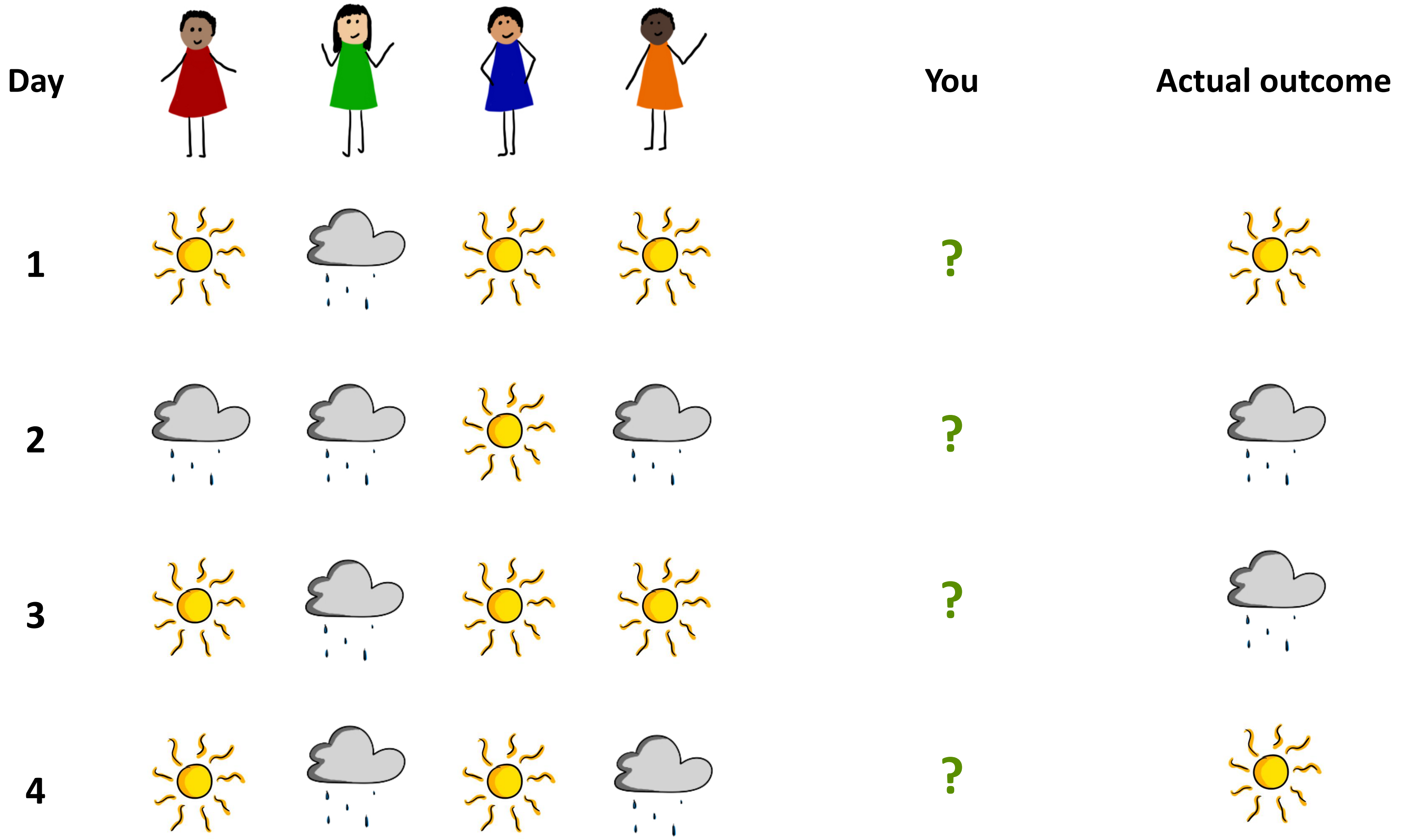
Memory Bounds for the Expert Problem

Vaidehi Srinivas
Ziyu (Neil) Xu
David P. Woodruff
Samson Zhou

Prediction with Expert Advice

a fundamental problem of **sequential prediction**



Quantifying Performance

In general, predicting the future is impossible.

We judge our algorithm based on **regret**.

Definition (regret)

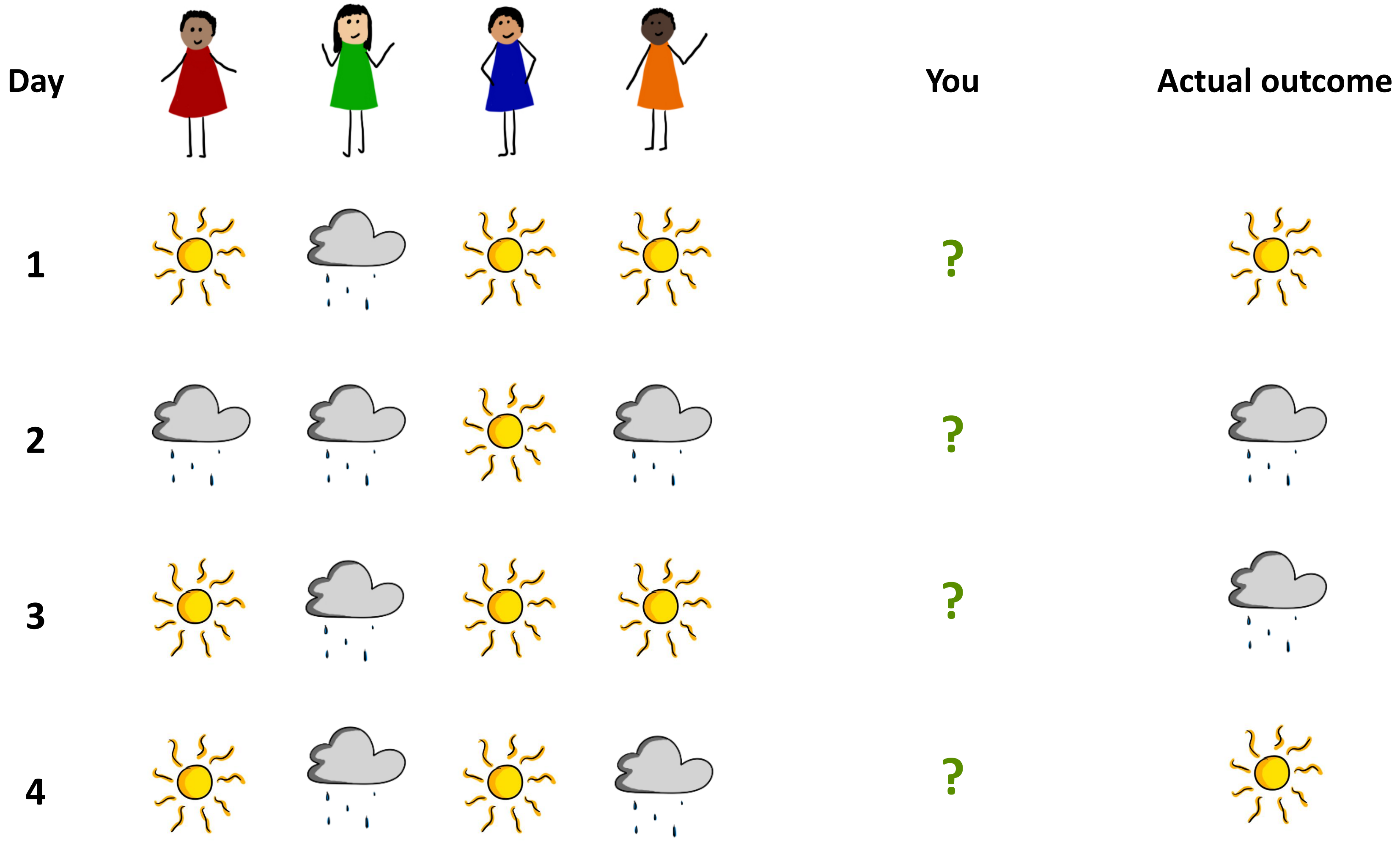
# of mistakes algorithm makes more than the best expert

