Online Nonsubmodular Minimization with Delayed Costs: From Full Information to Bandit Feedback

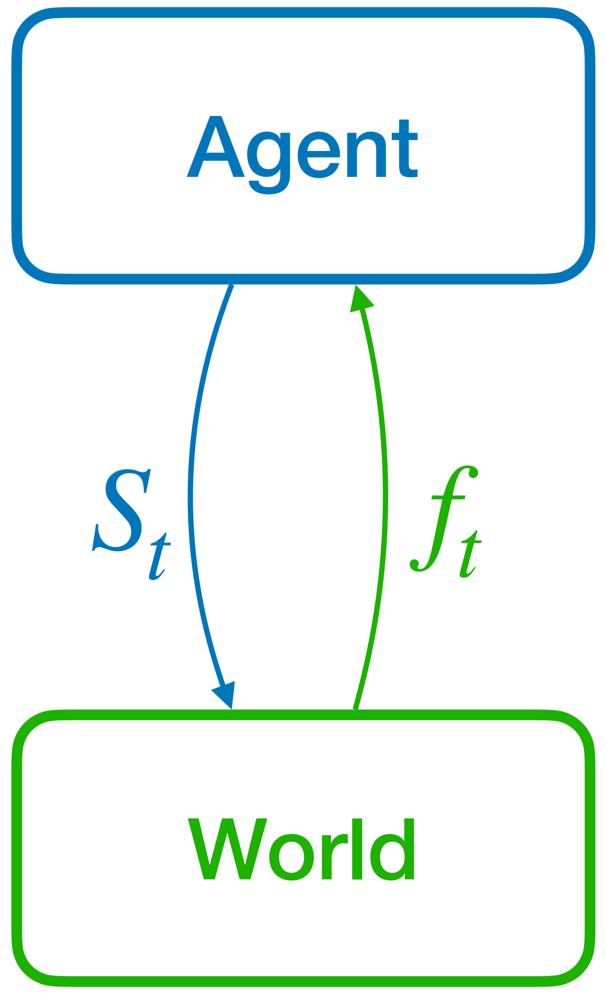
ICML-2022

Tianyi Lin[1,⋆], Aldo Pacchiano[2,⋆], Yaodong Yu[1,⋆], Michael I. Jordan[1]
(⋆ Equal contribution)
[1]UC Berkeley, [2]Microsoft Research

Preliminary - Online nonsubmodular learning

for round = 1, 2, …

- agent chooses a subset $S_t \subseteq [n]$
- agent suffers a cost $f_t(S_t)$ (f_t produced by the world)
- agent receives feedback (information about f_t)
- agent updates its model



