Approximability of all Finite CSPs with Linear Sketches




Chi-Ning Chou   Madhu Sudan   Santhoshini Velusamy   Sasha Golovnev

Harvard University   Georgetown University

Motivation

Instead of NP-hardness or UG-hardness

Unconditional hardness of approximating constraint satisfaction problems (CSPs) in the streaming model

Capture many common computation problems

Practically interesting and theoretically nice

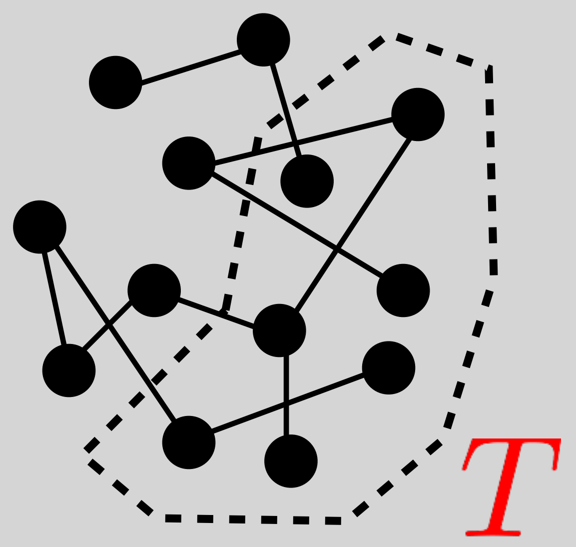
We characterize optimal approx. ratio for **all finite CSPs** in a weaker setting… !

Basic Definitions

Constraint Satisfaction Problem (CSP)

<u>Max-CUT as a CSP</u>

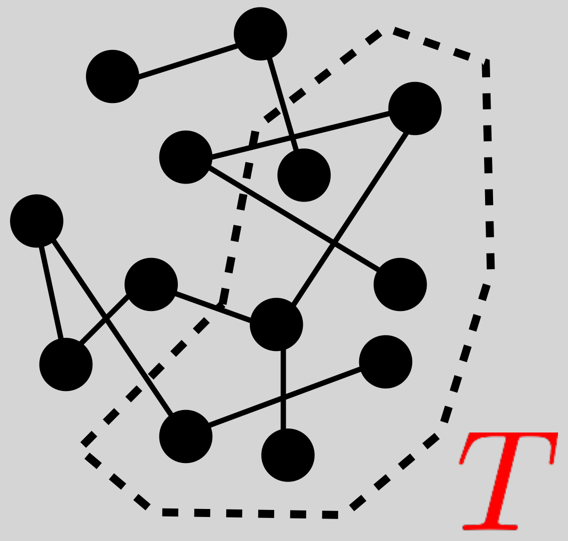
An undirected graph G



Constraint Satisfaction Problem (CSP)

Max-CUT as a CSP

An undirected graph G



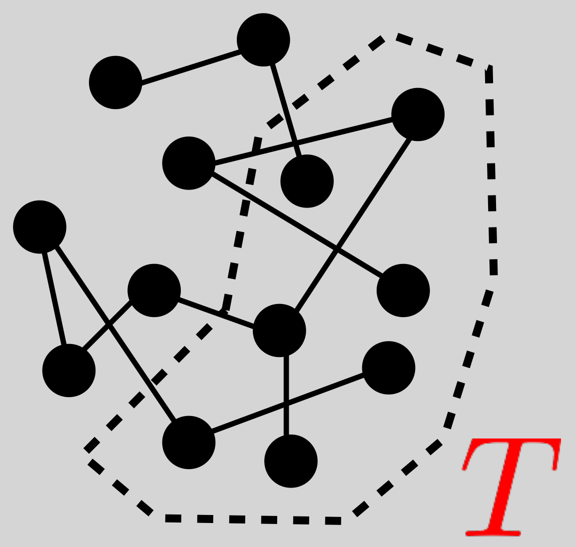
- **Variables:** $x_i = 1 \Leftrightarrow i \in T$

- **Variables:** $x_1, x_2, \ldots, x_n$ taking values in Σ (an alphabet set of finite size).

Constraint Satisfaction Problem (CSP)

Max-CUT as a CSP

An undirected graph G



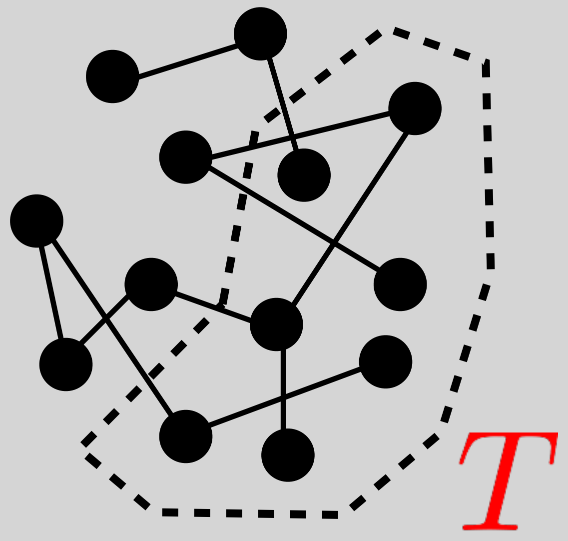
- **Variables:** $x_i = 1 \Leftrightarrow i \in T$
- **Constraints:** $(i, j) \in E \Rightarrow (\oplus, \{i, j\})$

- **Variables:** $x_1, x_2, \ldots, x_n$ taking values in Σ (an alphabet set of finite size).
- **Constraints:** (f, S) where $f \in \mathscr{F} \subset \{g : \Sigma^k \to \{0,1\}\}$ and $S \subset [n]$.

Constraint Satisfaction Problem (CSP)

Max-CUT as a CSP

An undirected graph G



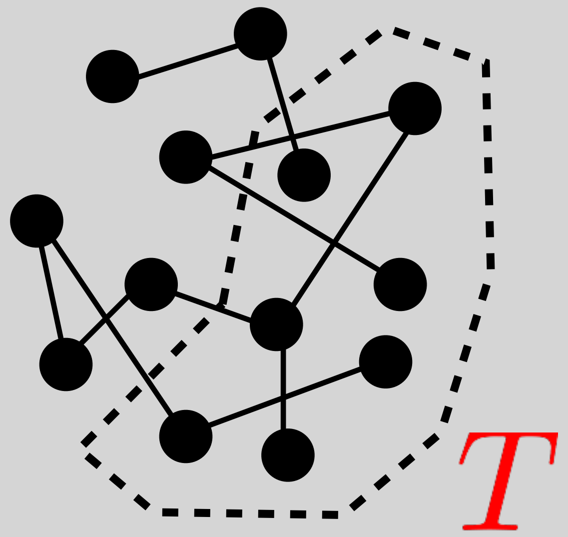
- **Variables:** $x_i = 1 \Leftrightarrow i \in T$
- **Constraints:** $(i, j) \in E \Rightarrow (\oplus, \{i, j\})$

- **Variables:** $x_1, x_2, \ldots, x_n$ taking values in Σ (an alphabet set of finite size).
- **Constraints:** (f, S) where $f \in \mathcal{F} \subset \{g : \Sigma^k \rightarrow \{0,1\}\}$ and $S \subset [n]$.
- **Input:** $\Psi = ((f_i, S_i))_{i \in [m]}$, number of constraints = m.

Constraint Satisfaction Problem (CSP)

Max-CUT as a CSP

An undirected graph G



- **Variables**: $x_i = 1 \Leftrightarrow i \in T$
- **Constraints**: $(i,j) \in E \Rightarrow (\oplus, \{i,j\})$
- **Value**: val_Ψ = max cut value

- **Variables**: $x_1, x_2, \ldots, x_n$ taking values in Σ (an alphabet set of finite size).
- **Constraints**: (f, S) where $f \in \mathcal{F} \subset \{g : \Sigma^k \to \{0,1\}\}$ and $S \subset [n]$.
- **Input**: $\Psi = ((f_i, S_i))_{i \in [m]}$, number of constraints = m.
- **Output**: The value of Ψ. Namely, the largest # of satisfied constraints.

  Formally, define $\text{val}_\Psi := \max_{\sigma:[n]\to\Sigma} \left| \{(f,S) \in \Psi : f(\sigma(x_S)) = 1\} \right| \in [0,m]$.

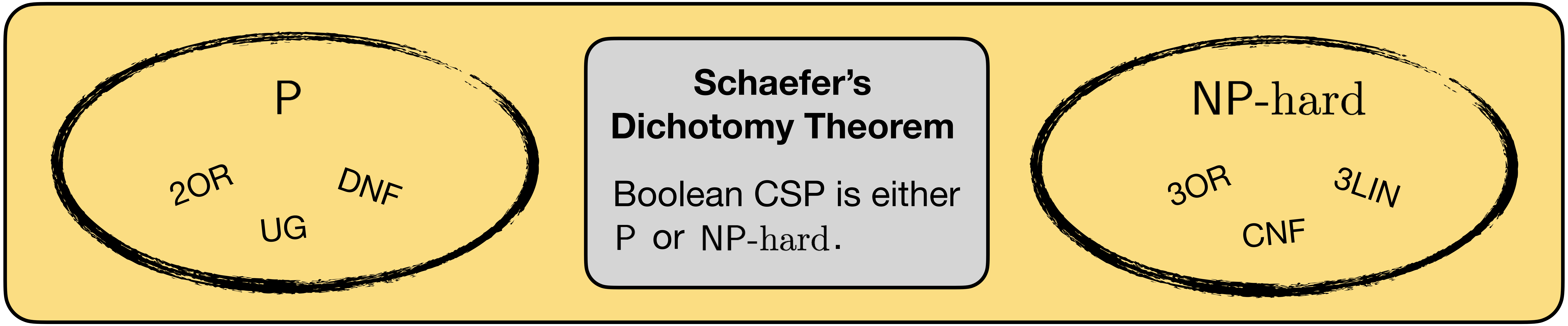
Constraint Satisfaction Problem (CSP)

- CSP is ubiquitous and has been extremely well-studied!
- Some CSPs are easy and some are hard to solve **exactly**.

- What about solving CSP **approximately**?

Constraint Satisfaction Problem (CSP)

- CSP is ubiquitous and has been extremely well-studied!
- Some CSPs are easy and some are hard to solve **exactly**.



- What about solving CSP **approximately**?

Approximating CSP

- Approximation ⇔ Distinguishing instances with different values.

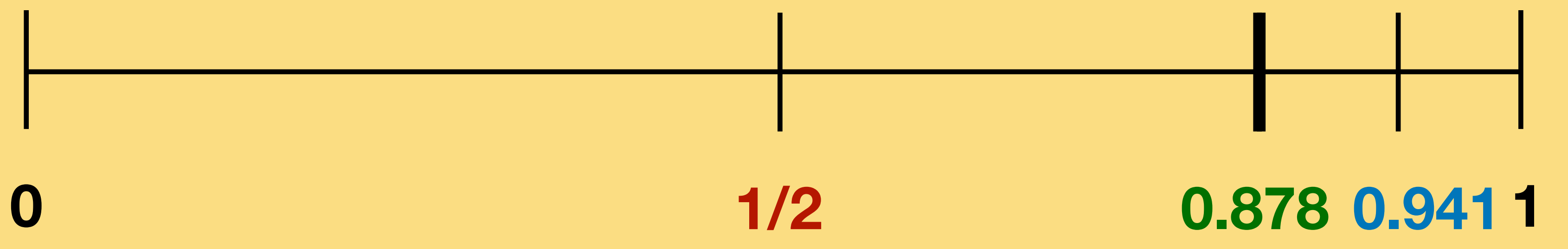
α-approximation: Let $\alpha \in (0,1]$. For any $v \in [0,m]$, can distinguish the following.

Yes: $\mathrm{val}_\Psi \geq v$ **No:** $\mathrm{val}_\Psi < \alpha \cdot v$

- $\alpha = 1$: the exact version; $\alpha = 1 - \epsilon, \forall \epsilon > 0$: fully approximation.
- **Algorithmic side**: Random sampling, SDP-based algorithms.
- **Hardness side**: NP-hardness or UG-hardness (through PCP theorem).
- Many fascinating results and open problems!

Approximating CSP

Max-CUT:



- Approximation ⇔ Distinguishing instances with different values.

α-approximation: Let $\alpha \in (0,1]$. For any $v \in [0,m]$, can distinguish the following.

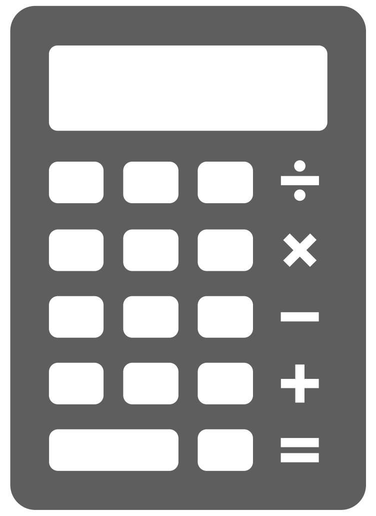
Yes: $\mathrm{val}_\Psi \geq v$ **No:** $\mathrm{val}_\Psi < \alpha \cdot v$

- $\alpha = 1$: the exact version; $\alpha = 1 - \epsilon, \forall \epsilon > 0$: fully approximation.
- **Algorithmic side**: Random sampling, SDP-based algorithms.
- **Hardness side**: NP-hardness or UG-hardness (through PCP theorem).
- Many fascinating results and open problems!

Unconditional Hardness

Through the Lens of Streaming Model

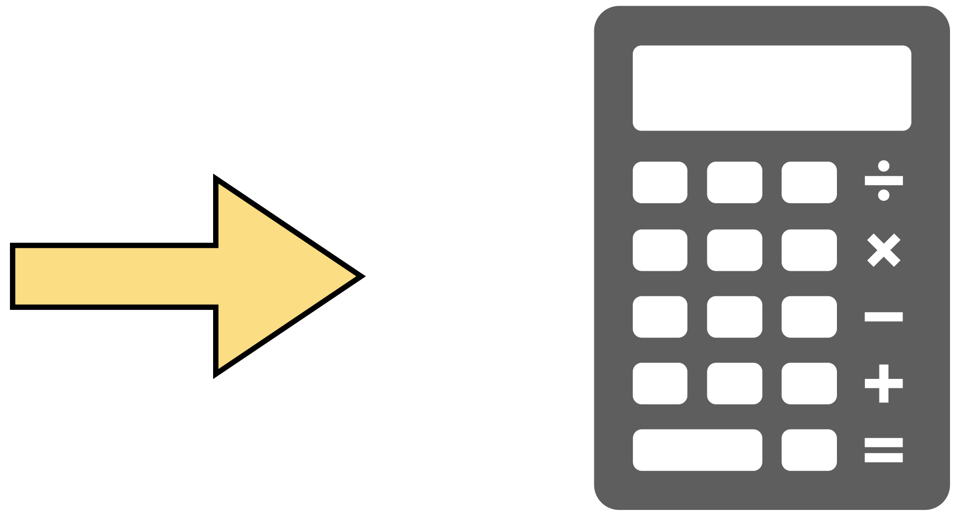
CSP in the Streaming Model



- Bounded space machine, i.e., only having $o(n)$ or even $O(\log n)$ space.
- The input (each constraint) arrives in a stream, i.e., see the input only once.
- **Observation**: Cannot even store an assignment (which requires n bits)!
- α**-approximation**: Output an integer v such that
  - there exists an assignment satisfying v constraints and
  - $v \geq \alpha \cdot \mathrm{val}_\Psi$.

CSP in the Streaming Model

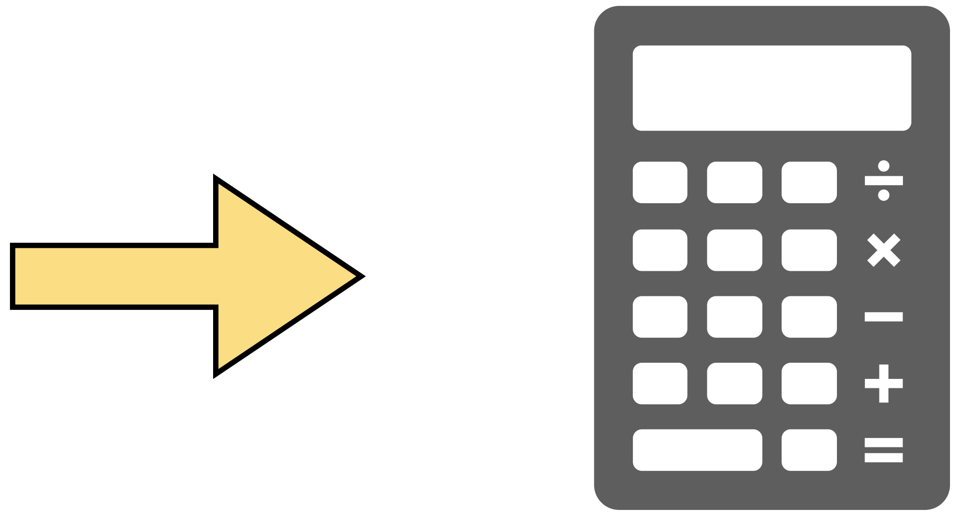
$x_1 \wedge x_6 \wedge \neg x_4$



- Bounded space machine, i.e., only having $o(n)$ or even $O(\log n)$ space.
- The input (each constraint) arrives in a stream, i.e., see the input only once.
- **Observation**: Cannot even store an assignment (which requires n bits)!
- α**-approximation**: Output an integer v such that
  - there exists an assignment satisfying v constraints and
  - $v \geq \alpha \cdot \mathrm{val}_\Psi$.

CSP in the Streaming Model

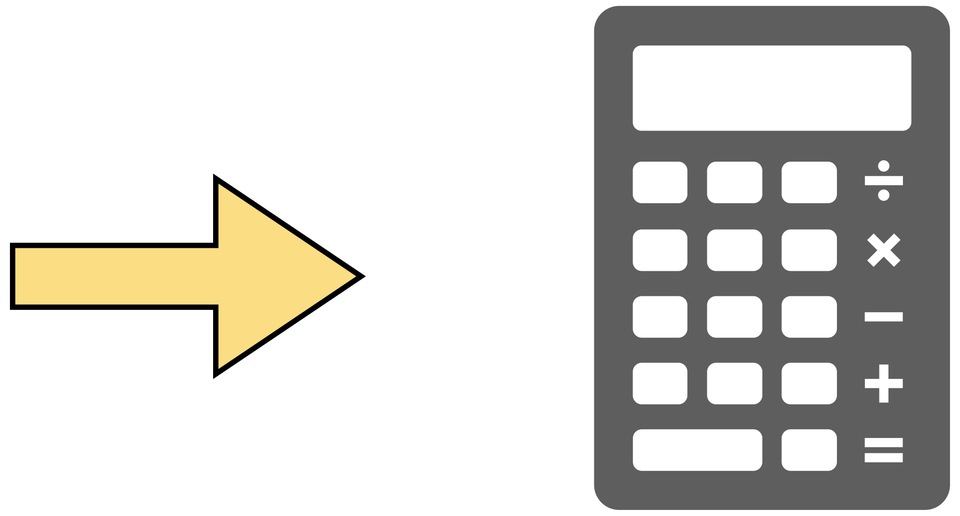
$$\neg x_5 \wedge x_2 \wedge \neg x_{23}$$



- Bounded space machine, i.e., only having $o(n)$ or even $O(\log n)$ space.
- The input (each constraint) arrives in a stream, i.e., see the input only once.
- **Observation**: Cannot even store an assignment (which requires n bits)!
- α**-approximation**: Output an integer v such that
  - there exists an assignment satisfying v constraints and
  - $v \geq \alpha \cdot \mathrm{val}_\Psi$.

