# Optimal Bounds for Noisy Computing

Nadim Ghaddar

University of Toronto

Joint work with Ziao Wang, Banghua Zhu, and Lele Wang

Banff International Research Station (BIRS) Workshop
Banff, Canada

March 11, 2024

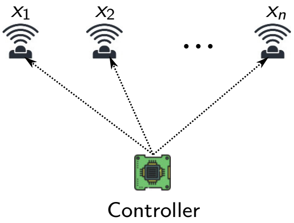
# Motivation

- Computing in the presence of noise

# Motivation

- Computing in the presence of noise

  E.g. $n$ sensors make noisy measurements of signals $x_1, \ldots, x_n$



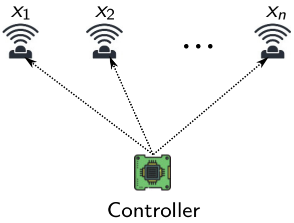

  Controller adaptively probes one of the sensors to make a measurement.

- Goal: Compute a function $f(x_1, \ldots, x_n)$ from noisy measurements

# Motivation

- Computing in the presence of noise

  E.g. $n$ sensors make noisy measurements of signals $x_1, \ldots, x_n$




  Controller adaptively probes one of the sensors to make a measurement.

- Goal: Compute a function $f(x_1, \ldots, x_n)$ from noisy measurements

- Applications
  - Fault tolerance
  - Active ranking
  - Recommendation systems
  - $\cdots$

# Problem Statement (OR Function)

- Let $\mathbf{x} = (x_1, \ldots, x_n) \in \{0,1\}^n$.
- **OR function:**

$$\mathrm{OR}(\mathbf{x}) = \begin{cases} 1, & \text{if } \exists\, i \in [n] : x_i = 1 \\ 0, & \text{otherwise.} \end{cases}$$

- **Goal:** Find an estimate of $\mathrm{OR}(\mathbf{x})$ using noisy readings.

# Problem Statement (OR Function)

- Let $\mathbf{x} = (x_1, \ldots, x_n) \in \{0,1\}^n$.
- **OR function:**

$$\mathrm{OR}(\mathbf{x}) = \begin{cases} 1, & \text{if } \exists\, i \in [n] : x_i = 1 \\ 0, & \text{otherwise.} \end{cases}$$

- **Goal:** Find an estimate of $\mathrm{OR}(\mathbf{x})$ using noisy readings.
  - At $k$th time step, submit query $U_k = x_i$ for some $i \in [n]$.
  - Receive noisy response

    $$Y_k = U_k \oplus Z_k,$$

    where $Z_k \sim \mathrm{Bern}(p)$, for some fixed and known $p < 1/2$.
  - After $T$ queries, compute estimate $\widehat{\mathrm{OR}}$ of $\mathrm{OR}(\mathbf{x})$.

# Problem Statement (OR Function)

- Let $\mathbf{x} = (x_1, \ldots, x_n) \in \{0,1\}^n$.
- **OR function:**

$$\mathrm{OR}(\mathbf{x}) = \begin{cases} 1, & \text{if } \exists\, i \in [n] : x_i = 1 \\ 0, & \text{otherwise.} \end{cases}$$

- **Goal:** Find an estimate of $\mathrm{OR}(\mathbf{x})$ using noisy readings.
  - At $k$th time step, submit query $U_k = x_i$ for some $i \in [n]$.
  - Receive noisy response

    $$Y_k = U_k \oplus Z_k,$$

    where $Z_k \sim \mathrm{Bern}(p)$, for some fixed and known $p < 1/2$.
  - After $T$ queries, compute estimate $\widehat{\mathrm{OR}}$ of $\mathrm{OR}(\mathbf{x})$.

- **Question:** How many queries are needed to find $\widehat{\mathrm{OR}}$ s.t.

$$\sup_{\mathbf{x}} \mathrm{P}(\widehat{\mathrm{OR}} \neq \mathrm{OR}(\mathbf{x})) \leq \delta?$$

# Related Work (OR Function)

- Noisy boolean decision trees
  - Computation of boolean functions in the presence of noise
  - $\Omega(n \log n)$ queries are necessary when querying strategy is non-adaptive[1][2][3]
  - $\mathcal{O}(n)$ queries are sufficient when querying strategy is adaptive using a tournament algorithm[4]

[1] R. L. Dobrushin and S. I. Ortyukov. "Lower bound for the redundancy of self-correcting arrangements of unreliable functional elements". In: *Problemy Peredachi Informatsii* 13.1 (1977), pp. 82–89.