f_t : a class of nonsubmodular functions with special structure

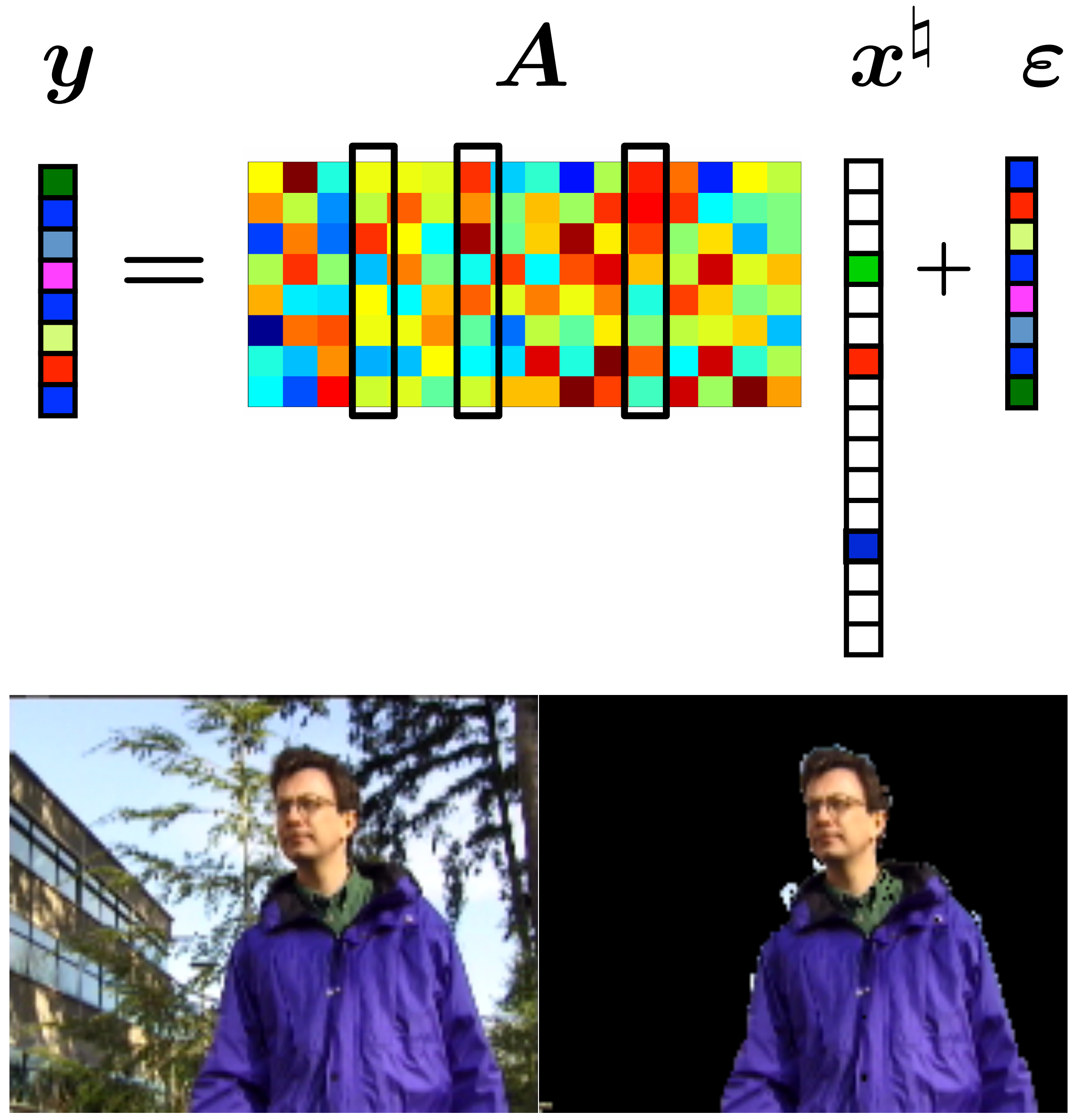
$$f_t(S) = \bar{f}_t(S) - \underline{f}_t(S), \quad \forall S \subseteq [n], t \in [T]$$

$\bar{f}_t(\cdot) : \alpha -$ weakly DR-submodular $\quad \underline{f}_t(\cdot) : \beta -$ weakly DR-supermodular