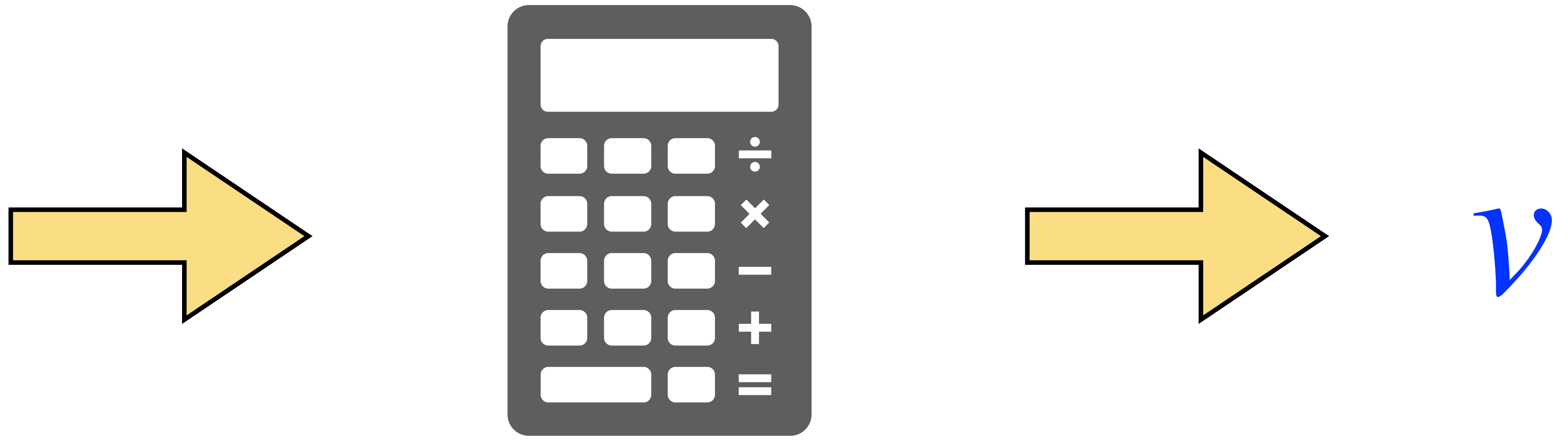
CSP in the Streaming Model

$$x_2 \wedge x_7 \wedge \neg x_{11}$$



- Bounded space machine, i.e., only having $o(n)$ or even $O(\log n)$ space.
- The input (each constraint) arrives in a stream, i.e., see the input only once.
- **Observation**: Cannot even store an assignment (which requires n bits)!
- α**-approximation**: Output an integer v such that
  - there exists an assignment satisfying v constraints and
  - $v \geq \alpha \cdot \mathrm{val}_\Psi$.

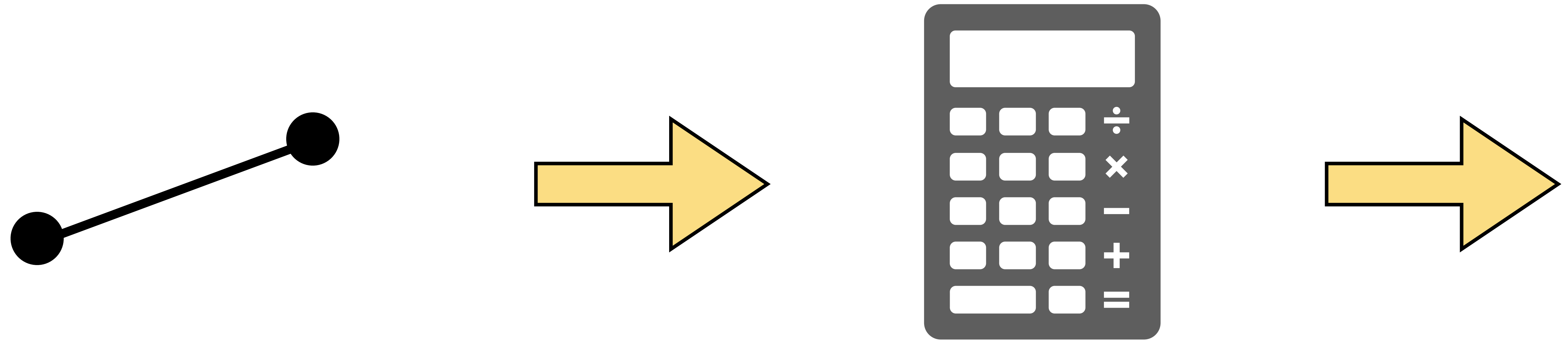
CSP in the Streaming Model



- Bounded space machine, i.e., only having $o(n)$ or even $O(\log n)$ space.
- The input (each constraint) arrives in a stream, i.e., see the input only once.
- **Observation**: Cannot even store an assignment (which requires n bits)!
- α**-approximation**: Output an integer v such that
  - there exists an assignment satisfying v constraints and
  - $v \geq \alpha \cdot \mathrm{val}_\Psi$.

Example: Max-CUT in the Streaming Model

- Trivial random sampling gives **1/2**-approximation in the streaming model!

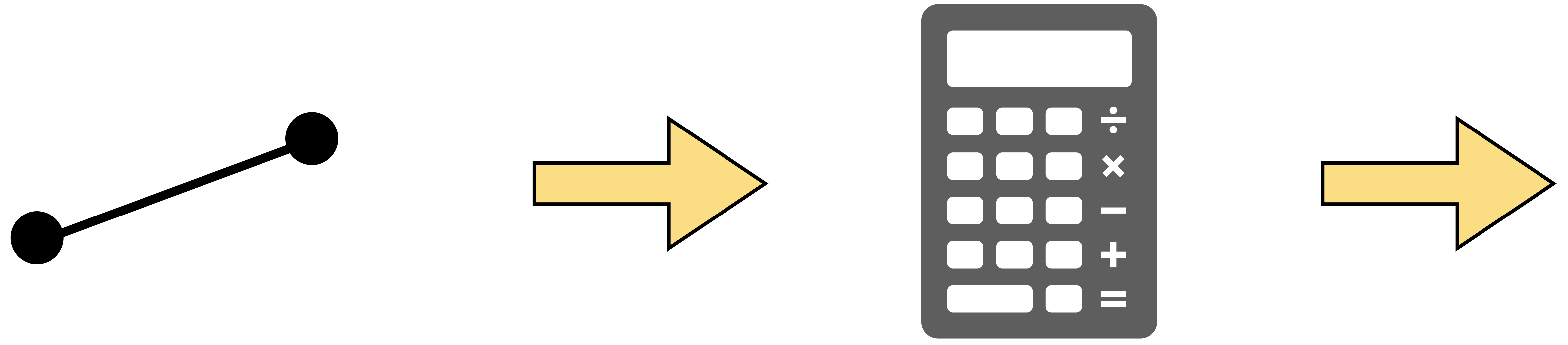


$$\frac{\#\ \text{edges}}{2}$$

- Use $O(\log n)$ space to record # edges.
- Why this would work?

Example: Max-CUT in the Streaming Model

- Trivial random sampling gives **1/2**-approximation in the streaming model!



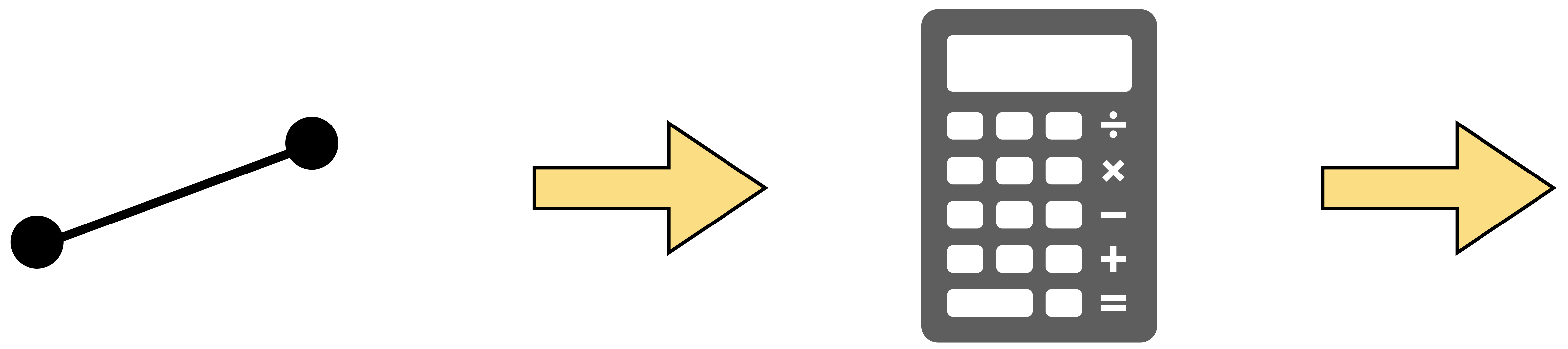
$$\frac{\#\text{ edges}}{2}$$

- Use $O(\log n)$ space to record # edges.
- Why this would work?

$$\mathop{\mathbb{E}}_{\sigma}[\mathsf{val}_{\mathcal{C}}(\sigma)]$$

Example: Max-CUT in the Streaming Model

- Trivial random sampling gives **1/2**-approximation in the streaming model!



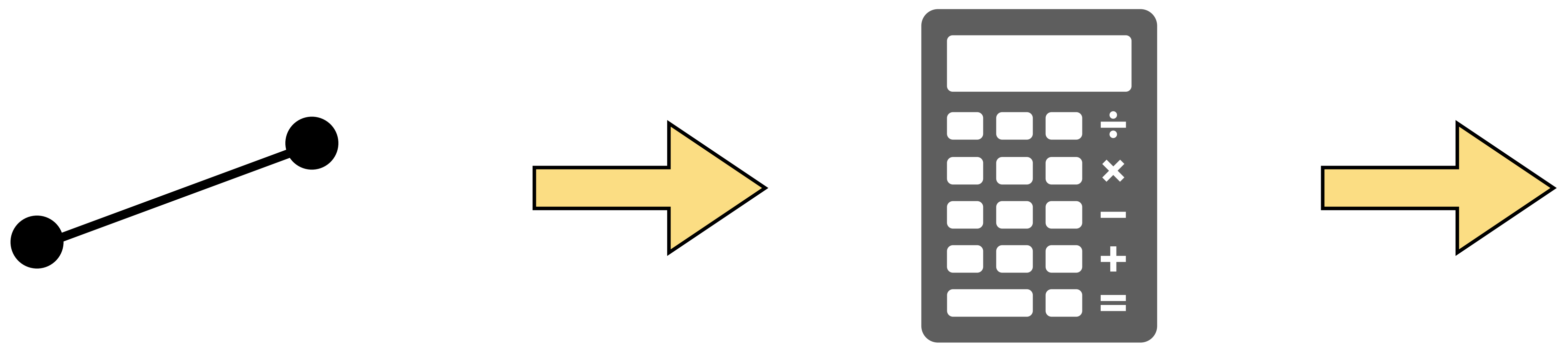
$$\frac{\#\ \text{edges}}{2}$$

- Use $O(\log n)$ space to record # edges.
- Why this would work?

$$\mathop{\mathbb{E}}_{\sigma}[\mathsf{val}_{\mathcal{C}}(\sigma)] = \sum_{e \in E(G)} \Pr_{\sigma}[e \text{ is a cut}]$$

Example: Max-CUT in the Streaming Model

- Trivial random sampling gives **1/2**-approximation in the streaming model!



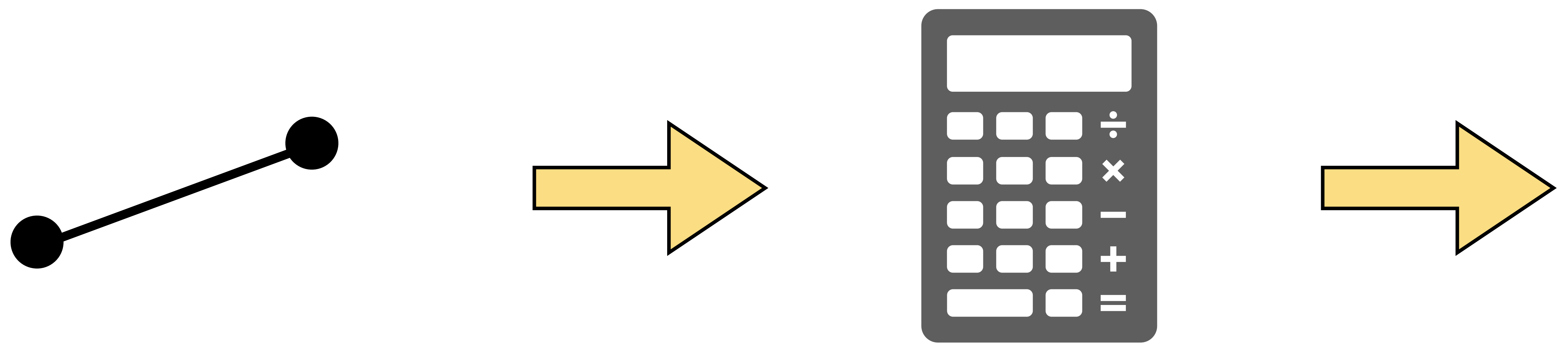
$$\frac{\#\ \text{edges}}{2}$$

- Use $O(\log n)$ space to record # edges.
- Why this would work?

$$\mathop{\mathbb{E}}_{\sigma}[\mathsf{val}_{\mathcal{C}}(\sigma)] = \sum_{e \in E(G)} \Pr_{\sigma}[e \text{ is a cut}] = \frac{|E(G)|}{2}$$

Example: Max-CUT in the Streaming Model

- Trivial random sampling gives **1/2**-approximation in the streaming model!



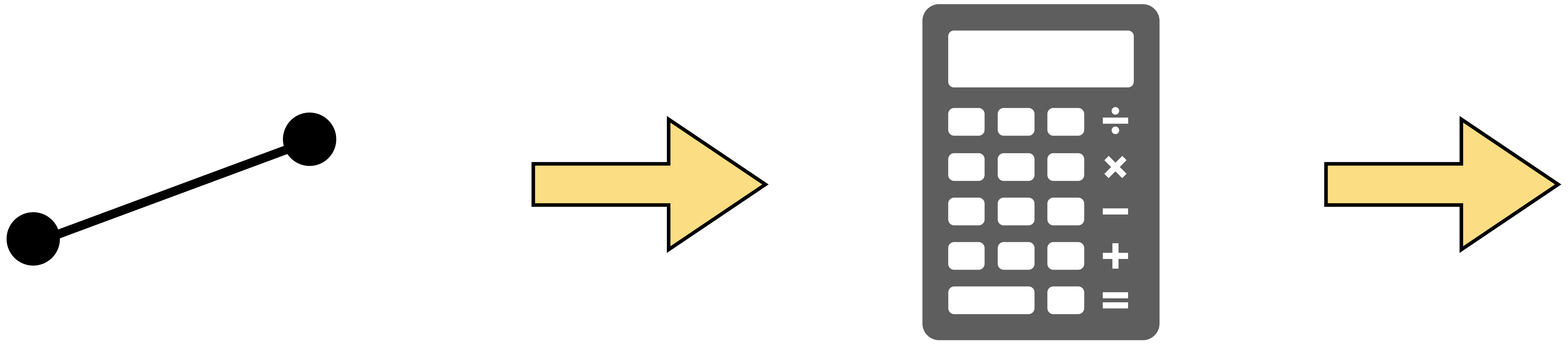
$$\frac{\#\ \text{edges}}{2}$$

- Use $O(\log n)$ space to record # edges.
- Why this would work?

$$\mathop{\mathbb{E}}_{\sigma}[\mathsf{val}_{\mathcal{C}}(\sigma)] = \sum_{e \in E(G)} \Pr_{\sigma}[e \text{ is a cut}] = \frac{|E(G)|}{2}$$

Random cut has value $\frac{\#\ \text{edges}}{2}$ $\Rightarrow$ Exist a cut having value $\frac{\#\ \text{edges}}{2}$

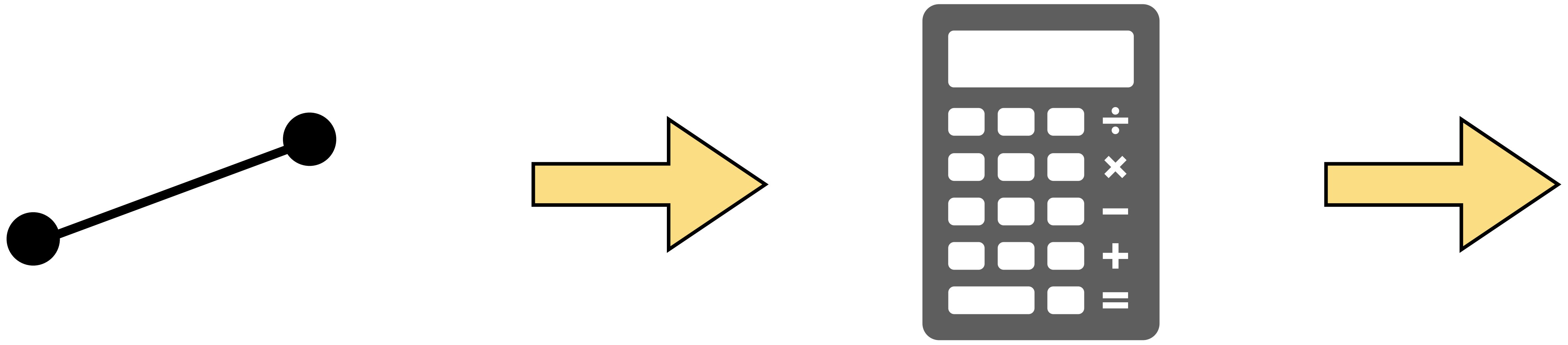
Trivial Random Sampling is Optimal for Max-CUT!



$$\frac{\#\ \text{edges}}{2}$$

Trivial Random Sampling is Optimal for Max-CUT!

