Distributed Online Frank-Wolfe under Delayed Feedback

Tuan-Anh Nguyen

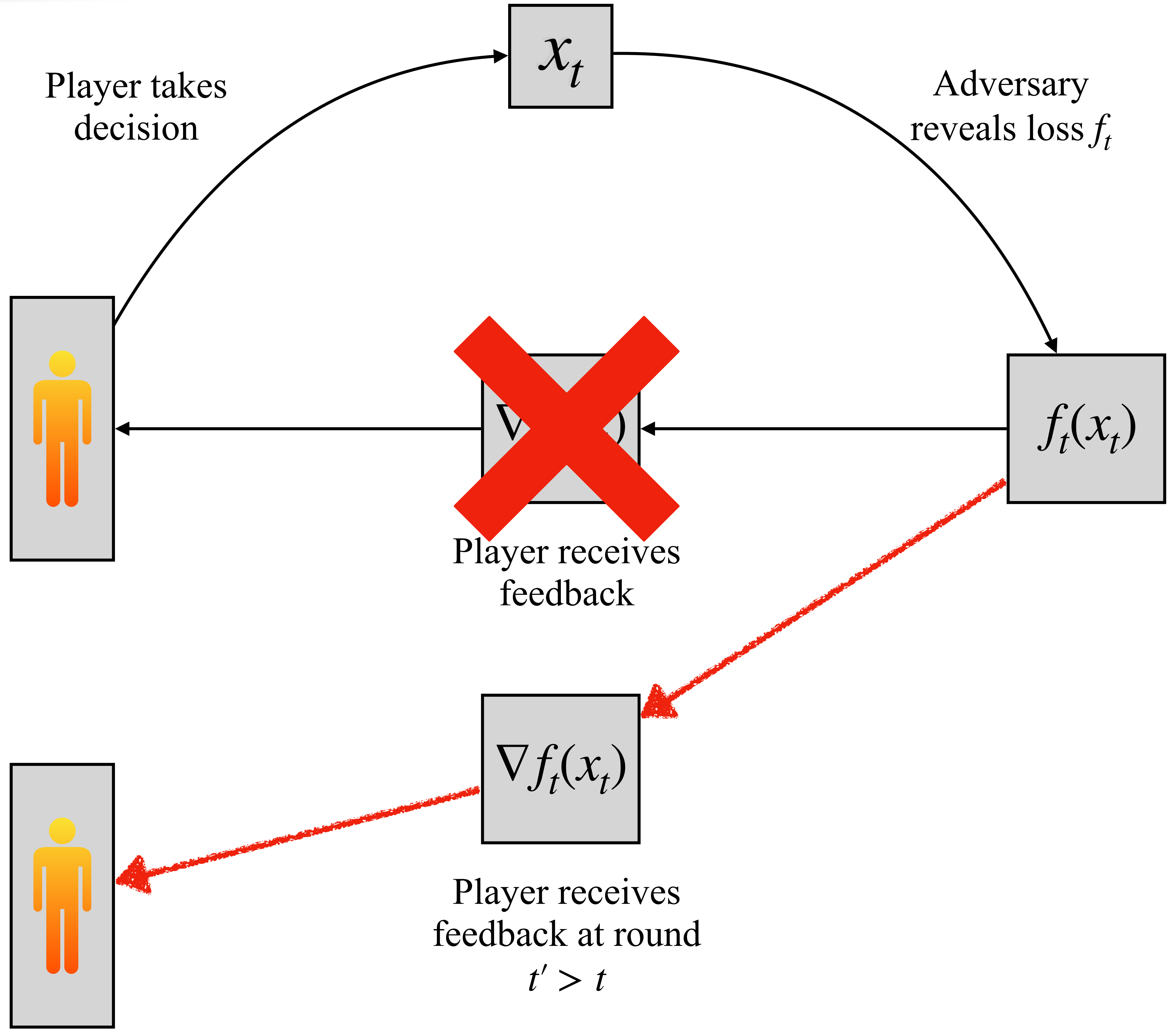
Journée de Recherche en Apprentissage Frugal
Grenoble, France