[2] N. Pippenger, G. D. Stamoulis, and J. N. Tsitsiklis. "On a lower bound for the redundancy of reliable networks with noisy gates". In: *IEEE Trans. Inf. Theory* 37.3 (1991), pp. 639–643.

[3] P. Gács and A. Gál. "Lower bounds for the complexity of reliable Boolean circuits with noisy gates". In: *IEEE Trans. Inf. Theory* 40.2 (1994), pp. 579–583.

[4] U. Feige, P. Raghavan, D. Peleg, and E. Upfal. "Computing with Noisy Information". In: *SIAM J. Comput.* 23.5 (1994), pp. 1001–1018.

[5] J.-Y. Audibert, S. Bubeck, and R. Munos. "Best arm identification in multi-armed bandits.". In: *COLT* (2010), pp. 41–53.

# Related Work (OR Function)

- Noisy boolean decision trees
  - Computation of boolean functions in the presence of noise
  - $\Omega(n \log n)$ queries are necessary when querying strategy is non-adaptive[1,2,3]
  - $\mathcal{O}(n)$ queries are sufficient when querying strategy is adaptive using a tournament algorithm[4]

- Multi-armed bandits
  - Evaluating OR function of $n$ bits is the same as evaluating their maximum.
  - Best arm identification problem
  - Reward is $\text{Bern}(p)$ when bit is 0, and $\text{Bern}(1-p)$ when bit is 1
  - $\mathcal{O}\left(\frac{n \log(1/\delta)}{(1-2p)^2}\right)$ queries are sufficient[5]

[1] R. L. Dobrushin and S. I. Ortyukov. "Lower bound for the redundancy of self-correcting arrangements of unreliable functional elements". In: *Problemy Peredachi Informatsii* 13.1 (1977), pp. 82–89.

[2] N. Pippenger, G. D. Stamoulis, and J. N. Tsitsiklis. "On a lower bound for the redundancy of reliable networks with noisy gates". In: *IEEE Trans. Inf. Theory* 37.3 (1991), pp. 639–643.

[3] P. Gács and A. Gál. "Lower bounds for the complexity of reliable Boolean circuits with noisy gates". In: *IEEE Trans. Inf. Theory* 40.2 (1994), pp. 579–583.

[4] U. Feige, P. Raghavan, D. Peleg, and E. Upfal. "Computing with Noisy Information". In: *SIAM J. Comput.* 23.5 (1994), pp. 1001–1018.

[5] J.-Y. Audibert, S. Bubeck, and R. Munos. "Best arm identification in multi-armed bandits.". In: *COLT* (2010), pp. 41–53.

# Related Work (OR Function)

- Noisy boolean decision trees
  - Computation of boolean functions in the presence of noise
  - $\Omega(n \log n)$ queries are necessary when querying strategy is non-adaptive[1,2,3]
  - $\mathcal{O}(n)$ queries are sufficient when querying strategy is adaptive using a tournament algorithm[4]
- Multi-armed bandits
  - Evaluating OR function of $n$ bits is the same as evaluating their maximum.
  - Best arm identification problem
  - Reward is $\text{Bern}(p)$ when bit is 0, and $\text{Bern}(1-p)$ when bit is 1
  - $\mathcal{O}\left(\frac{n \log(1/\delta)}{(1-2p)^2}\right)$ queries are sufficient[5]

**Dependence on $p$ is not tight in prior work.**

[1] R. L. Dobrushin and S. I. Ortyukov. "Lower bound for the redundancy of self-correcting arrangements of unreliable functional elements". In: *Problemy Peredachi Informatsii* 13.1 (1977), pp. 82–89.

[2] N. Pippenger, G. D. Stamoulis, and J. N. Tsitsiklis. "On a lower bound for the redundancy of reliable networks with noisy gates". In: *IEEE Trans. Inf. Theory* 37.3 (1991), pp. 639–643.

[3] P. Gács and A. Gál. "Lower bounds for the complexity of reliable Boolean circuits with noisy gates". In: *IEEE Trans. Inf. Theory* 40.2 (1994), pp. 579–583.

[4] U. Feige, P. Raghavan, D. Peleg, and E. Upfal. "Computing with Noisy Information". In: *SIAM J. Comput.* 23.5 (1994), pp. 1001–1018.

[5] J.-Y. Audibert, S. Bubeck, and R. Munos. "Best arm identification in multi-armed bandits.". In: *COLT* (2010), pp. 41–53.

