t-i}(w_0^{t-i}, \mathbf{w}^{t-i})}{\partial h_m(w_m^{t-i}; x_m^{t-i})} \cdot \frac{\partial h_m(w_m^{t-i}; x_m^{t-i})}{\partial w_m^{t-i}}}_{\text{Client Buffer, } i = 0, 1 \cdots l-1}
\end{aligned} \tag{5}$$

After receiving v_m^t from the server, the clients update their buffer by enqueuing the $v_m^t \cdot \frac{\partial h_m(w_m^t; x_m^t)}{\partial w_m^t}$. Finally, the client is updated with $w_m^{t+1} \leftarrow w_m^t - \eta_m \nabla_{w_m} S_{t,l,\alpha}(w_0^t, \mathbf{w}^t)$.

3.3 Event-driven online VFL framework

The event-driven online VFL framework is designed, and the procedures for both clients and servers are formalized in the algorithm 1.[3] When an event occurs at round t, the activated client m will receive the data x_m^t. Subsequently, it sends the embedding $h_m(w_m^t; x_m^t)$ to the server. The server then requests embeddings from the passive clients $m \in \bar{A}_t \subset [M]$. After gathering the embedding,

[3] A synchronous version of algorithm 1 is provided in the Appendix C.1.

the server calculates the partial derivative of $f^t(\cdot)$ w.r.t. its local model w_0^t and w.r.t. the output of the activated clients $h_m(w_m^t; x_m^t)$. Subsequently, the server updates the buffer for the sequence of partial derivatives in DLR. Following this, the server updates its local model with the partial derivative w.r.t. its model w_0 (Eq. 4) and sends the partial derivative w.r.t. the client's output v_m^t to the activated client. After receiving from the server, each activated client $m \in A_t$ calculates the partial derivative w.r.t. their parameter via chain rule (Eq. 5), and then they update their parameter accordingly. At each round, we denote the set of the activated client as $A_t \subset [M]$, where $[M]$ represents the set of all clients. The set of passive clients is denoted by $\bar{A}_t = [M] \setminus A_t$.

Algorithm 1 Event-driven online VFL

```
Input: window length $l$, coefficient $\alpha$, learning rate $\{\eta_m\}_{m=0}^M$,
Output: server model $w_0$, client models $w_m \in [M]$
initialize model $w_m$ for all participants $m \in \{0, 1, ...M\}$
Client procedure:
  if activated by an event then:
    sampling the environment to obtain $x_m^t$
    send $h_m(w_m^t; x_m^t)$ to the server
    receive $v_m^t$ from the server
    enqueue $v_m^t \cdot \frac{\partial h_m(w_m^t; x_m^t)}{\partial w_m^t}$ into the client's buffer
    calculate $\nabla_{w_m} \tilde{S}_{t,l,\alpha}(w_0^t, \mathbf{w}^t)$ with client's buffer
    update the parameter $w_m^{t+1} \leftarrow w_m^t - \eta_m \cdot \nabla_{w_m} S_{t,l,\alpha}(w_0^t, \mathbf{w}^t)$
  else if the server's query $t$ is received then:
    sampling the environment to obtain $x_m^t$
    send $h_m(w_m^t; x_m^t)$ to the server
    enqueue $\mathbf{0}$ to the client's buffer
Server procedure:
  when server receives $h_m(w_m^t; x_m^t)$ from activated client $m \in A_t$, do:
    send query $t$ to all passive client $m \in \bar{A}_t$
    calculate $\nabla_{w_0} f^t(w_0^{t-i}, \mathbf{w}^{t-i})$
    updates its model $w_0^{t+1} \leftarrow w_0^t - \eta_0 \cdot \nabla_{w_0} S_{t,l,\alpha}(w_0^t, \mathbf{w}^t)$
    sends $v_m = \frac{\partial S_{t,l,\alpha}(w_0^t, \tilde{\mathbf{w}}^t)}{\partial h_m(w_m^t; x_m^t)}$ to all activated clients $m \in A_t$
```

4 Regret analysis

4.1 Assumption

Assumption 1 to 5 are the assumptions for analysis of the dynamic local regret bound under the non-convex case. Specifically, Assumption 1 is used for modeling the smoothness of the loss function $f(\cdot)$, with which we can link the difference of the gradients with the difference of the input in the definition domain. These are the basic assumptions for solving the non-convex optimization problem in VFL (Liu et al., 2019; Zhang et al., 2021a; Chen et al., 2020; Castiglia et al., 2022). Assumption 2 and assumption 3 are the common assumptions in the analysis of stochastic non-convex optimization (Aydore et al., 2019; Hazan et al., 2017). The unbiased gradient assumption means that the expected value of the stochastic gradient equals the true gradient for the underlying distribution of the sample. The bounded variance assumption ensures that the variability in the stochastic gradient estimates is limited. Assumption 4 is used for bounding the magnitude of the gradient for all participant's models. This is a common assumption for the non-convex optimization of VFL (Gu et al., 2020; Castiglia et al., 2022; Wang et al., 2023; Zhang et al., 2021a) and the regret analysis for online learning in VFL (Wang & Xu, 2023). This assumption is specifically employed to bound the difference between the gradient with missing elements (due to the event-driven framework) and the ideal gradient without such omissions. Assumption 5 assumes that the loss value is bounded, which is mainly used to bound the values of the difference between the sliding window average and to simplify the theoretical result. This is a common assumption in the regret analysis in online learning under non-convex case (Aydore et al., 2019; Hazan et al., 2017).

Assumption 1. ***Lipschitz Gradient:*** *∇f^t is L-Lipschitz continuous w.r.t. all the parameter, i.e., there exists a constant L for $\forall\ [w_0, \mathbf{w}], [w_0', \mathbf{w}']$ such that*

$$\left\|\nabla_{[w_0,\mathbf{w}]} f^t(w_0, \mathbf{w}) - \nabla_{[w_0,\mathbf{w}]} f^t(w_0', \mathbf{w}')\right\| \leq L \left\|[w_0, \mathbf{w}] - [w_0', \mathbf{w}']\right\| \tag{6}$$

Assumption 2. ***Unbiased gradient:*** *The stochastic gradient $\hat{\nabla} f^t(w_0, \mathbf{w})$ obtained at each iteration is an unbiased estimator of the true gradient of the function $f^t(w_0, \mathbf{w})$, i.e., for all $[w_0, \mathbf{w}]$, we have*

$$\mathbb{E}\left[\hat{\nabla} f^t(w_0, \mathbf{w})\right] = \nabla f^t(w_0, \mathbf{w}), \tag{7}$$

where $\nabla f^t(w_0, \mathbf{w})$ is the true gradient of the global objective function.

Assumption 3. ***Bounded variance:*** *The variance of the stochastic gradient $\hat{\nabla} f^t(w_0, \mathbf{w})$ obtained at each iteration is bounded, i.e., there exists a constant σ^2 such that for all $[w_0, \mathbf{w}]$, we have*

$$\mathbb{E}\left[\left\|\hat{\nabla} f^t(w_0, \mathbf{w}) - \nabla f^t(w_0, \mathbf{w})\right\|^2\right] \leq \sigma^2. \tag{8}$$

This ensures that the noise in the gradient estimation is controlled.

Assumption 4. ***Bounded gradient:*** *The gradient of the objective function $f^t(w_0, \mathbf{w})$ is bounded, i.e. there exist positive constants $\mathbf{G}$ such that the following inequalities hold.*

$$\left\|\nabla_{[w_0,\mathbf{w}]} f^t\left(w_0, \mathbf{w}\right)\right\| \leq \mathbf{G} \tag{9}$$

Assumption 5. ***Bounded loss:*** *For all $w_0 \in \mathbb{R}^{d_0}$ and $\mathbf{w} = [w_1, \cdots w_M]$, where $w_m \in \mathbb{R}^{d_m}$ for $m \in [M]$, the loss $f^t(w_0, \mathbf{w})$ is bounded:*

$$|f^t(w_0, \mathbf{w})| < D \tag{10}$$

4.2 Theorem

Theorem 1. ***Dynamic local regret bound:*** *Under Assumption 1 - Assumption 5, solving the event-driven online vertical federated learning problem with Algorithm 1. Select constant $\eta_t = \eta$, define $p_{\min} = \min p_m$, $p_{\max} = \max p_m$, let $\alpha \to 1^-$.*

$$DLR_l(T) \leq \frac{T}{W p_{\min}} \left(\frac{8D}{\eta_t} + 2L\eta_t p_{\max}\sigma^2\right) + 2TL\eta_t \frac{p_{\max}}{p_{\min}} \mathbf{G}^2 \tag{11}$$

Remark 1. The last term in Eq. 11 arises from bounding the error between the gradient with missing elements in the event-driven framework and the ideal gradient without missing elements.

Corollary 1. *Select $l = T^{\frac{1}{2}}$ and $\eta_t = T^{-\frac{1}{4}}$. Let $\alpha \to 1^-$.*

$$DLR_l(T) = \mathcal{O}(T^{\frac{3}{4}}) \tag{12}$$

Remark 2. Corollary 1 suggests a sub-linear growth rate compared to the linear regret bound $\mathcal{O}(T)$, which demonstrates an effective regret minimization.

The proof of the Theorem 1 is in Appendix A. For the completeness of our analysis, we further provide the prevalent regret analysis of the event-driven online VFL in the convex case in Appendix B.

5 Experiment

5.1 Experiment setup

Supplementary experimental details, including the algorithms of the baselines specifically adapted for VFL, are provided in Appendix C. Furthermore, supplementary experiments addressing secondary aspects, such as ablation studies on the DLR parameter (l, α) and experiments on other datasets (SUSY, HIGGS), can be found in Appendix D.

Dataset We leveraged the Infinite MNIST (i-MNIST) dataset (Loosli et al., 2007) to assess the performance of the proposed methodologies in the context of VFL. The i-MNIST dataset extends the MNIST dataset by providing an endless stream of handwritten digit images along with their corresponding labels. To convert the i-MNIST dataset into a distributed dataset, each image was first flattened into a one-dimensional vector. This vector was then divided into four equal segments to ensure an even distribution of features across the clients. Each client was assigned one of the four feature partitions, while the server was assigned the label. Experiments on other practical online learning datasets, including the SUSY and HIGGS datasets, are provided in Appendix D.2.

Model architecture In our online VFL framework, there were four clients and one server. On the client side, the models consisted of a one-layer perceptron that took flattened features as input and generated 64-dimensional embeddings using ReLU activation. On the server side, a two-layer multi-layer perceptron was employed. The first layer took the concatenated client embeddings as input, produced an output of 256 units, and applied ReLU activation. The subsequent layer generated class logits using softmax activation. The loss function used by the server was cross-entropy.

Baselines The baselines comprised three different online VFL frameworks and three add-ons to those frameworks concerning the activation of participants, resulting in a total of nine baselines. The first online VFL framework applied online gradient descent in VFL (referred to as "OGD" for brevity). The second online VFL framework adapted Static Local Regret (Hazan et al., 2017) to online VFL (referred to as "SLR"). The third online VFL framework was our VFL framework incorporating dynamic local regret (referred to as "DLR").

The three activation add-ons included "Full" activation, "Random" activation, and "Event" activation. The "Full" activation represented the most common scenario where all clients were activated and received the synchronous data stream. The "Random" activation sets the activation probability for all clients to a constant value, i.e. $p_m = p$, which directly interpreted the theoretical result from theorem 1. The "Event" activation was an activation mechanism that we designed to simulate the activation in the sensor network. Many sensors are characterized by activation in response to detecting peaks (Suh, 2007) or stimuli (Heemels et al., 2012). Inspired by this, we implemented the "Event" activation in which a client is activated when the average of its input features exceeds a threshold, denoted as Γ, which mimics the stimuli from the surroundings.

To be more specific, the OGD-Full[4] framework was adapted from (Wang & Xu, 2023), customized to suit our general VFL setting. The OGD-Random and OGD-Event[5] were obtained by incorporating partial activation into the OGD-based online VFL. We also designed the SLR[6] baselines by incorporating Local Regret (Hazan et al., 2017) into the VFL with different activation schemes. DLR-Full was the model obtained by directly adapting the DLR to the synchronous VFL framework. DLR-Random and DLR-Event were the main contributions of our work, which incorporate partial client activation ("Random" or "Event") into VFL.

Training procedure We followed the standard online learning setting, wherein at each round t, client m received only the corresponding feature of a single sample, rather than a batch. Each trial comprised a total of 2,000,000 non-repetitive samples. The learning rate η was tuned from $\{1, 0.1, 0.01, 0.001, \ldots\}$. The length of the exponential weighted sliding window for the DLR was tuned from $\{10, 50, 100, 150\}$. The activation probability p for the "Random" activation was selected from $\{0.25, 0.5, 0.75\}$. The activation threshold Γ was tuned from $\{-0.2, 0, 0.2, 0.4, 0.6, 0.8\}$.

We conducted experiments on both a stationary data stream and a synthetic non-stationary data stream. In section 5.2. we used the original i-MNIST dataset as the stationary data stream, where each class had an equal probability of being sampled ($\frac{1}{10}$) at each round. In section 5.3 we synthesized a non-stationary data stream by altering the class sampling probabilities every 50 rounds. At each stage, the sampling probability for each class was drawn from a uniform distribution $U(0, 1)$ and then normalized. Data was then sampled based on the updated probabilities.

Metrics To evaluate the algorithm's performance, we employed the run-time error rate (prequential) during online learning. This error rate was averaged and reported every 20,000 samples, providing insights into performance at each stage of the training. Additionally, the accumulated error rate, representing the overall error for the entire training process, served as a metric for evaluating overall performance. In terms of computational efficiency, we reported the total computation time for all clients, including both active and passive clients at each round. In terms of communication efficiency, we recorded the total communication cost for the VFL framework throughout the entire training process, specifically measuring the size of all communication messages exchanged between the server and the clients.

[4] Algorithm for OGD-Full are provided in Algorithm 4 in Appendix C.2.

[5] Algorithm for OGD-Event are provided in Algorithm 2 in Appendix B.

[6] Algorithm for SLR-Event are provided in Algorithm 5 in Appendix C.3.

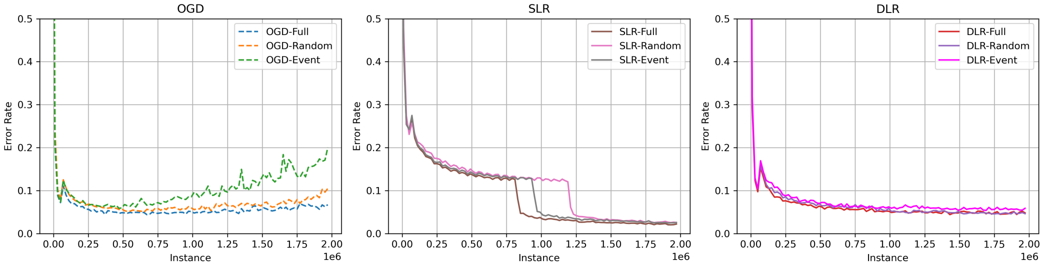

Figure 2: Run-time error rate under stationary data stream

Table 1: Performance metrics for online VFL under stationary data stream

	OGD			SLR			DLR		
Metric	Full	Random	Event	Full	Random	Event	Full	Random	Event
Accum. Error Rate	0.0575	0.0696	0.1027	0.0846	0.1083	0.0945	0.0618	**0.0649**	**0.0718**
Client Comp. (s)	4339.20	2652.38	3182.46	4565.42	2682.47	2992.73	4565.42	2682.47	2992.73
Client Comm. (MB)	1953.13	**1465.04**	**1686.86**	78125.02	58599.36	61565.37	1953.13	**1464.73**	**1539.13**

5.2 Result on stationary data stream

Figure 2 illustrates a comparison of the run-time error rates across all frameworks within a stationary data stream scenario. The x-axis shows the number of observed instances, while the y-axis represents the corresponding run-time error rates. From the analysis of the results, it is evident that in the "Full" activation framework, all models demonstrate stable convergence behavior. However, under the partial activation scheme, both SLR and DLR show superior convergence curves compared to OGD. Notably, DLR converges more rapidly than SLR, entering the convergence phase earlier.