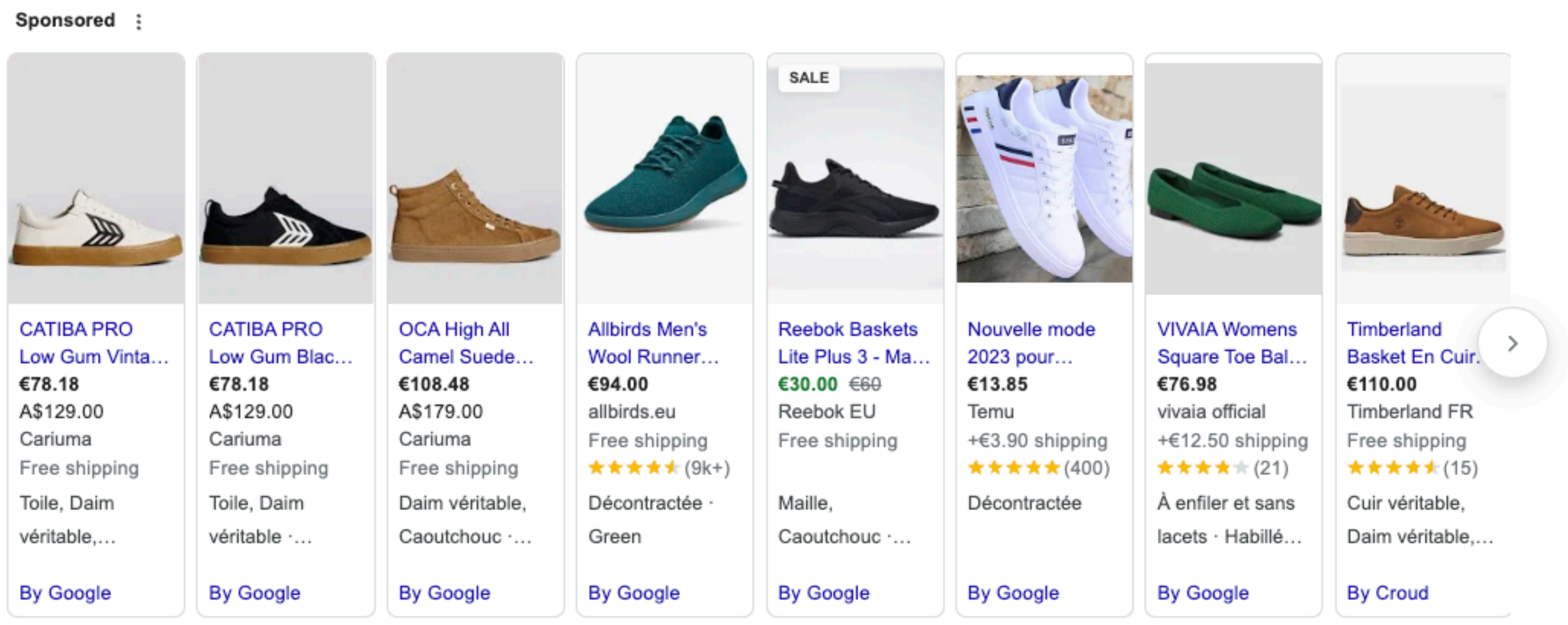
$$f(x)$$

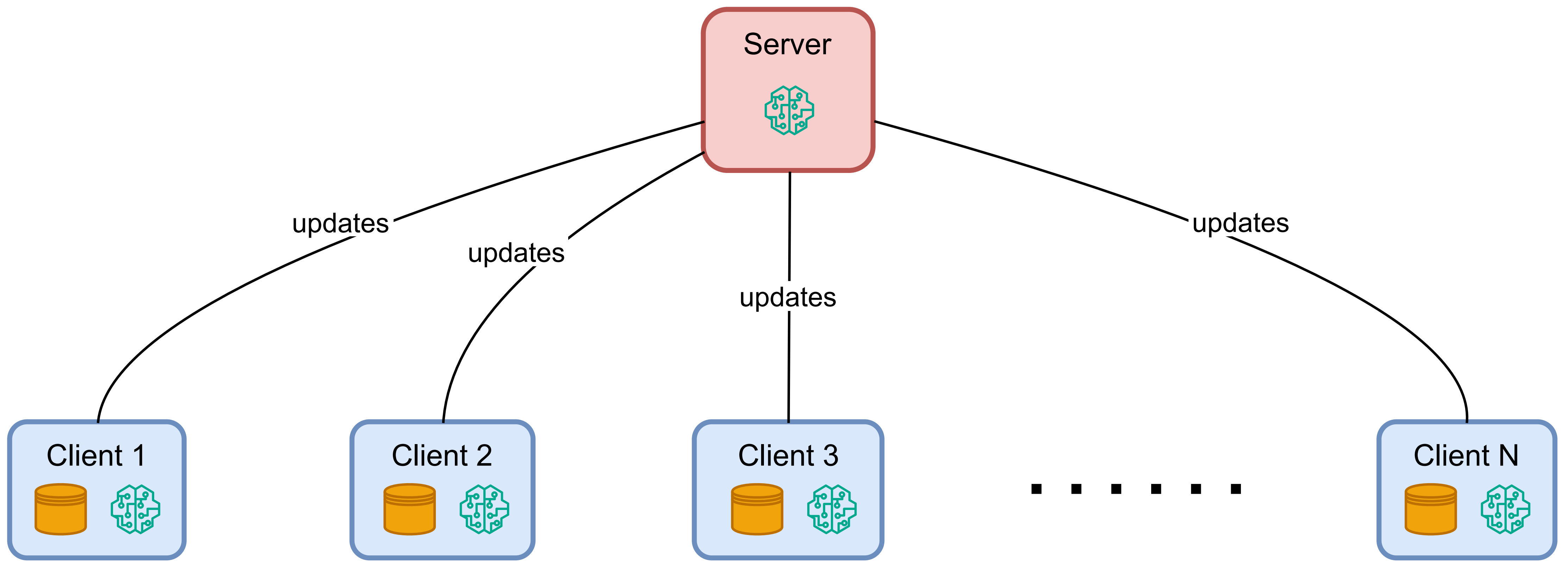
$\{SHIBA\}$

$\{HUSKY\}$

$$\min_{x \in \mathcal{K}} f(x) = \mathbb{E}_{a,b}\left[\ell(h_x(a), b))\right]$$







Player's goal : Determine a sequence of actions

$x_1, \ldots, x_T$ **minimising the cumulative loss** $\sum_{t=1}^{T} f_t(x_t)$

Why Frank-Wolfe ?

Vanilla Frank-Wolfe :