## Main Result

**Theorem 1 (OR function)**

*It is both sufficient and necessary to use*

$$(1 \pm o(1)) \frac{n \log \frac{1}{\delta}}{D_{\mathrm{KL}}(p \| 1-p)}$$

*queries in expectation to compute* OR *function with vanishing error probability* $\delta = o(1)$.

- $D_{\mathrm{KL}}(p \| 1-p)$: Kullback-Leibler (KL) divergence between Bern($p$) and Bern($1-p$)

# Main Result

**Theorem 1 (OR function)**

*It is both sufficient and necessary to use*

$$(1 \pm o(1)) \frac{n \log \frac{1}{\delta}}{D_{\mathrm{KL}}(p \| 1-p)}$$

*queries in expectation to compute* OR *function with vanishing error probability* $\delta = o(1)$.

- $D_{\mathrm{KL}}(p \| 1-p)$: Kullback-Leibler (KL) divergence between Bern($p$) and Bern($1-p$)
- Lower bound: Based on Le Cam's two point method
- Upper bound: Devise an adaptive querying strategy to compute the OR function

**Lemma (Le Cam's Two Point Lemma)**

*Let $(P_x)_{x \in \mathcal{X}}$ be a family of distributions, and let $\ell : \mathcal{X} \times \hat{\mathcal{X}} \to \mathbb{R}_+$ be any loss function. Let $x_1, x_2 \in \mathcal{X}$ satisfy that*

$$\ell(x_1, \hat{x}) + \ell(x_2, \hat{x}) \geq \Delta, \qquad \forall\, \hat{x} \in \hat{\mathcal{X}}.$$

*Then,*

$$\inf_{\hat{x}} \sup_{x \in \mathcal{X}} E_x[\ell(x, \hat{x})] \geq \frac{\Delta}{2} \left(1 - \| P_{x_1} - P_{x_2} \|_{TV}\right)$$

# Lower Bound: Le Cam's Two Point Method (1/3)

**Lemma (Le Cam's Two Point Lemma)**

*Let $(P_x)_{x \in \mathcal{X}}$ be a family of distributions, and let $\ell : \mathcal{X} \times \hat{\mathcal{X}} \to \mathbb{R}_+$ be any loss function. Let $x_1, x_2 \in \mathcal{X}$ satisfy that*

$$\ell(x_1, \hat{x}) + \ell(x_2, \hat{x}) \geq \Delta, \qquad \forall \hat{x} \in \hat{\mathcal{X}}.$$

*Then,*

$$\inf_{\hat{x}} \sup_{x \in \mathcal{X}} E_x[\ell(x, \hat{x})] \geq \frac{\Delta}{2} (1 - \| P_{x_1} - P_{x_2} \|_{TV})$$

- For computing the OR function, use $\mathcal{X} = \{0,1\}^n$, $\hat{\mathcal{X}} = \{0,1\}$ and

$$\ell(\mathbf{x}, \hat{x}) = \mathbb{1}\{\mathrm{OR}(\mathbf{x}) \neq \hat{x}\},$$

and $P_{\mathbf{x}}$ is the distribution of observations when the underlying sequence is $\mathbf{x}$.

# Lower Bound: Le Cam's Two Point Method (2/3)

- Consider length-$n$ sequences:
  - $\mathbf{x}_0 \triangleq$ all-zero sequence,
  - $\mathbf{x}_j \triangleq 1$ in $j$th position, and zeros everywhere else.
- For any $\hat{x} \in \{0, 1\}$,

$$\mathbb{1}\{\mathrm{OR}(\mathbf{x}_0) \neq \hat{x}\} + \mathbb{1}\{\mathrm{OR}(\mathbf{x}_j) \neq \hat{x}\} \geq 1$$

# Lower Bound: Le Cam's Two Point Method (2/3)

- Consider length-$n$ sequences:
  - $\mathbf{x}_0 \triangleq$ all-zero sequence,
  - $\mathbf{x}_j \triangleq 1$ in $j$th position, and zeros everywhere else.
- For any $\hat{x} \in \{0, 1\}$,

$$\mathbb{1}\{\mathrm{OR}(\mathbf{x}_0) \neq \hat{x}\} + \mathbb{1}\{\mathrm{OR}(\mathbf{x}_j) \neq \hat{x}\} \geq 1$$

- Le Cam's two point lemma implies

$$\inf_{\hat{x}} \sup_{\mathbf{x} \in \{0,1\}^n} \mathrm{P}(\hat{x} \neq \mathrm{OR}(\mathbf{x})) \geq \frac{1}{2}\left(1 - \|P_{\mathbf{x}_0} - P_{\mathbf{x}_j}\|_{TV}\right)$$