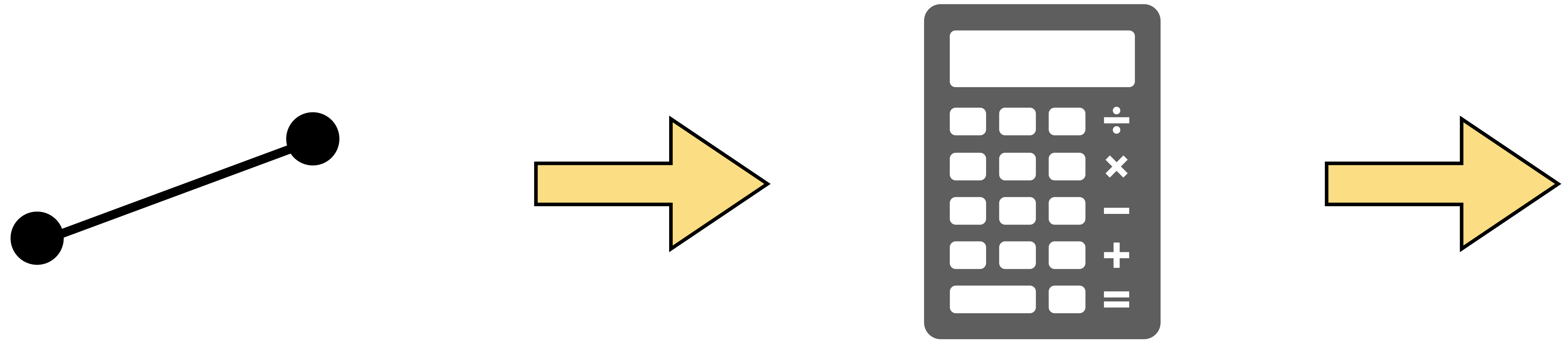
- Trivial random sampling gives **1/2**-approximation using $O(\log n)$ space.



$$\frac{\#\ \text{edges}}{2}$$

Trivial Random Sampling is Optimal for Max-CUT!

- Trivial random sampling gives **1/2**-approximation using $O(\log n)$ space.

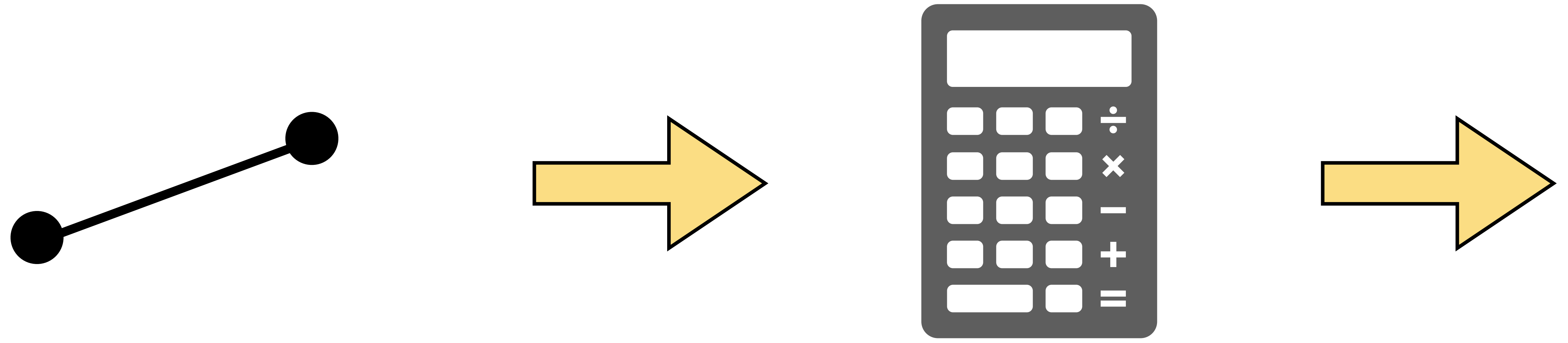


$$\frac{\#\ \text{edges}}{2}$$

- $\forall \epsilon > 0$, there's no (**1/2**+ϵ)-approximation streaming algorithm for Max-CUT!

Trivial Random Sampling is Optimal for Max-CUT!

- Trivial random sampling gives **1/2**-approximation using $O(\log n)$ space.



$$\frac{\#\ \text{edges}}{2}$$

- $\forall \epsilon > 0$, there's no (**1/2**+ϵ)-approximation streaming algorithm for Max-CUT!
  - ✦ [Kapralov-Khanna-Sudan 15]: $\Omega(\sqrt{n})$ space.
  - ✦ [Kapralov-Khanna-Sudan-Velingker 17]: 0.99-approx. needs $\Omega(n)$ space.
  - ✦ [Kapralov-Krachun 19]: $\Omega(n)$ space.

There's a SDP-based algorithm which gives **0.878**-approx.

More Recent Developments on the Streaming Complexity of CSPs

Paper	CSPs	Space Complexity	Type of Results
[KKS15]	Max-CUT	$\Omega(\sqrt{n})$	0.5-approx. hardness
[KKSV17]	Max-CUT	$\Omega(n)$	0.99-approx. hardness
[GVV17]	Max-DICUT	$O(\log n)$	0.4-approx. algorithm
[GT19]	Max-UG	$\Omega(\sqrt{n})$	Approx. resistance
[KK19]	Max-CUT	$\Omega(n)$	0.5-approx. hardness
[**C**GV20]	All Boolean 2-CSP	$O(\log n)$ v.s. $\Omega(\sqrt{n})$	Full classification
[**C**GSV21a]	All Boolean finite CSPs	$O(\log^3 n)$ v.s. $\Omega(\sqrt{n})$	Full classification
[**C**GSV21b]	All finite CSPs	$O(\log^3 n)$ v.s. $\Omega(\sqrt{n})$	Full classification
[SSV21]	All ordering CSPs	$O(\log^3 n)$ v.s. $\Omega(\sqrt{n})$	Approx. resistance
[**C**GSVV21]	All finite CSPs	$\Omega(n)$	Partial hardness

There are also lots of exciting recent works on graph problem and learning!

Our Results

Streaming algorithms with a certain "composable property", ask me offline for definition!

- We characterize the approximation ratio for every finite CSPs!
- In a slightly weaker setting of "sketching algorithms".

Classification Theorem (Informal)*

For every finite CSP, there exist α such that for every $\epsilon > 0$,

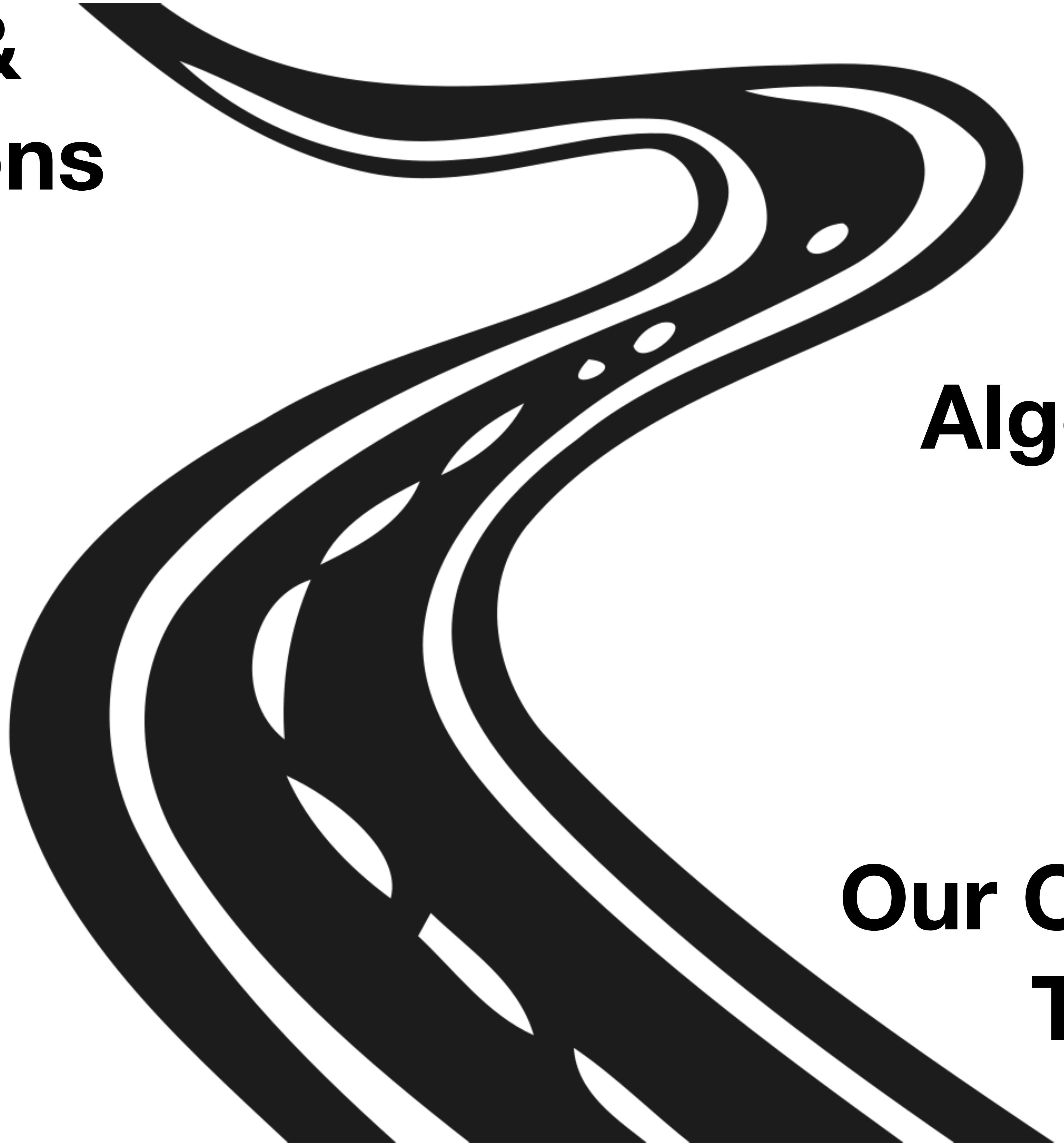
(i) there's an $(\alpha - \epsilon)$-approx. by linear sketches that uses $O(\log^3 n)$ space and

(ii) $(\alpha + \epsilon)$ -approx. using sketching algorithms requires $\Omega(\sqrt{n})$ space.

- More details stated in later slides.

Roadmap for Rest of the Talk

Conclusion & Future Directions



Algorithm

Hardness

Our Classification Theorem

Our Classification Theorem

And a Glimpse into the Proof

Our Classification Theorem for Approximating Finite CSP

For every finite $q, k \in \mathbb{N}$, every $\mathscr{F} \subset \{f : [q]^k \to \{0,1\}\}$, and every $0 \leq \beta < \gamma \leq 1$, we define two sets $\boxed{K_\gamma^Y(\mathscr{F}),\ K_\beta^N(\mathscr{F})}$ over $\mathbb{R}^{|\mathscr{F}|kq}$ and show that they are computable in PSPACE. *Will elaborate in next few slides!*

Our Classification Theorem for Approximating Finite CSP

For every finite $q, k \in \mathbb{N}$, every $\mathcal{F} \subset \{f : [q]^k \to \{0,1\}\}$, and every $0 \leq \beta < \gamma \leq 1$, we define two sets $\boxed{K_\gamma^Y(\mathcal{F}),\ K_\beta^N(\mathcal{F})}$ over $\mathbb{R}^{|\mathcal{F}|kq}$ and show that they are computable in PSPACE. *Will elaborate in next few slides!*

Classification Theorem

(i) If $K_\gamma^Y(\mathcal{F}) \cap K_\beta^N(\mathcal{F}) = \emptyset$, then (γ, β)-Max-CSP($\mathcal{F}$) can be solved by linear sketches in the dynamic setting using $O(\log^3 n)$ space;

(ii) If $K_\gamma^Y(\mathcal{F}) \cap K_\beta^N(\mathcal{F}) \neq \emptyset$, then $(\gamma - \epsilon, \beta + \epsilon)$-Max-CSP($\mathcal{F}$) by sketching algorithms in the insertion-only setting requires $\Omega(\sqrt{n})$ space $\forall \epsilon > 0$.

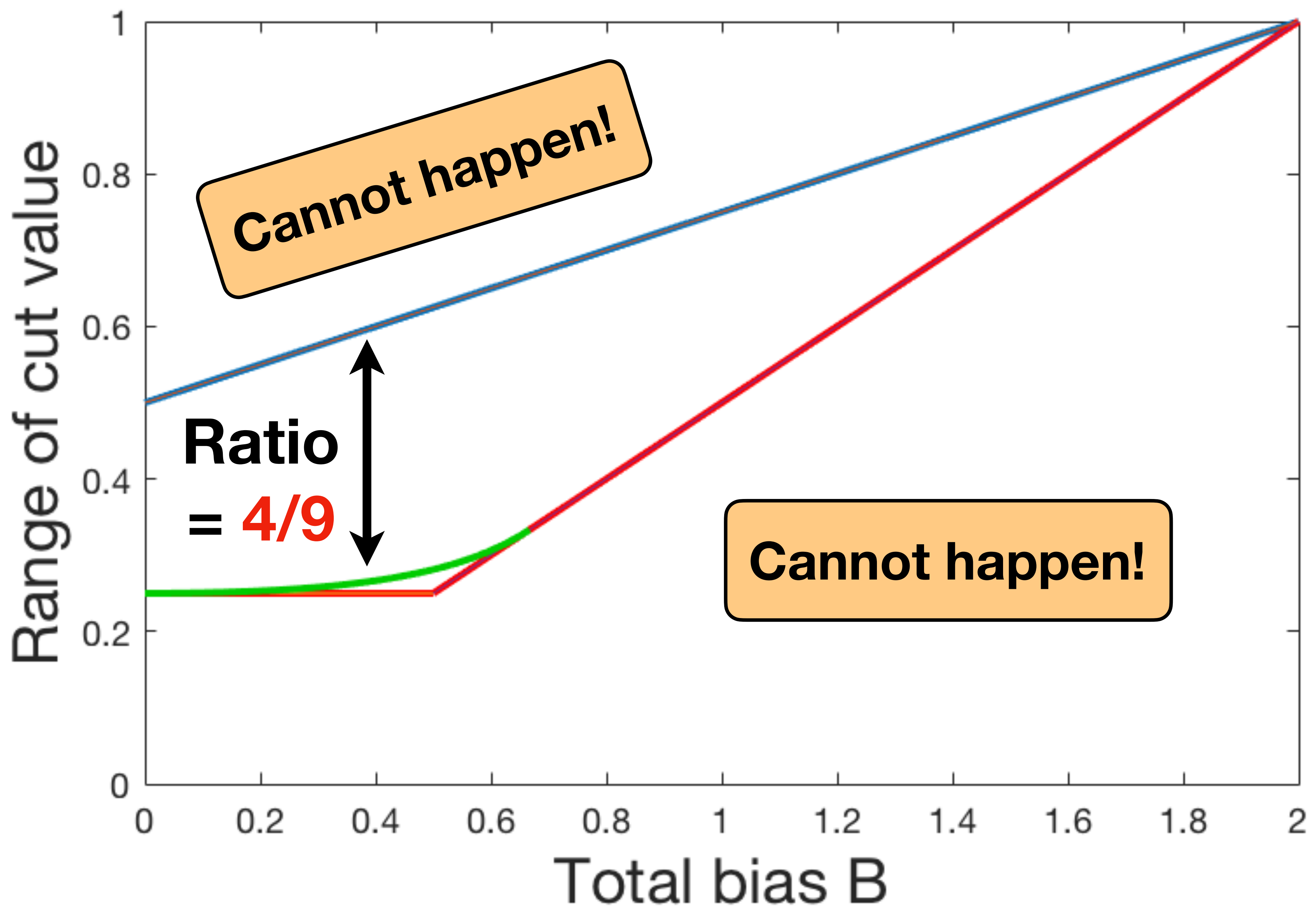
See our paper for more corollaries in some special settings!

Example: Max-DICUT [GVV17, **C**GV20]

ℓ_1 norm of the bias vector! Can be estimated using standard steaming tools.

Definition (bias and total bias):

$$\text{bias}(v) = \text{in-degree} - \text{out-degree} \quad \text{and} \quad B = \sum_v |\text{bias}(v)|$$

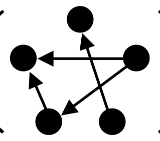
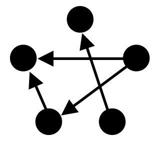


- **Blue line**: cut value upper bound.
- **Red line**: The cut value of greedy cut.
- **Green line**: Cut value achieved by random sampling with bias.
- **Streaming algorithm**: Estimate B and output max {**green line**, **red line**}.
- **Ratio**: When B = 2/5, the ratio is **4/9**.

Standard Ideas in Proving Streaming Lower bound

Standard Ideas in Proving Streaming Lower bound

The space of Max-CSP($\mathscr{F}$) instances

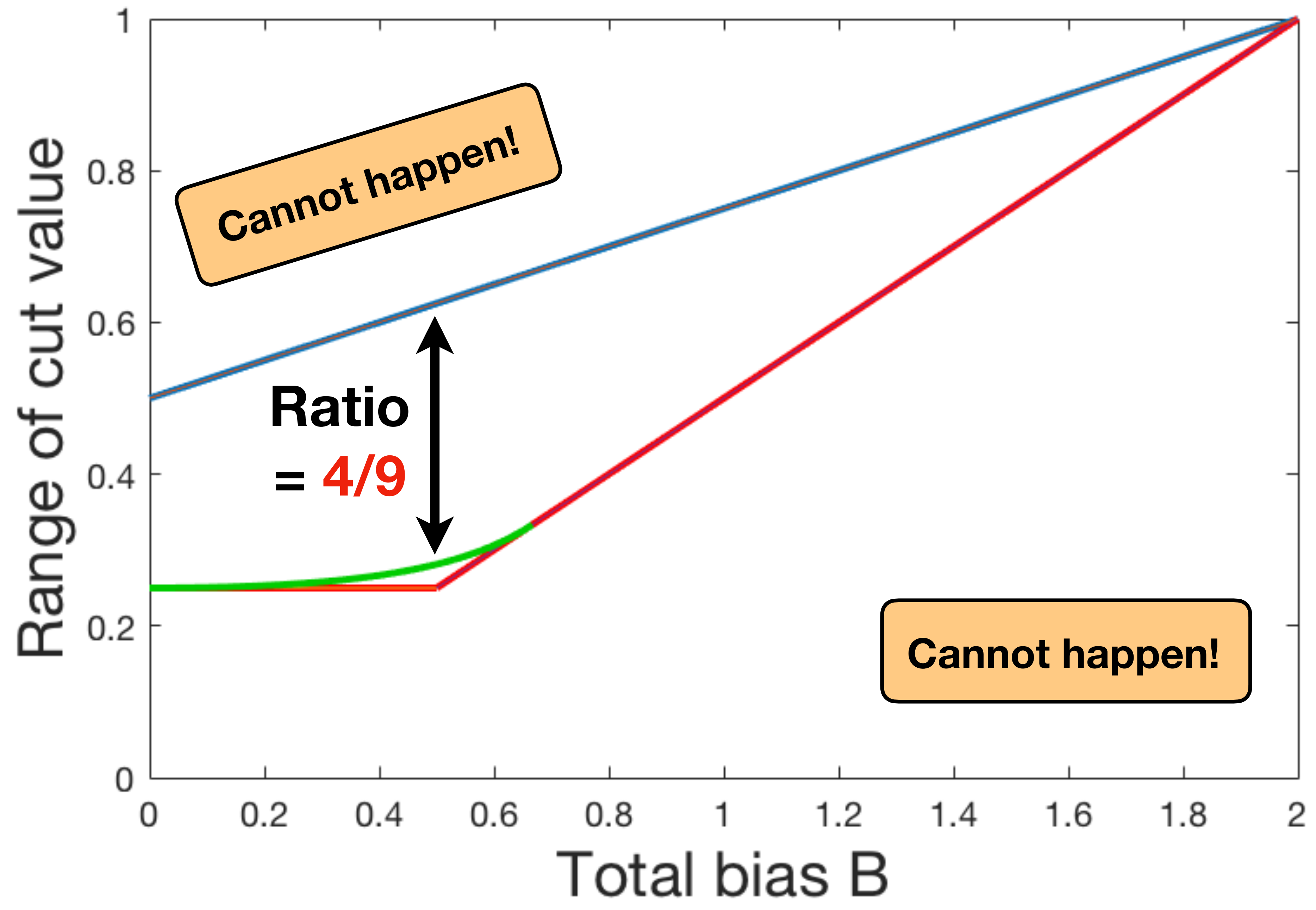
- **Distinguishing statistics:** $Q : I_{\mathscr{F}} \to \mathbb{R}$, e.g., $Q($$) =$ *total bias* of  [GVV17].