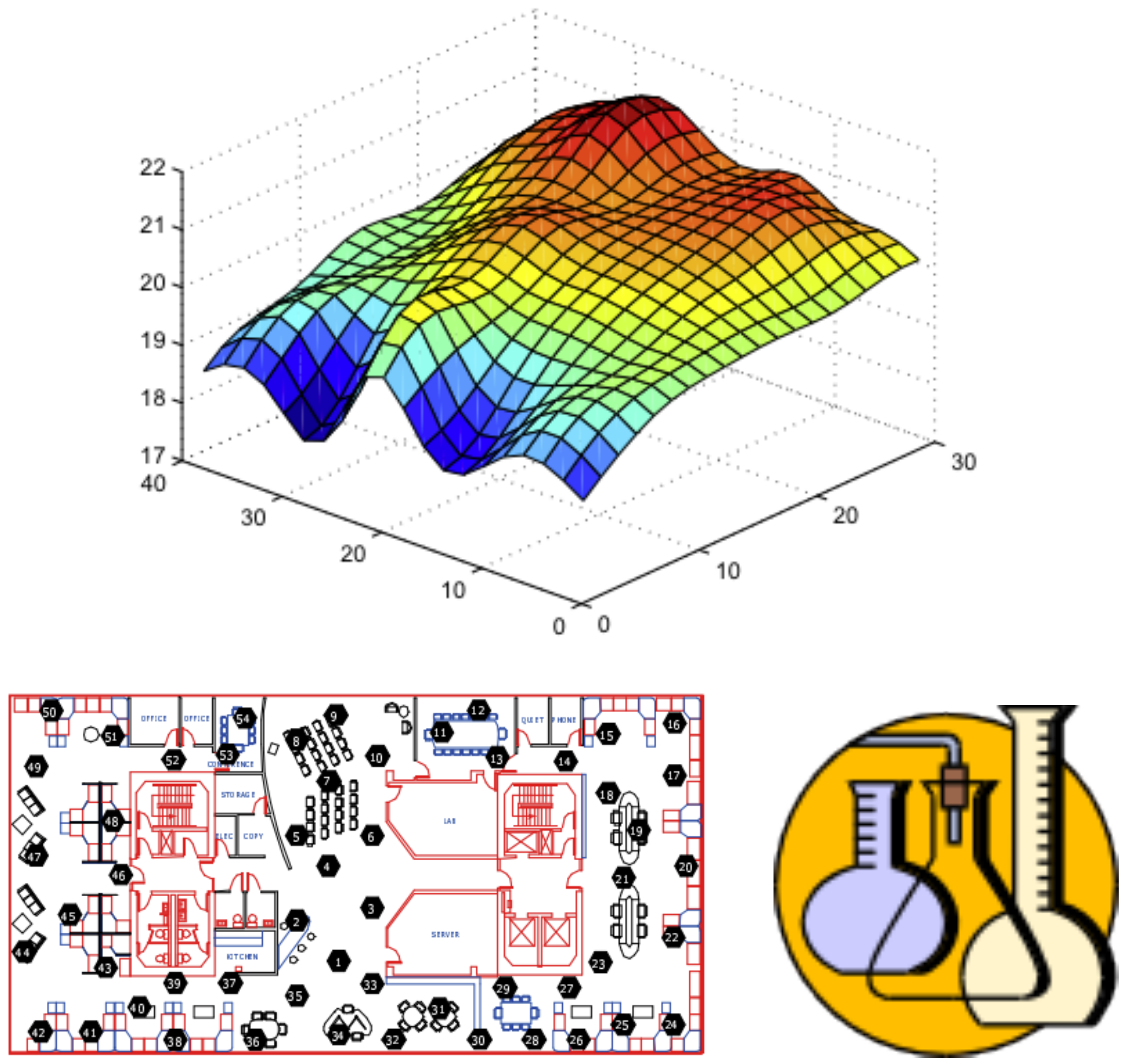
Examples: Structured Sparse Learning [El Halabi & Cevher, 2015], Batch Bayesian Optimization [El Halabi & Jegelka, 2020].

Preliminary - Online nonsubmodular learning

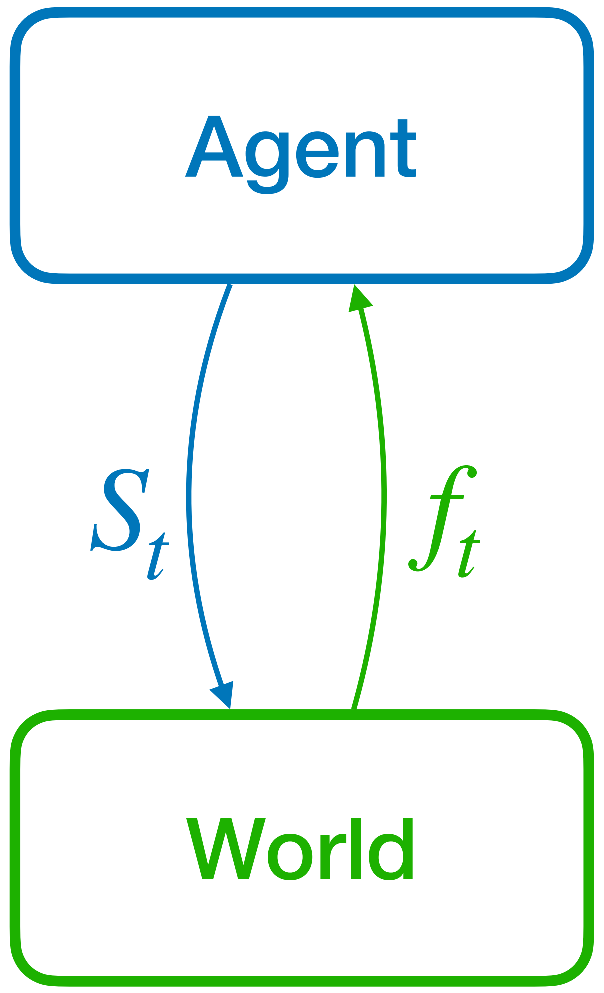
Examples: Structured Sparse Learning [El Halabi & Cevher, 2015], Batch Bayesian Optimization [El Halabi & Jegelka, 2020].