Definition (average regret)

$$\frac{\text{regret}}{\text{\# of days}}$$

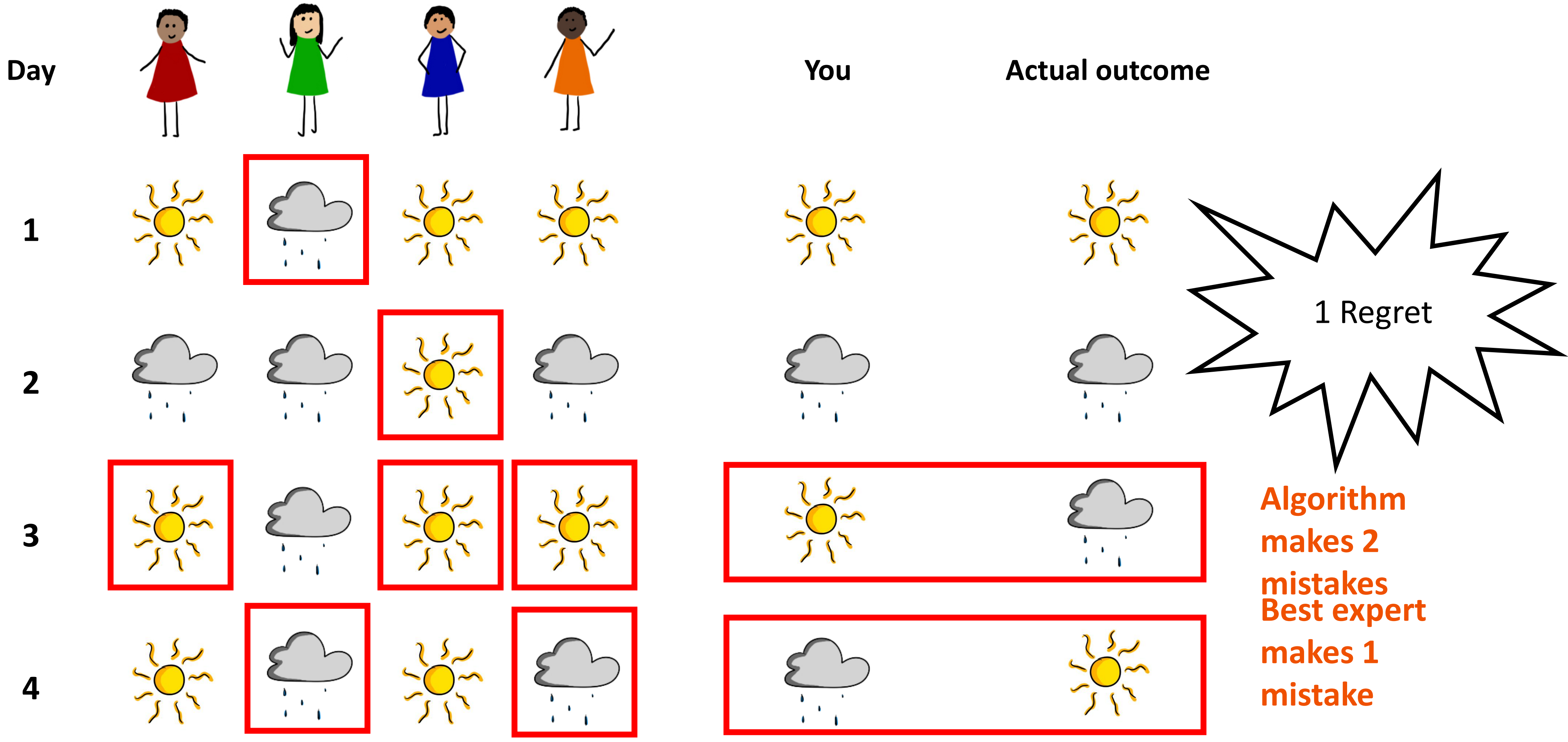
Prediction with Expert Advice

a fundamental problem of **sequential prediction**



Prediction with Expert Advice

a fundamental problem of **sequential prediction**



The Online Learning with Experts Problem

- n experts who decide either $\{0,1\}$ on each of T days ($n \gg T$)
- Algorithm takes advice from experts and predict either $\{0,1\}$ on each day
- Algorithm sees the outcome, which is either $\{0,1\}$, of each day and can use this information on future days
- The cost of the algorithm is the number of incorrect predictions
- (Average) Regret is the amortized additional cost of the algorithm compared to the cost M of the best expert

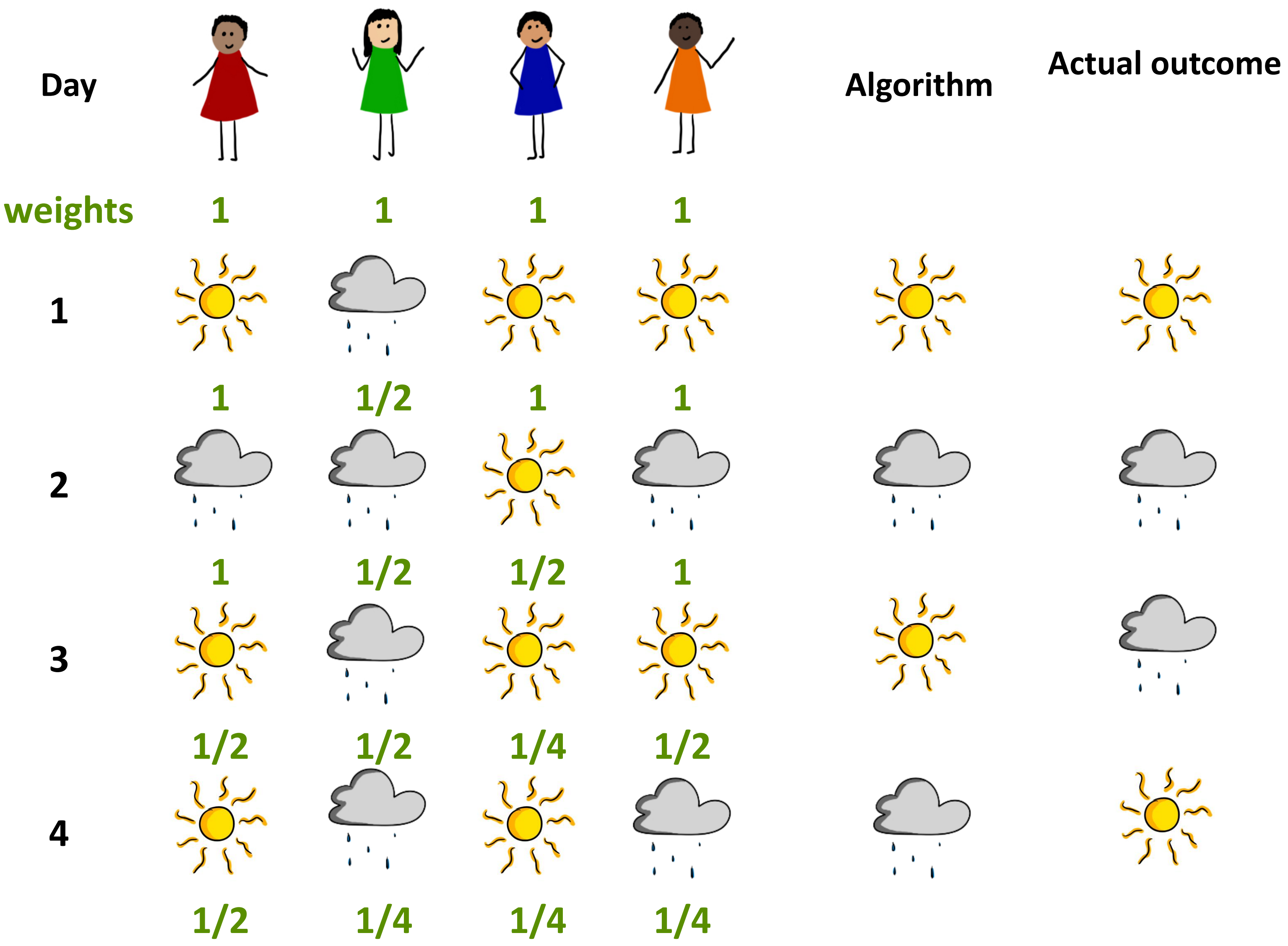
Applications of the Experts Problem

- Ensemble learning, e.g., AdaBoost

- Forecast and portfolio optimization

- Special case of online convex optimization

Weighted Majority (Littlestone, Warmuth 89)



Guarantee for Weighted Majority

Theorem (Deterministic Weighted Majority)

# of mistakes by deterministic weighted majority $\leq 2.41\,(M + \log_2 n)$

where M is the # of mistakes the best expert makes, n is # of experts.

- $\left(\frac{1}{2}\right)^M \leq$ sum of the weights $\leq \left(\frac{3}{4}\right)^m n$
- $m \leq \frac{M+\log_2 n}{\log_2 \frac{4}{3}}$

Guarantee for Weighted Majority

Theorem (Deterministic Weighted Majority)

$$\text{\# of mistakes by deterministic weighted majority} \leq 2.41\,(M + \log_2 n)$$

where M is the # of mistakes the best expert makes, n is # of experts.

Theorem (Randomized Weighted Majority, i.e, Multiplicative Weights)

For $\varepsilon > 0$, can construct algorithm A such that

$$E[\text{\# of mistakes by } A] \leq (1 + \varepsilon)\, M + \frac{\ln n}{\varepsilon}$$

Previous Work

- Weighted majority algorithm (Littlestone and Warmuth 89) down-weights each expert that is incorrect on each day and selects the weighted majority as the output for each day
- Weighted majority algorithm gets $O(M + \log n)$ total mistakes

- Randomized weighted majority algorithm (Littlestone and Warmuth 89) randomly follows each expert with probability proportional to the weight of the expert
- Randomized weighted majority algorithm achieves regret $O\left(\sqrt{\frac{\log n}{T}}\right)$

Memory Bounds for the Expert Problem

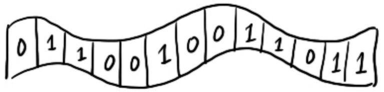
- These algorithms require $\Omega(n)$ memory to maintain weights for each expert – but what if n is very large and we want sublinear space?
- Can use no memory and just randomly guess each day – still good if the best expert makes a lot of mistakes but bad if the best expert makes very few mistakes

- What are the space/accuracy tradeoffs for the online learning with experts problem?