Table 1 presents the performance metrics for the entire training process, including the accumulated error rate, the total computational time for the client, and the total communication cost for the entire VFL framework. The accumulated error further illustrates that the DLR exhibits greater stability under the partial activation scheme. Specifically, the accumulated error rate for the DLR remains approximately 0.06 when employing partial activation, whereas the accumulated error rate for OGD undergoes a significant reduction from 0.0575 to 0.1027 when implementing the partial activation scheme. By comparing DLR and SLR, we observed that DLR converges more quickly, resulting in a lower average error rate. Specifically, due to the buffer design, SLR required the communication of the entire buffer between the server and the client in each round[7]. Consequently, the communication volume of SLR was an order of magnitude higher than that of DLR and OGD. Compared to "Full" activation, the partial activation approach results in a slightly higher error rate; however, it significantly reduces client computation and overall communication costs.

5.3 Result on non-stationary data stream

Figure 3 compares the run-time error rates across all the baselines under the non-stationary data stream case. The DLR and SLR frameworks generally outperformed OGD, demonstrating greater stability under the partial activation scheme for clients. Towards the end of the training process, OGD became increasingly unstable, particularly when partial activation add-ons were used. In contrast, both DLR and SLR maintained stability across all activation schemes. Although SLR was able to converge in non-stationary environments, it was less adaptable to dynamic environments compared to DLR, as its convergence was slower at the beginning of training.

Table 2 presents the performance metrics of the frameworks for comparison. Overall, the DLR method demonstrated a lower accumulated error rate compared to OGD and SLR, primarily due to its stability against non-stationary data streams and partial activation. The communication cost of SLR was significantly higher than that of OGD and DLR, consistent with previous experimental results. When comparing full activation with the partial activation scheme, it was observed that

[7] The step that incurs a high communication cost in SLR is highlighted in Algorithm 5 in Appendix C.3.

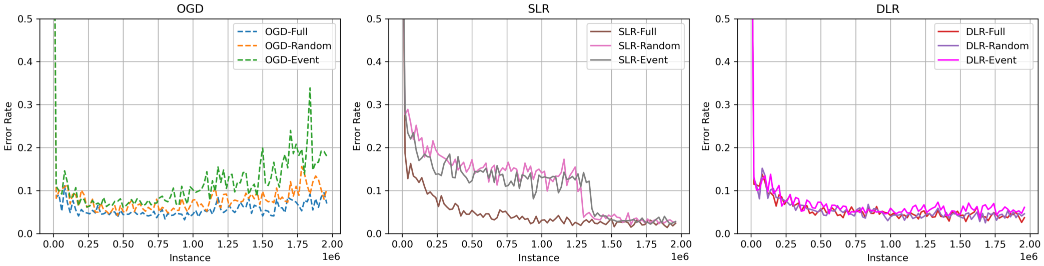

Figure 3: Run-time error rate under non-stationary data stream

Table 2: Performance metrics for online VFL under non-stationary data stream

	OGD			SLR			DLR		
Metric	Full	Random	Event	Full	Random	Event	Full	Random	Event
Accum. Error Rate	0.0592	0.0788	0.1191	0.0481	0.1159	0.1118	0.0553	**0.0561**	**0.0681**
Client Comp. (s)	4406.14	2822.29	3168.34	3497.49	2172.44	2103.50	4558.20	2872.79	2698.96
Total Comm. (MB)	1953.12	**1464.71**	**1543.32**	78124.61	58589.34	61748.22	1953.12	**1465.02**	**1413.52**

the partial activation approach typically resulted in a slightly higher total error rate during training. However, it significantly reduced both computational and communication costs, making it a more practical and efficient solution for real-world applications.

5.4 Enhancing computation-communication efficiency with partial activation

In the partial activation approach ("Random" and "Event"), fewer clients participated in the training of each epoch, therefore reducing both computation and communication costs compared to the framework with "Full" activation.

Activation probability p In the "Random" activation framework, the activation probability p determines the likelihood of a client being activated in each round. A higher value of p increases the probability of activation, while a lower value decreases it. We conducted the study on the activation probability within the DLR-Random Framework under the non-stationary data stream, using a window length of $l = 10$ and an attenuation coefficient $\alpha = 0.95$. We selected p from the range $\{0.25, 0.50, 0.75, 1.00\}$.

This observation indicates that the run-time error rate among the DLR-Random models remains consistent across varying activation probabilities. Table 3 presents the corresponding performance metrics of those trials. This indicates that a small activation probability proportionally decreases the computational cost for the client, albeit with a slight increase in the accumulated error rate. Additionally, both client computation time and communication decrease, as passive clients do not participate in the backward process.

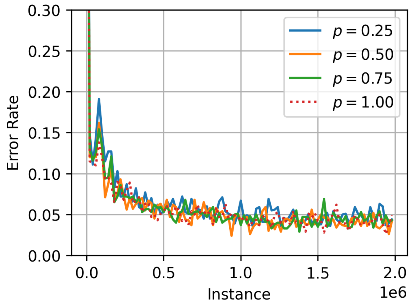

Figure 4: Activation probability p

Table 3: Performance metrics for different activation probability

Activation Probability	Accum. Error Rate	Client Comp. (s)	Total Comm. (MB)
0.25	0.0622	1903.91	1220.75
0.50	0.0561	2872.79	1465.02
0.75	0.0560	3788.07	1709.04
1.00	0.0553	4558.20	1953.12

Event activation threshold Γ Using the DLR-Event framework with a window length of $l = 10$ and an attenuation coefficient of $\alpha = 0.95$, we examined the impact of varying the activation threshold. Table 4 presents the performance metrics for the DLR-Event with different Γ. As the threshold increased, fewer clients were activated per round, leading to an overall increase in the framework's accumulated error rate while reducing its computational cost and communication costs. We further provided an examination of the activation rate for each client across various activation thresholds. As the threshold surpasses 0.6, few clients were activated in each round, the framework will mostly rely on the server for learning, with clients primarily offering a nearly invariant mapping of the input features. Conversely, as the threshold decreases, clients can be activated more frequently.

Table 4: Performance metrics for activation threshold

Threshold Γ	Accum. Error Rate	Client Comp. (s)	Total Comm. (MB)
0.6	0.1057	928.67	1017.76
0.2	0.0738	1852.15	1211.55
-0.2	0.0595	3752.07	1689.42

6 Limitations

Due to the inherent nature of VFL, wherein the clients possess non-intersecting feature sets, the involvement of passive clients in each round remains unavoidable. Omitting passive participants leads to missing input at the server's model, which is known as the "incomplete view" problem in VFL or the "non-overlapping sample" problem within the offline VFL setting. Although some research on VFL has attempted to address the incomplete view problem through knowledge distillation (KD) (Ren et al., 2022), self-supervised learning (SSL) (He et al., 2022; Li et al., 2022; Kang et al., 2022) and semi-supervised learning (Li et al., 2024), these approaches entail more complex models and higher computation-communication costs. For instance, Ren et al. (2022) train student models for each case with an incomplete view via KD, which imposes a significant computational cost, especially when the number of clients is large. Another example is the work by He et al. (2022), which implements SSL within the VFL framework. This approach requires communication between participants during the computation-intensive pretraining stage. Consequently, in scenarios where participants have limited computational resources, there is no perfect solution to this problem.

The future direction of event-driven online VFL focuses on examining the complete independent data streams for each client by addressing the incomplete view problem within the VFL framework. Resolving this problem will eliminate the need for passive client participation in each event, thereby reducing computation and communication costs and enabling the implementation of fully asynchronous online VFL.

7 Conclusion

Adapting online learning to vertical federated learning poses challenges due to the unique nature of VFL. The clients may not receive data streaming synchronously, as data generated by an "event" typically pertains to only a subset of the clients. To address these challenges, we proposed the Event-Driven Online Vertical Federated Learning framework. Within this framework, only a subset of clients is activated by the event in each round, with others passively participating in the learning process. Moreover, we incorporate dynamic local regret to address non-convex models in a non-stationary environment, which further enhances the adaptability of our framework for real-world applications. Through comprehensive regret analysis, we have derived the regret bound of $\mathcal{O}(T^{\frac{3}{4}})$ for dynamic local regret under the non-convex case. Through extensive experiments, we have demonstrated the effectiveness of our approach in reducing communication-computation costs and achieving strong performance in non-stationary environments. Overall, our contributions pave the way for more practical and adaptable implementations of VFL in real-world scenarios.

Acknowledgments

This work has been supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), Discovery Grants program.

References

Jacob D Abernethy, Elad Hazan, and Alexander Rakhlin. Competing in the dark: An efficient algorithm for bandit linear optimization. 2009.

Naman Agarwal, Alon Gonen, and Elad Hazan. Learning in non-convex games with an optimization oracle. In *Conference on Learning Theory*, pp. 18–29. PMLR, 2019.

Sergul Aydore, Tianhao Zhu, and Dean P Foster. Dynamic local regret for non-convex online forecasting. *Advances in neural information processing systems*, 32, 2019.

Jonas Beuchert, Friedrich Solowjow, Sebastian Trimpe, and Thomas Seel. Overcoming bandwidth limitations in wireless sensor networks by exploitation of cyclic signal patterns: An event-triggered learning approach. *Sensors*, 20(1):260, 2020.

Timothy J Castiglia, Anirban Das, Shiqiang Wang, and Stacy Patterson. Compressed-vfl: Communication-efficient learning with vertically partitioned data. In *International Conference on Machine Learning*, pp. 2738–2766. PMLR, 2022.

Tianyi Chen, Xiao Jin, Yuejiao Sun, and Wotao Yin. Vafl: a method of vertical asynchronous federated learning. *arXiv preprint arXiv:2007.06081*, 2020.

Nima Eshraghi and Ben Liang. Distributed online optimization over a heterogeneous network with any-batch mirror descent. In *International Conference on Machine Learning*, pp. 2933–2942. PMLR, 2020.

Wenjing Fang, Derun Zhao, Jin Tan, Chaochao Chen, Chaofan Yu, Li Wang, Lei Wang, Jun Zhou, and Benyu Zhang. Large-scale secure xgb for vertical federated learning. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 443–452, 2021.

Fangcheng Fu, Xupeng Miao, Jiawei Jiang, Huanran Xue, and Bin Cui. Towards communication-efficient vertical federated learning training via cache-enabled local updates. *arXiv preprint arXiv:2207.14628*, 2022.

Bhargav Ganguly and Vaneet Aggarwal. Online federated learning via non-stationary detection and adaptation amidst concept drift. *IEEE/ACM Transactions on Networking*, 2023.

Xiand Gao, Xiaobo Li, and Shuzhong Zhang. Online learning with non-convex losses and non-stationary regret. In *International Conference on Artificial Intelligence and Statistics*, pp. 235–243. PMLR, 2018.

Bin Gu, An Xu, and Cheng Deng. heng huang. 2020. privacy-preserving asynchronous federated learning algorithms for multi-party vertically collaborative learning. *arXiv preprint arXiv:2008.06233*, 2020.

Stephen Hardy, Wilko Henecka, Hamish Ivey-Law, Richard Nock, Giorgio Patrini, Guillaume Smith, and Brian Thorne. Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption. *arXiv preprint arXiv:1711.10677*, 2017.

Elad Hazan and Satyen Kale. Extracting certainty from uncertainty: Regret bounded by variation in costs. *Machine learning*, 80:165–188, 2010.

Elad Hazan, Alexander Rakhlin, and Peter Bartlett. Adaptive online gradient descent. *Advances in neural information processing systems*, 20, 2007.

Elad Hazan, Karan Singh, and Cyril Zhang. Efficient regret minimization in non-convex games. In *International Conference on Machine Learning*, pp. 1433–1441. PMLR, 2017.

Elad Hazan et al. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

Yuanqin He, Yan Kang, Xinyuan Zhao, Jiahuan Luo, Lixin Fan, Yuxing Han, and Qiang Yang. A hybrid self-supervised learning framework for vertical federated learning. *arXiv preprint arXiv:2208.08934*, 2022.

WPM Heemels Heemels, MCF Donkers, and Andrew R Teel. Periodic event-triggered control for linear systems. *IEEE Transactions on automatic control*, 58(4):847–861, 2012.

Amélie Héliou, Matthieu Martin, Panayotis Mertikopoulos, and Thibaud Rahier. Online non-convex optimization with imperfect feedback. *Advances in Neural Information Processing Systems*, 33: 17224–17235, 2020.

SC Hoi, D Sahoo, J Lu, and P Zhao. Online learning: A comprehensive survey. arxiv. *arXiv preprint arXiv:1802.02871*, 2018.

Songnam Hong and Jeongmin Chae. Communication-efficient randomized algorithm for multi-kernel online federated learning. *IEEE transactions on pattern analysis and machine intelligence*, 44(12):9872–9886, 2021.

Yaochen Hu, Di Niu, Jianming Yang, and Shengping Zhou. Fdml: A collaborative machine learning framework for distributed features. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2232–2240, 2019.

Zhenqi Huang, Sayan Mitra, and Nitin Vaidya. Differentially private distributed optimization. In *Proceedings of the 2015 international conference on distributed computing and networking*, pp. 1–10, 2015.

Yan Kang, Yang Liu, and Xinle Liang. Fedcvt: Semi-supervised vertical federated learning with cross-view training. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 13(4): 1–16, 2022.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International conference on machine learning*, pp. 5132–5143. PMLR, 2020.

Dohyeok Kwon, Jonghwan Park, and Songnam Hong. Tighter regret analysis and optimization of online federated learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020.

Wenguo Li, Xinling Guo, Xu Jiao, Tiancheng Huang, Xiaoran Yan, and Yao Yang. Vertical federated learning hybrid local pre-training. *arXiv preprint arXiv:2405.11884*, 2024.

Wenjie Li, Qiaolin Xia, Hao Cheng, Kouyin Xue, and Shu-Tao Xia. Vertical semi-federated learning for efficient online advertising. *arXiv preprint arXiv:2209.15635*, 2022.

Xiaoxiao Li, Meirui Jiang, Xiaofei Zhang, Michael Kamp, and Qi Dou. Fedbn: Federated learning on non-iid features via local batch normalization. *arXiv preprint arXiv:2102.07623*, 2021.

Peixi Liu, Guangxu Zhu, Wei Jiang, Wu Luo, Jie Xu, and Shuguang Cui. Vertical federated edge learning with distributed integrated sensing and communication. *IEEE Communications Letters*, 26(9):2091–2095, 2022.

Yang Liu, Yan Kang, Xinwei Zhang, Liping Li, Yong Cheng, Tianjian Chen, Mingyi Hong, and Qiang Yang. A communication efficient collaborative learning framework for distributed features. *arXiv preprint arXiv:1912.11187*, 2019.

Yang Liu, Zhuo Ma, Ximeng Liu, Siqi Ma, Surya Nepal, Robert H Deng, and Kui Ren. Boosting privately: Federated extreme gradient boosting for mobile crowdsensing. In *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*, pp. 1–11. IEEE, 2020.