# Lower Bound: Le Cam's Two Point Method (2/3)

- Consider length-$n$ sequences:
  - $\mathbf{x}_0 \triangleq$ all-zero sequence,
  - $\mathbf{x}_j \triangleq 1$ in $j$th position, and zeros everywhere else.
- For any $\hat{x} \in \{0, 1\}$,

$$\mathbb{1}\{\mathsf{OR}(\mathbf{x}_0) \neq \hat{x}\} + \mathbb{1}\{\mathsf{OR}(\mathbf{x}_j) \neq \hat{x}\} \geq 1$$

- Le Cam's two point lemma implies

$$\begin{aligned}\inf_{\hat{x}} \sup_{\mathbf{x}\in\{0,1\}^n} \mathsf{P}(\hat{x} \neq \mathsf{OR}(\mathbf{x})) &\geq \frac{1}{2}\left(1 - \|P_{\mathbf{x}_0} - P_{\mathbf{x}_j}\|_{TV}\right) \\ &\overset{(a)}{\geq} \frac{1}{4}\exp\left(-D_{\mathsf{KL}}(P_{\mathbf{x}_0}, P_{\mathbf{x}_j})\right)\end{aligned}$$

where:

- $(a)$: Bretagnolle-Huber inequality

# Lower Bound: Le Cam's Two Point Method (2/3)

- Consider length-$n$ sequences:
  - $\mathbf{x}_0 \triangleq$ all-zero sequence,
  - $\mathbf{x}_j \triangleq 1$ in $j$th position, and zeros everywhere else.
- For any $\hat{x} \in \{0, 1\}$,

$$\mathbb{1}\{\mathrm{OR}(\mathbf{x}_0) \neq \hat{x}\} + \mathbb{1}\{\mathrm{OR}(\mathbf{x}_j) \neq \hat{x}\} \geq 1$$

- Le Cam's two point lemma implies

$$\begin{aligned}\inf_{\hat{x}} \sup_{\mathbf{x}\in\{0,1\}^n} \mathrm{P}(\hat{x} \neq \mathrm{OR}(\mathbf{x})) &\geq \frac{1}{2}\left(1 - \|P_{\mathbf{x}_0} - P_{\mathbf{x}_j}\|_{TV}\right) \\ &\overset{(a)}{\geq} \frac{1}{4}\exp\left(-D_{\mathrm{KL}}(P_{\mathbf{x}_0}, P_{\mathbf{x}_j})\right) \\ &\overset{(b)}{=} \frac{1}{4}\exp\left(-\mathrm{E}_{\mathbf{x}_0}[T_j] D_{\mathrm{KL}}(p\|1-p)\right),\end{aligned}$$

where:
  - $(a)$: Bretagnolle-Huber inequality
  - $(b)$: Divergence decomposition ($T_j$ is the number of times bit $j$ is queried)

# Lower Bound: Le Cam's Two Point Method (3/3)

- Recall

$$\inf_{\hat{x}} \sup_{\mathbf{x} \in \{0,1\}^n} \mathrm{P}(\hat{x} \neq \mathrm{OR}(\mathbf{x})) \geq \frac{1}{4} \exp\left(-\mathrm{E}_{\mathbf{x}_0}[T_j] D_{\mathrm{KL}}(p \| 1-p)\right),$$

where $T_j$ is the number of times bit $j$ is queried.

- Bound holds for each $j$.

# Lower Bound: Le Cam's Two Point Method (3/3)

- Recall

$$\inf_{\hat{x}} \sup_{\mathbf{x}\in\{0,1\}^n} \mathrm{P}(\hat{x} \neq \mathrm{OR}(\mathbf{x})) \geq \frac{1}{4} \exp\left(-\mathrm{E}_{\mathbf{x}_0}[T_j] D_{\mathrm{KL}}(p \| 1-p)\right),$$

where $T_j$ is the number of times bit $j$ is queried.

- Bound holds for each $j$.

- Since $\sum_{j=1}^{n} \mathrm{E}_{\mathbf{x}_0}[T_j] \leq T$, there must exist $j^*$ s.t. $\mathrm{E}_{\mathbf{x}_0}[T_{j^*}] \leq T/n$. Thus,

$$\inf_{\hat{x}} \sup_{\mathbf{x}\in\{0,1\}^n} \mathrm{P}(\hat{x} \neq \mathrm{OR}(\mathbf{x})) \geq \frac{1}{4} \exp\left(-\frac{T \cdot D_{\mathrm{KL}}(p \| 1-p)}{n}\right),$$

which gives the lower bound.