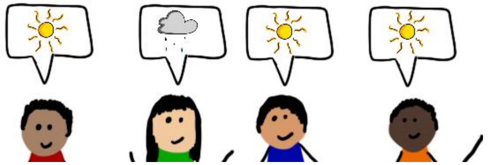
The Streaming Model

(the Jason Bourne model of computation)

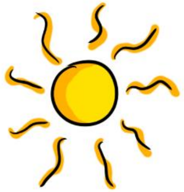
wake up with
no memory



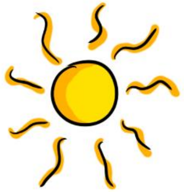
except a note
from your past
self
t ***s* bits**)



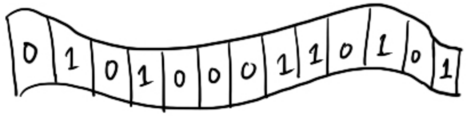
see expert predictions for today



make a prediction



see outcome



write a note to your future self
(at most ***s* bits**)

fall asleep and forget everything

repeat

The Streaming Model

The complete sequence of T days is the **data stream**.

$$(\text{prediction}_1, \text{outcome}_1), \ldots, (\text{prediction}_T, \text{outcome}_T)$$

Definition (Arbitrary Order Model)

An adversary chose the predictions and outcomes to trick you.

days

expert predictions

Definition (Random Order Model)

An adversary chose the predictions and outcomes to trick you,

then the order was randomly shuffled.

A Natural Idea

- What if we just identify the best expert?
- Find the best expert so far, follow it until a new best expert emerges, identify the new best expert, find it, repeat

- Must use $\Omega(n)$ space

Set Disjointness Communication Problem

- Set disjointness communication problem: Alice has a set $X \in \{0,1\}^n$ and Bob has a set $Y \in \{0,1\}^n$ and the promise is that either $|X \cap Y| = 0$ or $|X \cap Y| = 1$

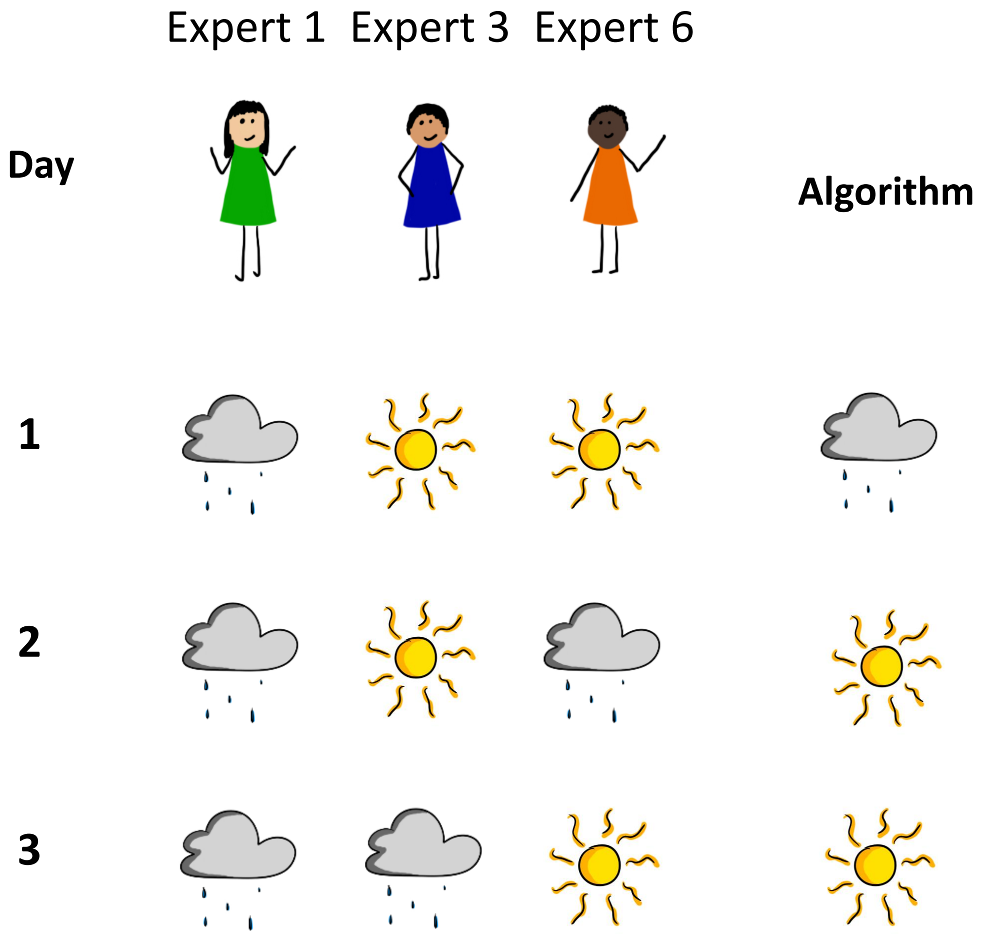
- Set disjointness requires total communication $\Omega(n)$

Reduction

- Suppose there exists an algorithm A that identifies the best expert. Alice creates a stream S so that each element of is X an expert that is correct on its own day

X

$$X = \{1,0,1,0,0,1,0,0,0,1\}$$
$$X = \{1,3,6,10\}$$



Reduction

- Alice runs the algorithm A on the stream S created by their set X and passes the state of A to Bob, who continues running the algorithm on the stream S' created by their set Y
- At the end, A will output an expert $i \in [n]$, and then Alice and Bob will check whether $X \cap Y = i$
- Solves set disjointness* so A must use $\Omega(n)$ space
- Not end of story: low-regret algorithm need not find best expert

Our Results (I)

- Any algorithm that achieves $\delta < \frac{1}{2+\sqrt{32 \ln 8}}$ (average) regret with probability at least $\frac{3}{4}$ must use $\Omega\left(\frac{n}{\delta^2 T}\right)$ space

- Lower bound holds for arbitrary-order, random-order, and i.i.d. streams

Our Results (II)

- There exists an algorithm that uses $O\left(\frac{n}{\delta^2 T}\log^2 n \log\frac{1}{\delta}\right)$ space achieves expected regret $\delta > \sqrt{\frac{8\log n}{T}}$ in the random-order model

- The algorithm is almost-tight with the space lower bounds and oblivious to M, the number of mistakes made by the best-expert
- Can achieve regret almost matching randomized weighted majority
- Result extends to general costs in $[0, \rho]$ with expected regret $\rho\delta$

Our Results (III)

- For $M < \frac{\delta^2 T}{1280 \log^2 n}$ and $\delta > \sqrt{\frac{128 \log^2 n}{T}}$, there exists an algorithm that uses $\tilde{O}\left(\frac{n}{\delta T}\right)$ space and achieves regret δ with probability $\frac{4}{5}$

- The algorithm *beats* the lower bounds, showing that the hardness comes from the best expert making a “lot” of mistakes
- Can achieve regret almost matching randomized weighted majority
- The algorithm oblivious to M, the number of mistakes made by the best-expert

Format

- Part 1: Background
- Part 2: Lower Bound
- Part 3: Arbitrary Model
- Part 4: Random-Order Model

Questions?