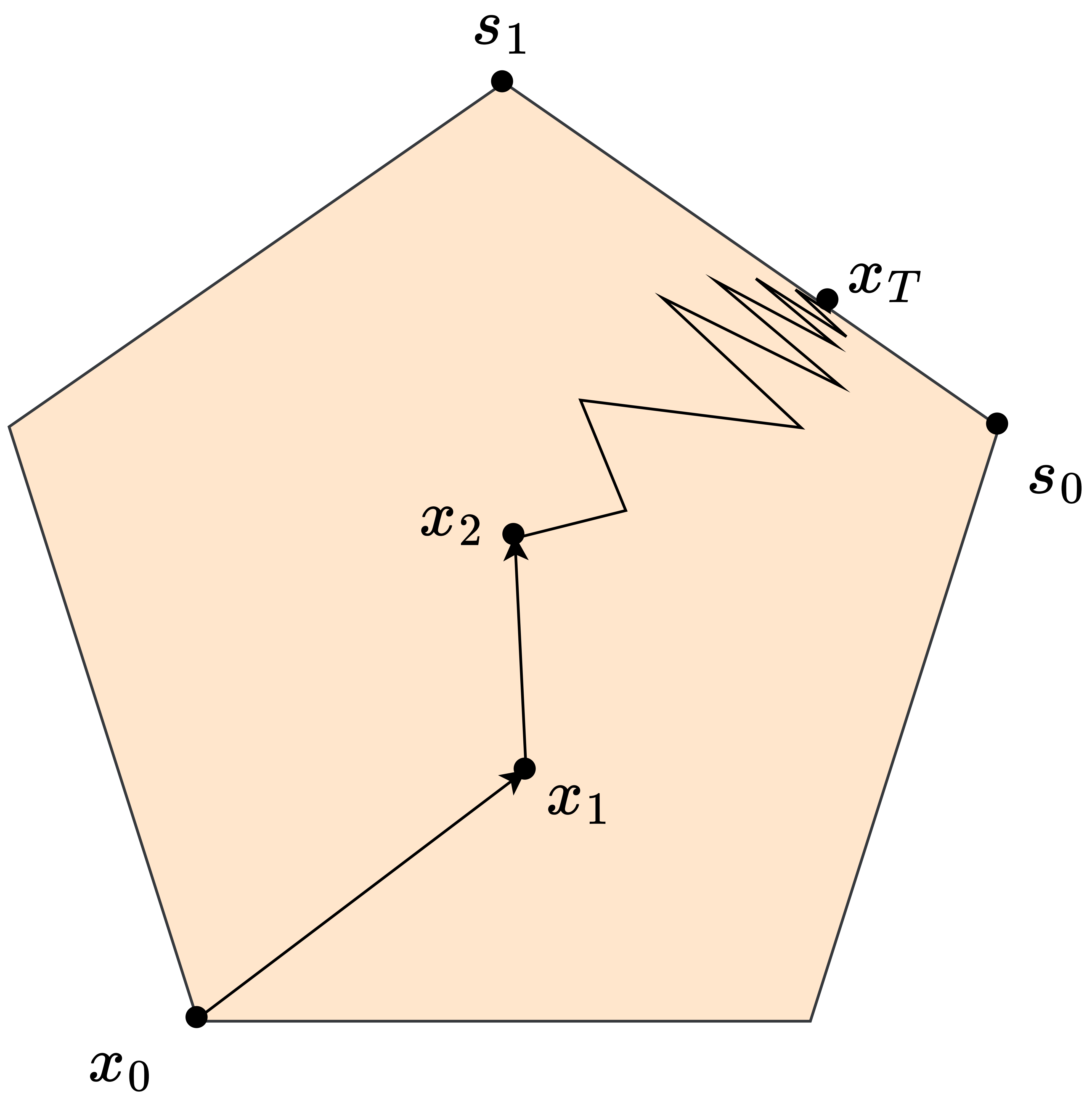
1.Linear Oracle: $s_t = \underset{s \in \mathcal{K}}{\operatorname{argmin}} \langle \nabla f(x_t), s \rangle$

2.Update : $x_{t+1} = x_t + \eta_t (s_t - x_t)$

Gradient Descent :

1.Update: $y_{t+1} = x_t - \eta \nabla f(x_t)$

2.Projection : $x_{t+1} = \Pi_{\mathcal{K}}(y_{t+1})$



Hazan, E. 2015, Introduction to online convex optimization. *Foundations and Trends in Optimization*.

Why Frank-Wolfe ?

Set	Linear minimization	Projection
n-dimensional ℓ_p-ball, $p \neq 1, 2, \infty$	$\mathcal{O}(n)$	$\tilde{\mathcal{O}}(n/\varepsilon^2)$
Nuclear norm ball of $n \times m$ matrices	$\mathcal{O}(\nu \ln(m+n)\sqrt{\sigma_1}/\sqrt{\varepsilon})$	$\mathcal{O}(mn \min\{m, n\})$
Flow polytope on a graph with m vertices and n edges with capacity bound on edges	$\mathcal{O}((n \log m)(n + m \log m))$	$\mathcal{O}(n^4 \log n)$
Birkhoff polytope ($n \times n$ doubly stochastic matrices)	$\mathcal{O}(n^3)$	$\tilde{\mathcal{O}}(n^2/\varepsilon^2)$

Gábor Braun, Alejandro Carderera, Cyrille W Combettes, Hamed Hassani, Amin Karbasi, Aryan Mokhtari, and Sebastian Pokutta, *Conditional Gradient Methods*, arXiv:2211.14103 [math.OC]

Vanilla Frank-Wolfe :