Gaëlle Loosli, Stéphane Canu, and Léon Bottou. Training invariant support vector machines using selective sampling. In Léon Bottou, Olivier Chapelle, Dennis DeCoste, and Jason Weston (eds.), *Large Scale Kernel Machines*, pp. 301–320. MIT Press, Cambridge, MA., 2007. URL `http://leon.bottou.org/papers/loosli-canu-bottou-2006`.

Othmane Marfoq, Giovanni Neglia, Richard Vidal, and Laetitia Kameni. Personalized federated learning through local memorization. In *International Conference on Machine Learning*, pp. 15070–15092. PMLR, 2022.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. PMLR, 2017.

H Brendan McMahan, Gary Holt, David Sculley, Michael Young, Dietmar Ebner, Julian Grady, Lan Nie, Todd Phillips, Eugene Davydov, Daniel Golovin, et al. Ad click prediction: a view from the trenches. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1222–1230, 2013.

Konstantin Mishchenko, Eduard Gorbunov, Martin Takáč, and Peter Richtárik. Distributed learning with compressed gradient differences. *arXiv preprint arXiv:1901.09269*, 2019.

Aritra Mitra, Hamed Hassani, and George J Pappas. Online federated learning. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pp. 4083–4090. IEEE, 2021.

Vaikkunth Mugunthan, Antigoni Polychroniadou, David Byrd, and Tucker Hybinette Balch. Smpai: Secure multi-party computation for federated learning. In *Proceedings of the NeurIPS 2019 Workshop on Robust AI in Financial Services*, 2019.

Tao Qi, Fangzhao Wu, Chuhan Wu, Lingjuan Lyu, Tong Xu, Hao Liao, Zhongliang Yang, Yongfeng Huang, and Xing Xie. Fairvfl: A fair vertical federated learning framework with contrastive adversarial learning. *Advances in Neural Information Processing Systems*, 35:7852–7865, 2022.

Thilina Ranbaduge and Ming Ding. Differentially private vertical federated learning. *arXiv preprint arXiv:2211.06782*, 2022.

Zhenghang Ren, Liu Yang, and Kai Chen. Improving availability of vertical federated learning: Relaxing inference on non-overlapping data. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 13(4):1–20, 2022.

Doyen Sahoo, Quang Pham, Jing Lu, and Steven C. H. Hoi. Online deep learning: Learning deep neural networks on the fly. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, Jun 2018. doi: 10.24963/ijcai.2018/369. URL `http://dx.doi.org/10.24963/ijcai.2018/369`.

Shai Shalev-Shwartz and Yoram Singer. A primal-dual perspective of online learning algorithms. *Machine Learning*, 69:115–142, 2007.

Arun Sai Suggala and Praneeth Netrapalli. Online non-convex learning: Following the perturbed leader is optimal. In *Algorithmic Learning Theory*, pp. 845–861. PMLR, 2020.

Young Soo Suh. Send-on-delta sensor data transmission with a linear predictor. *Sensors*, 7(4): 537–547, 2007.

Sebastian Trimpe and Raffaello D'Andrea. Event-based state estimation with variance-based triggering. *IEEE Transactions on Automatic Control*, 59(12):3266–3281, 2014.

Praneeth Vepakomma, Otkrist Gupta, Tristan Swedish, and Ramesh Raskar. Split learning for health: Distributed deep learning without sharing raw patient data. *arXiv preprint arXiv:1812.00564*, 2018.

Ganyu Wang, Bin Gu, Qingsong Zhang, Xiang Li, Boyu Wang, and Charles Ling. A unified solution for privacy and communication efficiency in vertical federated learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Ganyu Wang, Qingsong Zhang, Xiang Li, Boyu Wang, Bin Gu, and Charles X Ling. Secure and fast asynchronous vertical federated learning via cascaded hybrid optimization. *Machine Learning*, 113(9):6413–6451, 2024.

Heqiang Wang and Jie Xu. Online vertical federated learning for cooperative spectrum sensing. *arXiv preprint arXiv:2312.11363*, 2023.

Yujia Wang, Lu Lin, and Jinghui Chen. Communication-efficient adaptive federated learning. *arXiv preprint arXiv:2205.02719*, 2022.

Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H Yang, Farhad Farokhi, Shi Jin, Tony QS Quek, and H Vincent Poor. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security*, 15:3454–3469, 2020.

Kang Wei, Jun Li, Chuan Ma, Ming Ding, Sha Wei, Fan Wu, Guihai Chen, and Thilina Ranbaduge. Vertical federated learning: Challenges, methodologies and experiments. *arXiv preprint arXiv:2202.04309*, 2022.

Daniel Whiteson. HIGGS. UCI Machine Learning Repository, 2014a. DOI: https://doi.org/10.24432/C5V312.

Daniel Whiteson. SUSY. UCI Machine Learning Repository, 2014b. DOI: https://doi.org/10.24432/C54606.

Kai Yang, Tao Fan, Tianjian Chen, Yuanming Shi, and Qiang Yang. A quasi-newton method based vertical federated learning framework for logistic regression. *arXiv preprint arXiv:1912.00513*, 2019.

Ke Zhang, Ganyu Wang, Han Li, Yulong Wang, Hong Chen, and Bin Gu. Asynchronous vertical federated learning for kernelized auc maximization. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 4244–4255, 2024.

Qingsong Zhang, Bin Gu, Zhiyuan Dang, Cheng Deng, and Heng Huang. Desirable companion for vertical federated learning: New zeroth-order gradient based algorithm. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 2598–2607, 2021a.

Qingsong Zhang, Bin Gu, Cheng Deng, Songxiang Gu, Liefeng Bo, Jian Pei, and Heng Huang. Asysqn: Faster vertical federated learning algorithms with better computation resource utilization. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pp. 3917–3927, 2021b.

Qingsong Zhang, Bin Gu, Cheng Deng, and Heng Huang. Secure bilevel asynchronous vertical federated learning with backward updating. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 10896–10904, 2021c.

Jun Zhou, Chaochao Chen, Longfei Zheng, Huiwen Wu, Jia Wu, Xiaolin Zheng, Bingzhe Wu, Ziqi Liu, and Li Wang. Vertically federated graph neural network for privacy-preserving node classification. *arXiv preprint arXiv:2005.11903*, 2020.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th international conference on machine learning (icml-03)*, pp. 928–936, 2003.

A Regret analysis: non-convex case with dynamic local regret

Proof of Theorem 1:

Taking Expectation w.r.t x_t.

$$
\begin{aligned}
&\mathbb{E}_{x_t}\left[S_{t,l,\alpha}(w_0^{t+1},\mathbf{w}^{t+1}) - S_{t,l,\alpha}(w_0^{t},\mathbf{w}^{t})\right]\\
\overset{1)}{\leq}&\mathbb{E}_{x_t}\left\langle \nabla S_{t,l,\alpha}(w_0^{t},\mathbf{w}^{t}), [w_0^{t+1},\mathbf{w}^{t+1}] - [w_0^{t},\mathbf{w}^{t}]\right\rangle + \frac{L}{2}\mathbb{E}_{x_t}\left\|[w_0^{t+1},\mathbf{w}^{t+1}] - [w_0^{t},\mathbf{w}^{t}]\right\|^2\\
\overset{2)}{=}&-\eta_t\mathbb{E}_{x_t}\left\langle \nabla S_{t,l,\alpha}(w_0^{t},\mathbf{w}^{t}), \hat{\nabla}_{w_0} S_{t,l,\alpha}(w_0^{t},\mathbf{w}^{t})\right\rangle - \eta_t\mathbb{E}_{x_t}\left\langle \nabla S_{t,l,\alpha}(w_0^{t},\mathbf{w}^{t}), \sum_{m\in A_t}\hat{\nabla}_{w_m}\tilde{S}_{t,l,\alpha}(w_0^{t},\mathbf{w}^{t})\right\rangle\\
&+\frac{L\eta_t^2}{2}\mathbb{E}_{x_t}\left\|\hat{\nabla}_{w_0} S_{t,l,\alpha}(w_0^{t},\mathbf{w}^{t})\right\|^2 + \frac{L\eta_t^2}{2}\sum_{m\in A_t}\mathbb{E}_{x_t}\left\|\hat{\nabla}_{w_m}\tilde{S}_{t,l,\alpha}(w_0^{t},\mathbf{w}^{t})\right\|^2\\
=&\underbrace{-\eta_t\mathbb{E}_{x_t}\left\langle \nabla S_{t,l,\alpha}(w_0^{t},\mathbf{w}^{t}), \hat{\nabla}_{w_0} S_{t,l,\alpha}(w_0^{t},\mathbf{w}^{t})\right\rangle + \frac{L\eta_t^2}{2}\mathbb{E}_{x_t}\left\|\hat{\nabla}_{w_0} S_{t,l,\alpha}(w_0^{t},\mathbf{w}^{t})\right\|^2}_{a)}\\
&\underbrace{-\eta_t\mathbb{E}_{x_t}\left\langle \nabla S_{t,l,\alpha}(w_0^{t},\mathbf{w}^{t}), \sum_{m\in A_t}\hat{\nabla}_{w_m}\tilde{S}_{t,l,\alpha}(w_0^{t},\mathbf{w}^{t})\right\rangle + \frac{L\eta_t^2}{2}\sum_{m\in A_t}\mathbb{E}_{x_t}\left\|\hat{\nabla}_{w_m}\tilde{S}_{t,l,\alpha}(w_0^{t},\mathbf{w}^{t})\right\|^2}_{b)}
\end{aligned}
\tag{13}
$$

where 1) applies the Lipschitz Continuous of $S_{t,l,\alpha}(w_0^t,\mathbf{w}^t)$,
2) applies the update in one round of the event-driven online VFL.

For a)

$$
\begin{aligned}
&-\eta_t\mathbb{E}_{x_t}\left\langle \nabla S_{t,l,\alpha}(w_0^{t},\mathbf{w}^{t}), \hat{\nabla}_{w_0} S_{t,l,\alpha}(w_0^{t},\mathbf{w}^{t})\right\rangle + \frac{L\eta_t^2}{2}\mathbb{E}_{x_t}\left\|\hat{\nabla}_{w_0} S_{t,l,\alpha}(w_0^{t},\mathbf{w}^{t})\right\|^2\\
\overset{1)}{=}&-\eta_t\mathbb{E}_{x_t}\left\|\nabla S_{t,l,\alpha}(w_0^{t},\mathbf{w}^{t})\right\|^2\\
&+\frac{L\eta_t^2}{2}\mathbb{E}_{x_t}\left\|\hat{\nabla}_{w_0} S_{t,l,\alpha}(w_0^{t},\mathbf{w}^{t}) - \nabla_{w_0} S_{t,l,\alpha}(w_0^{t},\mathbf{w}^{t}) + \nabla_{w_0} S_{t,l,\alpha}(w_0^{t},\mathbf{w}^{t})\right\|^2\\
\overset{2)}{\leq}&-\eta_t\mathbb{E}_{x_t}\left\|\nabla S_{t,l,\alpha}(w_0^{t},\mathbf{w}^{t})\right\|^2\\
&+L\eta_t^2\mathbb{E}_{x_t}\left\|\hat{\nabla}_{w_0} S_{t,l,\alpha}(w_0^{t},\mathbf{w}^{t}) - \nabla_{w_0} S_{t,l,\alpha}(w_0^{t},\mathbf{w}^{t})\right\|^2 + L\eta_t^2\mathbb{E}_{x_t}\left\|\nabla_{w_0} S_{t,l,\alpha}(w_0^{t},\mathbf{w}^{t})\right\|^2\\
=&-\left(\eta_t - L\eta_t^2\right)\mathbb{E}_{x_t}\left\|\nabla S_{t,l,\alpha}(w_0^{t},\mathbf{w}^{t})\right\|^2\\
&+L\eta_t^2\mathbb{E}_{x_t}\left\|\hat{\nabla}_{w_0} S_{t,l,\alpha}(w_0^{t},\mathbf{w}^{t}) - \nabla_{w_0} S_{t,l,\alpha}(w_0^{t},\mathbf{w}^{t})\right\|^2
\end{aligned}
\tag{14}
$$

where 1) applies assumption 2 (unbiased gradient),
2) applies $\|a+b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$.

For b)

$$
\begin{aligned}
&-\eta_t\mathbb{E}_{x_t}\left\langle \nabla S_{t,l,\alpha}(w_0^{t},\mathbf{w}^{t}), \sum_{m\in A_t}\hat{\nabla}_{w_m}\tilde{S}_{t,l,\alpha}(w_0^{t},\mathbf{w}^{t})\right\rangle + \frac{L\eta_t^2}{2}\sum_{m\in A_t}\mathbb{E}_{x_t}\left\|\hat{\nabla}_{w_m}\tilde{S}_{t,l,\alpha}(w_0^{t},\mathbf{w}^{t})\right\|^2\\
=&-\eta_t\mathbb{E}_{x_t}\left\langle \nabla S_{t,l,\alpha}(w_0^{t},\mathbf{w}^{t}), \sum_{m\in A_t}\hat{\nabla}_{w_m}\tilde{S}_{t,l,\alpha}(w_0^{t},\mathbf{w}^{t}) - \sum_{m\in A_t}\hat{\nabla}_{w_m}S_{t,l,\alpha}(w_0^{t},\mathbf{w}^{t}) + \sum_{m\in A_t}\hat{\nabla}_{w_m}S_{t,l,\alpha}(w_0^{t},\mathbf{w}^{t})\right\rangle\\
&+\frac{L\eta_t^2}{2}\sum_{m\in A_t}\mathbb{E}_{x_t}\left\|\hat{\nabla}_{w_m}\tilde{S}_{t,l,\alpha}(w_0^{t},\mathbf{w}^{t}) - \hat{\nabla}_{w_m}S_{t,l,\alpha}(w_0^{t},\mathbf{w}^{t}) + \hat{\nabla}_{w_m}S_{t,l,\alpha}(w_0^{t},\mathbf{w}^{t})\right\|^2
\end{aligned}
$$