# Upper Bound: Proposed NoisyOR Algorithm (1/3)

- Proposed **NoisyOR** algorithm uses two subroutines:
  - EstimateSingleBit: estimates the value of a single bit using noisy queries
  - TournamentOR: existing algorithm that computes the $\mathrm{OR}$ function

# Upper Bound: Proposed NoisyOR Algorithm (1/3)

- Proposed **NoisyOR** algorithm uses two subroutines:
  - EstimateSingleBit: estimates the value of a single bit using noisy queries
  - TournamentOR: existing algorithm that computes the $\mathrm{OR}$ function

**Algorithm 1** EstimateSingleBit

**Input**: Single bit $x$, error probability $\delta$.
**Output**: Estimate of $x$.

```
Set t ← 1.
while true do
    Make noisy observation y_t of bit x.
    Set α ← P(X = 1 | Y^t = y^t).
    Set t ← t + 1.
    if α ≥ 1 − δ then return 1.
    else if α ≤ δ then return 0.
```

# Upper Bound: Proposed NoisyOR Algorithm (1/3)

- Proposed **NoisyOR** algorithm uses two subroutines:
  - EstimateSingleBit: estimates the value of a single bit using noisy queries
  - TournamentOR: existing algorithm that computes the OR function

**Algorithm 1** EstimateSingleBit

**Input**: Single bit $x$, error probability $\delta$.
**Output**: Estimate of $x$.

Set $t \leftarrow 1$.
**while true do**
  Make noisy observation $y_t$ of bit $x$.
  Set $\alpha \leftarrow \mathrm{P}(X = 1 \mid Y^t = y^t)$.
  Set $t \leftarrow t + 1$.
  **if** $\alpha \geq 1 - \delta$ **then return** 1.
  **else if** $\alpha \leq \delta$ **then return** 0.

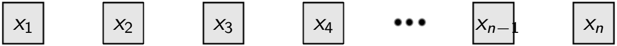
- EstimateSingleBit has error probability at most $\delta$ and uses at most

$$(1 + o(1)) \frac{\log(1/\delta)}{D_{\mathrm{KL}}(p \| 1 - p)}$$

queries in expectation.

# Upper Bound: Proposed NoisyOR Algorithm (2/3)

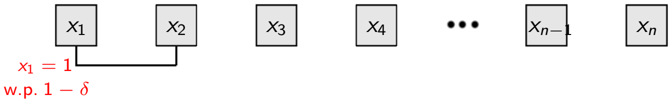
- TOURNAMENTOR: existing algorithm that computes the OR function[6][7]




[6] U. Feige, P. Raghavan, D. Peleg, and E. Upfal. "Computing with Noisy Information". In: *SIAM J. Comput.* 23.5 (1994), pp. 1001–1018.

[7] B. Zhu, Z. Wang, N. Ghaddar, J. Jiao, and L. Wang. *On the Optimal Bounds for Noisy Computing*. 2023. arXiv: 2306.11951.

- TournamentOR: existing algorithm that computes the OR function[6][7]

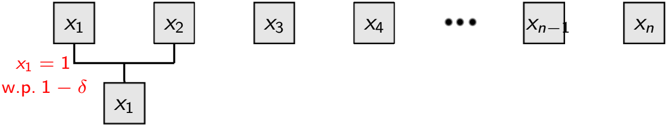



[6] U. Feige, P. Raghavan, D. Peleg, and E. Upfal. "Computing with Noisy Information". In: *SIAM J. Comput.* 23.5 (1994), pp. 1001–1018.

[7] B. Zhu, Z. Wang, N. Ghaddar, J. Jiao, and L. Wang. *On the Optimal Bounds for Noisy Computing*. 2023. arXiv: 2306.11951.

# Upper Bound: Proposed NoisyOR Algorithm (2/3)

- TournamentOR: existing algorithm that computes the OR function[6][7]

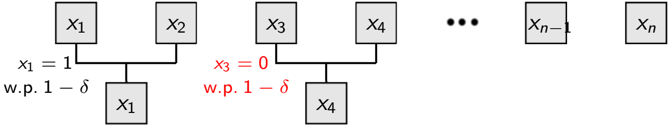



[6] U. Feige, P. Raghavan, D. Peleg, and E. Upfal. "Computing with Noisy Information". In: *SIAM J. Comput.* 23.5 (1994), pp. 1001–1018.