Lower Bound

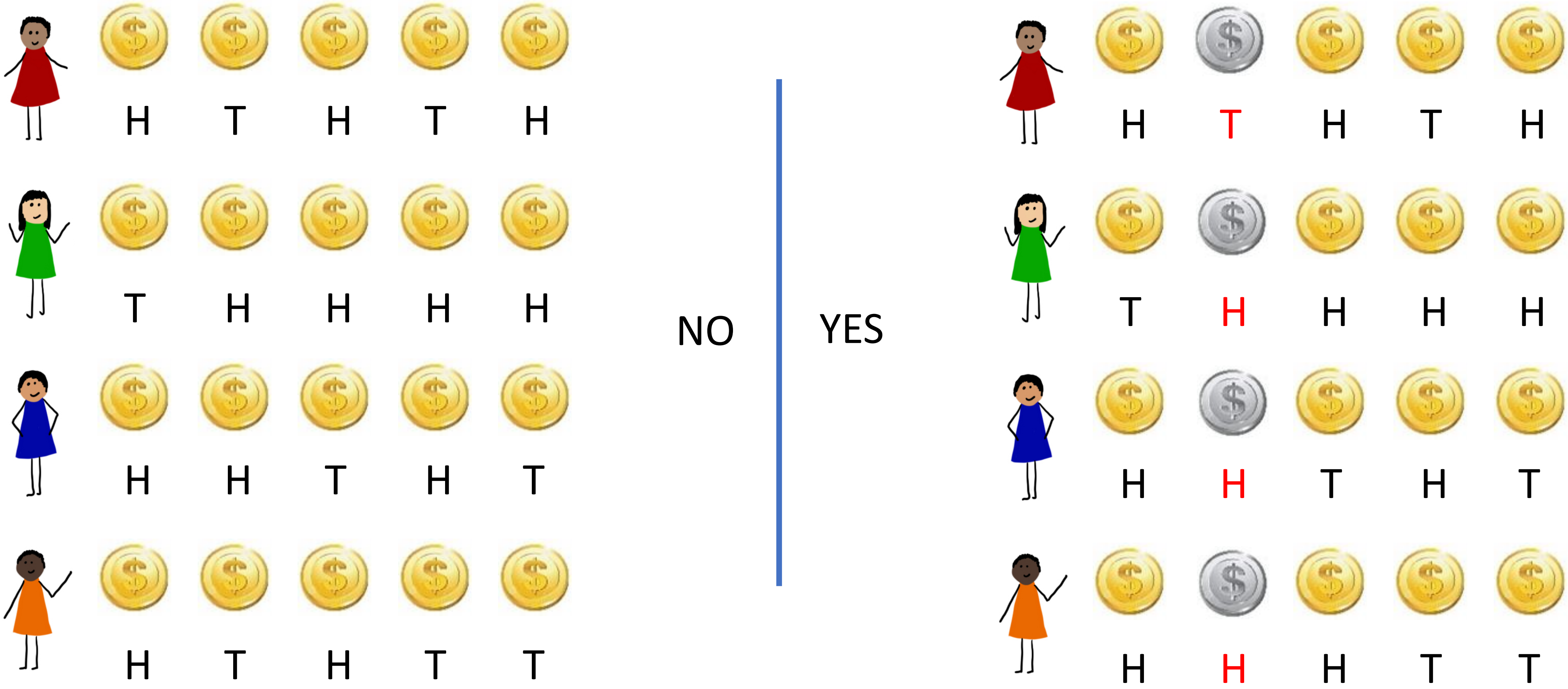
- Any algorithm that achieves $\delta < \frac{1}{2}$ (average) regret with probability at least $\frac{3}{4}$ must use $\Omega\left(\frac{n}{\delta^2 T}\right)$ space

- Lower bound holds for arbitrary-order, random-order, and i.i.d. streams

Communication Problem for Lower Bound

- Distributed detection problem
- ε-DIFFDIST problem: T players each hold n bits and must distinguish between two cases.
- Case 1: (NO) Every index for every player is drawn i.i.d. from a fair coin, i.e., a Bernoulli distribution with parameter $\frac{1}{2}$
- Case 2: (YES) An index $L \in [n]$ is selected arbitrarily. The L-th bit of each player is chosen i.i.d. from a Bernoulli distribution with parameter $\frac{1}{2} + \varepsilon$ and all the other bits are chosen i.i.d. from a fair coin

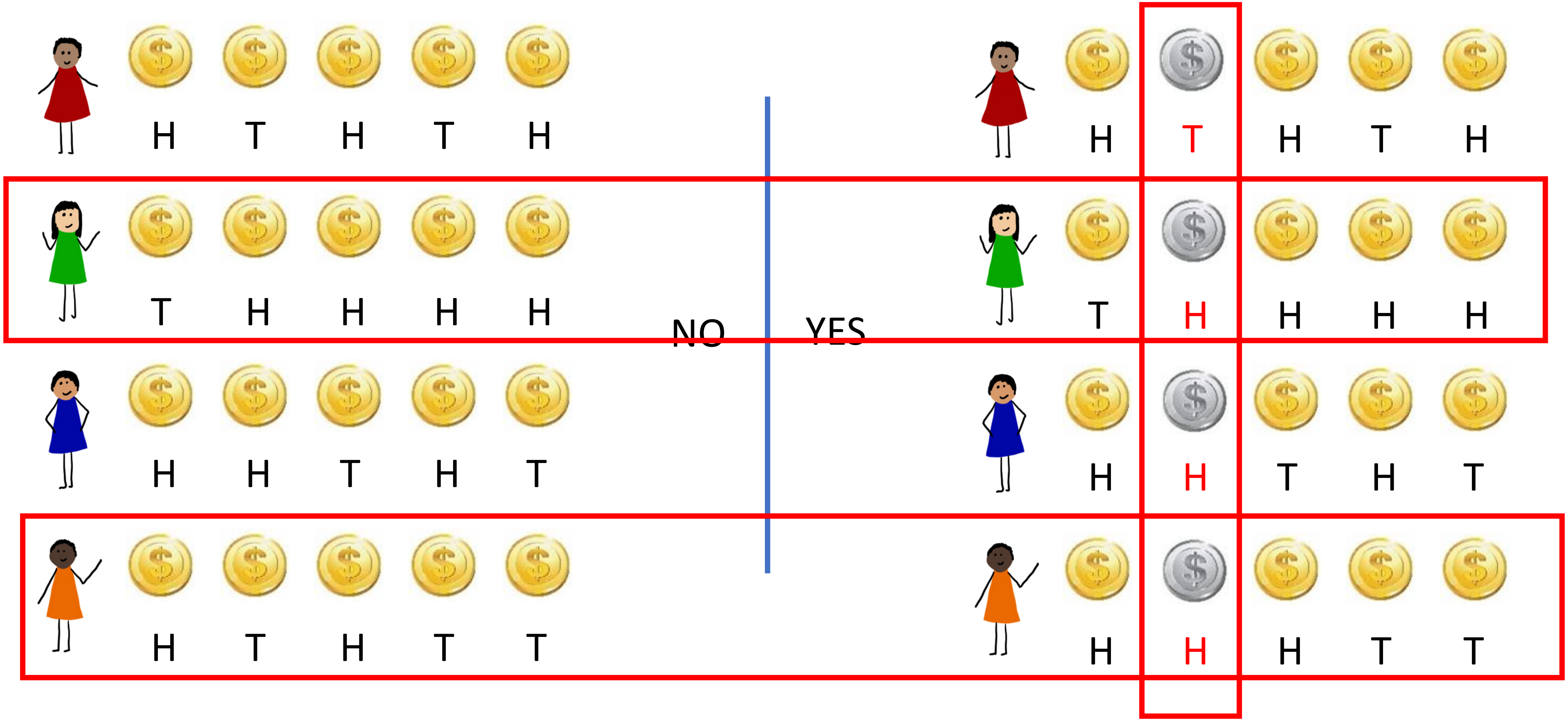
Communication Problem for Lower Bound



ε-DIFFDIST Problem

- ε-DIFFDIST problem: T players each hold n bits and must distinguish between two cases.
- Protocol: Randomly choose $\tilde{O}\left(\frac{1}{\varepsilon^2}\right)$ players and send all bits of those players, see whether some bit has bias at least $\frac{\varepsilon}{2}$

Communication Problem for Lower Bound



ε-DIFFDIST Problem

- ε-DIFFDIST problem: T players each hold n bits and must distinguish between two cases.
- Protocol: Randomly choose $\tilde{O}\left(\frac{1}{\varepsilon^2}\right)$ players and send all bits of those players, see whether some bit has bias at least $\frac{\varepsilon}{2}$
- Communication of protocol: $\tilde{O}\left(\frac{n}{\varepsilon^2}\right)$
- Theorem: $\Omega\left(\frac{n}{\varepsilon^2}\right)$ communication is necessary

ε-DIFFDIST Problem

- Theorem: $\Omega\left(\frac{n}{\varepsilon^2}\right)$ communication is necessary

- Fact: $\Omega\left(\frac{1}{\varepsilon^2}\right)$ samples are necessary to distinguish between a fair coin, i.e., a Bernoulli distribution with parameter $\frac{1}{2}$ and a coin with bias ε

- Intuition: players must solve the single coin problem on each of the n coins

ε-DIFFDIST Problem

- Intuition: players must solve the single coin problem on each of the n coins
- Formally, all the coins are independent in the NO distribution
- Independence implies entropy is additive so the mutual information must n times the mutual information in single coin problem
- Fact: $\Omega\left(\frac{1}{\varepsilon^2}\right)$ mutual information is necessary to distinguish between a single fair coin, i.e., a Bernoulli distribution with parameter $\frac{1}{2}$ and a coin with bias ε