$$
\begin{aligned}
&\overset{1)}{=} - \eta_t \mathbb{E}_{x_t} \left\| \sum_{m \in A_t} \nabla_{w_m} S_{t,l,\alpha}(w_0^t, \mathbf{w}^t) \right\|^2 \\
&\quad - \eta_t \mathbb{E}_{x_t} \left\langle \nabla S_{t,l,\alpha}(w_0^t, \mathbf{w}^t), \sum_{m \in A_t} \hat{\nabla}_{w_m} \tilde{S}_{t,l,\alpha}(w_0^t, \mathbf{w}^t) - \sum_{m \in A_t} \hat{\nabla}_{w_m} S_{t,l,\alpha}(w_0^t, \mathbf{w}^t) \right\rangle \\
&\quad + \frac{L\eta_t^2}{2} \sum_{m \in A_t} \mathbb{E}_{x_t} \left\| \hat{\nabla}_{w_m} \tilde{S}_{t,l,\alpha}(w_0^t, \mathbf{w}^t) - \hat{\nabla}_{w_m} S_{t,l,\alpha}(w_0^t, \mathbf{w}^t) + \hat{\nabla}_{w_m} S_{t,l,\alpha}(w_0^t, \mathbf{w}^t) \right\|^2 \\
&\overset{2)}{=} - \eta_t \mathbb{E}_{x_t} \left\| \sum_{m \in A_t} \nabla_{w_m} S_{t,l,\alpha}(w_0^t, \mathbf{w}^t) \right\|^2 \\
&\quad - \eta_t \mathbb{E}_{x_t} \left\langle \nabla S_{t,l,\alpha}(w_0^t, \mathbf{w}^t), \sum_{m \in A_t} \hat{\nabla}_{w_m} \tilde{S}_{t,l,\alpha}(w_0^t, \mathbf{w}^t) - \sum_{m \in A_t} \hat{\nabla}_{w_m} S_{t,l,\alpha}(w_0^t, \mathbf{w}^t) \right\rangle \\
&\quad + L\eta_t^2 \sum_{m \in A_t} \mathbb{E}_{x_t} \left\| \hat{\nabla}_{w_m} \tilde{S}_{t,l,\alpha}(w_0^t, \mathbf{w}^t) - \hat{\nabla}_{w_m} S_{t,l,\alpha}(w_0^t, \mathbf{w}^t) \right\|^2 \\
&\quad + L\eta_t^2 \sum_{m \in A_t} \mathbb{E}_{x_t} \left\| \hat{\nabla}_{w_m} S_{t,l,\alpha}(w_0^t, \mathbf{w}^t) \right\|^2 \\
&\overset{3)}{=} - \eta_t \mathbb{E}_{x_t} \left\| \sum_{m \in A_t} \nabla_{w_m} S_{t,l,\alpha}(w_0^t, \mathbf{w}^t) \right\|^2 \\
&\quad - \eta_t \mathbb{E}_{x_t} \left\langle \nabla S_{t,l,\alpha}(w_0^t, \mathbf{w}^t), \sum_{m \in A_t} \hat{\nabla}_{w_m} \tilde{S}_{t,l,\alpha}(w_0^t, \mathbf{w}^t) - \sum_{m \in A_t} \hat{\nabla}_{w_m} S_{t,l,\alpha}(w_0^t, \mathbf{w}^t) \right\rangle \\
&\quad + L\eta_t^2 \sum_{m \in A_t} \mathbb{E}_{x_t} \left\| \hat{\nabla}_{w_m} \tilde{S}_{t,l,\alpha}(w_0^t, \mathbf{w}^t) - \hat{\nabla}_{w_m} S_{t,l,\alpha}(w_0^t, \mathbf{w}^t) \right\|^2 \\
&\quad + L\eta_t^2 \sum_{m \in A_t} \mathbb{E}_{x_t} \left\| \nabla_{w_m} S_{t,l,\alpha}(w_0^t, \mathbf{w}^t) \right\|^2 + L\eta_t^2 \sum_{m \in A_t} \mathbb{E}_{x_t} \left\| \hat{\nabla}_{w_m} S_{t,l,\alpha}(w_0^t, \mathbf{w}^t) - \nabla_{w_m} S_{t,l,\alpha}(w_0^t, \mathbf{w}^t) \right\|^2 \\
&= - \left(\eta_t - L\eta_t^2 \right) \mathbb{E}_{x_t} \left\| \sum_{m \in A_t} \nabla_{w_m} S_{t,l,\alpha}(w_0^t, \mathbf{w}^t) \right\|^2 \\
&\quad - \eta_t \mathbb{E}_{x_t} \left\langle \nabla S_{t,l,\alpha}(w_0^t, \mathbf{w}^t), \sum_{m \in A_t} \hat{\nabla}_{w_m} \tilde{S}_{t,l,\alpha}(w_0^t, \mathbf{w}^t) - \sum_{m \in A_t} \hat{\nabla}_{w_m} S_{t,l,\alpha}(w_0^t, \mathbf{w}^t) \right\rangle \\
&\quad + L\eta_t^2 \sum_{m \in A_t} \mathbb{E}_{x_t} \left\| \hat{\nabla}_{w_m} \tilde{S}_{t,l,\alpha}(w_0^t, \mathbf{w}^t) - \hat{\nabla}_{w_m} S_{t,l,\alpha}(w_0^t, \mathbf{w}^t) \right\|^2 \\
&\quad + L\eta_t^2 \sum_{m \in A_t} \mathbb{E}_{x_t} \left\| \hat{\nabla}_{w_m} S_{t,l,\alpha}(w_0^t, \mathbf{w}^t) - \nabla_{w_m} S_{t,l,\alpha}(w_0^t, \mathbf{w}^t) \right\|^2
\end{aligned}
\tag{15}
$$

where 1) applies assumption 2 (unbiased gradient),
2) applies $\|a+b\|^2 \le 2\|a\|^2 + 2\|b\|^2$,
3) $\mathbb{E}(X^2) = \mathbb{E}(X)^2 + \mathrm{Var}(X)$.

Plug in a) and b) and taking expectation w.r.t. the activated client $m \in A_t$, where the activation probability of client m is p_m, we derive:

$$
\begin{aligned}
&\mathbb{E}_m \mathbb{E}_{x_t} \left[S_{t,l,\alpha}(w_0^{t+1}, \mathbf{w}^{t+1}) - S_{t,l,\alpha}(w_0^t, \mathbf{w}^t) \right] \\
&\le - \left(\eta_t - L\eta_t^2 \right) \mathbb{E}_{x_t} \left\| \nabla_{w_0} S_{t,l,\alpha}(w_0^t, \mathbf{w}^t) \right\|^2
\end{aligned}
$$

$$
\begin{aligned}
&+ L\eta_t^2 \mathbb{E}_{x_t} \left\| \hat{\nabla}_{w_0} S_{t,l,\alpha}(w_0^t, \mathbf{w}^t) - \nabla_{w_0} S_{t,l,\alpha}(w_0^t, \mathbf{w}^t) \right\|^2 \\
&- \left(\eta_t - L\eta_t^2\right) \sum_{m\in[M]} p_m \mathbb{E}_{x_t} \left\| \nabla_{w_m} S_{t,l,\alpha}(w_0^t, \mathbf{w}^t) \right\|^2 \\
&- \eta_t \mathbb{E}_{x_t} \sum_{m\in[M]} p_m \left\langle \nabla_{w_m} S_{t,l,\alpha}(w_0^t, \mathbf{w}^t), \hat{\nabla}_{w_m} \tilde{S}_{t,l,\alpha}(w_0^t, \mathbf{w}^t) - \hat{\nabla}_{w_m} S_{t,l,\alpha}(w_0^t, \mathbf{w}^t) \right\rangle \\
&+ L\eta_t^2 \sum_{m\in[M]} p_m \mathbb{E}_{x_t} \left\| \hat{\nabla}_{w_m} \tilde{S}_{t,l,\alpha}(w_0^t, \mathbf{w}^t) - \hat{\nabla}_{w_m} S_{t,l,\alpha}(w_0^t, \mathbf{w}^t) \right\|^2 \\
&+ L\eta_t^2 \sum_{m\in[M]} p_m \mathbb{E}_{x_t} \left\| \hat{\nabla}_{w_m} S_{t,l,\alpha}(w_0^t, \mathbf{w}^t) - \nabla_{w_m} S_{t,l,\alpha}(w_0^t, \mathbf{w}^t) \right\|^2 \\
\overset{1)}{\leq} &- \left(\eta_t - L\eta_t^2\right) p_{\min} \mathbb{E}_{x_t} \left\| \nabla S_{t,l,\alpha}(w_0^t, \mathbf{w}^t) \right\|^2 \\
&+ L\eta_t^2 p_{\max} \mathbb{E}_{x_t} \left\| \hat{\nabla} S_{t,l,\alpha}(w_0^t, \mathbf{w}^t) - \nabla S_{t,l,\alpha}(w_0^t, \mathbf{w}^t) \right\|^2 \\
&- \eta_t \mathbb{E}_{x_t} \sum_{m\in[M]} p_m \left\langle \nabla_{w_m} S_{t,l,\alpha}(w_0^t, \mathbf{w}^t), \hat{\nabla}_{w_m} \tilde{S}_{t,l,\alpha}(w_0^t, \mathbf{w}^t) - \hat{\nabla}_{w_m} S_{t,l,\alpha}(w_0^t, \mathbf{w}^t) \right\rangle \\
&+ L\eta_t^2 \sum_{m\in[M]} p_m \mathbb{E}_{x_t} \left\| \hat{\nabla}_{w_m} \tilde{S}_{t,l,\alpha}(w_0^t, \mathbf{w}^t) - \hat{\nabla}_{w_m} S_{t,l,\alpha}(w_0^t, \mathbf{w}^t) \right\|^2 \\
\overset{2)}{\leq} &- \left(\eta_t - L\eta_t^2\right) p_{\min} \mathbb{E}_{x_t} \left\| \nabla S_{t,l,\alpha}(w_0^t, \mathbf{w}^t) \right\|^2 \\
&+ L\eta_t^2 p_{\max} \cdot \frac{\sigma^2}{W^2} \cdot \frac{1-\alpha^{2l}}{1-\alpha^2} \\
&- \eta_t \mathbb{E}_{x_t} \sum_{m\in[M]} p_m \left\langle \nabla_{w_m} S_{t,l,\alpha}(w_0^t, \mathbf{w}^t), \hat{\nabla}_{w_m} \tilde{S}_{t,l,\alpha}(w_0^t, \mathbf{w}^t) - \hat{\nabla}_{w_m} S_{t,l,\alpha}(w_0^t, \mathbf{w}^t) \right\rangle \\
&+ L\eta_t^2 \underbrace{\sum_{m\in[M]} p_m \mathbb{E}_{x_t} \left\| \hat{\nabla}_{w_m} \tilde{S}_{t,l,\alpha}(w_0^t, \mathbf{w}^t) - \hat{\nabla}_{w_m} S_{t,l,\alpha}(w_0^t, \mathbf{w}^t) \right\|^2}_{c)} \\
\overset{3)}{\leq} &- \left(\eta_t - L\eta_t^2\right) p_{\min} \mathbb{E}_{x_t} \left\| \nabla S_{t,l,\alpha}(w_0^t, \mathbf{w}^t) \right\|^2 \\
&+ L\eta_t^2 p_{\max} \cdot \frac{\sigma^2}{W^2} \cdot \frac{1-\alpha^{2l}}{1-\alpha^2} \\
&- \eta_t \mathbb{E}_{x_t} \sum_{m\in[M]} p_m \left\langle \nabla_{w_m} S_{t,l,\alpha}(w_0^t, \mathbf{w}^t), \hat{\nabla}_{w_m} \tilde{S}_{t,l,\alpha}(w_0^t, \mathbf{w}^t) - \hat{\nabla}_{w_m} S_{t,l,\alpha}(w_0^t, \mathbf{w}^t) \right\rangle \\
&+ L\eta_t^2 p_{\max} \mathbf{G}^2
\end{aligned}
\tag{16}
$$

where 1) note that m is in different dimension,
2) applies assumption 3 (bounded variance), and α is a weighted average of l independently, therefore $\mathbb{E}_{x_t} \left\| \hat{\nabla} S_{t,l,\alpha}(w_0^t, \mathbf{w}^t) - \nabla S_{t,l,\alpha}(w_0^t, \mathbf{w}^t) \right\|^2 \leq \frac{\sigma^2}{W^2} \cdot \frac{1-\alpha^{2l}}{1-\alpha^2}$,
3) plug in c).

For c)

$$
\begin{aligned}
&\sum_{m\in[M]} p_m \mathbb{E}_{x_t} \left\| \hat{\nabla}_{w_m} \tilde{S}_{t,l,\alpha}(w_0^t, \mathbf{w}^t) - \hat{\nabla}_{w_m} S_{t,l,\alpha}(w_0^t, \mathbf{w}^t) \right\|^2 \\
= &\sum_{m\in[M]} p_m \mathbb{E}_{x_t} \left\| \frac{1}{W} \sum_{i=0}^{l-1} \alpha^i \nabla_{w_m} f^{t-i}(w_0^{t-i}, \mathbf{w}^{t-i}) \cdot (1 - \gamma_m^{t-i}) \right\|^2
\end{aligned}
$$

$$
\begin{aligned}
&=\frac{1}{W^2}\sum_{m\in[M]} p_m \mathbb{E}_{x_t}\left\|\sum_{i=0}^{l-1}\alpha^i\hat{\nabla}_{w_m} f^{t-i}(w_0^{t-i},\mathbf{w}^{t-i})\cdot(1-\gamma_m^{t-i})\right\|^2\\
&\overset{1)}{\le}\frac{1}{W^2}\sum_{m\in[M]} p_m\left(\sum_{i=0}^{l-1}\alpha^i(1-\gamma_m^{t-i})\right)^2\left\|\hat{\nabla}_{w_m} f^{t-i}(w_0^{t-i},\mathbf{w}^{t-i})\right\|^2\\
&\le\frac{1}{W^2}\sum_{m\in[M]} p_m\left(\sum_{i=0}^{l-1}\alpha^i\right)^2\left\|\hat{\nabla}_{w_m} f^{t-i}(w_0^{t-i},\mathbf{w}^{t-i})\right\|^2\\
&\overset{2)}{\le}\sum_{m\in[M]} p_m\left\|\hat{\nabla}_{w_m} f^{t-i}(w_0^{t-i},\mathbf{w}^{t-i})\right\|^2\\
&\overset{3)}{\le}\sum_{m\in[M]} p_{\max}\left\|\hat{\nabla}_{w_m} f^{t-i}(w_0^{t-i},\mathbf{w}^{t-i})\right\|^2\\
&\overset{4)}{\le}p_{\max}\mathbf{G}^2
\end{aligned}
\tag{17}
$$

where 1) by triangle inequality $\left\|\sum_{i=0}^{l-1}\alpha^i\nabla_{w_m} f^{t-i}(w_0^{t-i},\mathbf{w}^{t-i})\cdot(1-\gamma_m^{t-i})\right\| \le \sum_{i=0}^{l-1}\left\|\alpha^i\nabla_{w_m} f^{t-i}(w_0^{t-i},\mathbf{w}^{t-i})\cdot(1-\gamma_m^{t-i})\right\| \le \sum_{i=0}^{l-1}\alpha^i(1-\gamma_m^{t-i})\left\|\nabla_{w_m} f^{t-i}(w_0^{t-i},\mathbf{w}^{t-i})\right\|$,
2) $W=\sum_{i=0}^{l-1}\alpha^i$,
3) denote $\max_m\{p_m\}=p_{\max}$,
4) applies assumption 4 (bounded gradient).