Standard Ideas in Proving Streaming Lower bound

The space of Max-CSP($\mathscr{F}$) instances

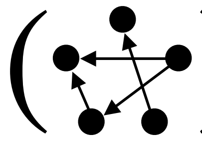
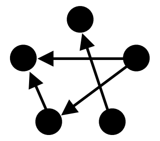
- **Distinguishing statistics:** $Q : I_{\mathscr{F}} \to \mathbb{R}$, e.g., $Q(\;)$ = *total bias* of [GVV17].

- **Algorithmic side:** Statistics Q s.t.
  (i) $Q(\Psi)$ can be estimated in small space;
  (ii) For every q, $\min\limits_{Q(\Psi)=q} \mathsf{val}_\Psi > \alpha \cdot \max\limits_{Q(\Psi)=q} \mathsf{val}_\Psi$.



Standard Ideas in Proving Streaming Lower bound

The space of Max-CSP($\mathcal{F}$) instances

- **Distinguishing statistics:** $Q : I_{\mathcal{F}} \to \mathbb{R}$, e.g., $Q($$)$ = *total bias* of  [GVV17].

- **Algorithmic side:** Statistics Q s.t.

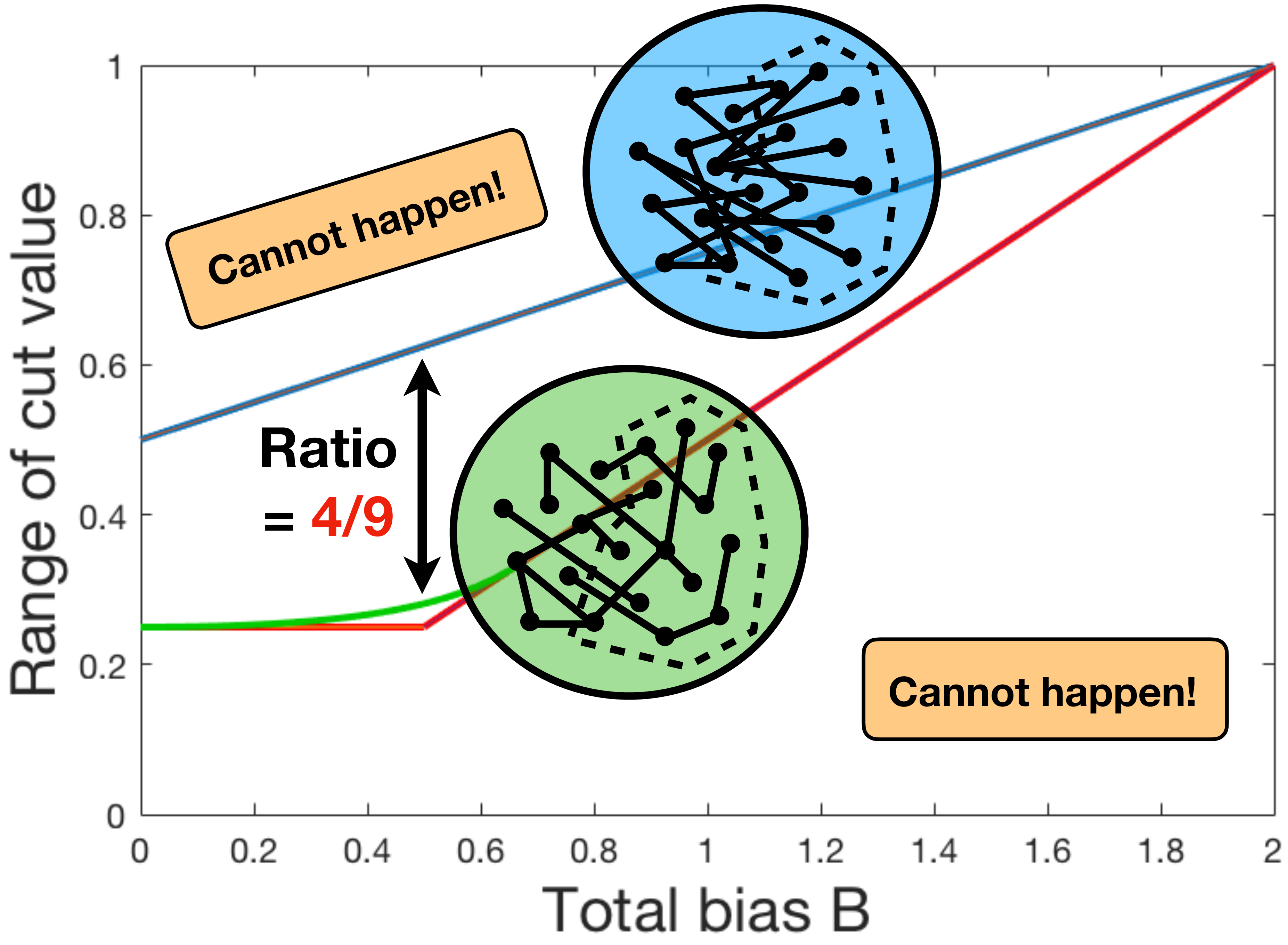
  (i) $Q(\Psi)$ can be estimated in small space;

  (ii) For every q, $\min_{Q(\Psi)=q} \mathrm{val}_\Psi > \alpha \cdot \max_{Q(\Psi)=q} \mathrm{val}_\Psi$.

- **Hardness side:**

  $\mathcal{D}^Y$ over $\operatorname{argmax}_{Q(\Psi)=q} \mathrm{val}_\Psi$ and $\mathcal{D}^N$ over $\operatorname{argmin}_{Q(\Psi)=q} \mathrm{val}_\Psi$

  s.t. no streaming algorithm with $o(\sqrt{n})$ space can distinguish them.



Technical challenges: Understand the extreme instances of the statistics.

Our Key Insight: From Combinatorics to Analysis and Geometry

Our Key Insight: From Combinatorics to Analysis and Geometry

Q: How to systematically find a desirable distinguishing statistics?

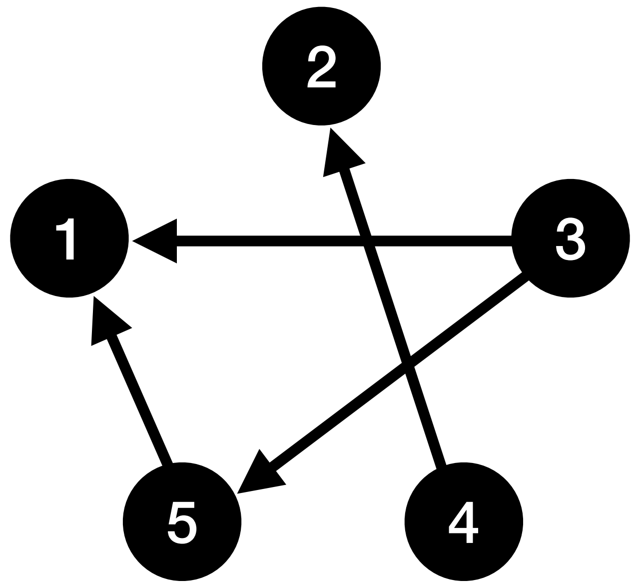
A: Unclear! Previous works used different “combinatorial properties”.

Our Key Insight: From Combinatorics to Analysis and Geometry

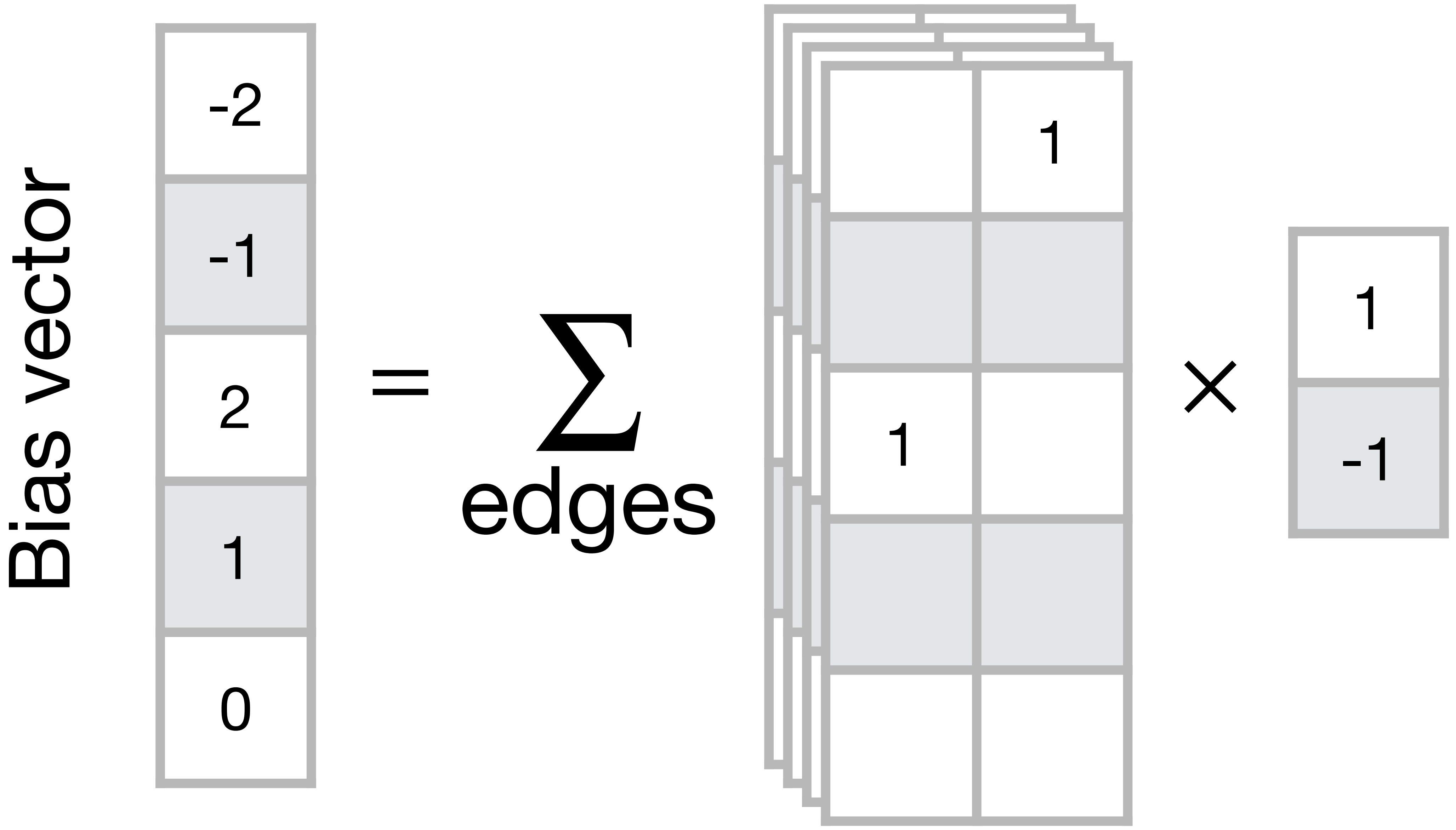
Q: How to systematically find a desirable distinguishing statistics?

A: Unclear! Previous works used different “combinatorial properties”.

Key idea 1: Generalize bias to certain “analytical properties”.



ℓ_1 norm of bias vector can be estimated in $O(\log n)$ space!



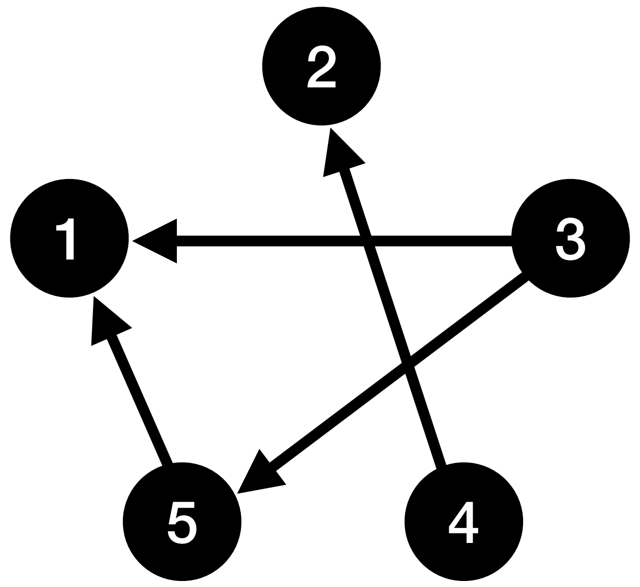
$$\text{Bias vector} \begin{bmatrix} -2 \\ -1 \\ 2 \\ 1 \\ 0 \end{bmatrix} = \sum_{\text{edges}} \begin{bmatrix} & 1 \\ & \\ 1 & \\ & \\ & \end{bmatrix} \times \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

Our Key Insight: From Combinatorics to Analysis and Geometry

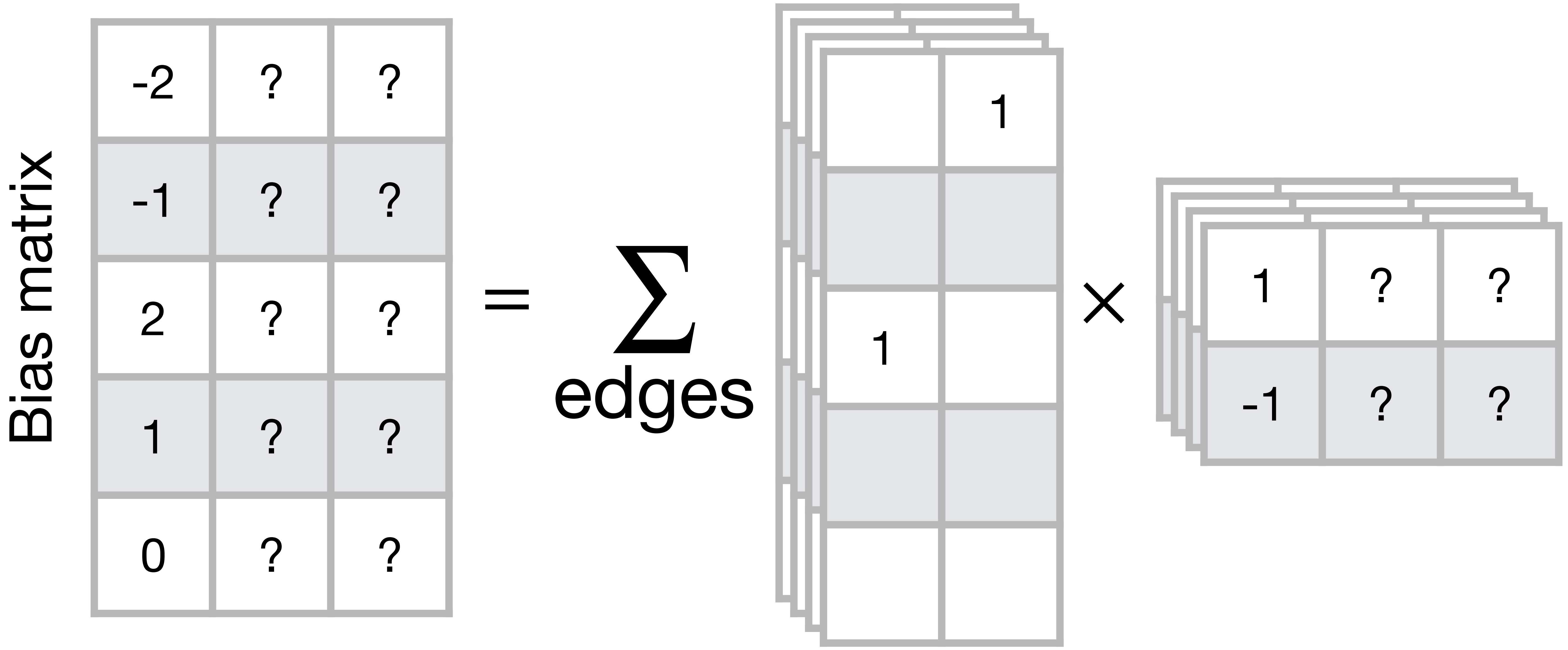
Q: How to systematically find a desirable distinguishing statistics?

A: Unclear! Previous works used different “combinatorial properties”.

Key idea 1: Generalize bias to certain “analytical properties”.



$\ell_{p,q}$ norm of bias vector can be estimated in $O(\log^{O(1)} n)$ space!