Structured sparse learning



Batch Bayesian optimization



Figures from [Mairal et al., 2010, Krause et al., 2008]

Preliminary - Regret

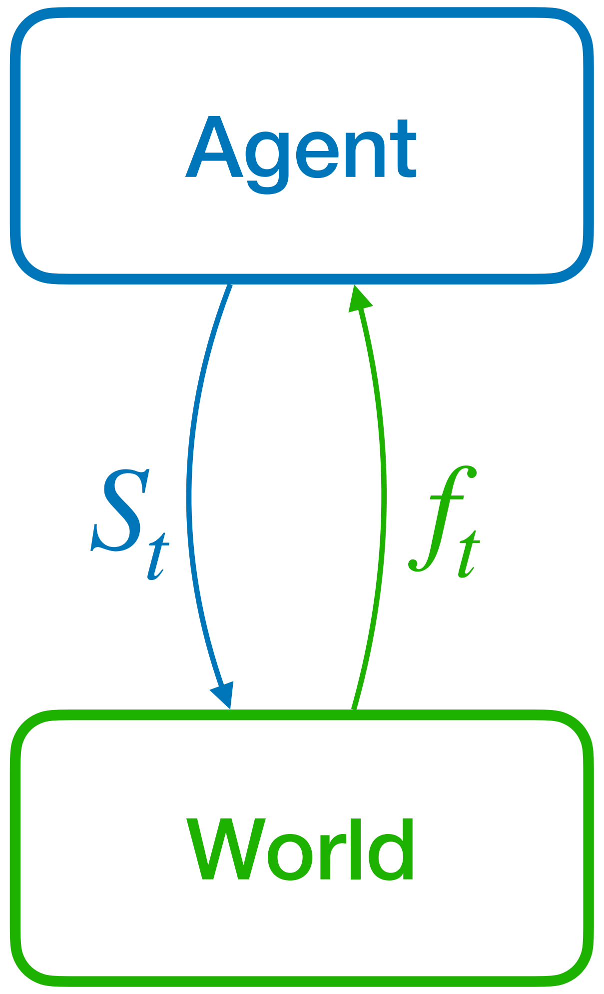
Regret:

$$\text{Regret}(T) = \sum_{t=1}^{T} f_t(S_t) - \min_{S \subseteq [n]} \sum_{t=1}^{T} f_t(S)$$

(α, β)-Regret (*this work*):

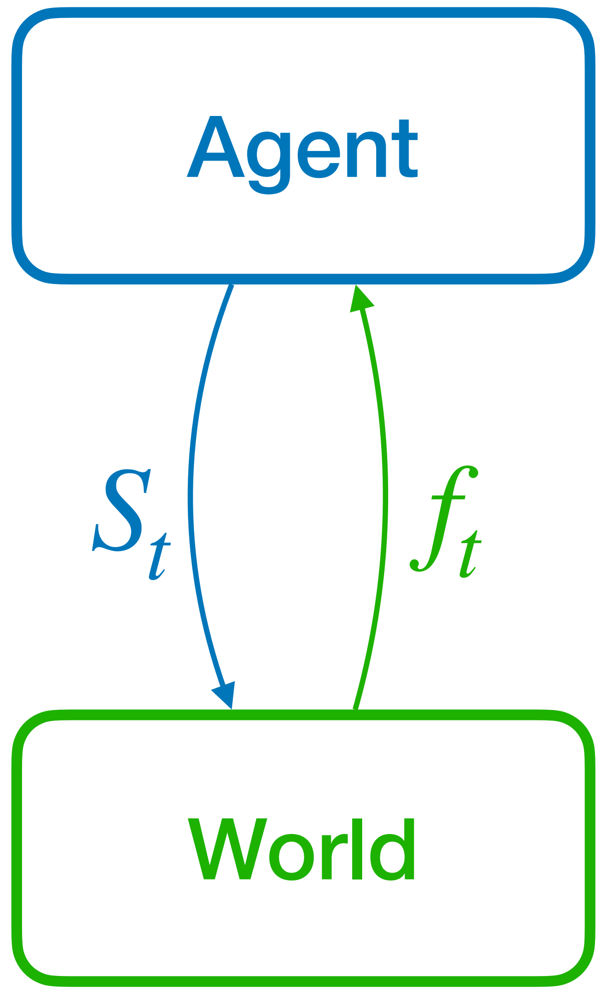
$$\text{Regret}_{\alpha,\beta}(T) = \sum_{t=1}^{T} f_t(S_t) - \sum_{t=1}^{T} \left(\frac{1}{\alpha} \cdot \bar{f}_t(S_T^\star) - \beta \cdot \underline{f_t}(S_T^\star) \right)$$