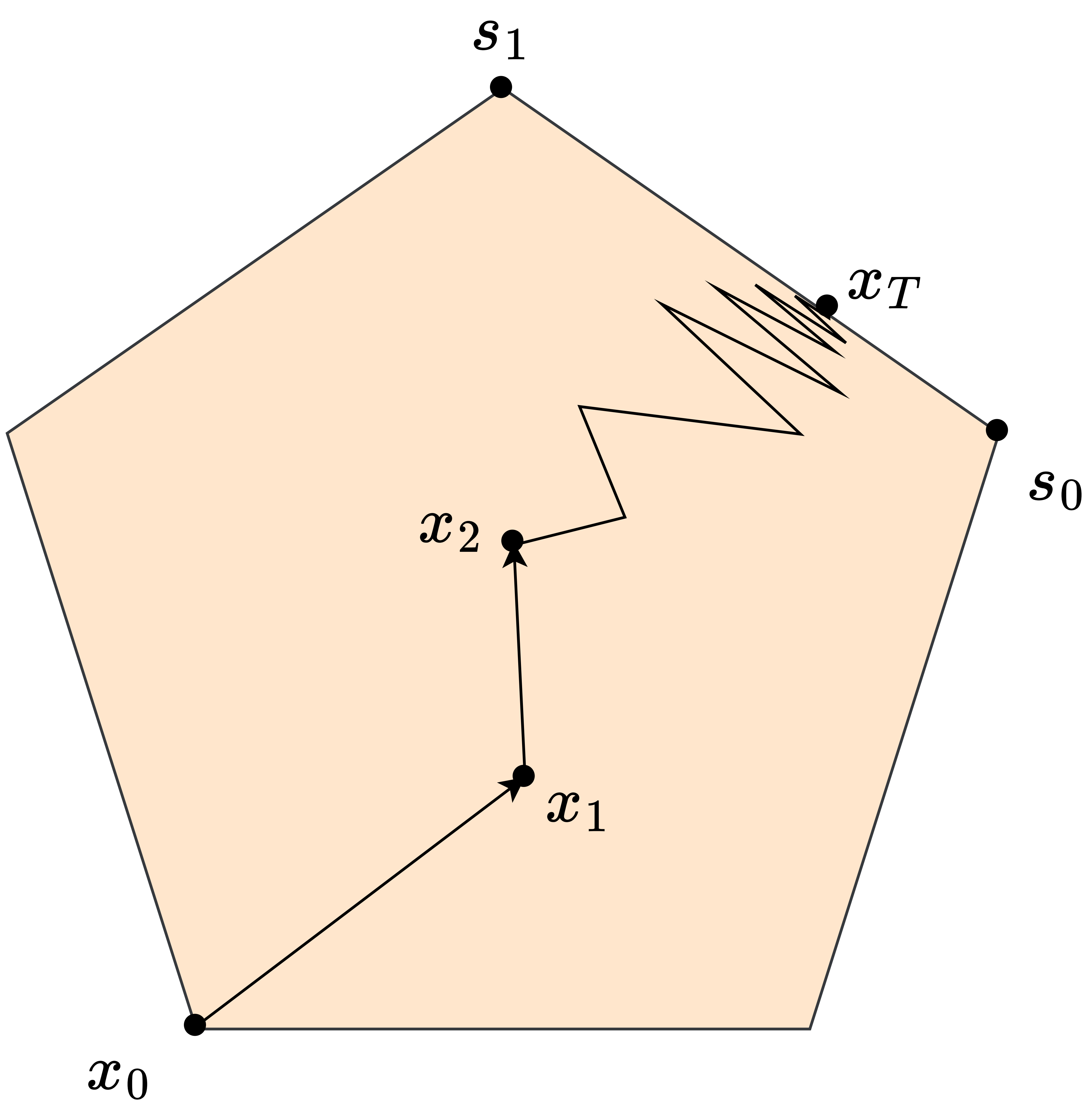
1.Linear Oracle: $s_t = \underset{s \in \mathcal{K}}{\operatorname{argmin}} \langle \nabla f(x_t), s \rangle$

2.Update : $x_{t+1} = x_t + \eta_t (s_t - x_t)$

Online Linear Oracle $\mathcal{O}$:

Sequence of linear loss function $\langle g_1, \cdot \rangle, \langle g_2, \cdot \rangle, \ldots$

$$s_t = \underset{s \in \mathcal{K}}{\operatorname{argmin}} \left\{ \zeta \sum_{l=1}^{t-1} \langle g_l, s \rangle + \langle u, s \rangle \right\}$$



Hazan, E. 2015, Introduction to online convex optimization. *Foundations and Trends in Optimization*.

Delay Mechanism

- $\mathbf{F}_t = \{s \leq t; s + d_s - 1 = t\}$ (or $\mathbf{F}_t = \emptyset$)
- $\mathbf{F}_t^i = \left\{s \leq t; s + d_s^i - 1 = t\right\}, \forall i \in [n]$

Regret :

$$R_T = \sum_{t=1}^{T} f_t(x_t) - \min_{x \in \mathcal{K}} \sum_{t=1}^{T} f_t(x)$$