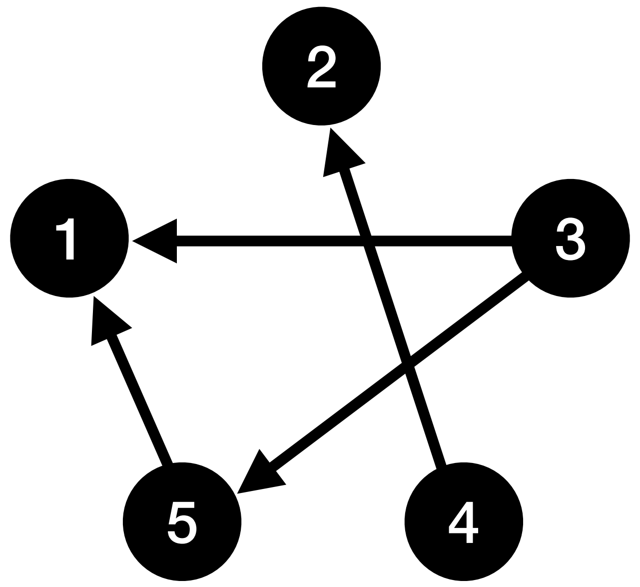


Our Key Insight: From Combinatorics to Analysis and Geometry

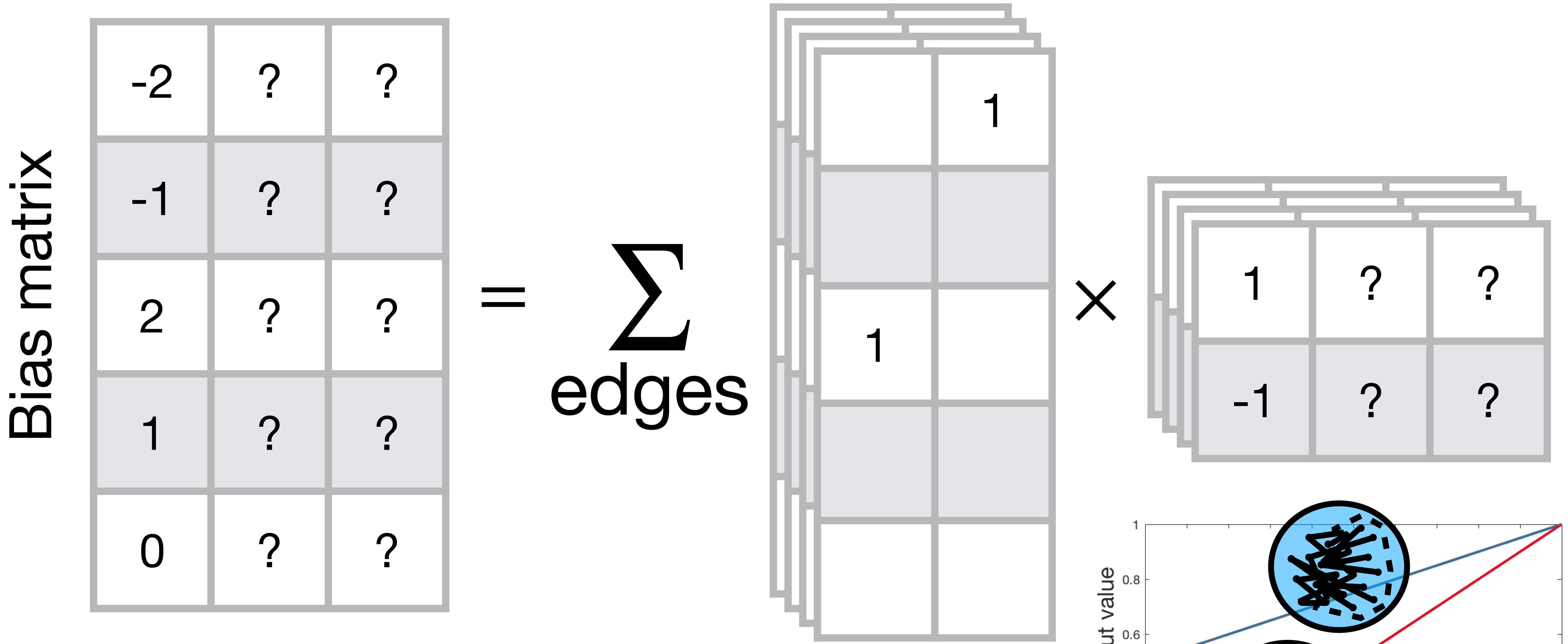
Q: How to systematically find a desirable distinguishing statistics?

A: Unclear! Previous works used different “combinatorial properties”.

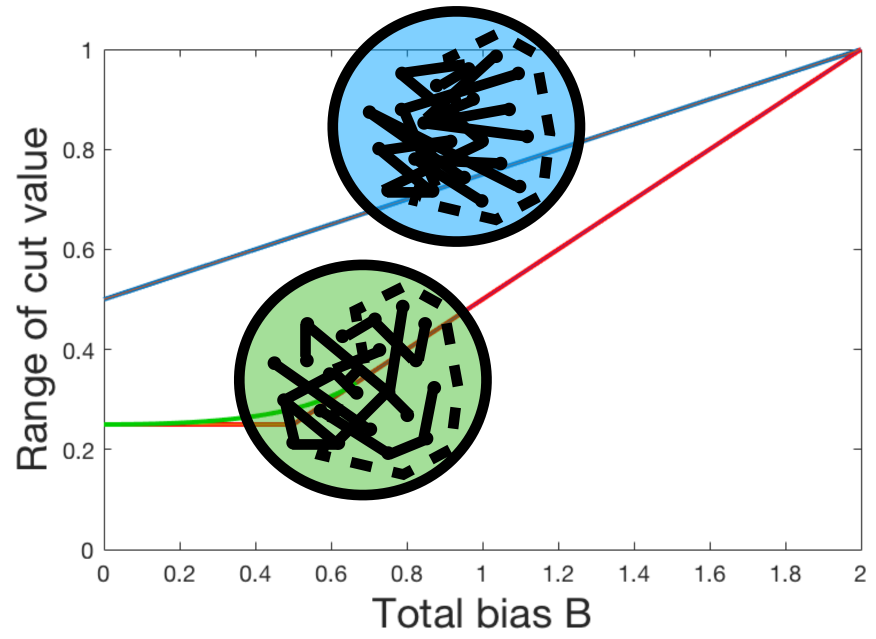
Key idea 1: Generalize bias to certain “analytical properties”.



$\ell_{p,q}$ norm of bias vector can be estimated in $O(\log^{O(1)} n)$ space!



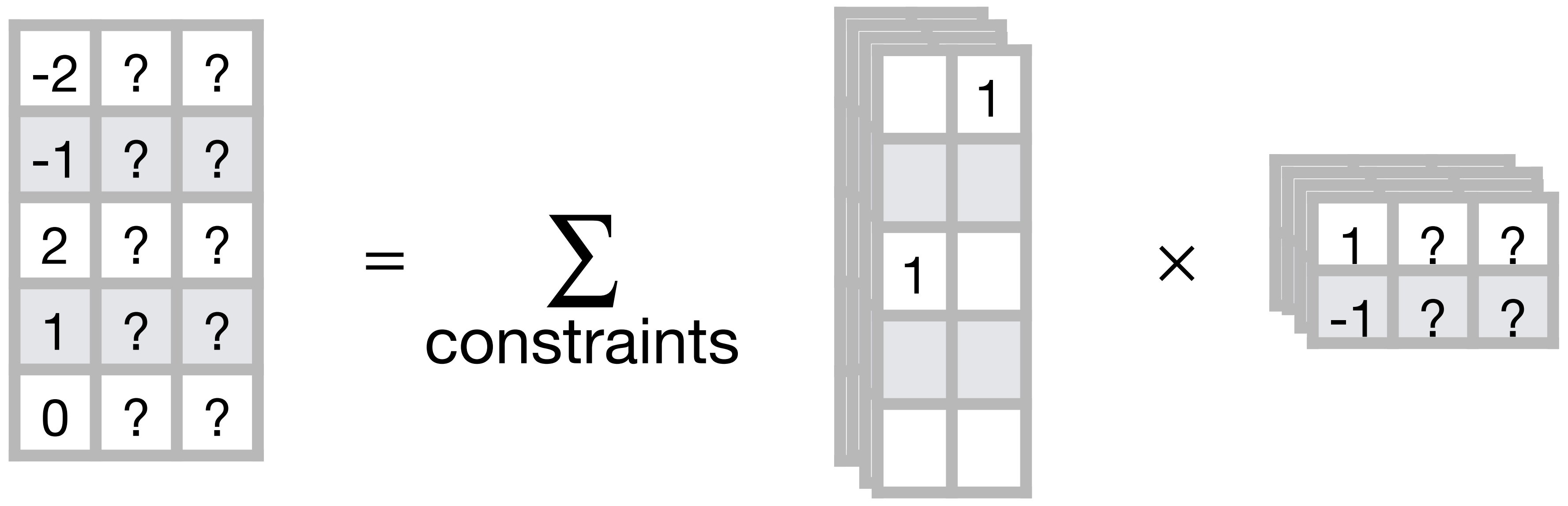
Key idea 2: Use the “geometry of $\mathscr{F}$” to generate tight instances.



Here Come the Convex Sets!

Key idea 1: Generalizing bias to the $\ell_{p,q}$ norm of a bias matrix.

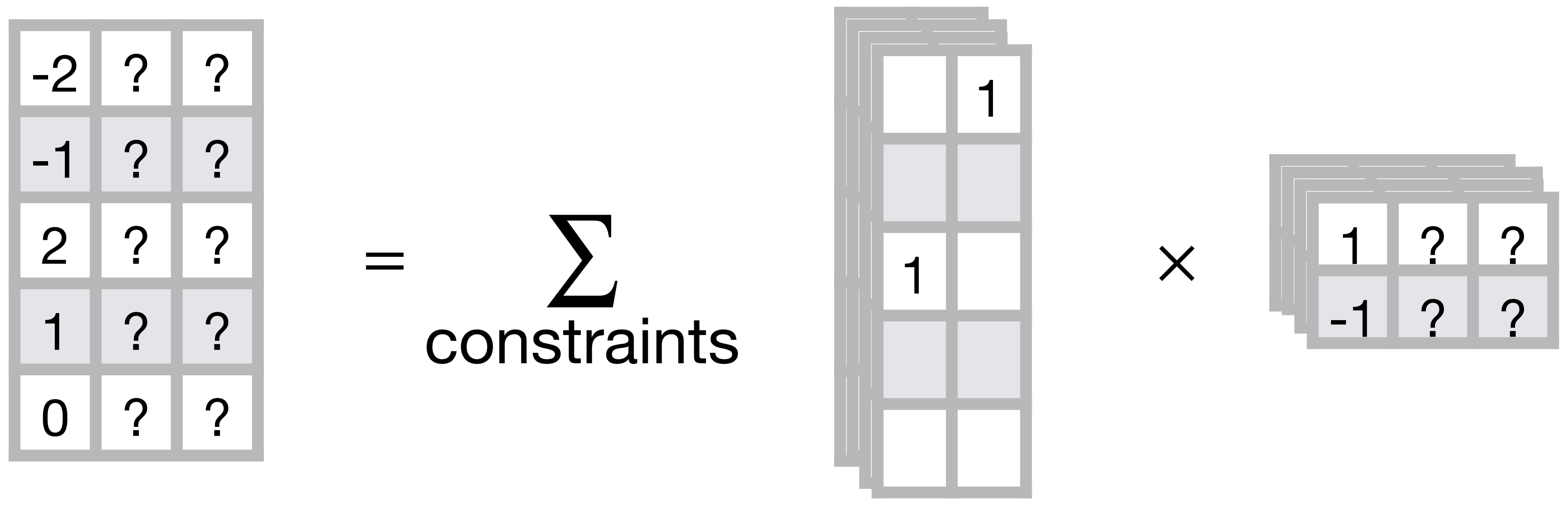
Bias matrix



Key idea 2: The "geometry" of $\mathscr{F}$ in $\mathscr{F} \times [q]^k$.

Here Come the Convex Sets!

Key idea 1: Generalizing bias to the $\ell_{p,q}$ norm of a bias matrix.

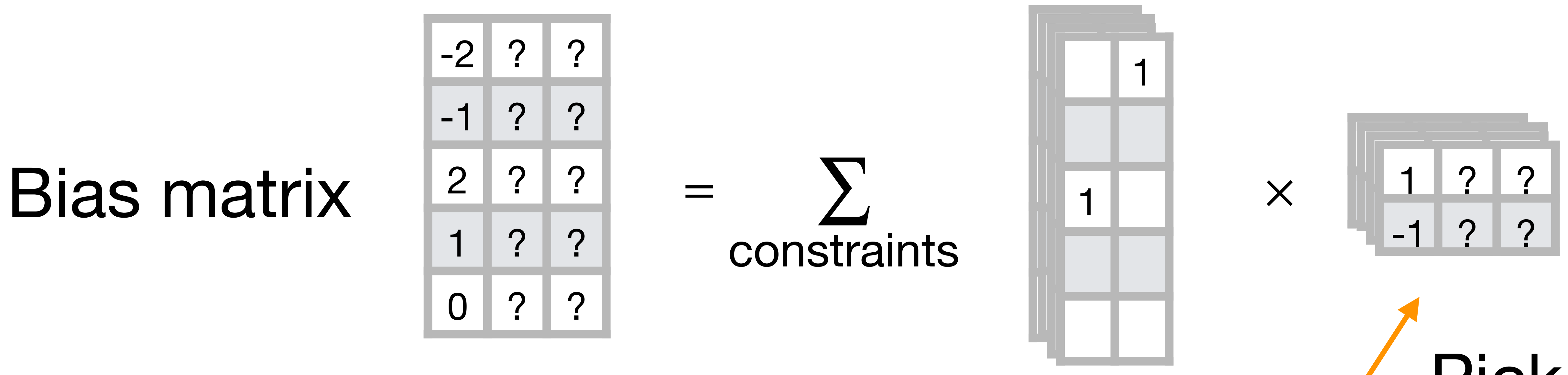
Bias matrix 

Key idea 2: The “geometry” of $\mathscr{F}$ in $\mathscr{F} \times [q]^k$.

Distribution over $\mathscr{F} \times [q]^k$ $\xrightarrow{\text{generates}}$ CSP instances

Here Come the Convex Sets!

Key idea 1: Generalizing bias to the $\ell_{p,q}$ norm of a bias matrix.



Pick the“weight matrices” to be the “**marginal**” of a distribution!

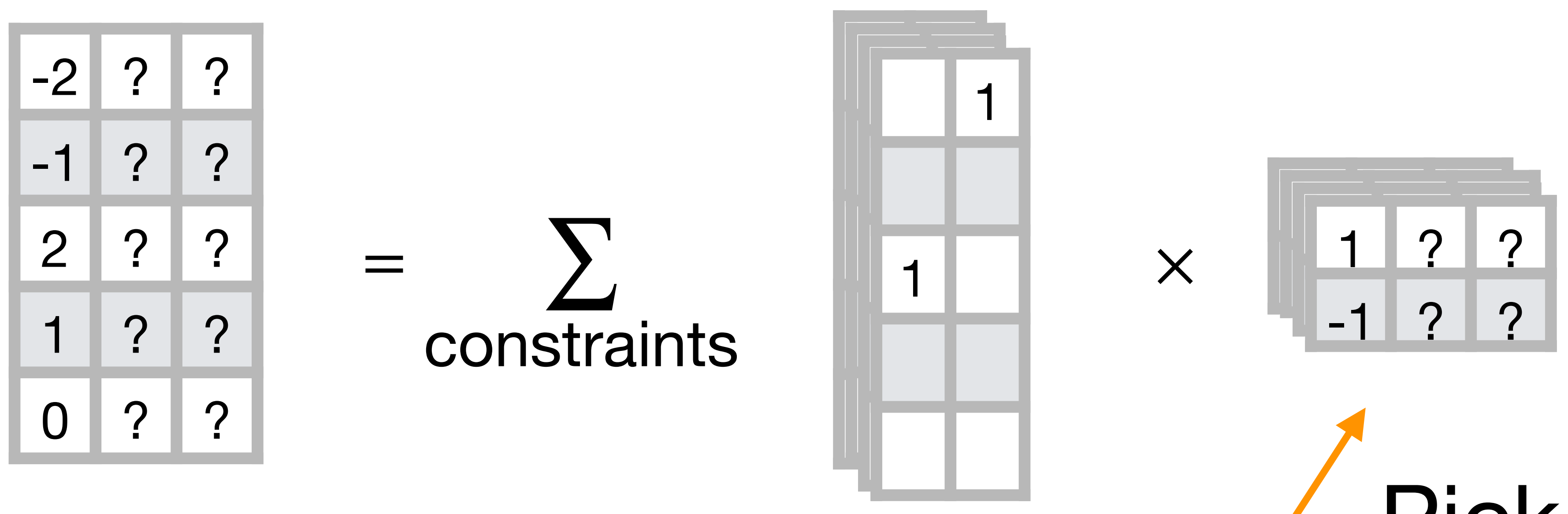
Key idea 2: The “geometry” of $\mathscr{F}$ in $\mathscr{F} \times [q]^k$.

Distribution over $\mathscr{F} \times [q]^k$ $\xrightarrow{\text{generates}}$ CSP instances

Here Come the Convex Sets!

Key idea 1: Generalizing bias to the $\ell_{p,q}$ norm of a bias matrix.

Bias matrix



Pick the"weight matrices" to be the "**marginal**" of a distribution!

Key idea 2: The "geometry" of $\mathscr{F}$ in $\mathscr{F} \times [q]^k$.

Distribution over $\mathscr{F} \times [q]^k \xrightarrow{\text{generates}}$ CSP instances

- K^Y_γ := marginals of distributions generating instances with value at least γ achieved by a **planted assignment.**
- K^N_β := marginals of distributions with value at most β.

Here Come the Convex Sets!

Classification Theorem

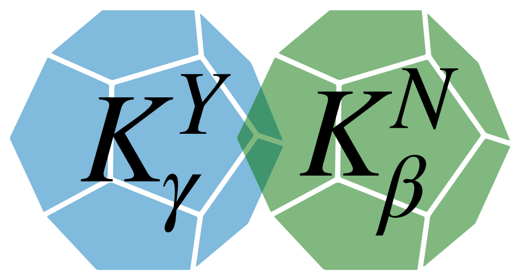
(i) If $K_\gamma^Y(\mathscr{F}) \cap K_\beta^N(\mathscr{F}) = \emptyset$, then (γ, β)-Max-CSP($\mathscr{F}$) can be solved by linear sketches in the dynamic setting using $O(\log^3 n)$ space;

(ii) If $K_\gamma^Y(\mathscr{F}) \cap K_\beta^N(\mathscr{F}) \neq \emptyset$, then $(\gamma - \epsilon, \beta + \epsilon)$-Max-CSP($\mathscr{F}$) by sketching algorithms in the insertion-only setting requires $\Omega(\sqrt{n})$ space $\forall \epsilon > 0$.

- K_γ^Y := marginals of distributions generating instances with value at least γ achieved by a **planted assignment.**
- K_β^N := marginals of distributions with value at most β.