$$S_T^\star = \text{argmin}_{S \subseteq [n]} \sum_{t=1}^{T} f_t(S)$$



This work

- Information about f_t:
  - Agent observes the whole function f_t (*full information setting*)
  - Agent only observes the value of $f_t(S_t)$ (*bandit feedback setting*)
- Delay between decision and feedback:
  - Agent receives information about f_t at round t (*non-delay setting*)
  - Agent receives information about f_t at round $t+d$, where d is the delay (*delay setting*)



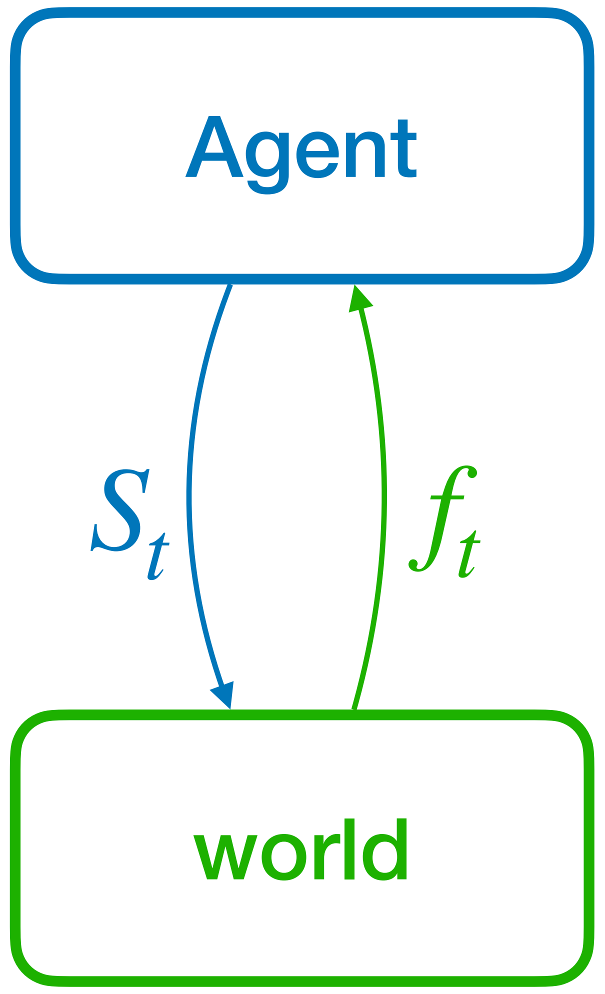
Question: Can we design online learning algorithms when the cost functions are nonsubmodular with delayed costs?

✔ *Yes, In all four settings!*

Online approximation algorithm (full information, without delay)

Algorithm 1 Online Approximate Gradient Descent

Initialization: the point $x^1 \in [0,1]^n$ and the stepsize $\eta > 0$;
for $t = 1, 2, \ldots$ **do**
  Let $x^t_{\pi(1)} \geq \ldots x^t_{\pi(n)}$ be the sorted entries in the decreasing order with $A^t_i = \{\pi(1), \ldots, \pi(i)\}$ for all $i \in [n]$ and $A^t_0 = \emptyset$. Let $x^t_{\pi(0)} = 1$ and $x^t_{\pi(n+1)} = 0$.
  Let $\lambda^t_i = x^t_{\pi(i)} - x^t_{\pi(i+1)}$ for all $0 \leq i \leq n$.
  Sample S^t from the distribution $\mathbb{P}(S^t = A^t_i) = \lambda^t_i$ for all $0 \leq i \leq n$ and observe the new loss function f_t.
  Compute $g^t_{\pi(i)} = f_t(A^t_i) - f_t(A^t_{i-1})$ for all $i \in [n]$.
  Compute $x^{t+1} = P_{[0,1]^n}(x^t - \eta g^t)$.



for round = 1, 2, …

- agent choose a subset $S_t \subseteq [n]$
- agent suffer a cost $f_t(S_t)$ (f_t produced by the world)
- agent receives feedback (information about f_t)
- agent updates its model

Online approximation algorithm (without delay)

- *full information setting* :