ε-DIFFDIST Summary

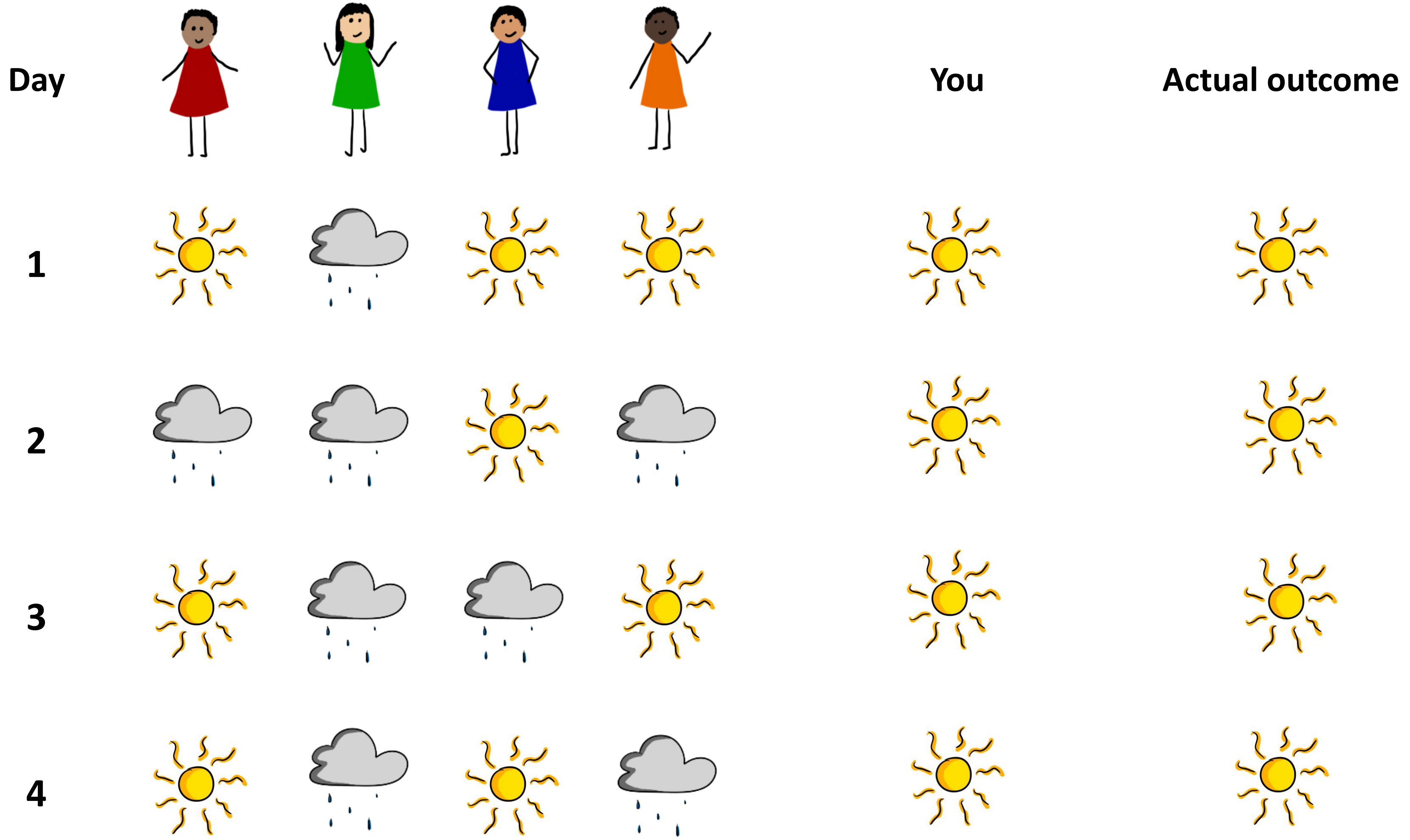
- ε-DIFFDIST problem: T players each hold n bits and must distinguish between two cases.
- Case 1: (NO) Every index for every player is drawn i.i.d. from a fair coin, i.e., a Bernoulli distribution with parameter $\frac{1}{2}$
- Case 2: (YES) An index $L \in [n]$ is selected arbitrarily. The L-th bit of each player is chosen i.i.d. from a Bernoulli distribution with parameter $\frac{1}{2} + \varepsilon$ and all the other bits are chosen i.i.d. from a fair coin
- Fact: $\Omega\left(\frac{n}{\varepsilon^2}\right)$ communication is necessary to solve the problem

Reduction Intuition

- Each player in the ε-DIFFDIST Problem corresponds to a different day
- Each bit in the ε-DIFFDIST Problem corresponds to a different expert

- Reduction: distinguishing whether there exists a slightly biased random bit corresponds to distinguishing whether there exists a slightly "better" expert
- We would like to use an online learning with experts algorithm for solving ε-DIFFDIST Problem for $\varepsilon = O(\delta)$ by sampling $\Omega\left(\frac{1}{\delta^2}\right)$ players

Reduction Challenge

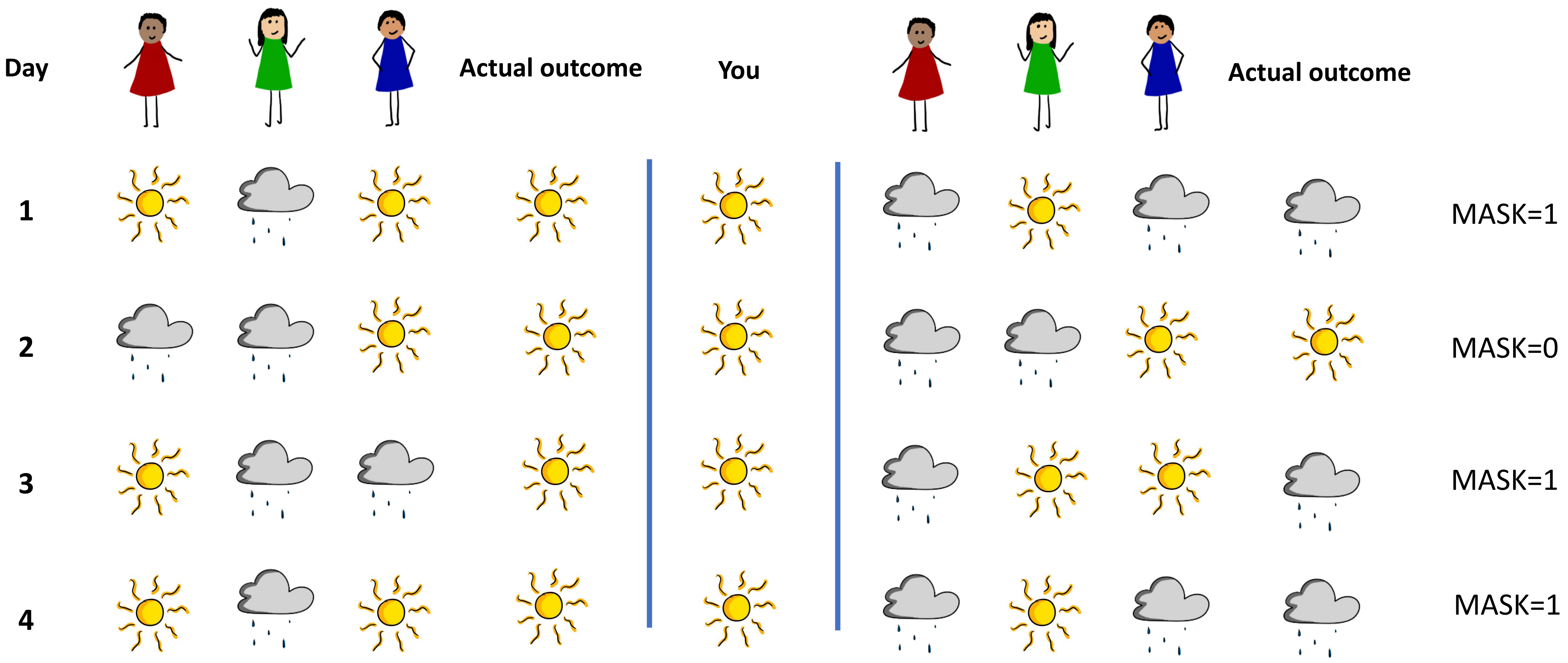


Reduction

- We would like to use an online learning with experts algorithm for solving ε-DIFFDIST Problem for $\varepsilon = O(\delta)$ by sampling $\Omega\left(\frac{1}{\delta^2}\right)$ players
- However, an algorithm with bad guarantees can still “luckily” have good cost

- Use masking argument – outcome of each day is masked by an independent fair coin flip on each day (expert advice also flipped)

Reduction Challenge



Reduction

- For constant $\delta < \frac{1}{2}$, if there is no biased coin, no expert will do better than $\frac{1}{2} + \frac{\delta}{2}$ with probability at least $\frac{1}{4}$
- For constant $\delta < \frac{1}{2}$, if there is a biased coin, an expert will do better than $\frac{1}{2} + \frac{\delta}{2}$ with probability at least $\frac{1}{4}$

Reduction Summary

- The online learning with experts algorithm with regret δ will be able to solve the ε-DIFFDIST Problem with probability at least $\frac{3}{4}$ for $\varepsilon = O(\delta)$, using $\Omega\left(\frac{n}{\delta^2}\right)$ total communication