Taking expectation w.r.t. time step $t\sim$ Unif($[T]$), and use $\mathbb{E}$ to denote the expectation $\mathbb{E}_t\mathbb{E}_m\mathbb{E}_{x_t}$. We can eliminate $\left\langle\nabla_{w_m} S_{t,l,\alpha}(w_0^t,\mathbf{w}^t),\hat{\nabla}_{w_m}\tilde{S}_{t,l,\alpha}(w_0^t,\mathbf{w}^t)-\hat{\nabla}_{w_m} S_{t,l,\alpha}(w_0^t,\mathbf{w}^t)\right\rangle$, because $\mathbb{E}_t\nabla\tilde{S}_{t,l,\alpha}(w_0^t,\mathbf{w}^t)=\mathbb{E}_t\nabla S_{t,l,\alpha}(w_0^t,\mathbf{w}^t)$.

$$
\begin{aligned}
\mathbb{E}\left[S_{t,l,\alpha}(w_0^{t+1},\mathbf{w}^{t+1})-S_{t,l,\alpha}(w_0^t,\mathbf{w}^t)\right]\le&-\left(\eta_t-L\eta_t^2\right)p_{\min}\mathbb{E}\left\|\nabla S_{t,l,\alpha}(w_0^t,\mathbf{w}^t)\right\|^2\\
&+L\eta_t^2 p_{\max}\cdot\frac{\sigma^2}{W^2}\cdot\frac{1-\alpha^{2l}}{1-\alpha^2}+L\eta_t^2 p_{\max}\mathbf{G}^2
\end{aligned}
\tag{18}
$$

Rearrange the above equation we have:

$$
\begin{aligned}
&\left(\eta_t-L\eta_t^2\right)p_{\min}\mathbb{E}\left\|\nabla S_{t,l,\alpha}(w_0^t,\mathbf{w}^t)\right\|^2\\
\le&\mathbb{E}\left[S_{t,l,\alpha}(w_0^t,\mathbf{w}^t)-S_{t,l,\alpha}(w_0^{t+1},\mathbf{w}^{t+1})\right]\\
&+L\eta_t^2 p_{\max}\cdot\frac{\sigma^2}{W^2}\cdot\frac{1-\alpha^{2l}}{1-\alpha^2}+L\eta_t^2 p_{\max}\mathbf{G}^2\\
\le&\underbrace{\mathbb{E}S_{t,l,\alpha}(w_0^t,\mathbf{w}^t)-\mathbb{E}S_{t+1,l,\alpha}(w_0^{t+1},\mathbf{w}^{t+1})}_{d)}+\underbrace{\mathbb{E}S_{t+1,l,\alpha}(w_0^{t+1},\mathbf{w}^{t+1})-\mathbb{E}S_{t,l,\alpha}(w_0^{t+1},\mathbf{w}^{t+1})}_{e)}\\
&+L\eta_t^2 p_{\max}\cdot\frac{\sigma^2}{W^2}\cdot\frac{1-\alpha^{2l}}{1-\alpha^2}+L\eta_t^2 p_{\max}\mathbf{G}^2\\
\overset{1)}{\le}&\frac{2D}{W}\cdot\frac{(1-\alpha^l)}{1-\alpha}+\frac{D}{W}\left[(1+\alpha^{l-1})+\frac{(1-\alpha^{l-1})(1+\alpha)}{1-\alpha}\right]\\
&+L\eta_t^2 p_{\max}\cdot\frac{\sigma^2}{W^2}\cdot\frac{1-\alpha^{2l}}{1-\alpha^2}+L\eta_t^2 p_{\max}\mathbf{G}^2
\end{aligned}
\tag{19}
$$

The remaining follow the same procedure as Aydore et al. (2019), for brevity, we use the lemma on their paper. For d) we apply the (Aydore et al., 2019, Lemma 3.3) and derive $\mathbb{E}S_{t,l,\alpha}(w_0^t,\mathbf{w}^t)-$

$\mathbb{E}S_{t+1,l,\alpha}(w_0^{t+1}, \mathbf{w}^{t+1}) \le \frac{2D}{W} \cdot \frac{(1-\alpha^l)}{1-\alpha}$. For e) we apply the (Aydore et al., 2019, Lemma 3.2) and derive $\mathbb{E}S_{t+1,l,\alpha}(w_0^{t+1}, \mathbf{w}^{t+1}) - \mathbb{E}S_{t,l,\alpha}(w_0^{t+1}, \mathbf{w}^{t+1}) \le \frac{D}{W}\left[(1+\alpha^{l-1}) + \frac{(1-\alpha^{l-1})(1+\alpha)}{1-\alpha}\right]$. 1) plugs in d) and e).

Divide both side by $(\eta_t - L\eta_t^2)p_{\min}$.

$$\begin{aligned}
&\mathbb{E}\left\|\nabla S_{t,l,\alpha}(w_0^t, \mathbf{w}^t)\right\|^2 \\
\le& \frac{\frac{2D}{W}\cdot\frac{(1-\alpha^l)}{1-\alpha} + \frac{D}{W}\left[(1+\alpha^{l-1}) + \frac{(1-\alpha^{l-1})(1+\alpha)}{1-\alpha}\right] + L\eta_t^2 p_{\max}\cdot\frac{\sigma^2}{W^2}\cdot\frac{1-\alpha^{2l}}{1-\alpha^2} + L\eta_t^2 p_{\max}\mathbf{G}^2}{(\eta_t - L\eta_t^2)p_{\min}} \\
\overset{1)}{\le}& \frac{4D}{\eta_t W p_{\min}}\cdot\frac{(1-\alpha^l)}{1-\alpha} + \frac{2D}{\eta_t W p_{\min}}\left[(1+\alpha^{l-1}) + \frac{(1-\alpha^{l-1})(1+\alpha)}{1-\alpha}\right] \\
&+ 2L\eta_t\frac{p_{\max}}{p_{\min}}\cdot\frac{\sigma^2}{W^2}\cdot\frac{1-\alpha^{2l}}{1-\alpha^2} + 2L\eta_t\frac{p_{\max}}{p_{\min}}\mathbf{G}^2 \\
=& \frac{2D}{\eta_t W p_{\min}}\left[\frac{2(1-\alpha^l)}{1-\alpha} + (1+\alpha^{l-1}) + \frac{(1-\alpha^{l-1})(1+\alpha)}{1-\alpha}\right] \\
&+ 2L\eta_t\frac{p_{\max}}{p_{\min}}\cdot\frac{\sigma^2}{W^2}\cdot\frac{1-\alpha^{2l}}{1-\alpha^2} + 2L\eta_t\frac{p_{\max}}{p_{\min}}\mathbf{G}^2
\end{aligned} \tag{20}$$

where 1) note that when $\eta_t \le \frac{1}{2L}$, $\eta_t - L\eta_t^2 \le \frac{1}{2}\eta_t$,

Follow the proof of (Aydore et al., 2019, Theorem 3.4), as $\alpha \to 1^-$

$$\begin{aligned}
&\lim_{\alpha\to 1^-}\mathbb{E}\left\|\nabla S_{t,l,\alpha}(w_0^t, \mathbf{w}^t)\right\|^2 \\
\le& \frac{8D}{\eta_t W p_{\min}} + 2L\eta_t\frac{p_{\max}}{p_{\min}}\cdot\frac{\sigma^2}{W^2}\cdot\frac{1-\alpha^{2l}}{1-\alpha^2} + 2L\eta_t\frac{p_{\max}}{p_{\min}}\mathbf{G}^2 \\
\le& \frac{1}{W p_{\min}}\left(\frac{8D}{\eta_t} + 2L\eta_t p_{\max}\sigma^2\right) + 2L\eta_t\frac{p_{\max}}{p_{\min}}\mathbf{G}^2
\end{aligned} \tag{21}$$

Summing from $t = 0, 1, ...T$ concludes the proof.

$$\begin{aligned}
DLR_l(T) =& \sum_{t=0}^{t}\lim_{\alpha\to 1^-}\mathbb{E}\left\|\nabla S_{t,l,\alpha}(w_0^t, \mathbf{w}^t)\right\|^2 \\
\le& \frac{T}{W p_{\min}}\left(\frac{8D}{\eta_t} + 2L\eta_t p_{\max}\sigma^2\right) + 2TL\eta_t\frac{p_{\max}}{p_{\min}}\mathbf{G}^2
\end{aligned} \tag{22}$$

□

Proof of Corollary 1:

Select $l = T^{\frac{1}{2}}$ and $\eta_t = T^{-\frac{1}{4}}$, note that $\lim_{\alpha\to 1^-} W = \lim_{\alpha\to 1^-}\frac{1-\alpha^l}{1-\alpha} = l$.

$$\begin{aligned}
&\sum_{t=0}^{t}\lim_{\alpha\to 1^-}\mathbb{E}\left\|\nabla S_{t,l,\alpha}(w_0^t, \mathbf{w}^t)\right\|^2 \\
\overset{1)}{\le}& \frac{T}{l p_{\min}}\left(\frac{8D}{\eta_t} + 2L\eta_t p_{\max}\sigma^2\right) + 2TL\eta_t\frac{p_{\max}}{p_{\min}}\mathbf{G}^2 \\
\overset{2)}{\le}& \frac{T}{\sqrt{T}p_{\min}}\left(8DT^{\frac{1}{4}} + 2LT^{-\frac{1}{4}}p_{\max}\sigma^2\right) + 2T^{\frac{3}{4}}L\eta_t\frac{p_{\max}}{p_{\min}}\mathbf{G}^2 \\
=& \frac{8D}{p_{\min}}T^{\frac{3}{4}} + \frac{2Lp_{\max}\sigma^2}{p_{\min}}T^{\frac{1}{4}} + \frac{2L\eta_t p_{\max}\mathbf{G}^2}{p_{\min}}T^{\frac{3}{4}} \\
=& \mathcal{O}(T^{\frac{3}{4}})
\end{aligned} \tag{23}$$

■

B Regret analysis for OGD-event in convex case

In the convex case, OGD is already an efficient algorithm to solve the online learning problem with partial client activation.

We start by defining Regret in the VFL framework below.

Definition 3. ***Regret for online convex optimization in VFL framework***

$$R_T = \sum_{t=1}^{T} f(w_0^t, \mathbf{w}_0^t) - \sum_{t=1}^{T} f(w_0^*, \mathbf{w}^*) \tag{24}$$

where $[w_0*, \mathbf{w}^*] = \underset{[w_0^*, \mathbf{w}^*]}{argmin} \sum_{t=1}^{T} f(w_0^*, \mathbf{w}^*)$

Following the online convex optimization (Hazan et al., 2016), we design the event-driven online VFL with online gradient descent (OGD-Event in Section 5.1). The algorithm is provided in the algorithm 2 below.

Algorithm 2 Event-driven online VFL with online gradient descent (OGD-Event)

```
Input:
Output: model parameter $w_m$ for all workers $m \in \{0, 1, ...M\}$.
Initialize $w_m$ for all participants $m \in \{0, 1, ...M\}$
for $t \in [T]$ do
    for $m \in A_t$ do
        client $m$ send $h_m(w_m; x_m)$ to the server.
    end for
    for $m \in \bar{A}_t$ do
        Server queries the embeddings from passive client $m$.
        Client $m$ send $h_m(w_m; x_m)$ to the server.
    end for
    The server updates its model $w_0 \leftarrow w_0 - \eta_0 \cdot \frac{\partial f(w_0, \mathbf{w})}{\partial w_0}$
    for $m \in A_t$ do
        Server send $v_m = \frac{\partial f(w_0, \mathbf{w})}{\partial h_m}$ to the client $m$.
        Client $m$ update parameter $w_m \leftarrow w_m - \eta_m v_m \cdot \frac{\partial h_m}{\partial w_m}$
    end for
end for
```

We use a different set of assumptions on convexity, the diameter of the space, and the client's delay to make it fit the regret analysis framework for online convex optimization with Algorithm 2.

Assumption 6. ***Convexity:*** *for any* $[w_0, \mathbf{w}]$ *and* $[w_0', \mathbf{w}']$, $w_0, w_0' \in \mathbb{R}^{d_0}$, $w_m, w_m' \in R^{d_m}$, $f^t(w_0, \mathbf{w})$ *satisfy*

$$f^t(w_0', \mathbf{w}') \geq f^t(w_0, \mathbf{w}) + \left\langle \nabla f^t(w_0, \mathbf{w}), [w_0', \mathbf{w}'] - [w_0, \mathbf{w}] \right\rangle \tag{25}$$

Assumption 7. ***Bounded space diameter:*** *For any* $[w_0, \mathbf{w}]$ *and* $[w_0', \mathbf{w}']$, *satisfy*

$$\|[w_0, \mathbf{w}] - [w_0', \mathbf{w}']\| \leq D \tag{26}$$

Assumption 8. Independent client activation: *The activated client* m_t *for the global iteration t is independent of* $m_0, \cdots, m_{t-1}$ *and satisfies* $\mathbb{P}(m_t = m) := p_m$.

Assumption 9. Uniformly bounded delay: *For each client* m, *the delay at each global iteration* t *is bounded by a constant* τ. *i.e.* $\tau_m^t \leq \tau$.

Theorem 2. *Under Assumptions 6 $\sim$ 9, to solve the online VFL problem with partial client participation using Algorithm 2, the following inequality holds.*

$$R_T \leq \frac{3\mathbf{G}^2 D + 4D^3 + 2L^2\tau^2\mathbf{G}^2 D}{2\mathbf{G}\min\{p_m\}} \cdot \sqrt{T} \tag{27}$$

where the learning rate is chosen as $\eta_t = \frac{D}{\mathbf{G}\sqrt{t}}$.

Remark 3. The regret is sublinear $O(\sqrt{T})$.

Proof of Theorem 2:

First, we bound the update of the participants. For notation brevity, we use $\nabla_{w_0} f(w_0, \mathbf{w})$ and $\nabla_{w_m} f(w_0, \mathbf{w})$ are the partial derivative w.r.t. the corresponding parameter in the space of the parameter of the global model, where the position of all other parameters are filled with 0.