[7] B. Zhu, Z. Wang, N. Ghaddar, J. Jiao, and L. Wang. *On the Optimal Bounds for Noisy Computing*. 2023. arXiv: 2306.11951.

# Upper Bound: Proposed NoisyOR Algorithm (2/3)

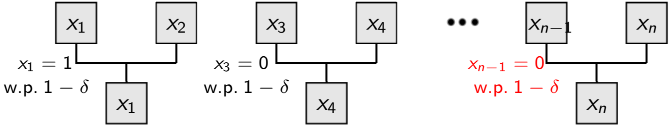
- TournamentOR: existing algorithm that computes the OR function[6][7]




[6] U. Feige, P. Raghavan, D. Peleg, and E. Upfal. "Computing with Noisy Information". In: *SIAM J. Comput.* 23.5 (1994), pp. 1001–1018.
[7] B. Zhu, Z. Wang, N. Ghaddar, J. Jiao, and L. Wang. *On the Optimal Bounds for Noisy Computing*. 2023. arXiv: 2306.11951.

# Upper Bound: Proposed NoisyOR Algorithm (2/3)

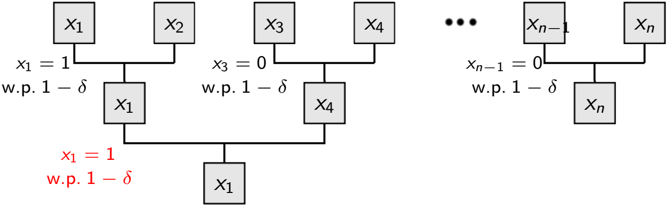
- TOURNAMENTOR: existing algorithm that computes the OR function[6][7]




[6] U. Feige, P. Raghavan, D. Peleg, and E. Upfal. "Computing with Noisy Information". In: *SIAM J. Comput.* 23.5 (1994), pp. 1001–1018.

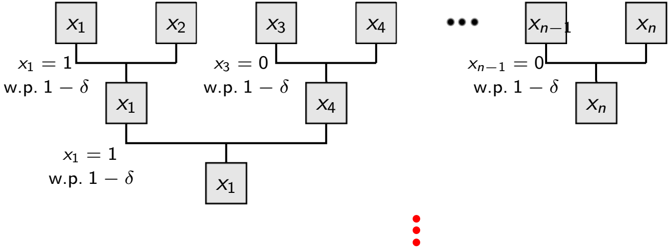
[7] B. Zhu, Z. Wang, N. Ghaddar, J. Jiao, and L. Wang. *On the Optimal Bounds for Noisy Computing*. 2023. arXiv: 2306.11951.

- TOURNAMENTOR: existing algorithm that computes the OR function[6][7]




[6] U. Feige, P. Raghavan, D. Peleg, and E. Upfal. "Computing with Noisy Information". In: *SIAM J. Comput.* 23.5 (1994), pp. 1001–1018.
[7] B. Zhu, Z. Wang, N. Ghaddar, J. Jiao, and L. Wang. *On the Optimal Bounds for Noisy Computing*. 2023. arXiv: 2306.11951.

- TOURNAMENTOR: existing algorithm that computes the OR function[6][7]

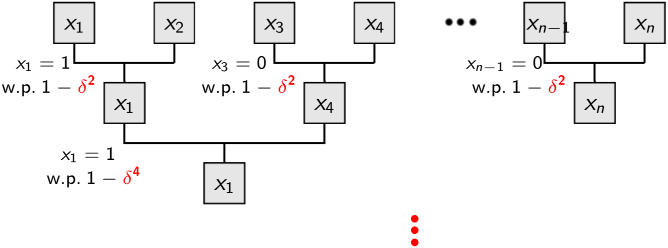



[6] U. Feige, P. Raghavan, D. Peleg, and E. Upfal. "Computing with Noisy Information". In: *SIAM J. Comput.* 23.5 (1994), pp. 1001–1018.

[7] B. Zhu, Z. Wang, N. Ghaddar, J. Jiao, and L. Wang. *On the Optimal Bounds for Noisy Computing*. 2023. arXiv: 2306.11951.