- Any algorithm that achieves $\delta < \frac{1}{2}$ regret with probability at least $\frac{3}{4}$ must use $\Omega\left(\frac{n}{\delta^2 T}\right)$ space

Format

- ❖ Part 1: Background
- ❖ Part 2: Lower Bound
- ❖ Part 3: Arbitrary Model
- ❖ Part 4: Random-Order Model

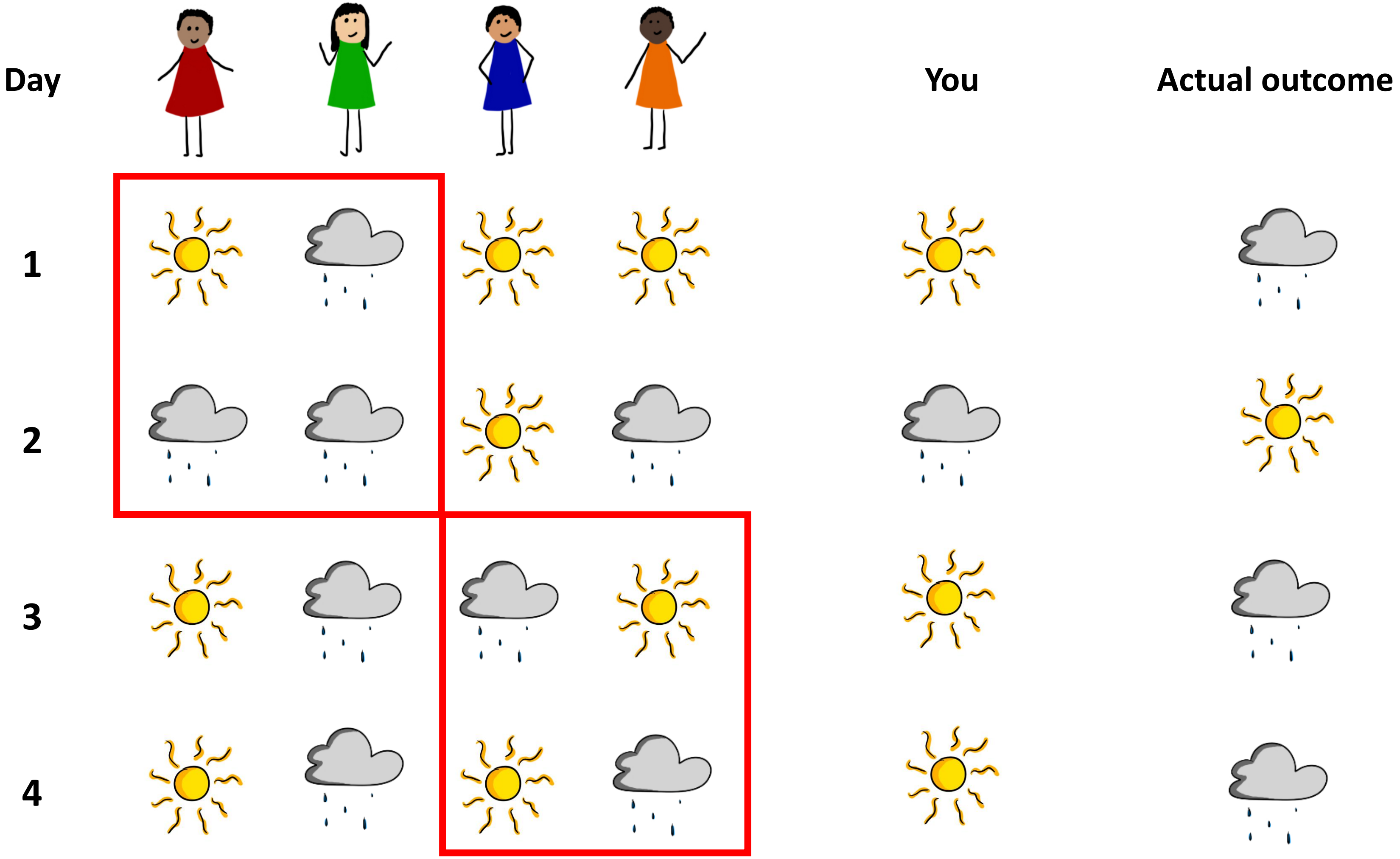
Questions?

“Low-Mistake” Regime

- For $M < \frac{\delta^2 T}{1280 \log^2 n}$ and $\delta > \sqrt{\frac{128 \log^2 n}{T}}$, there exists an algorithm that uses $\tilde{O}\left(\frac{n}{\delta T}\right)$ space and achieves regret δ with probability $\frac{4}{5}$

- We know there is a really accurate expert. What if we iteratively pick “pools” of experts and delete them if they run “poorly”?

Reduction Problem



No Mistake Regime

- If iteratively pick pool of k experts ("rounds") and output the majority vote of the pool while deleting any incorrect expert, each pool will have at most $O(\log k)$ errors
- If best expert makes no mistakes, use $\frac{n}{k}$ pools to achieve regret δT means setting $k = \tilde{O}\left(\frac{n}{\delta T}\right)$

No Mistake Regime Summary

- Algorithm: Iteratively pick pool of $k = \tilde{O}\left(\frac{n}{\delta T}\right)$ experts ("rounds") and output the majority vote of the pool while deleting any incorrect expert

- If the number of rounds is small, the pools must have done well so the overall regret is small
- The number of rounds cannot be large because at some point the best expert would have been sampled and retained

"Low-Mistake" Regime

- Algorithm: Iteratively pick pool of $k = \tilde{O}\left(\frac{n}{\delta T}\right)$ experts and output the majority vote of the pool while deleting any incorrect expert

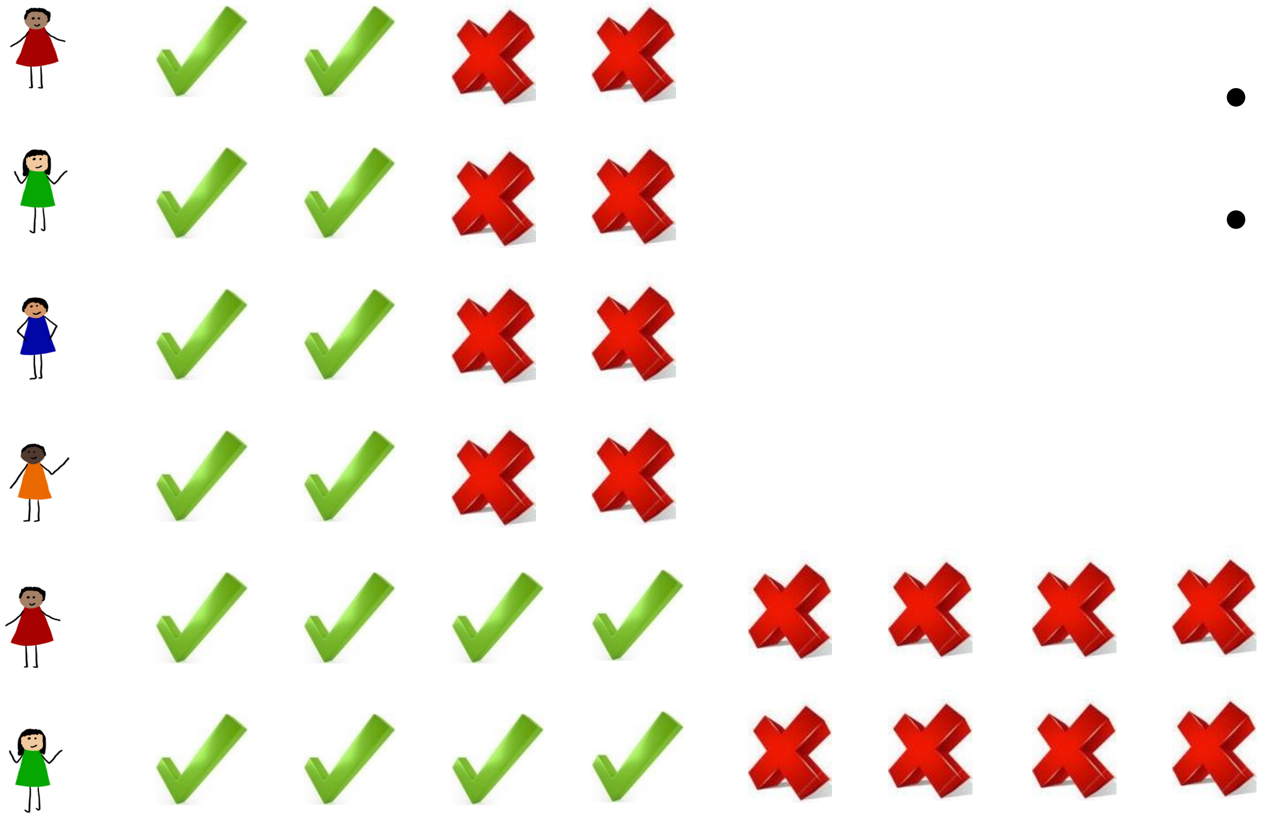
- If best expert makes M mistakes, use $\frac{nM}{k}$ pools to achieve regret δT means setting $k = \tilde{O}\left(\frac{nM}{\delta T}\right)$, which is too large