  - $\text{Regret}_{\alpha,\beta}(T) = O(\sqrt{nT} + \sqrt{T\log(1/\delta)})$ **with probability** $1-\delta$.
  - $\mathbb{E}[\text{Regret}_{\alpha,\beta}(T)] = O(\sqrt{nT})$

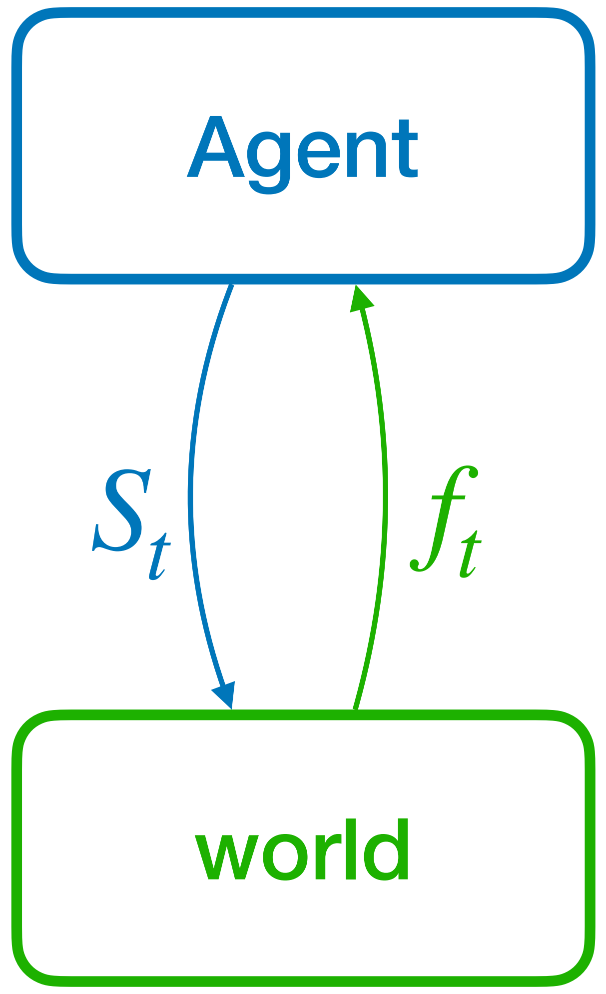
- *bandit feedback setting* :

  - $\text{Regret}_{\alpha,\beta}(T) = O(nT^{2/3} + \sqrt{n\log(1/\delta)}T^{2/3})$ **with probability** $1-\delta$.
  - $\mathbb{E}[\text{Regret}_{\alpha,\beta}(T)] = O(nT^{2/3})$

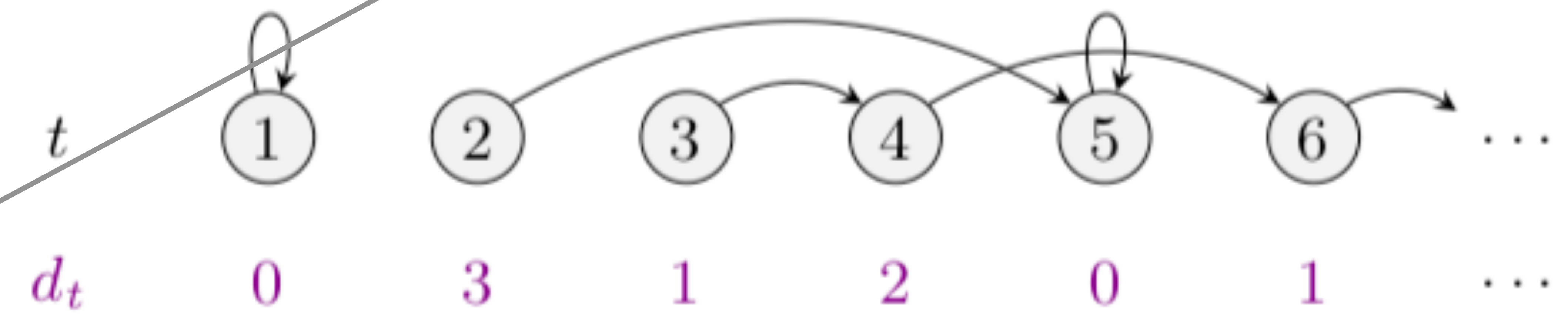
Delay online approximation algorithm (full information, with delay)