$$
\begin{aligned}
&\left\|[w_0^{t+1}, \mathbf{w}^{t+1}] - [w_0^*, \mathbf{w}^*]\right\|^2 \\
&\overset{1)}{=} \left\|[w_0^t, \mathbf{w}^t] - \eta_t \nabla_{w_0} f(w_0^t, \tilde{\mathbf{w}}^t) - \eta_t \sum_{m \in A_t} \nabla_{w_m} f(w_0^t, \tilde{\mathbf{w}}^t) - [w_0^*, \mathbf{w}^*]\right\|^2 \\
&\overset{2)}{=} \left\|[w_0^t, \mathbf{w}^t] - [w_0^*, \mathbf{w}^*]\right\|^2 + \eta_t^2 \left\|\nabla_{w_0} f(w_0, \tilde{\mathbf{w}}) + \sum_{m \in A_t} \nabla_{w_m} f(w_0^t, \tilde{\mathbf{w}}^t)\right\|^2 \\
&\quad - 2\eta_t \left\langle \nabla_{w_0} f(w_0^t, \tilde{\mathbf{w}}^t) + \sum_{m \in A_t} \nabla_{w_m} f(w_0^t, \tilde{\mathbf{w}}^t), [w_0^t, \mathbf{w}^t] - [w_0^*, \mathbf{w}^*] \right\rangle \\
&\overset{3)}{\leq} \left\|[w_0^t, \mathbf{w}^t] - [w_0^*, \mathbf{w}^*]\right\|^2 + \eta_t^2 \mathbf{G}^2 \\
&\quad - 2\eta_t \left\langle \nabla_{w_0} f(w_0^t, \tilde{\mathbf{w}}^t) + \sum_{m \in A_t} \nabla_{w_m} f(w_0^t, \tilde{\mathbf{w}}^t), [w_0^t, \mathbf{w}^t] - [w_0^*, \mathbf{w}^*] \right\rangle \\
&= \left\|[w_0^t, \mathbf{w}^t] - [w_0^*, \mathbf{w}^*]\right\|^2 + \eta_t^2 \mathbf{G}^2 \\
&\quad - 2\eta_t \left\langle \nabla_{w_0} f(w_0^t, \tilde{\mathbf{w}}^t), [w_0^t, \mathbf{w}^t] - [w_0^*, \mathbf{w}^*] \right\rangle \\
&\quad - 2\eta_t \left\langle \sum_{m \in A_t} \nabla_{w_m} f(w_0^t, \tilde{\mathbf{w}}^t), [w_0^t, \mathbf{w}^t] - [w_0^*, \mathbf{w}^*] \right\rangle \\
&= \left\|[w_0^t, \mathbf{w}^t] - [w_0^*, \mathbf{w}^*]\right\|^2 + \eta_t^2 \mathbf{G}^2 \\
&\quad - 2\eta_t \left\langle \nabla_{w_0} f(w_0^t, \tilde{\mathbf{w}}^t) - \nabla_{w_0} f(w_0^t, \mathbf{w}^t) + \nabla_{w_0} f(w_0^t, \mathbf{w}^t), [w_0^t, \mathbf{w}^t] - [w_0^*, \mathbf{w}^*] \right\rangle \\
&\quad - 2\eta_t \left\langle \sum_{m \in A_t} \nabla_{w_m} f(w_0^t, \tilde{\mathbf{w}}^t) - \sum_{m \in A_t} \nabla_{w_m} f(w_0^t, \mathbf{w}^t) + \sum_{m \in A_t} \nabla_{w_m} f(w_0^t, \mathbf{w}^t), [w_0^t, \mathbf{w}^t] - [w_0^*, \mathbf{w}^*] \right\rangle \\
&= \left\|[w_0^t, \mathbf{w}^t] - [w_0^*, \mathbf{w}^*]\right\|^2 + \eta_t^2 \mathbf{G}^2 \\
&\quad + 2 \left\langle \nabla_{w_0} f(w_0^t, \mathbf{w}^t) - \nabla_{w_0} f(w_0^t, \tilde{\mathbf{w}}^t), \eta_t \left([w_0^t, \mathbf{w}^t] - [w_0^*, \mathbf{w}^*]\right) \right\rangle - 2\eta_t \left\langle \nabla_{w_0} f(w_0^t, \mathbf{w}^t), [w_0^t, \mathbf{w}^t] - [w_0^*, \mathbf{w}^*] \right\rangle \\
&\quad + 2 \left\langle \sum_{m \in A_t} \nabla_{w_m} f(w_0^t, \mathbf{w}^t) - \sum_{m \in A_t} \nabla_{w_m} f(w_0^t, \tilde{\mathbf{w}}^t), \eta_t \left([w_0^t, \mathbf{w}^t] - [w_0^*, \mathbf{w}^*]\right) \right\rangle \\
&\quad - 2\eta_t \left\langle \sum_{m \in A_t} \nabla_{w_m} f(w_0^t, \mathbf{w}^t), [w_0^t, \mathbf{w}^t] - [w_0^*, \mathbf{w}^*] \right\rangle \\
&\overset{4)}{\leq} \left\|[w_0^t, \mathbf{w}^t] - [w_0^*, \mathbf{w}^*]\right\|^2 + \eta_t^2 \mathbf{G}^2 \\
&\quad + \left\|\nabla_{w_0} f(w_0^t, \mathbf{w}^t) - \nabla_{w_0} f(w_0^t, \tilde{\mathbf{w}}^t)\right\|^2 + \eta_t^2 \left\|\left([w_0^t, \mathbf{w}^t] - [w_0^*, \mathbf{w}^*]\right)\right\|^2 \\
&\quad - 2\eta_t \left\langle \nabla_{w_0} f(w_0^t, \mathbf{w}^t), [w_0^t, \mathbf{w}^t] - [w_0^*, \mathbf{w}^*] \right\rangle \\
&\quad + \left\|\sum_{m \in A_t} \nabla_{w_m} f(w_0^t, \mathbf{w}^t) - \sum_{m \in A_t} \nabla_{w_m} f(w_0^t, \tilde{\mathbf{w}}^t)\right\|^2 + \eta_t^2 \left\|\left([w_0^t, \mathbf{w}^t] - [w_0^*, \mathbf{w}^*]\right)\right\|^2 \\
&\quad - 2\eta_t \left\langle \sum_{m \in A_t} \nabla_{w_m} f(w_0^t, \mathbf{w}^t), [w_0^t, \mathbf{w}^t] - [w_0^*, \mathbf{w}^*] \right\rangle
\end{aligned}
$$

$$
\begin{aligned}
&\overset{5)}{=} \left\| [w_0^t, \mathbf{w}^t] - [w_0^*, \mathbf{w}^*] \right\|^2 + \eta_t^2 \mathbf{G}^2 \\
&\quad + \left\| \nabla_{w_0} f(w_0^t, \mathbf{w}^t) + \sum_{m \in A_t} \nabla_{w_m} f(w_0^t, \mathbf{w}^t) - \nabla_{w_0} f(w_0^t, \tilde{\mathbf{w}}^t) - \sum_{m \in A_t} \nabla_{w_m} f(w_0^t, \tilde{\mathbf{w}}^t) \right\|^2 \\
&\quad + \eta_t^2 \left\| \left([w_0^t, \mathbf{w}^t] - [w_0^*, \mathbf{w}^*] \right) \right\|^2 - 2\eta_t \left\langle \nabla_{w_0} f(w_0^t, \mathbf{w}^t), [w_0^t, \mathbf{w}^t] - [w_0^*, \mathbf{w}^*] \right\rangle \\
&\quad + \eta_t^2 \left\| \left([w_0^t, \mathbf{w}^t] - [w_0^*, \mathbf{w}^*] \right) \right\|^2 - 2\eta_t \left\langle \sum_{m \in A_t} \nabla_{w_m} f(w_0^t, \mathbf{w}^t), [w_0^t, \mathbf{w}^t] - [w_0^*, \mathbf{w}^*] \right\rangle \\
&\overset{6)}{\leq} \left\| [w_0^t, \mathbf{w}^t] - [w_0^*, \mathbf{w}^*] \right\|^2 + \eta_t^2 \mathbf{G}^2 \\
&\quad + L^2 \left\| \mathbf{w}^t - \tilde{\mathbf{w}}^t \right\|^2 \\
&\quad + 2\eta_t^2 \left\| \left([w_0^t, \mathbf{w}^t] - [w_0^*, \mathbf{w}^*] \right) \right\|^2 \\
&\quad - 2\eta_t \left\langle \nabla_{w_0} f(w_0^t, \mathbf{w}^t), [w_0^t, \mathbf{w}^t] - [w_0^*, \mathbf{w}^*] \right\rangle - 2\eta_t \left\langle \sum_{m \in A_t} \nabla_{w_m} f(w_0^t, \mathbf{w}^t), [w_0^t, \mathbf{w}^t] - [w_0^*, \mathbf{w}^*] \right\rangle \\
&\overset{7)}{\leq} \left\| [w_0^t, \mathbf{w}^t] - [w_0^*, \mathbf{w}^*] \right\|^2 + \eta_t^2 \mathbf{G}^2 + L^2 \tau^2 \mathbf{G}^2 \max_{i \in [\tau]} \left\{ \eta_{t-i}^2 \right\} \\
&\quad + 2\eta_t^2 \left\| \left([w_0^t, \mathbf{w}^t] - [w_0^*, \mathbf{w}^*] \right) \right\|^2 \\
&\quad - 2\eta_t \left\langle \nabla_{w_0} f(w_0^t, \mathbf{w}^t), [w_0^t, \mathbf{w}^t] - [w_0^*, \mathbf{w}^*] \right\rangle - 2\eta_t \left\langle \sum_{m \in A_t} \nabla_{w_m} f(w_0^t, \mathbf{w}^t), [w_0^t, \mathbf{w}^t] - [w_0^*, \mathbf{w}^*] \right\rangle \\
&\overset{8)}{\leq} \left\| [w_0^t, \mathbf{w}^t] - [w_0^*, \mathbf{w}^*] \right\|^2 + \eta_t^2 \mathbf{G}^2 + L^2 \tau^2 \mathbf{G}^2 \max_{i \in [\tau]} \left\{ \eta_{t-i}^2 \right\} + 2\eta_t^2 D^2 \\
&\quad - 2\eta_t \left\langle \nabla_{w_0} f(w_0^t, \mathbf{w}^t), [w_0^t, \mathbf{w}^t] - [w_0^*, \mathbf{w}^*] \right\rangle - 2\eta_t \left\langle \sum_{m \in A_t} \nabla_{w_m} f(w_0^t, \mathbf{w}^t), [w_0^t, \mathbf{w}^t] - [w_0^*, \mathbf{w}^*] \right\rangle
\end{aligned}
\tag{28}
$$

where 1) is the partial activation of clients and server optimization step at time step t, 2) note that $\{\nabla_{w_m} f(w_0^t, \tilde{\mathbf{w}}^t)\}_{m \in \{0\} \cup A_t}$ are in the non-intersect dimensions, 3) applies assumption 4 (bounded gradient), 4) applies $\langle a, b \rangle \leq \frac{1}{2} \|a\|^2 + \frac{1}{2} \|b\|^2$, 5) $\{\nabla_{w_m} f(w_0^t, \tilde{\mathbf{w}}^t)\}_{m \in \{0\} \cup A_t}$ are in the non-intersect dimensions, 6) applies assumption assumption 1 (Lipschitz Gradient). 7) applies Eq. 29 below, 8) applies assumption 7 (bounded parameter diameter).

$$
\begin{aligned}
&\left\| \mathbf{w}^t - \tilde{\mathbf{w}}^t \right\|^2 \\
&\overset{1)}{\leq} \left\| \sum_{i=1}^{\tau} \left(\mathbf{w}^{t+1-i} - \mathbf{w}^{t-i} \right) \right\|^2 \\
&\overset{2)}{\leq} \tau \sum_{i=1}^{\tau} \left\| \mathbf{w}^{t+1-i} - \mathbf{w}^{t-i} \right\|^2 \\
&= \tau \sum_{i=1}^{\tau} \left\| -\eta_{t-i} \sum_{m \in A_{t-i}} \nabla_{w_m} f(w_0^{t-i}, \mathbf{w}^{t-i}) \right\|^2 \\
&= \tau \sum_{i=1}^{\tau} \eta_{t-i}^2 \left\| \sum_{m \in A_{t-i}} \nabla_{w_m} f(w_0^{t-i}, \mathbf{w}^{t-i}) \right\|^2 \\
&\overset{3)}{\leq} \tau \mathbf{G}^2 \sum_{i=1}^{\tau} \eta_{t-i}^2
\end{aligned}
$$

$$\leq \tau^2 \mathbf{G}^2 \max_{i\in[\tau]} \left\{\eta_{t-i}^2\right\} \tag{29}$$

where 1) applies assumption 9 (uniformly bounded delay), 2) by Cauchy-Schwarz inequality, $\left(\sum_{i=0}^{n-1} x_i\right)^2 = \left(\sum_{i=0}^{n-1} 1 \cdot x_i\right)^2 \leq n \sum_{i=0}^{n-1} x_i^2$, 3) applies assumption 4 (bounded gradient), noting that the gradients are in non-intersect dimensions.

Rearrange Eq. 28:

$$\begin{aligned}&2\eta_t \left\langle \nabla_{w_0} f(w_0^t, \mathbf{w}^t), [w_0^t, \mathbf{w}^t] - [w_0^*, \mathbf{w}^*]\right\rangle + 2\eta_t \left\langle \sum_{m\in A_t} \nabla_{w_m} f(w_0^t, \mathbf{w}^t), [w_0^t, \mathbf{w}^t] - [w_0^*, \mathbf{w}^*]\right\rangle \\ \leq& \left\|[w_0^t, \mathbf{w}^t] - [w_0^*, \mathbf{w}^*]\right\|^2 - \left\|[w_0^{t+1}, \mathbf{w}^{t+1}] - [w_0^*, \mathbf{w}^*]\right\|^2 + \eta_t^2 \mathbf{G}^2 + L^2\tau^2\mathbf{G}^2 \max_{i\in[\tau]}\left\{\eta_{t-i}^2\right\} + 2\eta_t^2 D^2\end{aligned} \tag{30}$$

Taking expectation w.r.t. the activation client sets A_t from both sides.

$$\begin{aligned}&2\eta_t \left\langle \nabla_{w_0} f(w_0^t, \mathbf{w}^t), [w_0^t, \mathbf{w}^t] - [w_0^*, \mathbf{w}^*]\right\rangle + 2\eta_t \left\langle \sum_{m=1}^{M} p_m \nabla_{w_m} f(w_0^t, \mathbf{w}^t), [w_0^t, \mathbf{w}^t] - [w_0^*, \mathbf{w}^*]\right\rangle \\ \leq& \mathbb{E}\left\|[w_0^t, \mathbf{w}^t] - [w_0^*, \mathbf{w}^*]\right\|^2 - \mathbb{E}\left\|[w_0^{t+1}, \mathbf{w}^{t+1}] - [w_0^*, \mathbf{w}^*]\right\|^2 + \eta_t^2 \mathbf{G}^2 + L^2\tau^2\mathbf{G}^2 \max_{i\in[\tau]}\left\{\eta_{t-i}^2\right\} + 2\eta_t^2 D^2\end{aligned} \tag{31}$$

Taking the minimum of p_m:

$$\begin{aligned}&2\eta_t \min\left\{p_m\right\} \left\langle \nabla_{w_0} f(w_0^t, \mathbf{w}^t), [w_0^t, \mathbf{w}^t] - [w_0^*, \mathbf{w}^*]\right\rangle + 2\eta_t \min\left\{p_m\right\} \left\langle \sum_{m=1}^{M} \nabla_{w_m} f(w_0^t, \mathbf{w}^t), [w_0^t, \mathbf{w}^t] - [w_0^*, \mathbf{w}^*]\right\rangle \\ \leq& \mathbb{E}\left\|[w_0^t, \mathbf{w}^t] - [w_0^*, \mathbf{w}^*]\right\|^2 - \mathbb{E}\left\|[w_0^{t+1}, \mathbf{w}^{t+1}] - [w_0^*, \mathbf{w}^*]\right\|^2 + \eta_t^2 \mathbf{G}^2 + L^2\tau^2\mathbf{G}^2 \max_{i\in[\tau]}\left\{\eta_{t-i}^2\right\} + 2\eta_t^2 D^2\end{aligned} \tag{32}$$

Combining the gradient from different dimensions, and rearranging the equation:

$$\begin{aligned}&\left\langle \nabla_{[w_0,\mathbf{w}]} f(w_0^t, \mathbf{w}^t), [w_0^t, \mathbf{w}^t] - [w_0^*, \mathbf{w}^*]\right\rangle \\ \leq& \frac{1}{2\eta_t \min\left\{p_m\right\}} \left(\mathbb{E}\left\|[w_0^t, \mathbf{w}^t] - [w_0^*, \mathbf{w}^*]\right\|^2 - \mathbb{E}\left\|[w_0^{t+1}, \mathbf{w}^{t+1}] - [w_0^*, \mathbf{w}^*]\right\|^2\right) \\ &+ \frac{1}{2\eta_t \min\left\{p_m\right\}} \left(\eta_t^2 \mathbf{G}^2 + L^2\tau^2\mathbf{G}^2 \max_{i\in[\tau]}\left\{\eta_{t-i}^2\right\} + 2\eta_t^2 D^2\right) \\ \overset{1)}{\leq}& \frac{1}{2\min\left\{p_m\right\}} \left(\frac{Q}{\eta_t} + \eta_t \mathbf{G}^2 + \frac{L^2\tau^2\mathbf{G}^2 \max_{i\in[\tau]}\left\{\eta_{t-i}^2\right\}}{\eta_t} + 2\eta_t D^2\right)\end{aligned} \tag{33}$$

where 1) denote $Q = \mathbb{E}\left\|[w_0^t, \mathbf{w}^t] - [w_0^*, \mathbf{w}^*]\right\|^2 - \mathbb{E}\left\|[w_0^{t+1}, \mathbf{w}^{t+1}] - [w_0^*, \mathbf{w}^*]\right\|^2$ for brevity.