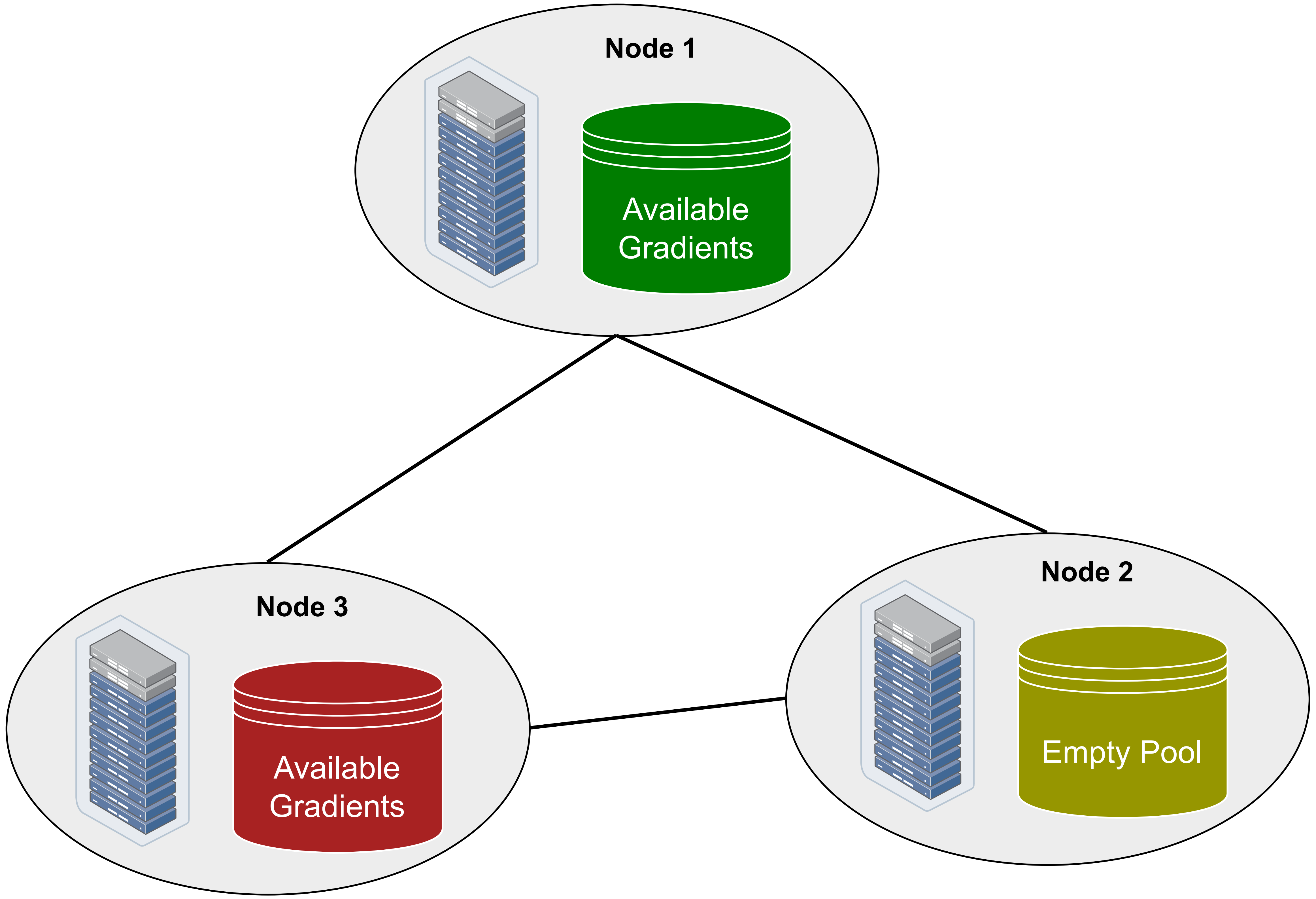


Figure : Given a time t, each agent holds a distinct pool of available gradient feedback that is ready for computation at the current time.

Centralized Algorithm

For some round t

Prediction

For K rounds, do

$$s_k \in \mathcal{O}_k$$

$$x_{k+1} = x_k + \eta_k (s_k - x_k)$$

Follow the Perturbed Leader

$$h_{t-1,k} = \zeta \sum_{l=1}^{t-1} \langle g_{l,k}, s \rangle + \langle n, s \rangle$$

$$s_k = \operatorname*{argmin}_{s \in \mathcal{K}} h_{t-1,k}$$

Play $x_t = x_{K+1}$ and receives $\mathbf{F}_t$

Update

For K rounds, do

$$g_k = \sum_{s \in \mathbf{F}_t} \nabla f_s(x_{s,k})$$

$$\text{Update } \mathcal{O}_k \text{ with } g_k$$

$$h_{t-1,k} + \zeta \langle g_k, . \rangle$$

Decentralized Algorithm

For some round t at agent i

Prediction

For K rounds, do

$$s_{i,k} \in \mathcal{O}_{i,k}$$

$$x_{i,k+1} = \boxed{y_{i,k}} + \eta_k(s_{i,k} - \boxed{y_{i,k}})$$

$$y_{i,k} = \sum_{j=1}^{n} w_{ij} x_{j,k}$$

Play $x_{i,t} = x_{i,K+1}$ and receives $\mathbf{F}_t^i$

Update

For K rounds, do

Surrogate gradient $g_{i,k+1}$ (1)

Local gradient average (2)

Update $\mathcal{O}_{i,k}$ with $d_{i,k}$

$$(1) \sum_{s \in \mathbf{F}_t^i} \left[\nabla f_{i,s}(x^i_{s,k+1}) - f_{i,s}(x^i_{s,k}) \right] + d_{i,k}$$

$$(2) \quad d_{i,k} = \sum_{j=1}^{n} w_{ij} g_{j,k}$$

Some comments :

- K Online Linear Oracles $\mathcal{O}_1, \ldots, \mathcal{O}_K$ throughout the learning process
- Oracles provide estimations of the upcoming gradients' direction from feedbacks on previous rounds
- Oracles receive delayed feedback from the algorithms
- Mixed delayed feedbacks from neighbouring agents in distributed setting

Impact of delayed feedback to the oracle's output

- s_t : oracle's output with delayed feedback
- $\hat{s}_t$: oracle's output without delayed feedback

$$\|s_t - \hat{s}_t\| = O\left(\zeta \sum_{s<t} \mathbf{I}_{\{s+d_s>t\}}\right)$$

$$\|s_t^i - \hat{s}_t^i\| = O\left(\zeta\sqrt{n}\left(\frac{\lambda}{\rho}+1\right)\frac{1}{n}\sum_{i=1}^{n}\sum_{s<t}\mathbf{I}_{\{s+d_s^i>t\}}\right)$$

Informal Theorem 1 :

- Given $\zeta = \frac{1}{G\sqrt{B}}, \eta_k = min\left(1, \frac{A}{k}\right), K = \sqrt{T}$

$$R_T = O\left(DG\sqrt{B} + R_{T,\mathcal{O}}\right)$$

Additional term related to delayed feedback

Regret of the oracle

$B = \sum_{t=1}^{T} d_t$, sum of all delay value over T rounds

$$F_t(x) = \frac{1}{n}\sum_{i=1}^{n} f_{i,t}(x) \qquad\qquad R_T = \sum_{t=1}^{T} F_t(x_t) - \sum_{t=1}^{T} F_t(x^*)$$

Informal Theorem 2 :

- Given $\zeta = \dfrac{1}{G\sqrt{B}}$, $\eta_k = min\left(1, \dfrac{A}{k}\right)$, $K = \sqrt{T}$

$$R_T = O\left(\sqrt{n}DG\left(\frac{\lambda}{\rho}+1\right)\sqrt{B} + R_{T,\mathcal{O}}\right)$$