Algorithm 3 Delay Online Approximate Gradient Descent

1: **Initialization:** the point $x^1 \in [0,1]^n$ and the stepsize $\eta_t > 0$; $\mathcal{P}_0 \leftarrow \emptyset$ and $f_\infty = 0$.
2: **for** $t = 1, 2, \ldots$ **do**
3: Let $x^t_{\pi(1)} \geq \ldots x^t_{\pi(n)}$ be the sorted entries in the decreasing order with $A^t_i = \{\pi(1), \ldots, \pi(i)\}$ for all $i \in [n]$ and $A^t_0 = \emptyset$. Let $x^t_{\pi(0)} = 1$ and $x^t_{\pi(n+1)} = 0$.
4: Let $\lambda^t_i = x^t_{\pi(i)} - x^t_{\pi(i+1)}$ for all $0 \leq i \leq n$.
5: Sample S^t from the distribution $\mathbb{P}(S^t = A^t_i) = \lambda^t_i$ for $0 \leq i \leq n$ and observe the new loss function f_t.
6: Compute $g^t_{\pi(i)} = f_t(A^t_i) - f_t(A^t_{i-1})$ for all $i \in [n]$ and then trigger a delay $d_t \geq 0$.
7: Let $\mathcal{R}_t = \{s : s + d_s = t\}$ and $\mathcal{P}_t \leftarrow \mathcal{P}_{t-1} \cup \mathcal{R}_t$. Take $q_t = \min \mathcal{P}_t$ and set $\mathcal{P}_t \leftarrow \mathcal{P}_t \setminus \{q_t\}$.
8: Compute x^{t+1} using Eq. (11).



Delay $d = o(t^\gamma)$, $\gamma < 1$. (delay can be *unbounded*)



Online approximation algorithm (with delay)

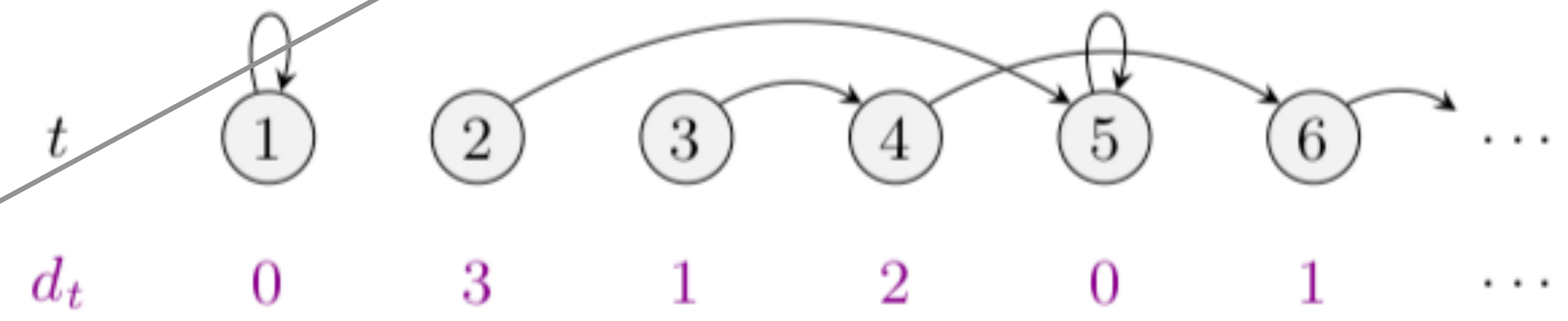
- *full information setting* :

  - $\text{Regret}_{\alpha,\beta}(T) = O(\sqrt{nT^{1+\gamma}} + \sqrt{T\log(1/\delta)})$ **with probability** $1-\delta$.
  - $\mathbb{E}[\text{Regret}_{\alpha,\beta}(T)] = O(\sqrt{nT^{1+\gamma}})$

- *bandit feedback setting* :

  - $\text{Regret}_{\alpha,\beta}(T) = O(nT^{(2+\gamma)/3} + \sqrt{n\log(1/\delta)}T^{(4-\gamma)/6})$ **with probability** $1-\delta$.
  - $\mathbb{E}[\text{Regret}_{\alpha,\beta}(T)] = O(nT^{(2+\gamma)/3})$

Delay $d = o(t^{\gamma})$, $\gamma < 1$. (delay can be *unbounded*)



Thank you!

For more information, please refer to our paper (link: https://arxiv.org/abs/2205.07217) and come to our poster!