By convexity (assumption 6):

$$f^t(w_0^t, \mathbf{w}^t) - f^t(w_0^*, \mathbf{w}^*) \leq \left\langle \nabla_{[w_0,\mathbf{w}]} f^t, [w_0^t, \mathbf{w}^t] - [w_0^*, \mathbf{w}^*]\right\rangle \tag{34}$$

Summing over $t = 1...T$, and if we set a diminish learning rate $\eta_t = \frac{D}{\mathbf{G}\sqrt{t}}$ with ($\frac{1}{\eta_0} \triangleq 0$):

$$\begin{aligned}&\sum_{t=1}^{T} \left(f^t(w_0^t, \mathbf{w}^t) - f^t(w_0^*, \mathbf{w}^*)\right) \\ \leq& \sum_{t=1}^{T} \left(\left\langle \nabla_{[w_0,\mathbf{w}]} f^t, [w_0^t, \mathbf{w}^t] - [w_0^*, \mathbf{w}^*]\right\rangle\right)\end{aligned}$$

$$
\begin{aligned}
&\overset{1)}{\leq}\frac{1}{2\min\{p_m\}}\sum_{t=1}^{T}\left(\frac{Q}{\eta_t}+\eta_t\mathbf{G}^2+\frac{L^2\tau^2\mathbf{G}^2\max_{i\in[\tau]}\left\{\eta_{t-i}^2\right\}}{\eta_t}+2\eta_t D^2\right)\\
&\overset{2)}{=}\frac{1}{2\min\{p_m\}}\left\{\sum_{t=1}^{T}\frac{\mathbb{E}\left\|[w_0^t,\mathbf{w}^t]-[w_0^*,\mathbf{w}^*]\right\|^2-\mathbb{E}\left\|[w_0^{t+1},\mathbf{w}^{t+1}]-[w_0^*,\mathbf{w}^*]\right\|^2}{\eta_t}+\sum_{t=1}^{T}\left[\eta_t(\mathbf{G}^2+2D^2+L^2\tau^2\mathbf{G}^2)\right]\right\}\\
&\overset{3)}{\leq}\frac{1}{2\min\{p_m\}}\left\{\sum_{t=1}^{T}\mathbb{E}\left\|[w_0^t,\mathbf{w}^t]-[w_0^*,\mathbf{w}^*]\right\|^2\left(\frac{1}{\eta_t}-\frac{1}{\eta_{t+1}}\right)+\sum_{t=1}^{T}\left[\eta_t(\mathbf{G}^2+2D^2+L^2\tau^2\mathbf{G}^2)\right]\right\}\\
&\overset{4)}{\leq}\frac{1}{2\min\{p_m\}}\left\{\sum_{t=1}^{T}D^2\left(\frac{1}{\eta_t}-\frac{1}{\eta_{t+1}}\right)+\sum_{t=1}^{T}\left[\eta_t(\mathbf{G}^2+2D^2+L^2\tau^2\mathbf{G}^2)\right]\right\}\\
&=\frac{1}{2\min\{p_m\}}\left[D^2\frac{1}{\eta_T}+(\mathbf{G}^2+2D^2+L^2\tau^2\mathbf{G}^2)\sum_{t=1}^{T}\eta_t\right]\\
&\overset{5)}{\leq}\frac{1}{2\min\{p_m\}}\left[\mathbf{G}D\sqrt{T}+(\mathbf{G}D+\frac{2D^3}{\mathbf{G}}+L^2\tau^2\mathbf{G}D)\cdot 2\sqrt{T}\right]\\
&\leq\frac{3\mathbf{G}^2D+4D^3+2L^2\tau^2\mathbf{G}^2D}{2\mathbf{G}\min\{p_m\}}\sqrt{T}
\end{aligned}
\tag{35}
$$

where 1) applies Eq. 33,
2) η_t is diminish. Expand Q,
3) $\frac{1}{\eta_0}\triangleq 0$, and $\left\|[w_0^{T+1},\mathbf{w}^{T+1}]-[w_0^*,\mathbf{w}^*]\right\|^2\geq 0$,
4) applies assumption 7 (bounded parameter diameter).
5) $\sum_{t=1}^{T}\frac{1}{\sqrt{t}}\leq 2\sqrt{T}$.

The proof of Theorem 2 is complete. ■

C Extra details

C.1 Algorithm 1 in a synchronous manner

We also provide a synchronous version of the algorithm 1 below.

Algorithm 3 Event-driven online VFL on Dynamic Local Regret

```
Input: hyperparameter $l$, $\alpha$, $\eta$
Output: server model $w_0$, client models $w_m \in [M]$
initialize model $w_m$ for all participants $m \in \{0, 1, ...M\}$
for $t \in [T]$ do
    for $m \in A_t$ do
        activated client $m$ send $h_m(w_m^t; x_m^t)$ to the server.
    end for
    for $m \in \bar{A}_t$ do
        server queries the passive client $m$.
        passive client $m$ send $h_m(w_m^t; x_m^t)$ to the server.
    end for
    server updates its model $w_0^{t+1} \leftarrow w_0^t - \eta_0 \cdot \nabla_{w_0} S_{t,l,\alpha}(w_0^t, \mathbf{w}^t)$
    for $m \in A_t$ do
        server send $v_m = \frac{\partial S_{t,l,\alpha}(w_0^t, \mathbf{w}^t)}{\partial h_m(w_m^t; x_m^t)}$ to the client $m$.
        client $m$ update parameter $w_m^{t+1} \leftarrow w_m^t - \eta_m \cdot v_m \cdot \frac{\partial h_m}{\partial w_m}$
    end for
end for
```

C.2 Algorithm: OGD-Full

Below we introduce the algorithm for the OGD-Full baseline in Algorithm 4 adapted from Wang & Xu (2023), and transformed into a partial gradient-based approach similar to the split neural network method proposed by Vepakomma et al. (2018). It is important to highlight that the utilization of multiple local updates in Wang & Xu (2023) is not applicable within the context of our study on the general VFL framework because the multiple local update approach requires that clients possess labeled data, which does not align with the setting in our paper.

Algorithm 4 Online VFL with Online Gradient Descent (OGD-Full)

```
Input: Learning rate $\eta$
Output: Model parameter $w_m$ for all workers $m \in \{0, 1, ...M\}$.
Initialize $w_m$ for all participants $m \in \{0, 1, ...M\}$
for $t \in [T]$ do
    for $m \in [M]$ do
        Client $m$ send $h_m(w_m; x_m)$ to the server.
    end for
    The server updates its model $w_0 \leftarrow w_0 - \eta_0 \cdot \frac{\partial f(w_0, \mathbf{w})}{\partial w_0}$
    for $m \in [M]$ do
        Server send $v_m = \frac{\partial f(w_0, \mathbf{w})}{\partial h_m}$ to the client $m$.
        Client $m$ update parameter $w_m \leftarrow w_m - \eta_m v_m \cdot \frac{\partial h_m}{\partial w_m}$
    end for
end for
```

C.3 Algorithm: event-driven online VFL using static local regret

We also provide a Static Time-Smoothed Stochastic Gradient Descent algorithm (Aydore et al., 2019; Hazan et al., 2017) which is adapted to Event-Driven Online VFL. Note that the static slide window is $F_{t,l}(w_0^t, \mathbf{w}^t) = \frac{1}{l}\sum_{i=0}^{l} f^{t-i}(w_0^t, \mathbf{w}^t)$, where the time step for the parameter is t.

Algorithm 5 Event-driven online VFL on Static Local Regret

```
Input: Hyperparameter $l$, $\alpha$, $\eta$, fixed model $w^*$
Output: Server model $w_0$, client models $w_m \in [M]$
Initialize model $w_m$ for all participants $m \in \{0, 1, \dots, M\}$
for $t \in [T]$ do
    for $m \in A_t$ do
        Activated client $m$ sends a window of embeddings $\left\{h_m(w_m^t; x_m^{t-i})\right\}_{i=0}^{l-1}$ to the server.
    end for
    for $m \in \bar{A}_t$ do
        Server queries the passive client $m$, for the window of embeddings of length $l$
        Passive clients $m$ sends the window of embeddings $\left\{h_m(w_m^t; x_m^{t-i})\right\}_{i=0}^{l-1}$ to the server.
    end for
    Server updates its model $w_0^{t+1} \leftarrow w_0^t - \eta_0 \cdot \nabla_{w_0} F_{t,l}(w_0^t, \mathbf{w}^t)$
    # $\nabla_{w_0} F_{t,l}(w_0^t, \mathbf{w}^t) = \frac{1}{l}\sum_{i=0}^{l} \nabla_{w_0} f^{t-i}(w_0^t, \mathbf{w}^t)$
    for $m \in A_t$ do
        Server sends a window of gradient $\left\{\nabla_{h_m} f^{t-i}(w_0^t, \mathbf{w}^t)\right\}_{i=0}^{l-1}$ to client $m$.
        Client $m$ updates parameter $w_m^{t+1} \leftarrow w_m^t - \eta_m \cdot \nabla_{w_m} F_{t,l}(w_0^t, \mathbf{w}^t)$
        # $\nabla_{w_m} F_{t,l}(w_0^t, \mathbf{w}^t) = \frac{1}{l}\sum_{i=0}^{l} \nabla_{w_m} f^{t-i}(w_0^t, \mathbf{w}^t) =$
        $\frac{1}{l}\sum_{i=0}^{l} \nabla_{h_m} f^{t-i}(w_0^t, \mathbf{w}^t) \cdot \nabla_{w_m} h_m(w_m^t; x_m^{t-i})$
    end for
end for
```

D Supplement experiments

D.1 Ablation study of DLR parameters

In order to achieve a thorough comprehension of applying DLR in the VFL framework, we conduct an ablation study on several key parameters, including the sliding window length l, and attenuation coefficient α.

Sliding window length l The sliding window length l determines the length of the exponential average window of the DLR. A larger window implies that a greater number of past gradients influence the update at the current time. In the DLR-Full framework, employed under the concept drift data stream and with an attenuation coefficient of $\alpha = 0.95$, we vary the window length $l = 10, 50, 100, 150$, and conduct the training. It's worth noting that $0.95^{10} \approx 0.60$, $0.95^{50} \approx 0.08$, and $0.95^{100} \approx 0.006$ represent three different shapes of the exponential sliding average window.

Figure 5 presents the run-time error rate of different window lengths in the DLR-Full framework. As observed in the figure, the DLR is stable across different window lengths, which is consistent with the result reported by Aydore et al. (2019).

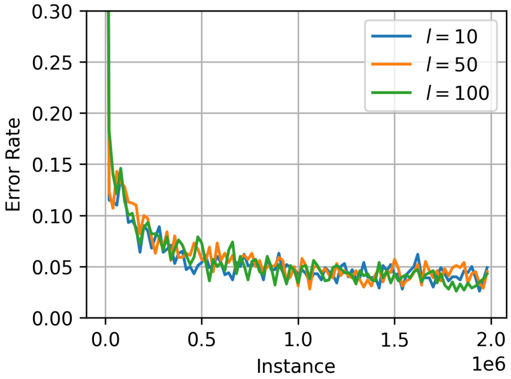

Figure 5: Ablation study on window length l

Attenuation coefficient α The attenuation coefficient of dynamic local regret influences the weighting of past gradients: a larger α indicates that the influence of past gradients decreases more slowly, while a smaller α leads to a quicker attenuation of the effect of old parameters. For our experiment, we utilize a sliding window length of 100 and vary α across 0.99, 0.95, and 0.9 in the DLR-Full framework. Notably, $0.99^{100} \approx 0.37$, $0.95^{90} \approx 0.01$, and $0.9^{44} \approx 0.01$ represent three distinct shapes of the exponential averaging window. Figure 6 illustrates the run-time error rate for different values of α. We observed that better results were obtained for $\alpha = 0.95$ and 0.9. Therefore, we recommend using exponential average windows where the tail approximates 0.

D.2 Experiment on other online learning dataset

SUSY dataset SUSY (Whiteson, 2014b) is a physics dataset from the UCI repository. It is a classification problem to distinguish between a signal process that produces supersymmetric particles and a background process that does not. Eighteen features are used for each sample to determine whether it belongs to the signal or background class. The first 8 features are kinematic properties measured by the particle detectors in the accelerator. The last ten features are functions of the first 8 features; these are high-level features derived by physicists to help discriminate between the two classes. The preprocess method follows the same process in Sahoo et al. (2018). In the experiment, two clients are employed, and each holds half of the feature set. Each client models are composed of four fully-connected layers with the following neuron configurations: 32, 64, 98, and an output layer containing 128 neurons. After each hidden layer, a Rectified Linear Unit (ReLU) activation function is applied to introduce non-linearity. On the server side, a five-layer MLP is employed. This model integrates the concatenated outputs from the client models and processes them through

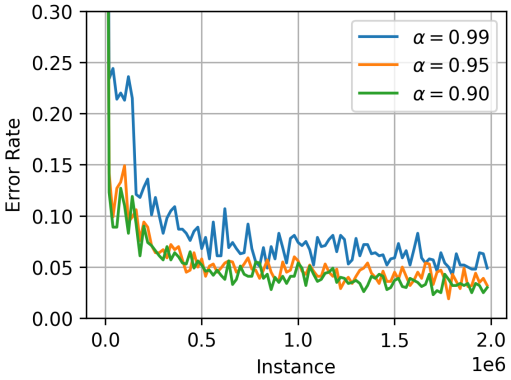

Figure 6: Attenuation coefficient α

successive fully-connected layers with neuron counts of 256, 128, 64, and 32, ending with an output layer with 2 neurons. We tested various learning rates within the range of [0.01, 0.003, 0.001] to identify the optimal setting for our models under different experimental conditions. The exponential weighted sliding window's length of the DLR is tuned from $\{10, 50\}$, and the attenuation coefficient α from $\{0.95, 0.99\}$. The activation probability p for the "Random" activation is selected from $\{0.25, 0.5, 0.75\}$. The activation threshold Γ is tuned from $\{-1, -0.5, 0, 0.5\}$. The results for both the stationary and non-stationary data streams are depicted in Figures 7 and 8, respectively. These figures demonstrate that the DLR model achieves stable convergence in both Full and Partial activation modes, consistently outperforming OGD across varying data conditions. Additionally, DLR is learning faster than SLR in both scenarios.

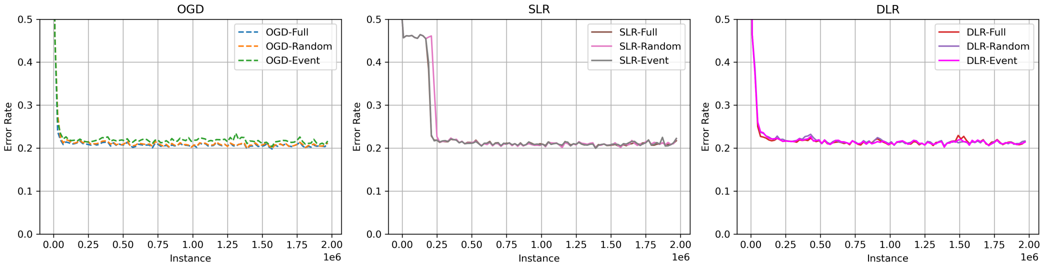

Figure 7: SUSY dataset, stationary case

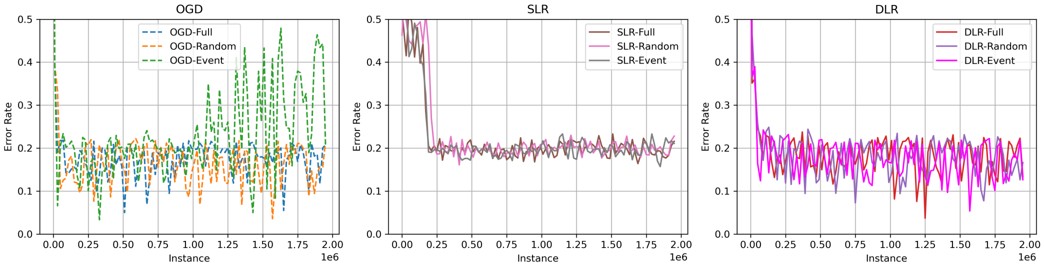

Figure 8: SUSY dataset, non-stationary case

HIGGS dataset HIGGS (Whiteson, 2014a) is also a physics dataset from UCI repository. It is a classification challenge aimed at distinguishing between a signal process that produces Higgs bosons and a background process that does not. Each sample comprises 28 features: 21 low-level features representing kinematic properties measured by particle detectors in the accelerator and 7 high-level features derived from the first 21. The model architecture employed in this experiment is identical to that used in the SUSY study, with the sole difference being the number of input features. The hyperparameter tuning procedure also remains consistent with the previous experiment. The results for both stationary and non-stationary conditions are depicted in Figure 9 and Figure 10, respectively. These results illustrate that the DLR model achieves comparable performance under stationary conditions and exhibits enhanced stability under non-stationary conditions. Furthermore, DLR demonstrates faster learning compared to SLR, with less initial stagnation at the early phases of training.