$B = \sum_{t=1}^{T} \frac{1}{n} \sum_{i=1}^{n} d_{i,t}$, sum of average delay values over n agents

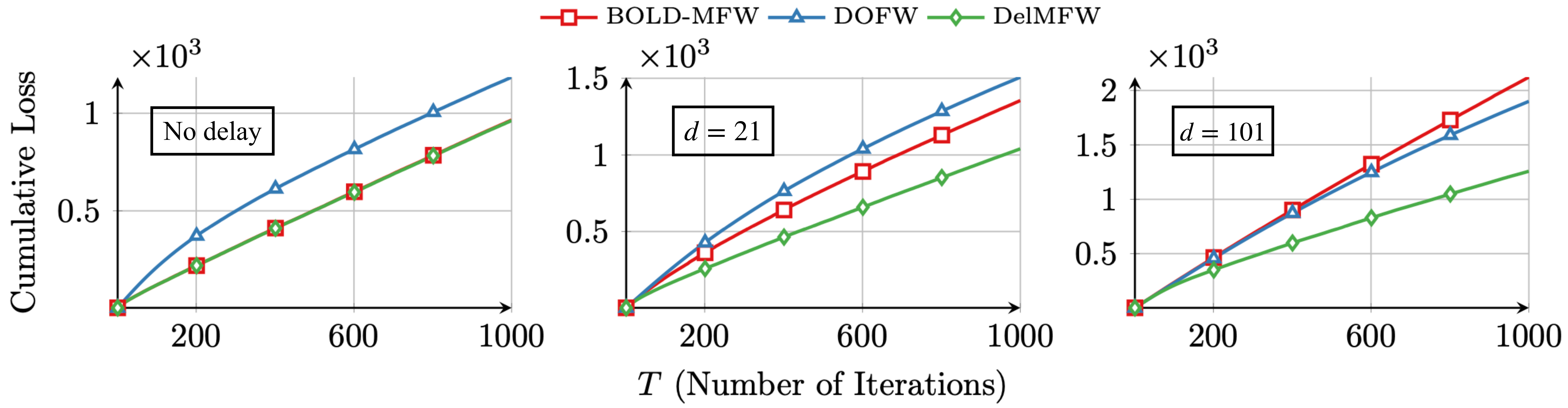
Related Works

- Joulani et al. 2013. Online learning under delayed feedback. Proceedings of the 30th International Conference on Machine Learning.
- Quanrud et al. 2015. Online learning with adversarial delays. Advances in Neural Information Processing Systems.
- Wan et al. 2022. Online frank-wolfe with arbitrary delays. Advances in Neural Information Processing Systems.

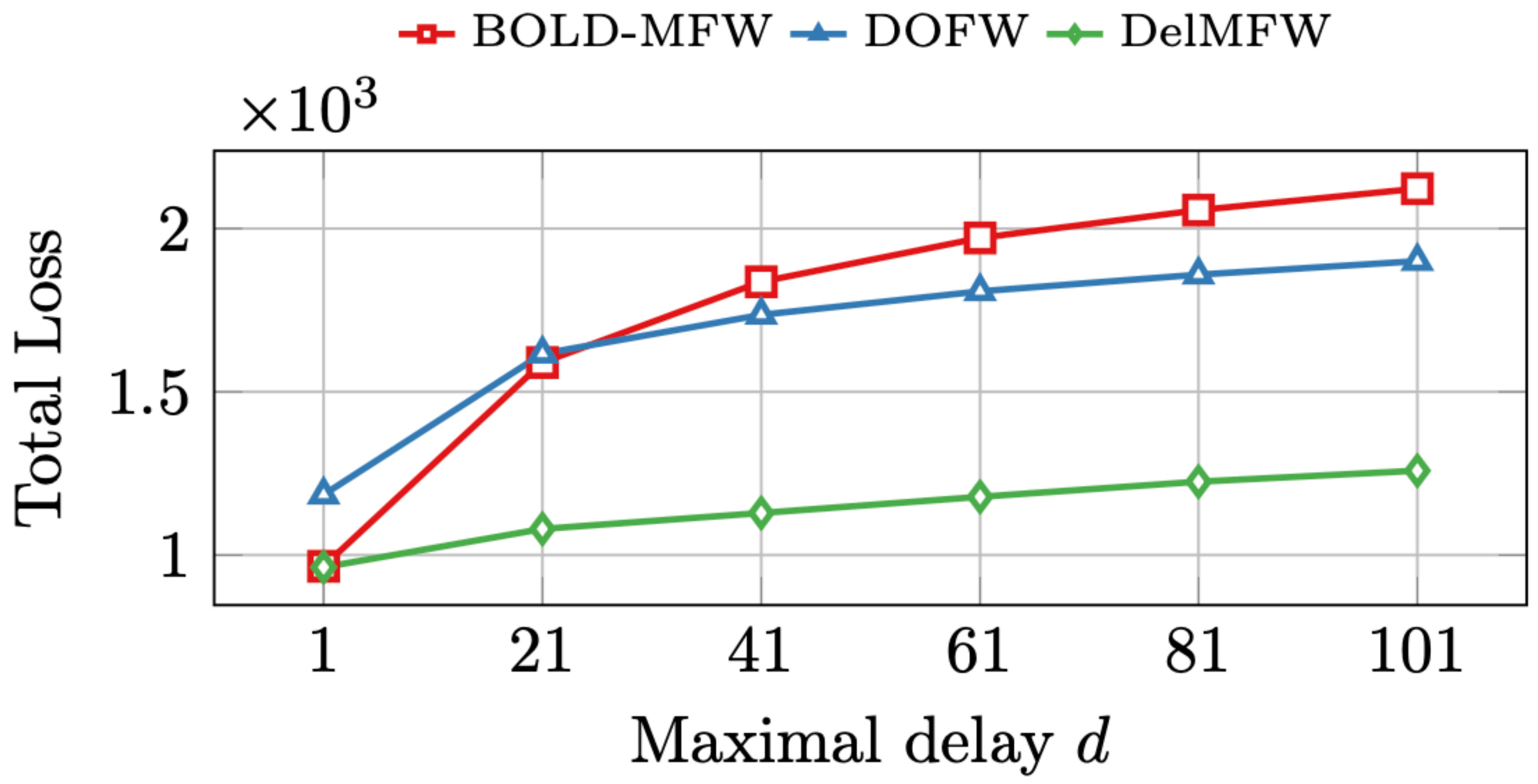
Table 1: Comparisons to previous algorithms DGD [Quanrud and Khashabi, 2015] and DOFW [Wan et al., 2022] on centralized online convex optimization with delays bounded by d. Our algorithms are in bold.