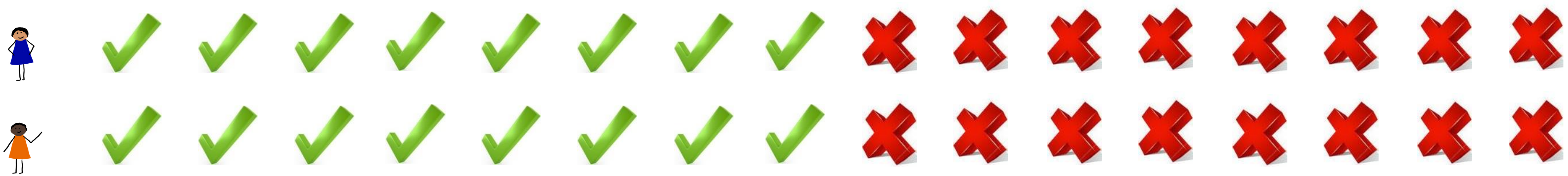
“Low-Mistake” Fix-Its

- Fix #1: Randomly sample pools of experts instead of iteratively picking pools
- Problem #1: Cannot guarantee that the best expert will be retained

- Fix #2: Delete experts that have erred with fraction more than $1 - \delta$
- Problem #2: “Build-up” of errors

A Really Bad Case Study





- Suppose $\delta = \frac{1}{2}$
- Example shows that the pool of k sampled experts can make roughly $T - T/k$ errors

“Low-Mistake” Regime

- Algorithm: Repeatedly sample a pool of $k = \tilde{O}\left(\frac{n}{\delta T}\right)$ experts and output the majority vote of the pool while deleting any expert with lower than $1 - \frac{\delta}{8 \log n}$ accuracy since it was sampled

WANT TO SHOW

- If the number of rounds is small, the pools must have done well so the overall regret is small
- The number of rounds cannot be large because at some point the best expert would have been sampled and retained

"Low-Mistake" Regime: First Property

- Algorithm: Repeatedly sample a pool of $k = \tilde{O}\left(\frac{n}{\delta T}\right)$ experts and output the majority vote of the pool while deleting any expert with lower than $1 - \frac{\delta}{8 \log n}$ accuracy since it was sampled
- Lemma: For $\delta > \sqrt{\frac{128 \log^2 n}{T}}$, a pool that is used for t days can only make $\frac{t\delta}{2} + 4 \log n$ mistakes
- For the algorithm to make $T\delta$ mistakes, need at least $\frac{T\delta}{8 \log n}$ rounds

"Low-Mistake" Regime: Second Property

- For the algorithm to make $T\delta$ mistakes, need at least $\frac{T\delta}{8\log n}$ rounds
- "BAD" day: the best expert is deleted by the pool if it is sampled on that day
- $|\mathrm{BAD}| \leq \frac{8M\log n}{\delta}$ and $M < \frac{\delta^2 T}{1280\log^2 n}$, so the remaining rounds must be sampled on "GOOD" days and avoid the best expert
- Must avoid sampling the best expert on at least $\frac{T\delta}{16\log n}$ rounds
- $O\left(\frac{n\log^2 n}{\delta T}\right)$ experts sampled in each round → low probability

Idealized Analysis

- Analysis: conditioned on the number of rounds being small, the algorithm makes a small number of mistakes
- Subtle pitfall: if the best expert is sampled on a good day, then there will be no more rounds, so the conditional probability analysis is difficult

Simple Decoupling Argument

- Instead of conditioning on total number of rounds, consider a decoupling argument, where we draw the day of the next round over a matching distribution
- Define a set of random variables $d_1, d_2, \ldots$ for each round's day
- Given d_i, draw d_{i+1} from the distribution of possible days for the next round based on possible experts sampled in the pool conditioned on entire history

- Results in a Poisson process, then apply same analysis bounding the number of resamplings on "GOOD" days

Arbitrary Order Model Summary

- Algorithm: Repeatedly sample a pool of $k = \tilde{O}\left(\frac{n}{\delta T}\right)$ experts and output the majority vote of the pool while deleting any expert with lower than $1 - \frac{\delta}{8 \log n}$ accuracy since it was sampled

- If the number of rounds is small, the pools must have done well so the overall regret is small
- The number of rounds cannot be large because at some point the best expert would have been sampled and retained

Format

- ❖ Part 1: Background
- ❖ Part 2: Lower Bound
- ❖ Part 3: Arbitrary Model
- ❖ Part 4: Random-Order Model

Questions?

Random-Order Streams

- There exists an algorithm that uses $O\left(\frac{n}{\delta^2 T}\log^2 n \log\frac{1}{\delta}\right)$ space achieves expected regret $\delta > \sqrt{\frac{8\log n}{T}}$ in the random-order model

TAKING A STEP BACK

- We used majority vote of remaining experts in sampled pool
- Instead of removing experts, could just downweight them and run deterministic weighted majority
- Why not randomized weighted majority, i.e., multiplicative weights?

Random-Order Streams

- Algorithm: Repeatedly sample a pool of $k = \tilde{O}\left(\frac{n}{\delta^2 T}\right)$ experts and run multiplicative weights on pool, resample if the expected cost of the pool over t time "is bad".

WANT TO SHOW

- If the number of rounds is small, the pools must have done well so the overall regret is small
- The number of rounds cannot be large because at some point the best expert would have been sampled and retained

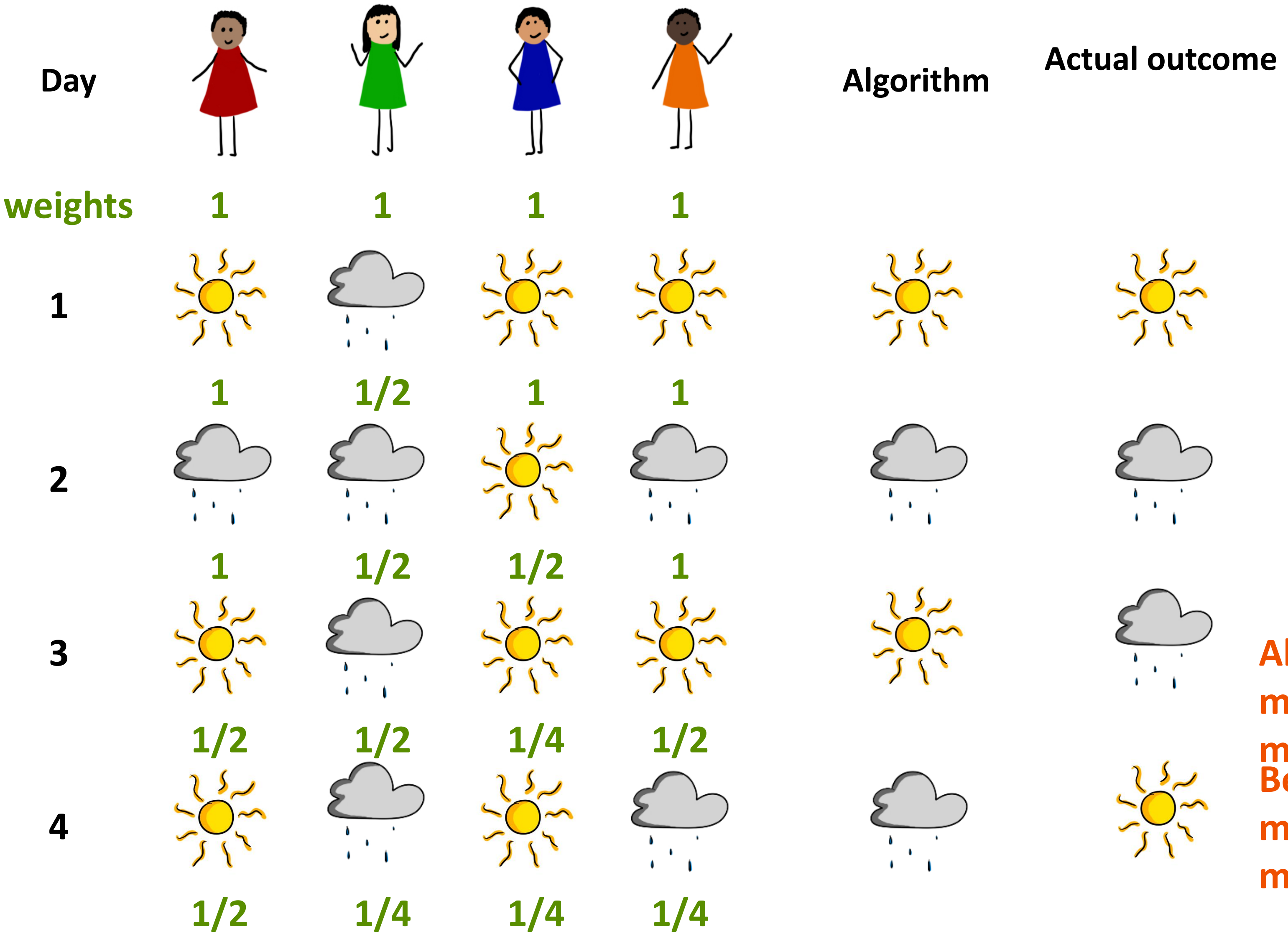
Multiplicative Weights Algorithm

Algorithm 4 The multiplicative weights algorithm.