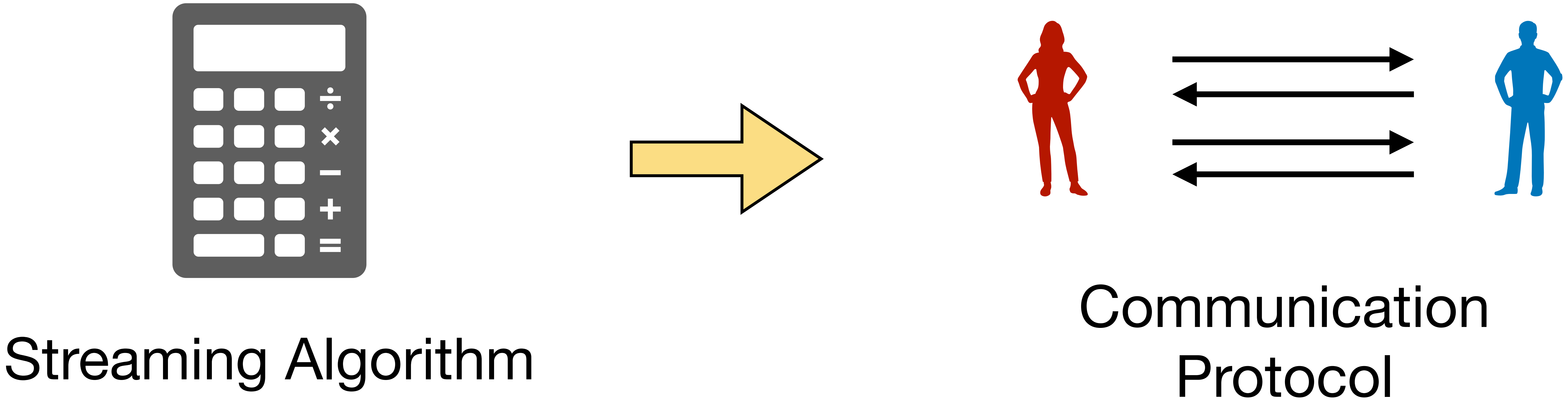
Hardness

$$K_\gamma^Y(\mathcal{F}) \cap K_\beta^N(\mathcal{F}) \neq \emptyset \;\; \Rightarrow \;\; \text{Hard!}$$



Streaming Lower Bounds via Communication Complexity

- Unconditional lower bounds from *communication games*.
- **High-level idea**:

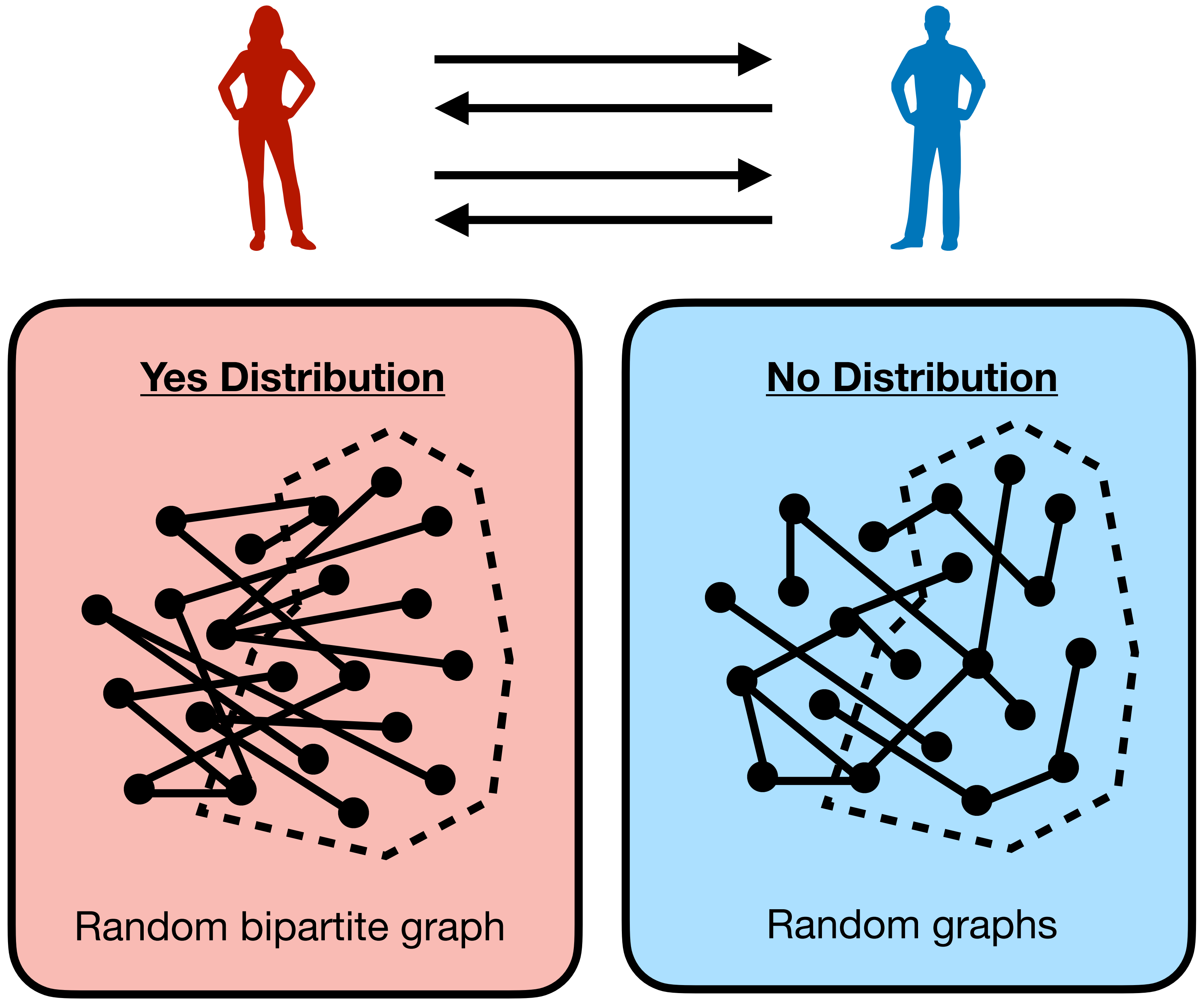


- **Usage**: Alice and Bob insert some inputs to the streaming algorithm and send the “*configuration*” as the message.
- Space complexity of streaming algorithm $\geq$ communication complexity.

A Bird-Eye View of Our Lower Bound Proof

A sequence of reductions from communication games to streaming problems!

Communication Games



Boolean Hidden Matching problem
[GKK+09]

Generalize to $k > 2$

Streaming CSPs

$$\Psi = \left\{(f_i, S_i)\right\}_{i\in[m]}$$

Produce CSP constraints

Simultaneous Signal Detection problem

Increase the # of (hyper)edges

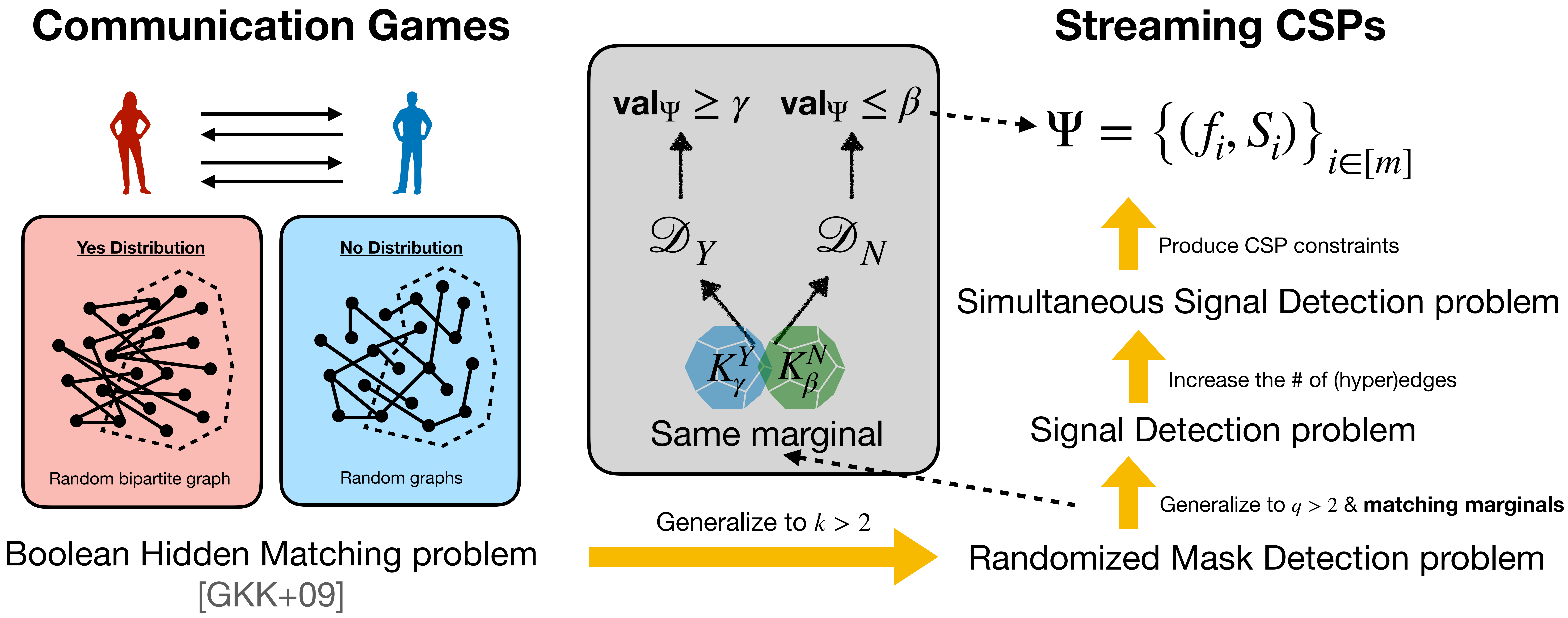
Signal Detection problem

Generalize to $q > 2$ & **matching marginals**

Randomized Mask Detection problem

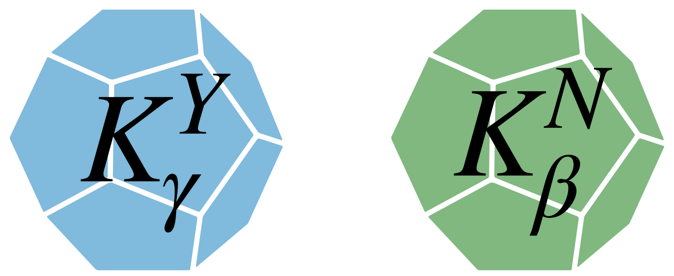
A Bird-Eye View of Our Lower Bound Proof

A sequence of reductions from communication games to streaming problems!

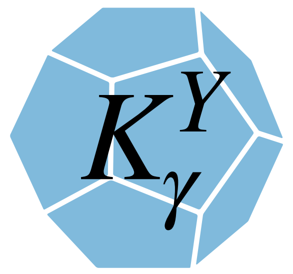


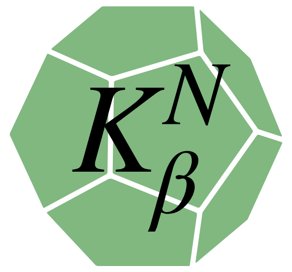
Algorithm

$$K_\gamma^Y(\mathscr{F}) \cap K_\beta^N(\mathscr{F}) = \emptyset \quad \Rightarrow \quad \exists \text{ Algorithm!}$$



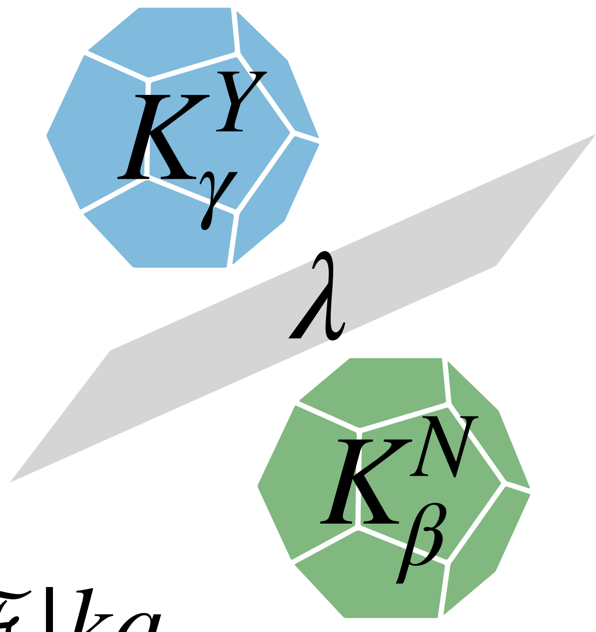
Key Ideas and a Sketch of the Analysis for our Algorithm





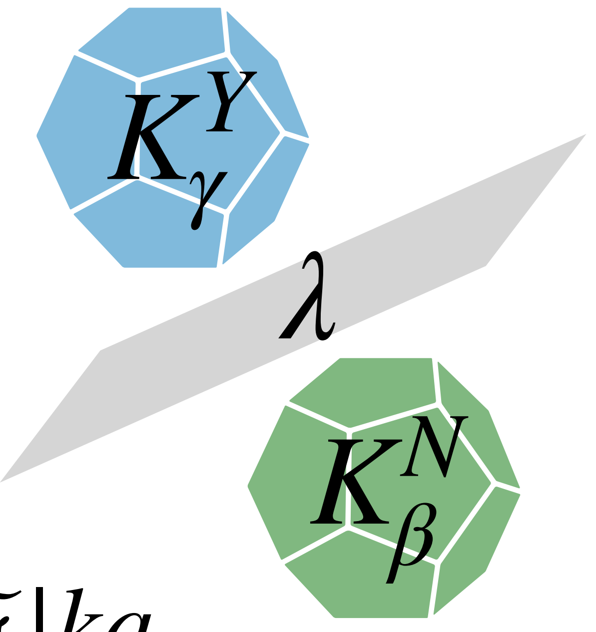
- **Recall**: We generalize the bias vector of Max-DICUT to bias matrix.

Key Ideas and a Sketch of the Analysis for our Algorithm



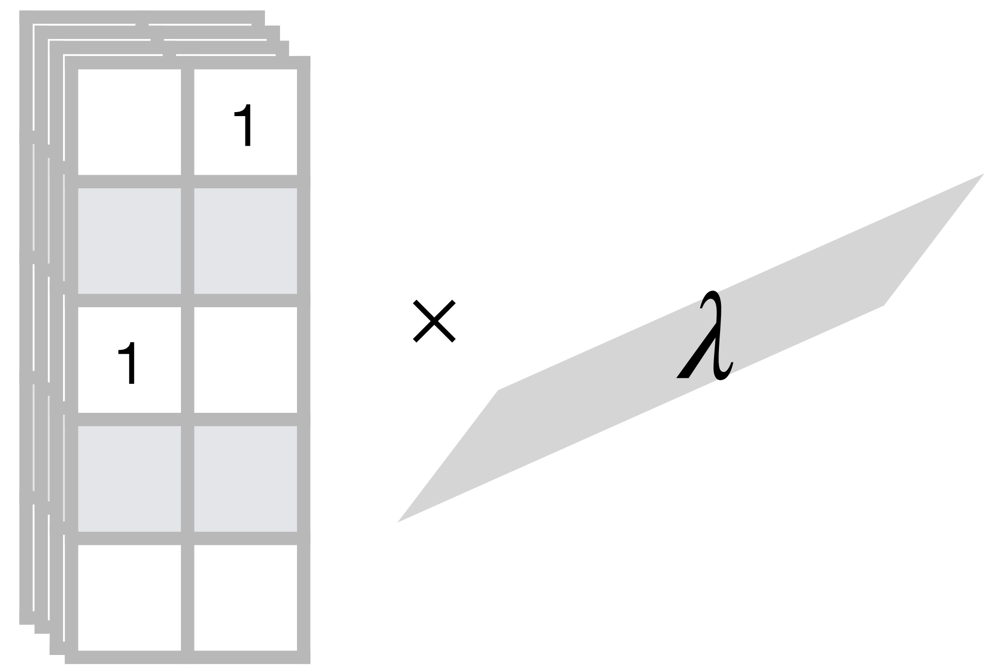
- **Recall**: We generalize the bias vector of Max-DICUT to bias matrix.
- **Observation:** K_γ^Y and K_β^N are convex $\Rightarrow \exists$ separating vector $\lambda \in \mathbb{R}^{|\mathscr{F}|kq}$.

Key Ideas and a Sketch of the Analysis for our Algorithm



- **Recall**: We generalize the bias vector of Max-DICUT to bias matrix.
- **Observation:** K_γ^Y and K_β^N are convex $\Rightarrow \exists$ separating vector $\lambda \in \mathbb{R}^{|\mathcal{F}|kq}$.
- The $\ell_{1,\infty}$ norm of the bias matrix using λ is a good distinguishing statistics!

$$Q(\Psi) = \frac{1}{m} \sum_{i\in[m]}$$



(2) Take the ℓ_1 norm

(1) Take the max

Desired properties of $Q(\Psi)$:

(i) $Q(\Psi)$ can be estimated in $O(\log^3 n)$ space;
- Tool from the streaming literature [AKO11].

(ii) For every q, $\min_{Q(\Psi)=q} \mathrm{val}_\Psi > \alpha \cdot \max_{Q(\Psi)=q} \mathrm{val}_\Psi$.
- A direct probability analysis utilizing the structure of S_γ^Y, S_β^N.

This might look like coming out of nowhere…
but it's actually a very natural choice if knowing the previous analysis!

Conclusion & Future Directions

Conclusion

Classification Theorem

For every finite $q, k \in \mathbb{N}$, every $\mathcal{F} \subset \{f : [q]^k \to \{0,1\}\}$, and every $0 \leq \beta < \gamma \leq 1$, the following hold.