# Upper Bound: Proposed NoisyOR Algorithm (2/3)

- TOURNAMENTOR: existing algorithm that computes the OR function[6][7]

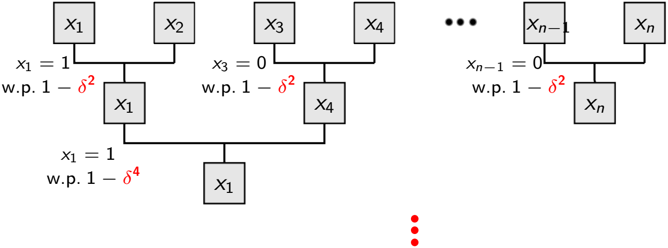



[6] U. Feige, P. Raghavan, D. Peleg, and E. Upfal. "Computing with Noisy Information". In: *SIAM J. Comput.* 23.5 (1994), pp. 1001–1018.

[7] B. Zhu, Z. Wang, N. Ghaddar, J. Jiao, and L. Wang. *On the Optimal Bounds for Noisy Computing*. 2023. arXiv: 2306.11951.

# Upper Bound: Proposed NoisyOR Algorithm (2/3)

- TOURNAMENTOR: existing algorithm that computes the OR function[6][7]




- TOURNAMENTOR has error probability at most $\delta$ and uses at most $\mathcal{O}(n)$ queries.

[6] U. Feige, P. Raghavan, D. Peleg, and E. Upfal. "Computing with Noisy Information". In: *SIAM J. Comput.* 23.5 (1994), pp. 1001–1018.

[7] B. Zhu, Z. Wang, N. Ghaddar, J. Jiao, and L. Wang. *On the Optimal Bounds for Noisy Computing*. 2023. arXiv: 2306.11951.

# Upper Bound: Proposed NoisyOR Algorithm (3/3)

- Proposed **NoisyOR** algorithm

**Algorithm 2** NoisyOR

**Input**: Bit sequence $\mathbf{x} = (x_1, \ldots, x_n)$, error probability $\delta$.
**Output**: Estimate of OR($\mathbf{x}$).

```
Set y ← ∅.
for i ∈ [n] do
    if EstimateSingleBit(x_i, δ/2) = 1 then
        Append x_i to y.
if length(y) = 0 then
    return 0.
else if length(y) ≥ max(log n, nδ log 1/δ) then
    return 1.
else
    return TournamentOR(y, δ/2)
```

# Upper Bound: Proposed NoisyOR Algorithm (3/3)

- Proposed **NoisyOR** algorithm

**Algorithm 2** NoisyOR

**Input**: Bit sequence $\mathbf{x} = (x_1, \ldots, x_n)$, error probability $\delta$.
**Output**: Estimate of OR($\mathbf{x}$).

```
Set y ← ∅.
for i ∈ [n] do
    if EstimateSingleBit(x_i, δ/2) = 1 then
        Append x_i to y.
if length(y) = 0 then
    return 0.
else if length(y) ≥ max(log n, nδ log 1/δ) then
    return 1.
else
    return TournamentOR(y, δ/2)
```

# Upper Bound: Proposed NoisyOR Algorithm (3/3)

- Proposed **NoisyOR** algorithm

**Algorithm 2** NoisyOR

**Input**: Bit sequence $\mathbf{x} = (x_1, \ldots, x_n)$, error probability $\delta$.
**Output**: Estimate of $\mathrm{OR}(\mathbf{x})$.

```
Set y ← ∅.
for i ∈ [n] do
    if EstimateSingleBit(x_i, δ/2) = 1 then
        Append x_i to y.
if length(y) = 0 then
    return 0.
else if length(y) ≥ max(log n, nδ log(1/δ)) then
    return 1.
else
    return TournamentOR(y, δ/2)
```

# Upper Bound: Proposed NoisyOR Algorithm (3/3)

- Proposed **NoisyOR** algorithm

**Algorithm 2** NoisyOR

**Input**: Bit sequence $\mathbf{x} = (x_1, \ldots, x_n)$, error probability $\delta$.
**Output**: Estimate of OR($\mathbf{x}$).