Input: Number n of experts, number T of rounds, parameter ε

1: Initialize $w_i^{(1)} = 1$ for all $i \in [n]$.
2: **for** $t \in [T]$ **do**
3: $\quad p_i^{(t)} \leftarrow \frac{w_i^{(t)}}{\sum_{i \in [n]} w_i^{(t)}}$
4: $\quad$ Follow the advice of expert i with probability $p_i^{(t)}$.
5: $\quad$ Let $c_i^{(t)}$ be the cost for the decision of expert $i \in [n]$.
6: $\quad w_i^{(t+1)} \leftarrow w_i^{(t)} \left(1 - \varepsilon c_i^{(t)}\right)$
7: **end for**

Weighted Majority (Littlestone, Warmuth 89)



Algorithm makes 2 mistakes

Best expert makes 1 mistake

Multiplicative Weights Algorithm

Algorithm 4 The multiplicative weights algorithm.

Input: Number n of experts, number T of rounds, parameter ε

1: Initialize $w_i^{(1)} = 1$ for all $i \in [n]$.
2: **for** $t \in [T]$ **do**
3: $\quad p_i^{(t)} \leftarrow \frac{w_i^{(t)}}{\sum_{i \in [n]} w_i^{(t)}}$
4: $\quad$ Follow the advice of expert i with probability $p_i^{(t)}$.
5: $\quad$ Let $c_i^{(t)}$ be the cost for the decision of expert $i \in [n]$.
6: $\quad w_i^{(t+1)} \leftarrow w_i^{(t)} \left(1 - \varepsilon c_i^{(t)}\right)$
7: **end for**

- Theorem (Arora, Hazan, Kale 2012): Expected cost of the algorithm is $\sum_{t=1}^{T} \sum_{i=1}^{n} c_i^{(t)} p_i^{(t)} \leq \frac{\ln n}{\varepsilon} + (1+\varepsilon) \sum_{t=1}^{T} c_i^{(t)}$ for each $i \in [n]$ (and in particular the best expert), i.e, $\leq \frac{\ln n}{\varepsilon} + (1+\varepsilon) M$
- ε is trade-off term between multiplicative and additive error

Multiplicative Weights Algorithm

- Structural lemma: Let $X_1, \ldots, X_t$ be independent random variables in $[0,1]$ with expectation α and X be their sum. Then $\Pr\left[|X - \alpha t| \geq 4\sqrt{t \log T} + \frac{4 \log T}{\alpha}\right] \leq \frac{1}{T^2}$

- Casework: If $t \leq \frac{1}{\alpha^2}$, then deviation should not be more than $\frac{4 \log T}{\alpha}$
- Casework: If $t \geq \frac{1}{\alpha^2}$, then deviation should not be more than $4\sqrt{t \log T}$

Random-Order Streams: First Property

- Structural lemma: Let $X_1, \ldots, X_t$ be independent random variables in $[0,1]$ with expectation α and X be their sum. Then $\Pr\left[|X - \alpha t| \geq 4\sqrt{t \log T} + \frac{4 \log T}{\alpha}\right] \leq \frac{1}{T^2}$
- By the guarantee for multiplicative weights for $\varepsilon = \frac{\delta}{2}$, the cost of each pool is at most $\left(1 + \frac{\delta}{2}\right)\left(\alpha t + 4\sqrt{t \log T} + \frac{4 \log T}{\alpha}\right) + \frac{2 \ln n}{\delta}$
- For $\delta > \sqrt{\frac{16 \log^2 n}{T}}$, $\delta > \frac{M}{T}$, number of rounds must be at least $\Omega\left(\frac{\delta^2 T}{\log n}\right)$

Random-Order Streams: Second Property

- Number of rounds must be at least $\Omega\left(\frac{\delta^2 T}{\log n}\right)$
- Must avoid sampling the best expert on at least $\Omega\left(\frac{\delta^2 T}{\log n}\right)$ rounds
- $O\left(\frac{n \log^2 n}{\delta^2 T}\right)$ experts sampled in each round → low probability

- Must use same "decoupling" argument
- Similar analysis for $\delta \leq \frac{M}{T}$

Summary of Multiplicative Weights Algorithm

- There exists an algorithm that uses $O\left(\frac{n}{\delta^2 T}\log^2 n\right)$ space achieves expected regret $\delta > \sqrt{\frac{16\log^2 n}{T}}$, $\delta > \frac{M}{T}$ in the random-order model (assuming the number of mistakes M made by the best expert is known)

- Similar analysis for $\delta \leq \frac{M}{T}$ (gets $O\left(\frac{n}{\delta M}\log^2 n\right)$ space)
- Remove the assumption that M is known?

Removing the Assumption on M

- Do a binary search for $\frac{M}{T}$ with γ as the running estimate
- Proceed through $\ell = 2\log\frac{1}{\delta}$ epochs, each of length $\frac{\delta T}{\ell}$
- Run previous algorithm on with estimated cost $\gamma \cdot \frac{\delta T}{\ell}$ and target regret $O(1)$ until we have a $(1 + O(\delta))$-approximation of $\frac{M}{T}$ by γ

- Since regret is lower, space usage increases by a factor of $O(\ell)$ for $\delta \leq \frac{M}{T}$

Summary of Random-Order Model

- Given $\delta > \sqrt{\frac{16 \log^2 n}{T}}$, there exists an algorithm in the random-order model that uses achieves expected regret δ and uses $O\left(\frac{n}{\delta^2 T} \log^2 n\right)$ space for $\delta > \frac{M}{T}$ and $O\left(\frac{n}{\delta M} \log^2 n \log \frac{1}{\delta}\right)$ space for $\delta \leq \frac{M}{T}$

- Generalizes to other sequential prediction algorithms!

Follow the Perturbed Leader

Algorithm 2 The follow the perturbed leader algorithm (FPL*) from [KV05], instantiated for the experts problem.

Input: Number n of experts, number T of rounds, parameter ε

for $t \in [T]$ **do**
    for $i \in [n]$ **do**
        Choose $p_i^{(t)}$ independently, according to $\pm(2r/\varepsilon)$, where r is drawn from a standard exponential distribution
    end for
    Follow the expert i for whom the sum of their total cost so far and $p_i^{(t)}$ is the lowest
end for

Follow the Perturbed Leader

Algorithm 2 The follow the perturbed leader algorithm (FPL*) from [KV05], instantiated for the experts problem.

Input: Number n of experts, number T of rounds, parameter ε

```
for t ∈ [T] do
    for i ∈ [n] do
        Choose p_i^(t) independently, according to ±(2r/ε), where r is drawn from a standard exponential distribution
    end for
    Follow the expert i for whom the sum of their total cost so far and p_i^(t) is the lowest
end for
```

- Theorem (Kalai and Vempala 2005): Expected cost of the algorithm is $\frac{O(\ln n)}{\varepsilon} + (1+\varepsilon)\sum_{t=1}^{T} c_i^{(t)}$ for each $i \in [n]$ (and in particular the best expert), i.e, $\leq \frac{O(\ln n)}{\varepsilon} + (1+\varepsilon)M$
- ε is trade-off term between multiplicative and additive error

Summary of Results

- Any algorithm that achieves $\delta < \frac{1}{2+\sqrt{32 \ln 8}}$ regret with probability at least $\frac{3}{4}$ must use $\Omega\left(\frac{n}{\delta^2 T}\right)$ space
- There exists an algorithm that uses $O\left(\frac{n}{\delta^2 T}\log^2 n\right)$ space and achieves expected regret $\delta > \sqrt{\frac{16 \log^2 n}{T}}$ in the random-order model
- For $M < \frac{\delta^2 T}{1280 \log^2 n}$ and $\delta > \sqrt{\frac{128 \log^2 n}{T}}$, there exists an algorithm that uses $\tilde{O}\left(\frac{n}{\delta T}\right)$ space and achieves regret δ with probability $\frac{4}{5}$

Summary of Results II

- If the cost is between $[0, \rho]$, the expected regret is $\rho\delta$ for both models
- General regret in random-order model (multiplicative weights) → use same algorithm
- General regret in arbitrary-order model (weighted majority) → output random expert

Future Work?

- What bounds for arbitrary-order streams can be shown when the best expert incurs unrestricted cost? For constant δ, can we tolerate a constant fraction of mistakes?
- What happens when predictions are real values and evaluated against the correct answer with respect to a loss function?
- Extra constraints are imposed on the experts, i.e., side information