(i) If $K_\gamma^Y(\mathcal{F}) \cap K_\beta^N(\mathcal{F}) = \emptyset$, then (γ, β)-Max-CSP($\mathcal{F}$) can be solved by linear sketches in the dynamic setting using $O(\log^3 n)$ space;

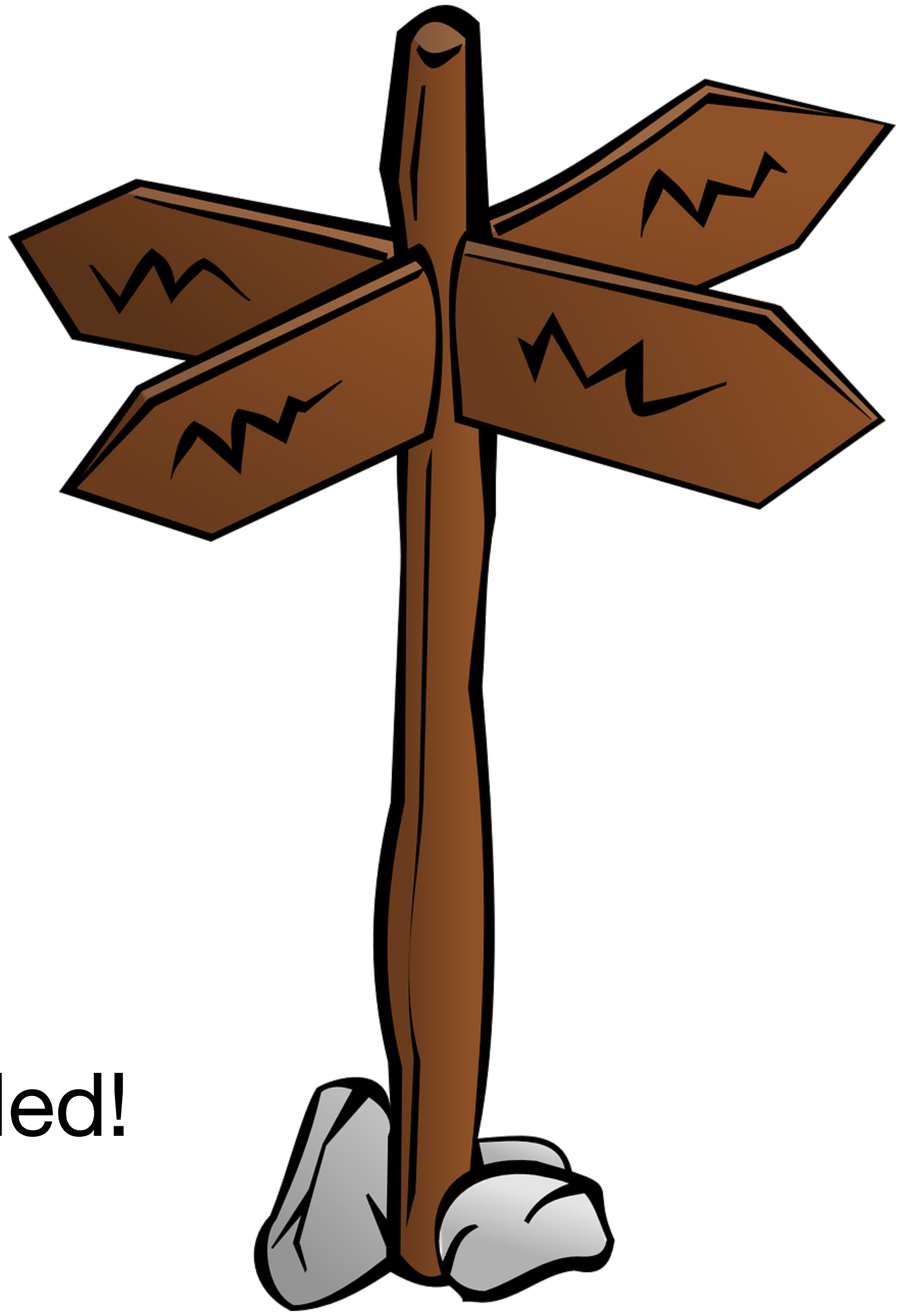
(ii) If $K_\gamma^Y(\mathcal{F}) \cap K_\beta^N(\mathcal{F}) \neq \emptyset$, then $(\gamma - \epsilon, \beta + \epsilon)$-Max-CSP($\mathcal{F}$) by sketching algorithms in the insertion-only setting requires $\Omega(\sqrt{n})$ space $\forall \epsilon > 0$.

Main technical contributions: (i) Identifying the right convex sets and the communication games, (ii) design a sequence of cool reductions.

What I Skipped

- How to establish the lower bound for uniform marginal case?
  - The standard Fourier analysis boils down to a combinatorial counting problem.
- How does the polarization technique work?
  - For each marginal μ, there's a polarized distribution $\mathscr{D}_\mu$ s.t. for every $\mathscr{D}$ with $\mu(\mathscr{D}) = \mu$, there's a finite path a indistinguishable distributions connecting $\mathscr{D}$ and $\mathscr{D}_\mu$.
- How to increase the # of (hyper)edges?
  - We only know how to do this via data processing inequality or through simultaneous communication model where the former can only handle uniform marginal and the latter only gives lower bound against sketching algorithms.
- Why the lower bounds only hold for sketching algorithms?
  - In fact, the CSP distributions generated by SD problem **is distinguishable** by a streaming algorithm with logarithmic space when the marginal is not uniform! New communication game and idea are needed.
- The analysis of our linear sketches?
  - It's mainly standard probabilistic analysis and heavily relying on our good choices of the convex sets.
- Examples of the instantiation of our classification theorem?
  - See our paper for examples on Max-DICUT, Max-UG, and Max-Coloring!

Future Directions



Multi-pass lower bound

- Closer to bounded space model
- Technically challenging

Insertion-only + general streaming lower bound

- New communication game is needed!
- Any possible separation!?

Linear space lower bound

- Full classification in linear space?
- Separation from $\sqrt{n}$ space?

Applications & Instantiations of our classification theorem

- Simplify our characterization in interesting cases?
- The communication games could be of interest in other area?

Different streaming models

More on the convex sets

Appendix

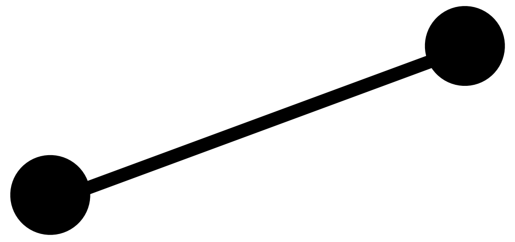
More on the hardness side

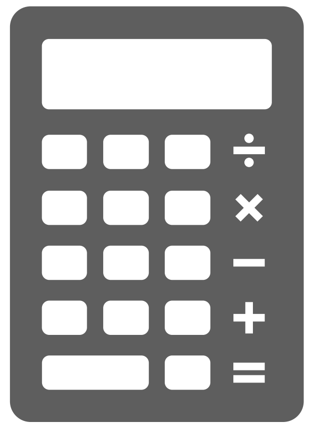
Different Streaming Models

Various Streaming Settings

Harder	**Algorithm**	Easier
Easier	**Hardness**	Harder

Input Models | **Streaming Models**





Insertion-only	+,+,+,+,+,…	Linear sketches	$M(x)$ (with $M(x \circ y) = M(x) + M(y)$)
Strict turnstile (a.k.a. Dynamic)	+,+,-,+,-,…	Sketches	$C(x)$ (with $C(x \circ y) = f(C(x), C(y))$)
Turnstile	-,-,+,-,+,…	Streaming algorithm	Anything!

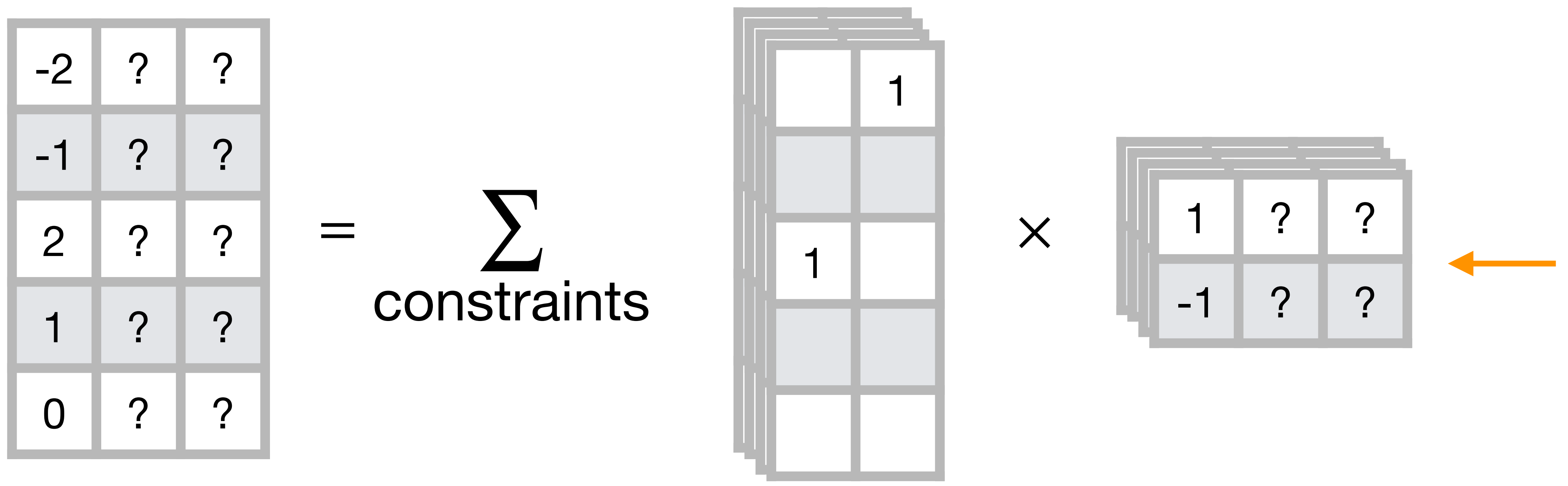
More on the Convex Sets

Hardness Side: Connecting Bias Matrix to Constraint Space

The intuition of using constraint space came from the hardness proof which will be explained later…

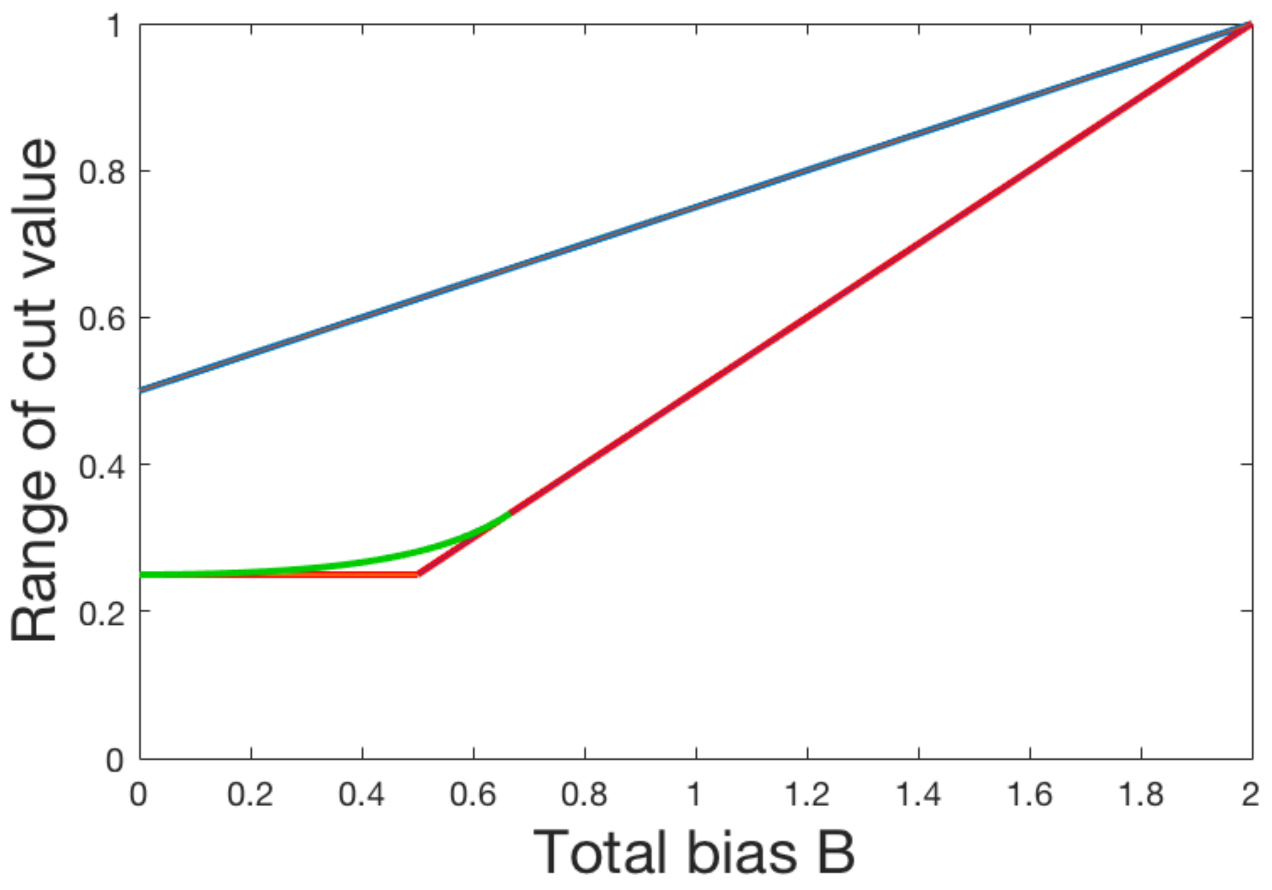
- Constraint space: $\mathscr{F} \times [q]^k$ containing tuples of the form $(f, \mathbf{a})$.

Bias matrix



Pick the dimension of these "weight matrices" to be $k \times q$

- An assignment $\mathbf{b} \in [q]^{kq}$ has value $\mathsf{val}(f, \mathbf{a})(\mathbf{b}) = f(b_{1,a_1}, b_{2,a_2}, \ldots, b_{k,a_k})$.
  - (Planted assignment) Let $\mathbb{I} \in [q]^{kq}$ with $\mathbb{I}_{i,\sigma} = \sigma$.
  - (Random assignment) Sample $\mathbf{b}$ with $b_{i,\sigma} \sim \mathscr{P}_\sigma$ for every $\mathscr{P}_\sigma$ over $[q]$.
- Yes/No distributions over the constraint space:
  - $S^Y_\gamma(\mathscr{F}) := \left\{ \mathscr{D} \mid \mathbb{E}_{(f,\mathbf{a} \sim \mathscr{D})}[\mathsf{val}(f, \mathbf{a})(\mathbb{I})] \geq \gamma \right\}$.
  - $S^N_\beta(\mathscr{F}) := \left\{ \mathscr{D} \mid \mathbb{E}_{(f,\mathbf{a}) \sim \mathscr{D}}[\mathbb{E}_{\mathbf{b}, b_{i,\sigma} \sim \mathscr{P}_\sigma}[\mathsf{val}(f, \mathbf{a})(\mathbf{b})]] \leq \beta, \forall \mathscr{P}_\sigma \right\}$.



Here Come the Convex Sets!

- Use the Yes/No distributions $\mathscr{D}^Y$ & $\mathscr{D}^N$ to generate boundary instances.
- **Key hardness idea:**

"Marginals" of $\mathscr{D}^Y$ and $\mathscr{D}^N$ match $\Rightarrow$ Can't distinguish in $o(\sqrt{n})$ space.

- For a distribution over $\mathscr{F} \times [q]^k$, define marginal $\mu(\mathscr{D}) = (\mu_{f,i,\sigma}) \in \mathbb{R}^{|\mathscr{F}|kq}$ by

$$\mu_{f,i,\sigma} = \Pr_{(g,\mathbf{a})\sim\mathscr{D}}[g = f \text{ and } a_i = \sigma].$$

- $K_\gamma^Y(\mathscr{F}) := \left\{\mu(\mathscr{D}) \mid \mathscr{D} \in S_\gamma^Y(\mathscr{F})\right\}$.
- $K_\beta^N(\mathscr{F}) := \left\{\mu(\mathscr{D}) \mid \mathscr{D} \in S_\beta^N(\mathscr{F})\right\}$.

Classification Theorem

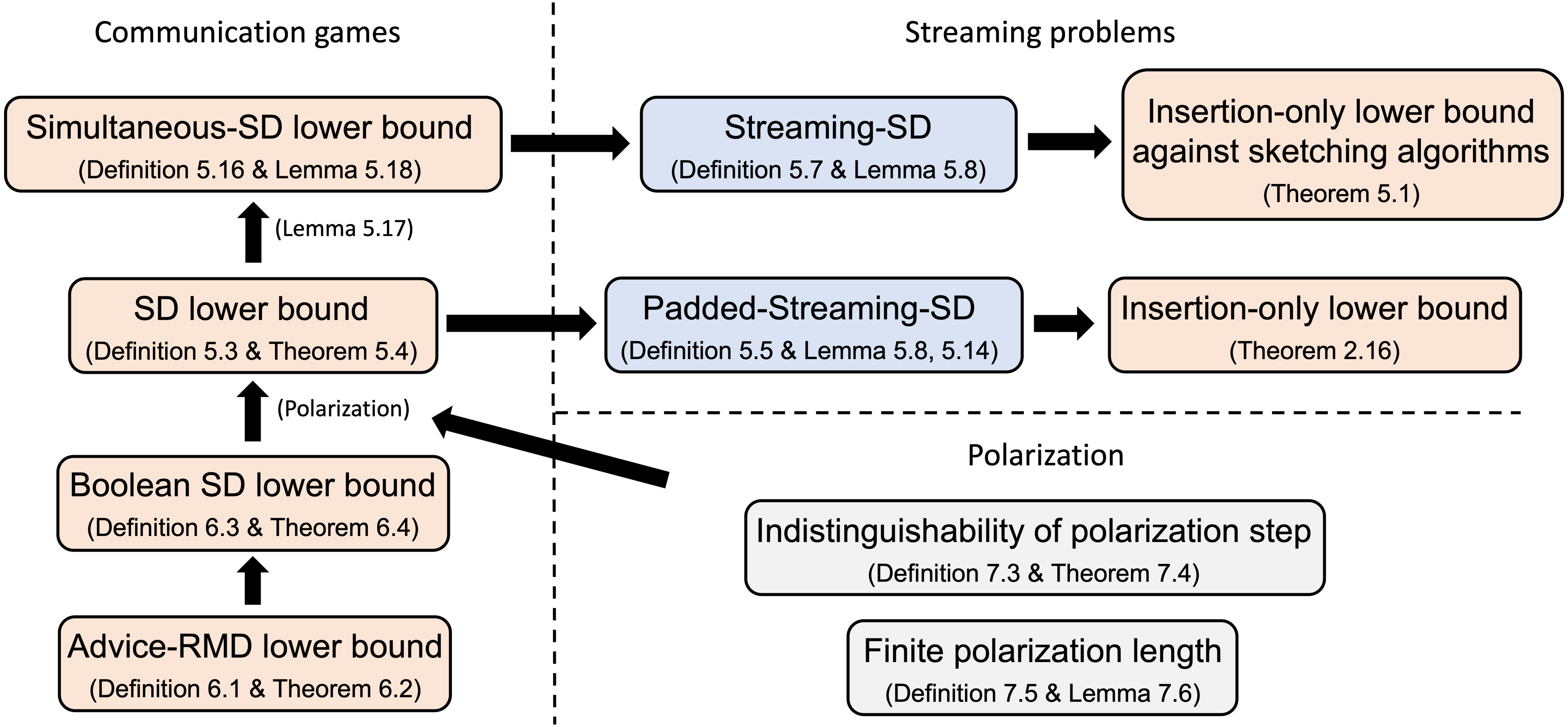
(i) If $K_\gamma^Y(\mathscr{F}) \cap K_\beta^N(\mathscr{F}) = \varnothing$, then (γ, β)-Max-CSP($\mathscr{F}$) can be solved by linear sketches in the dynamic setting using $O(\log^3 n)$ space;

(ii) If $K_\gamma^Y(\mathscr{F}) \cap K_\beta^N(\mathscr{F}) \neq \varnothing$, then $(\gamma - \epsilon, \beta + \epsilon)$-Max-CSP($\mathscr{F}$) by sketching algorithms in the insertion-only setting requires $\Omega(\sqrt{n})$ space $\forall \epsilon > 0$.