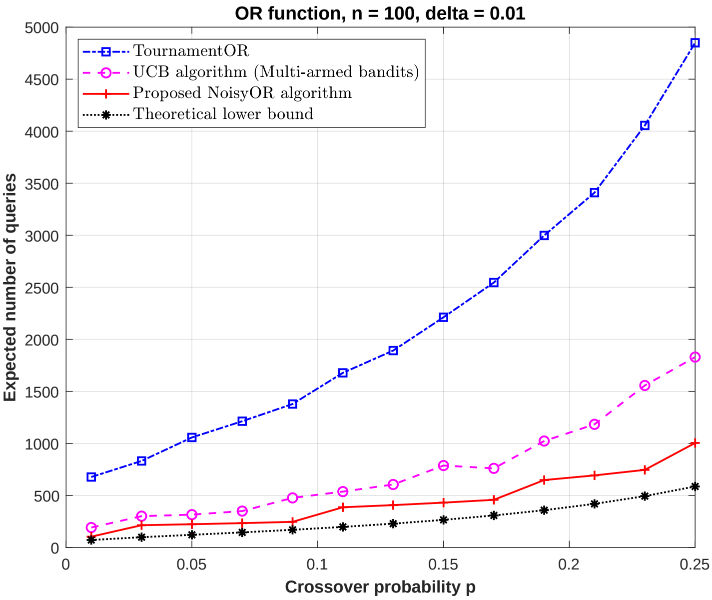
```
Set y ← ∅.
for i ∈ [n] do
    if EstimateSingleBit(x_i, δ/2) = 1 then
        Append x_i to y.
if length(y) = 0 then
    return 0.
else if length(y) ≥ max(log n, nδ log 1/δ) then
    return 1.
else
    return TournamentOR(y, δ/2)
```

- NoisyOR has error probability at most $\delta$ and uses at most

$$(1 + o(1)) \frac{n \log(1/\delta)}{D_{\mathrm{KL}}(p \| 1 - p)}$$

queries in expectation.

# Numerical Experiments



# Beyond the OR Function (1/2)

- Threshold function: For $\mathbf{x} \in \{0,1\}^n$,

$$\mathrm{TH}_k(\mathbf{x}) \triangleq \begin{cases} 1 & \text{if } \sum_{i=1}^n x_i \geq k, \\ 0 & \text{otherwise.} \end{cases}$$

Notice that $\mathrm{OR}(\mathbf{x}) = \mathrm{TH}_1(\mathbf{x})$.

**Theorem 2 ($\mathrm{TH}_k$ function)**

*For $k = o(n)$, it is both sufficient and necessary to use*

$$(1 \pm o(1)) \frac{n \log \frac{\mathbf{k}}{\delta}}{D_{\mathrm{KL}}(p \| 1-p)}$$

*queries in expectation to compute $\mathrm{TH}_k$ with a vanishing error probability $\delta = o(1)$.*

# Beyond the OR Function (2/2)

- **Noisy Comparison Model:** When $\mathbf{x} \in \mathbb{R}^n$,
  - At $k$th time step, query $(U_k, V_k) \triangleq (x_i, x_j)$ for $i \neq j$.
  - Receive noisy response $Y_k = \mathbb{1}_{\{U_k < V_k\}} \oplus Z_k$, where $Z_k \sim \mathrm{Bern}(p)$.

[8] Z. Wang, N. Ghaddar, B. Zhu, and L. Wang. *Noisy Sorting Capacity*. 2023. arXiv: 2202.01446.

[9] Y. Gu and Y. Xu. "Optimal Bounds for Noisy Sorting". In: *STOC 2023*, 1502–1515.

# Beyond the OR Function (2/2)

- **Noisy Comparison Model:** When $\mathbf{x} \in \mathbb{R}^n$,
  - At $k$th time step, query $(U_k, V_k) \triangleq (x_i, x_j)$ for $i \neq j$.
  - Receive noisy response $Y_k = \mathbb{1}_{\{U_k < V_k\}} \oplus Z_k$, where $Z_k \sim \mathrm{Bern}(p)$.

| Function | Description | Optimal Query complexity ($\delta = o(1)$) |
|---|---|---|
| MAX | Returns index of maximum element | $\frac{n \log \frac{1}{\delta}}{D_{\mathrm{KL}}(p \| 1-p)}$ |
| SEARCH | Takes $w$ as input and returns $i$ s.t. $x_i < w < x_{i+1}$ ($\mathbf{x}$ is sorted) | $\frac{\log n}{1 - H(p)}$ |
| SORT[8][9] | Sorts $\mathbf{x}$ | $\left[\frac{1}{1-H(p)} + \frac{1}{D_{\mathrm{KL}}(p \| 1-p)}\right] n \log n$ |

[8] Z. Wang, N. Ghaddar, B. Zhu, and L. Wang. *Noisy Sorting Capacity*. 2023. arXiv: 2202.01446.

[9] Y. Gu and Y. Xu. "Optimal Bounds for Noisy Sorting". In: *STOC 2023*, 1502–1515.

# Final Remarks

- Optimal bounds for noisy computing: OR, $\text{TH}_k$, MAX, SEARCH, SORT functions
- Extensions:
  - General channel models
  - Different performance metric
  - Unknown $p$ and/or query-dependent $p$
- Arxiv version: https://arxiv.org/abs/2309.03986
- Any questions?