Algorithm	Centralized	Distributed	Adversarial Delay	Projection-free	Regret
DGD	Yes	-	Yes	-	$\mathcal{O}(\sqrt{dT})$
DOFW	Yes	-	Yes	Yes	$\mathcal{O}(T^{3/4} + dT^{1/4})$
DeLMFW	Yes	-	Yes	Yes	$\mathcal{O}(\sqrt{dT})$
De2MFW	-	Yes	Yes	Yes	$\mathcal{O}(\sqrt{dT})$

DeLMFW

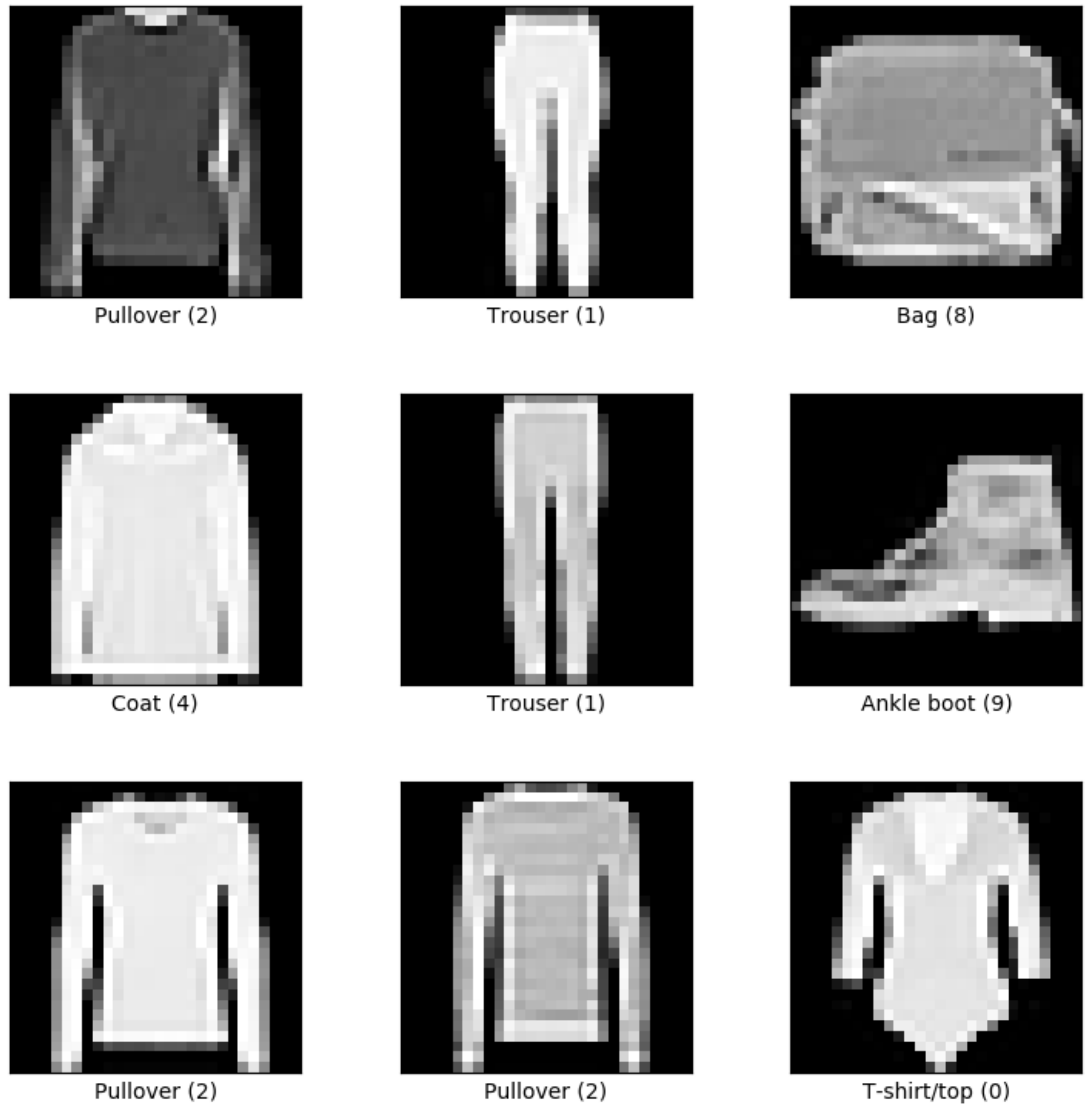


BOLD-MFW
DOFW
DelMFW
Cumulative Loss
$\times 10^3$
No delay
$d = 21$
$d = 101$
T (Number of Iterations)