More on the Hardness Side

Structure of the Our Lower Bound Proof

A sequence of reductions from communication games to streaming problems!



* To make the reductions work, we work on the "advice" version of the games.

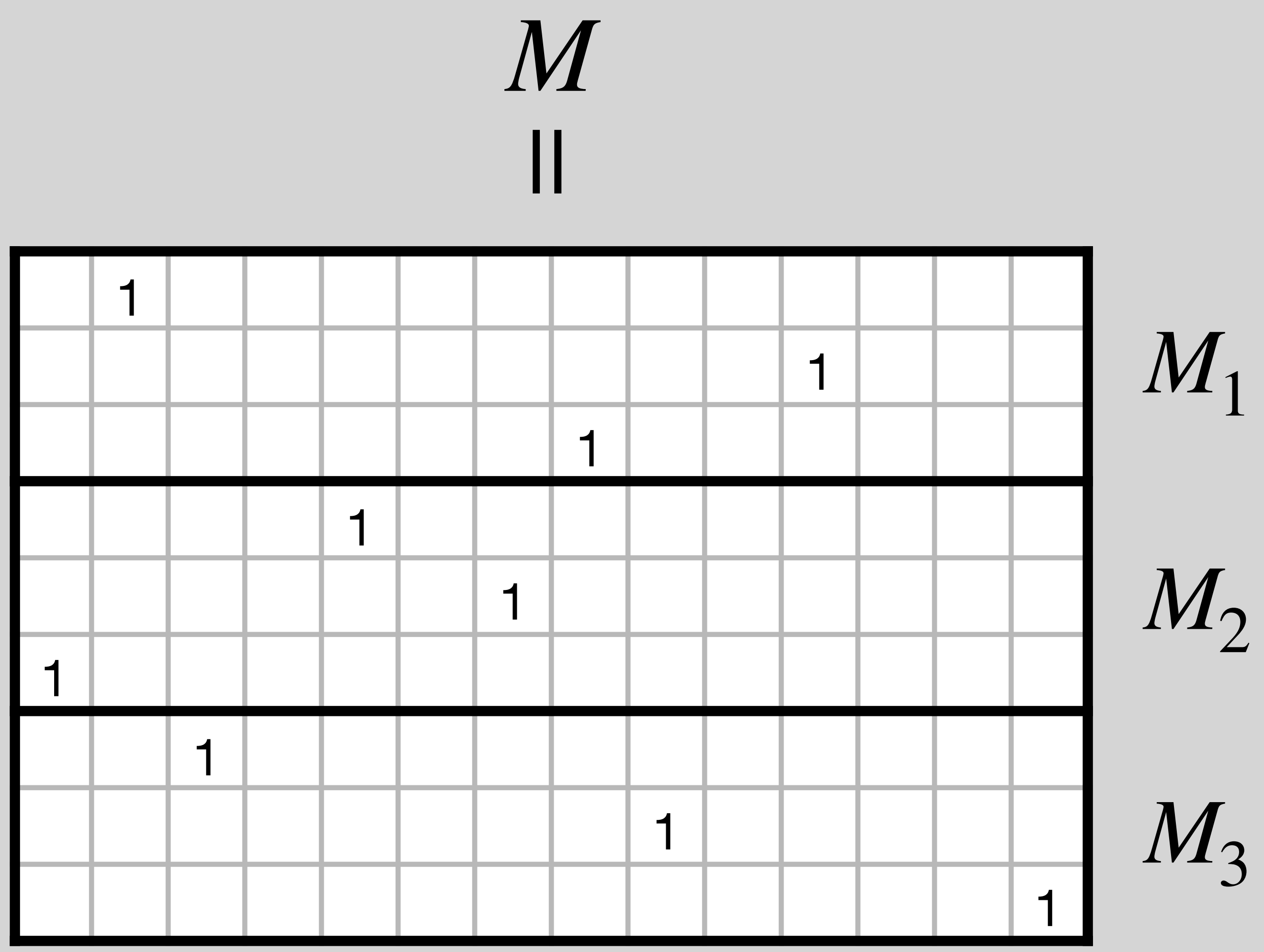
$(\mathscr{D}^Y, \mathscr{D}^N)$-Signal Detection (SD) Problem

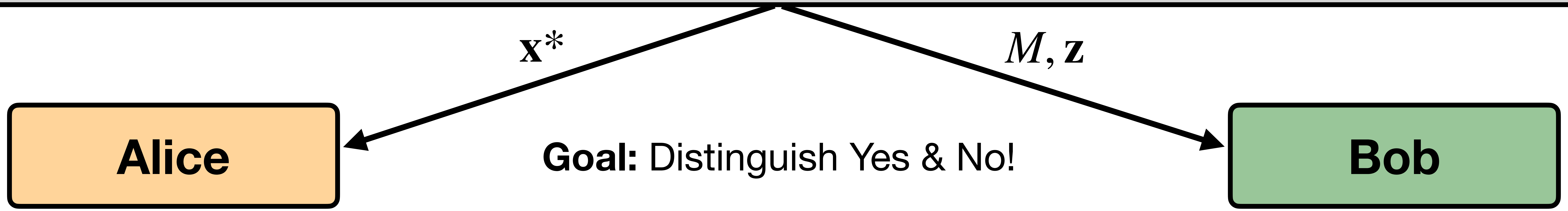
How communication games relate to streaming CSPs!

Let $n, k, q \in \mathbb{N}$, $\alpha \in (0,1)$ small enough, $\mathscr{D}^Y, \mathscr{D}^N$ distributions over $[q]^k$.

Generator

1. Sample $\mathbf{x}^* \sim$ Unif$([q]^n)$.
2. Sample a partial permutation matrix $M \in \{0,1\}^{k\alpha n \times n}$.
3. Sample $\mathbf{b} = (\mathbf{b}(1), \mathbf{b}(2), \ldots, \mathbf{b}(\alpha n))$ from one of the following:
   - (**Yes**) each $\mathbf{b}(i) \in [q]^k$ is sampled from $\mathscr{D}^Y$.
   - (**No**) each $\mathbf{b}(i) \in [q]^k$ is sampled from $\mathscr{D}^N$.
4. Let $\mathbf{z} \in \{0,1\}^{\alpha n}$ be given by $z_i = 1$ iff $M_i \mathbf{x}^* = \mathbf{b}(i)$.





Goal: Distinguish Yes & No!

* To make the reductions work, we work on the "advice" version of the games.

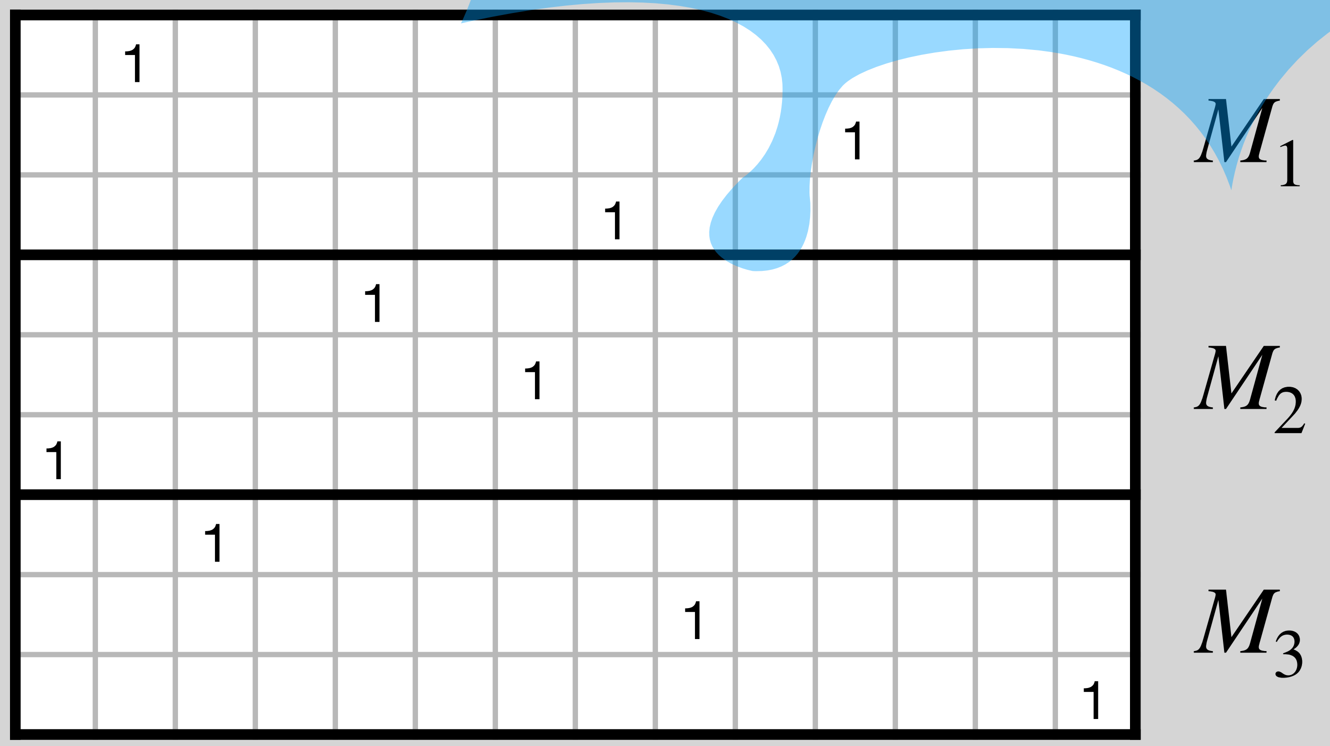
Reduce SD to Streaming CSP

Reject sampling!

Let $n, k, q \in \mathbb{N}$, $\alpha \in (0,1)$ small enough, $\mathscr{D}$ a distribution over $[q]^k$.

1. Sample $\mathbf{x}^* \sim$ Unif($[q]^n$).
2. Sample a partial permutation matrix $M \in \{0,1\}^{k\alpha n \times n}$.
3. Sample $\mathbf{b} = (\mathbf{b}(1), \mathbf{b}(2), \ldots, \mathbf{b}(\alpha n))$ s.t. each $\mathbf{b}(i) \sim \mathscr{D}$.
4. Let $\mathbf{z} \in \{0,1\}^{\alpha n}$ be given by $z_i = 1$ iff $M_i \mathbf{x}^* = \mathbf{b}(i)$.

$M =$



$\mathbf{x}^*$

Alice

$M, \mathbf{z}$

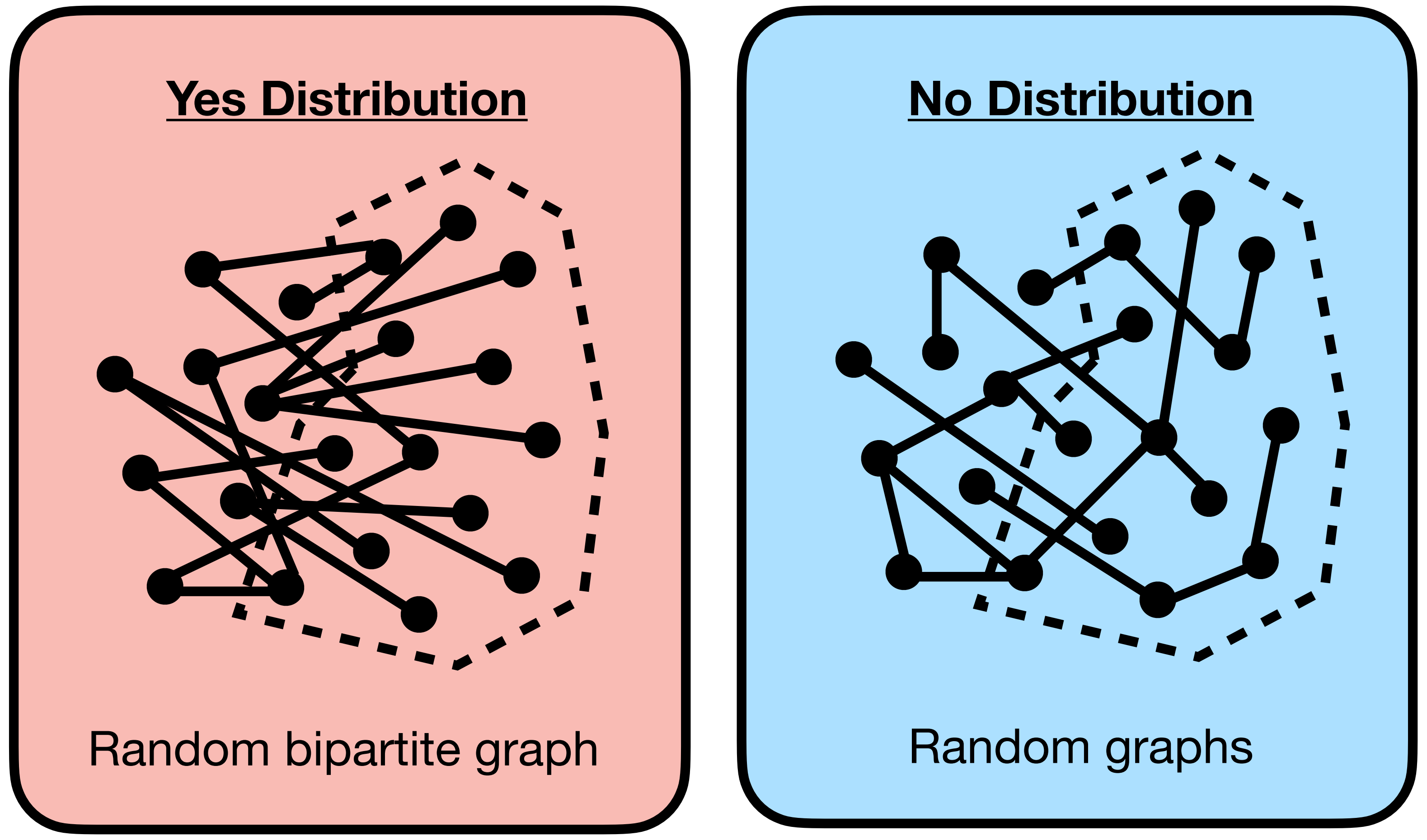
Bob

For each $i \in [\alpha n]$ s.t. $z_i = 1$, insert a constraint (f, M_i) to the streaming algorithm!

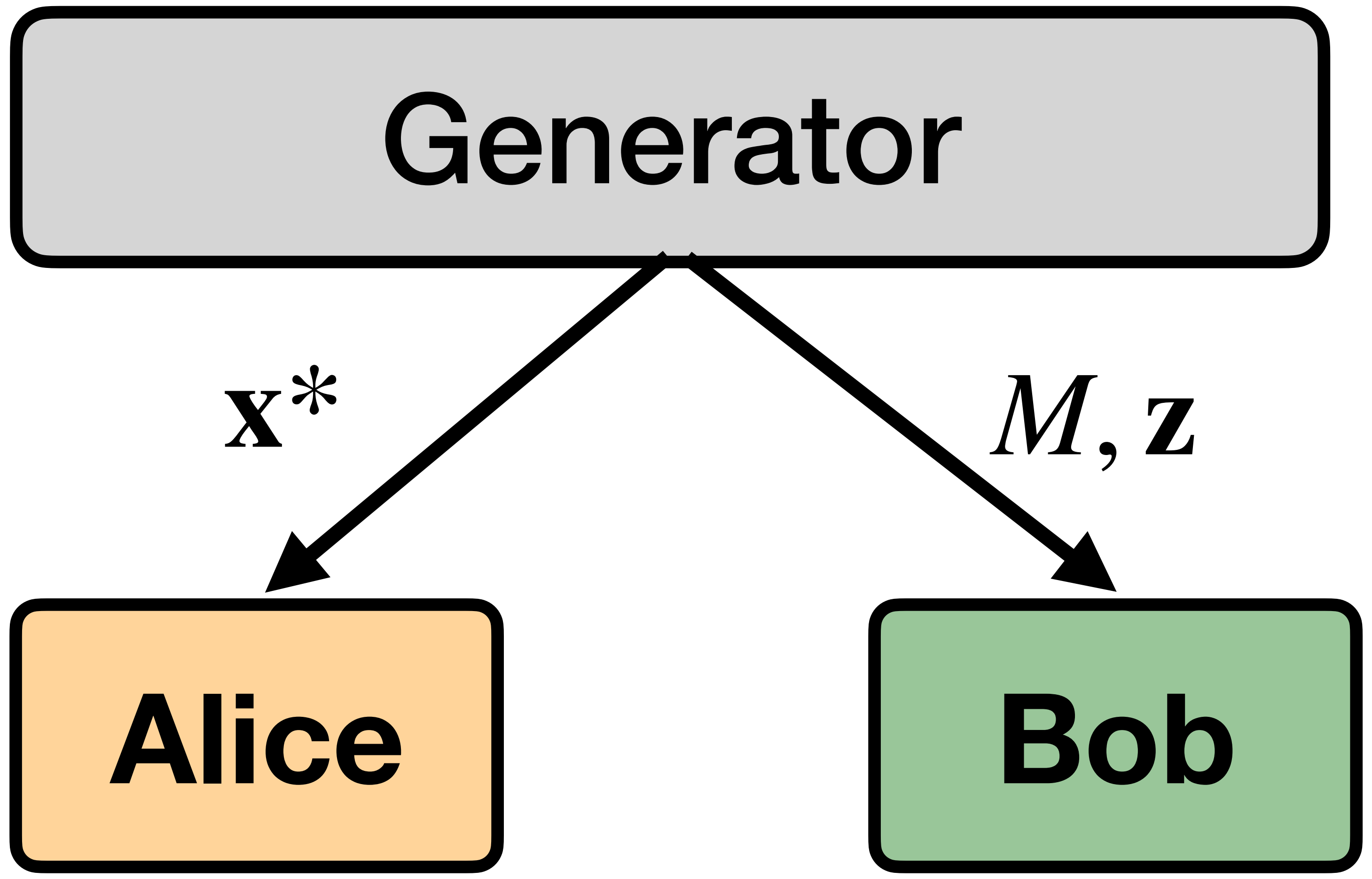
Observe that when setting the assignment to $\mathbf{x}^*$, the distribution outputted by Bob follows $(f, M\mathbf{x}^*) \sim (f, \mathbf{b})$ where $\mathbf{b} \sim \mathscr{D}$.

When Are the Communication Games Hard?

Boolean Hidden Matching problem
[GKK+09]



$(\mathscr{D}^Y, \mathscr{D}^N)$-SD
with $\boxed{\mu(\mathscr{D}^Y) = \mu(\mathscr{D}^N)}$
Corresponds to $K_\gamma^Y \cap K_\beta^N \neq \emptyset$!



The game is hard when both distributions have uniform marginal. (Analyzing the total variation distance via Fourier analysis.)

We develop a polarization technique for reduction between communication games while keeping the marginals of the two distributions the same