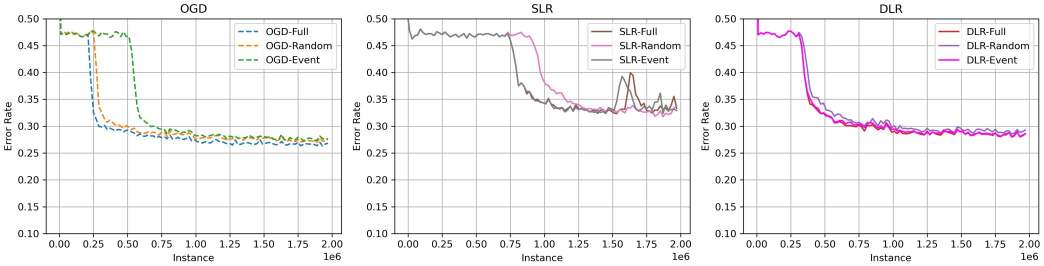

Figure 9: HIGGS dataset, stationary case

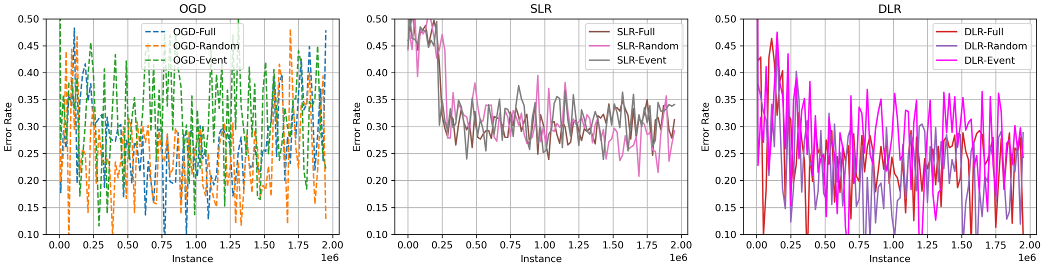

Figure 10: HIGGS dataset, non-stationary case

D.3 Additional experiment on event-driven online VFL

Scalability of the framework (number of total clients): To evaluate scalability, we increase the total number of clients in the experiment described in Section 5.2 on the stationary iMNIST dataset to 8 and 16 clients. Most settings remain consistent with those described in Section 5.1, with the number of clients increased to 8 and 16. Features are evenly distributed among the clients, and the input size of each client's model is adjusted accordingly. While the output size of the clients remains unchanged, the input size of the server model is scaled up to accommodate the increased number of clients. The results for 8 clients are presented in Figure 11, while the results for 16 clients are presented in Figure 12. Those figures demonstrate that when scale up the number of clients, the conclusion remain consistent: OGD is less stable under partial client activation, while DLR converges more rapidly than SLR. This consistency supports the robustness and scalability of our proposed framework.

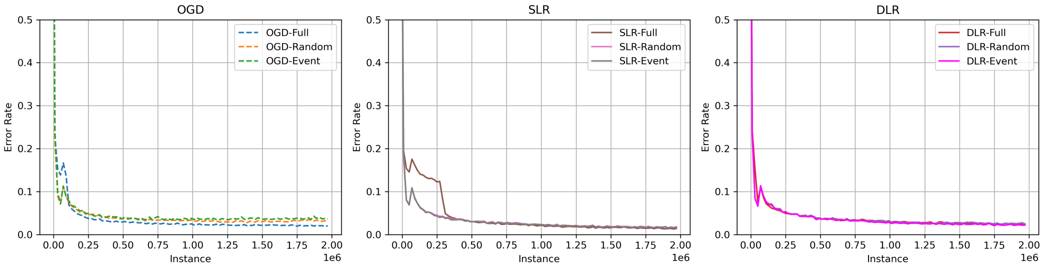

Figure 11: Run-time error rates on the iMNIST with 16 clients under stationary data stream

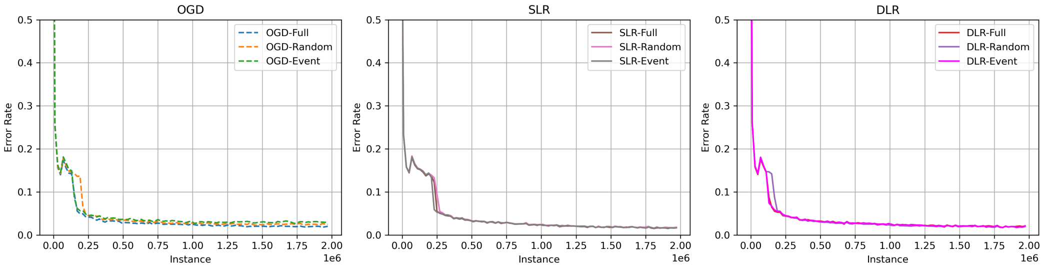

Figure 12: Run-time error rates on the iMNIST with 16 clients under stationary data stream

Number of activated clients: To further examine the impact of partial client activation, we conducted experiments in which a fixed number of clients were randomly selected to participate in each round. The experimental setup remains consistent with the configuration described in Section 5.1. The experiment involves 4 clients, with the activated clients randomly selected in each round. The number of activated clients varies from 1 to 4. The results are presented in Figure 13. The results suggest that increasing the number of activated clients enhances the stability of learning and reduces the variability in error rate over time. However, even with a single activated client, the model eventually achieves comparable performance, demonstrating robustness under limited client activation.

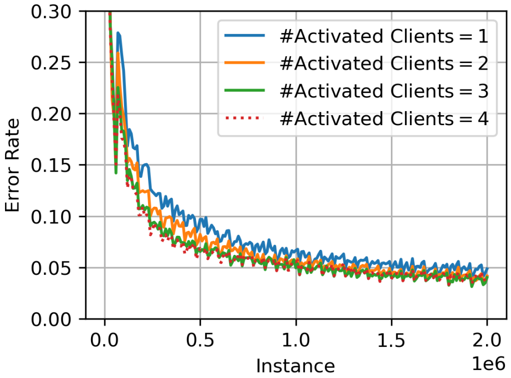

Figure 13: Number of activated client.

Client activation statistic: To further analyze and validate the partial client activation mechanism, we conducted additional experiments to monitor and record the activation frequency of each client throughout the training process. Table 5 presents the activation frequencies for each client under various event-driven framework settings described in Section 5.4, considering different activation probabilities p and activation thresholds Γ. Note that the activation frequency for each client is defined as the ratio of the number of iterations in which the client was activated to the total number of iterations. Additionally, we present results for extreme Γ values, including $\Gamma = +\infty$ (where no clients are activated) and $\Gamma = -\infty$ (where all clients are always activated). This table also provides insight into the selection of Γ values in Section 5.4, ensuring that the chosen Γ values cover a wide range of activation frequencies for the clients.

Table 5: Frequency of activation for each client under different settings

Setting	**Client 1**	**Client 2**	**Client 3**	**Client 4**
Random				
$p = 0.25$	0.249747	0.250473	0.250277	0.250142
$p = 0.5$	0.499507	0.499811	0.500266	0.500179
$p = 0.75$	0.749623	0.750214	0.750155	0.750257
$p = 1.0$	1.0	1.0	1.0	1.0
Event				
$\Gamma = +\infty\ (+100)$	0	0	0	0
$\Gamma = 0.6$	0.000006	0.066729	0.101827	0.000204
$\Gamma = 0.2$	0.006346	0.461186	0.456317	0.038649
$\Gamma = -0.2$	0.355166	0.982896	0.985216	0.596582
$\Gamma = -\infty\ (-100)$	1.0	1.0	1.0	1.0

E Related works

E.1 Vertical federated learning

Federated learning is a decentralized machine learning approach where models are trained collaboratively across multiple participants. In terms of how the data is distributed among the participants of federated learning, federated learning can be roughly categorized into Horizontal Federated Learning (HFL) and Vertical Federated Learning (VFL). HFL is centered on collaborative model training among the devices possessing the same feature set but with non-overlapping samples (McMahan et al., 2017; Karimireddy et al., 2020; Li et al., 2020; 2021; Marfoq et al., 2022; Mishchenko et al., 2019). In contrast, VFL involves collaboration among participants with non-overlapping feature sets but on the same sample (Vepakomma et al., 2018; Yang et al., 2019; Liu et al., 2019; Chen et al., 2020; Gu et al., 2020; Zhang et al., 2021c;b;a; Wang et al., 2023; Zhang et al., 2024; Qi et al., 2022).

Research on VFL is facing a wide variety of challenges. The primary concern in VFL lies in privacy and security. To protect the privacy, VFL frameworks commonly employ privacy protection methods such as differential privacy (DP) (Ranbaduge & Ding, 2022; Wei et al., 2020; Zhou et al., 2020; Huang et al., 2015; Zhou et al., 2020; Huang et al., 2015), homomorphic encryption (HE) (Hardy et al., 2017; Liu et al., 2020), secure multiparty computation (SMC)(Fang et al., 2021; Mugunthan et al., 2019; Gu et al., 2020). The second challenge encountered in VFL pertains to communication costs. This is due to the transmission of large intermediate model information through the network during VFL training. To enhance communication efficiency in federated learning, the most common method is to compress the communication of the VFL (Castiglia et al., 2022; Wang et al., 2022; 2023). Besides, another common strategy involves using multiple local updates on each participant to reduce the number of communication rounds (Liu et al., 2019; Fu et al., 2022). However, these methods often assume the client possesses additional information, such as data labels, which may not align with the general VFL framework. The third challenge of VFL involves adapting to a new data distribution efficiently. The distribution of data may change throughout the life cycle of VFL. Offline learning settings require retraining the VFL model when encountering concept drift, which is neither computationally efficient nor communication efficient within the VFL framework. Therefore, applying online learning to the VFL framework is essential (Wang & Xu, 2023). Moreover, online VFL also alleviates storage costs, particularly when clients have limited storage space. In the offline learning paradigm, both the client and the server possess large datasets. Certain types of VFL based on the offline learning paradigm are required to store the intermediate results corresponding to each sample which takes a large storage cost (Chen et al., 2020; Wang et al., 2023; Zhang et al., 2021a). In contrast, online learning obviates the need to store large datasets. Participants receive one sample from the environment at a time, making it suitable for VFL scenarios with low-capacity participants, such as sensor networks (Wang & Xu, 2023; Liu et al., 2022).

E.2 Online learning

Online learning is a paradigm wherein models are continually updated as fresh data becomes accessible, as opposed to being trained on a static dataset in a batch mode. In the well-established online convex optimization, the primary objective is to minimize regret (Hoi et al., 2018; Hazan et al., 2016), which is the difference between the cumulative loss of the player and the optimal solution. The most straightforward approach to online convex optimization involves directly optimizing regret, known as *follow the leader (FTL)* (Hazan et al., 2016). However, FTL can be unstable which motivates the need to stabilize the training through regularization. Therefore the idea of *follow the regularized leader (FTRL)* (Shalev-Shwartz & Singer, 2007; Abernethy et al., 2009; Hazan & Kale, 2010; McMahan et al., 2013) was proposed to address this problem. In FTL and FTRL, the learner is required to store all previously seen samples, which is inefficient in terms of memory and computation. Another prevalent algorithm for online convex optimization is online gradient descent (OGD) (Zinkevich, 2003). This iterative optimization algorithm updates the model using the gradient on the incoming data point. In theory, it achieves a sublinear regret of $O(\sqrt{T})$. Hazan et al. (2007) further introduced the adaptive OGD algorithm, which offers intermediate regret rates between $O(\sqrt{T})$ and $O(\log(T))$.

While traditional online learning predominantly addresses convex cases with shallow models, recent research has shifted its focus towards the non-convex case of online learning. Hazan et al. (2017) introduced the concept of *local regret* as a surrogate for regret analysis in non-convex online learning. Unlike the regret utilized in online convex optimization, the fundamental idea is to confine the regret within a sliding window, rendering it “local”. Aydore et al. (2019) further explores the concept of local regret, introducing *dynamic local regret* to address concept drift in the data stream. This approach incorporates an exponential average over the sliding window of local regret. Additionally, they utilize past gradients for the window, enhancing computational efficiency compared to the static local regret proposed by Hazan et al. (2017). Gao et al. (2018) present an online normalized gradient descent algorithm for scenarios with gradient information available and a bandit online normalized gradient descent algorithm when only loss function values are accessible. They demonstrate achieving a regret bound of $O(\sqrt{T + V_T T})$. Sahoo et al. (2018) propose online deep learning to tackle online learning within the deep learning paradigm. Agarwal et al. (2019) addresses online learning in a non-convex setting by leveraging an offline optimization oracle. Their study demonstrates that by enhancing the oracle model, online and statistical learning models achieve computational equivalence. Suggala & Netrapalli (2020) demonstrate achieving an $O(\sqrt{T})$ rate using the Follow the Perturbed Leader (FTPL) algorithm. Héliou et al. (2020) address online learning with non-convex losses, introducing a mixed-strategy learning policy based on dual averaging under the assumption of inexact model feedback for the loss function.

To facilitate a clear comparison, we present Table 6 which summarizes the regret bounds of the most relevant works, including both standalone online learning algorithms and the online VFL. Specifically, compared to the theoretical results of standalone DLR (Aydore et al., 2019), the additional constant term $(2W\mathbf{G})$ arises from the missing gradient elements of passive clients due to the dynamic partial activation of clients in the event-driven online VFL framework.

Table 6: Comparison of the regret bound

Method	**Online Convex Learning**	**Online Non-Convex Learning**
Standalone		
OGD (Hazan et al., 2016)	$O(\sqrt{T})$	-
SLR (Hazan et al., 2017)	-	$SLR_w(T) \leq \frac{T}{w}(8\beta M + \sigma^2)$
DLR (Aydore et al., 2019)	-	$DLR_w(T) \leq \frac{T}{W}(8\beta M + \sigma^2)$
Online VFL		
Online VFL (Wang & Xu, 2023)	$O(\sqrt{T})$	-
Event-Driven Online VFL (ours)	$O(\sqrt{T})$	$DLR_w(T) \leq \frac{T}{W} \cdot \frac{p_{max}}{p_{min}} \cdot \left(\frac{8\beta M}{p_{max}} + 2\sigma^2 + 2W\mathbf{G}\right)$