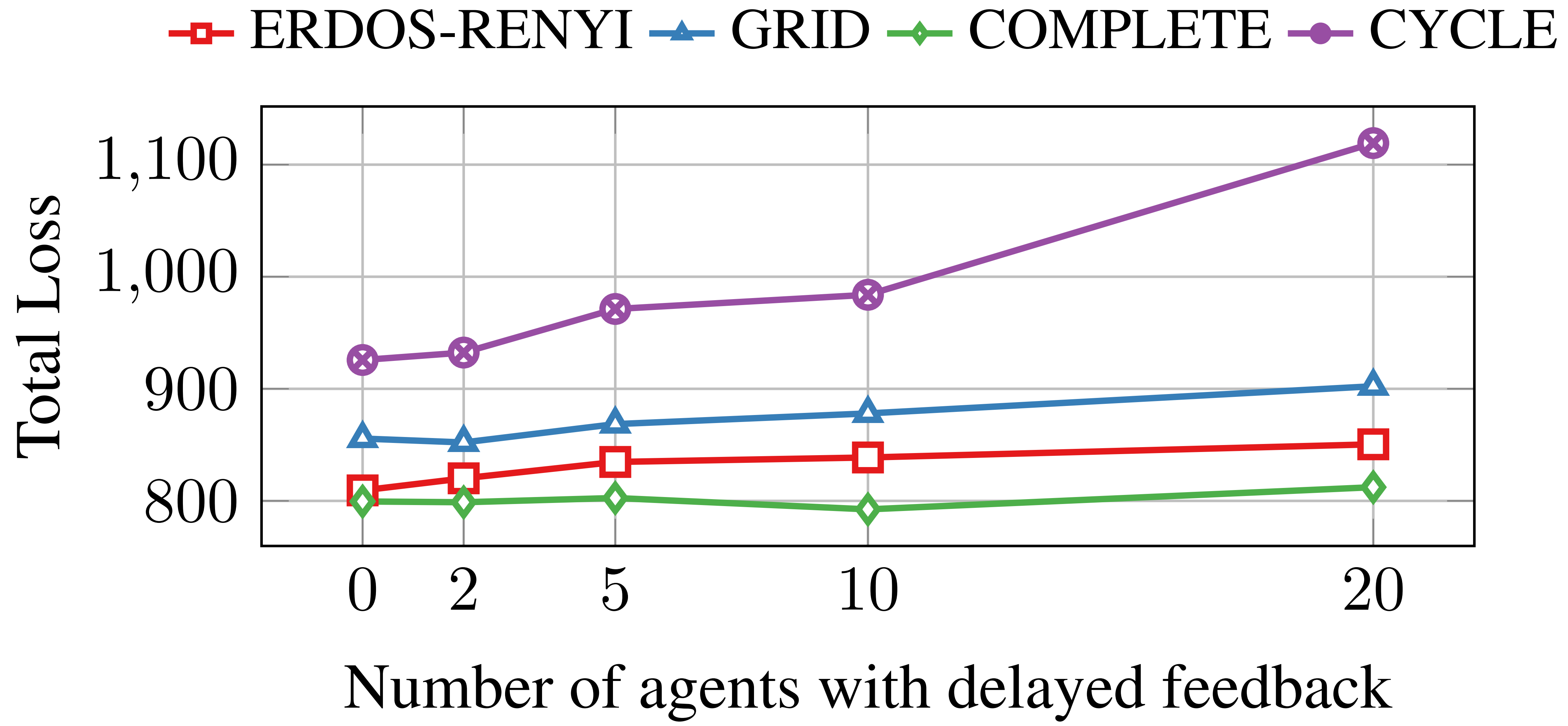


BOLD-MFW
DOFW
DelMFW
$\times 10^3$
Total Loss
Maximal delay d

De2MFW



$n = 30$



Topology / f	**Erdos Renyi**	**Grid**	**Complete**	**Cycle**
0	809.37	855.62	799.49	925.72
2	820.15 (+1.3%)	852.15 (-0.4%)	798.79 (-0.08%)	932.34 (+0.7%)
5	834.74 (+3.0%)	868.52 (+1.4%)	802.59 (+0.3%)	971.24 (+4.7%)
10	838.74 (+3.5%)	878.04 (+2.5%)	792.45 (-0.8%)	983.89 (+6.0%)
20	850.49 (+4.9%)	902.30 (+5.3%)	812.21 (+1.5%)	1119.24 (+18.9%)

Discussion

Positive Results :

- Distributed projection-free algorithm that handling delayed feedback
- Optimal Regret Bound in delay and non-delay setting

Limitation :

- Excessive gradient computation => high communication

Thank you

Follow the Perturbed Leader

Algorithm 17 FPL for linear losses

1: Input: $\eta > 0$, distribution $\mathcal{D}$ over $\mathbb{R}^n$, decision set $\mathcal{K} \subseteq \mathbb{R}^n$.
2: Sample $\mathbf{n}_0 \sim \mathcal{D}$. Let $\hat{\mathbf{x}}_1 \in \arg\min_{\mathbf{x}\in\mathcal{K}}\{-\mathbf{n}_0^\top \mathbf{x}\}$.
3: **for** $t = 1$ to T **do**
4: Predict $\hat{\mathbf{x}}_t$.
5: Observe the linear loss function, suffer loss $\mathbf{g}_t^\top \mathbf{x}_t$.
6: Update

$$\hat{\mathbf{x}}_t = \arg\min_{\mathbf{x}\in\mathcal{K}} \left\{ \eta \sum_{s=1}^{t-1} \mathbf{g}_s^\top \mathbf{x} + \mathbf{n}_0^\top \mathbf{x} \right\}$$

7: **end for**

Lemma 1 (Theorem 5.8 [Hazan, 2016]). *Given a sequence of linear loss function $f_1, \ldots, f_T$. Suppose that Assumptions 1 to 3 hold true. Let $\mathcal{D}$ be a the uniform distribution over hypercube $[0,1]^m$. The regret of FTPL is*

$$\mathcal{R}_{T,\mathcal{O}} \leq \zeta D G^2 T + \frac{1}{\zeta}\sqrt{m} D$$

where ζ is learning rate of algorithm.

Hazan, E. 2015, Introduction to online convex optimization. *Foundations and Trends in Optimization*.