Oblivious-Transfer Complexity of Noisy Coin-Toss via Secure Zero Communication Reductions

Saumya Goyal[1], Varun Narayanan[2], Manoj Prabhakaran[3]

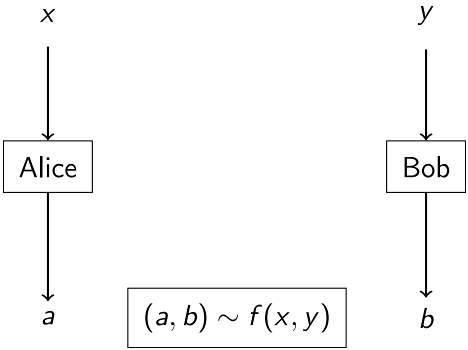
[1]Stanford University, [2]Technion, [3]IIT Bombay

November 2022

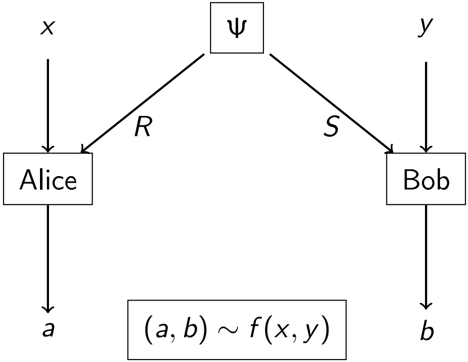
Outline

Table of Contents

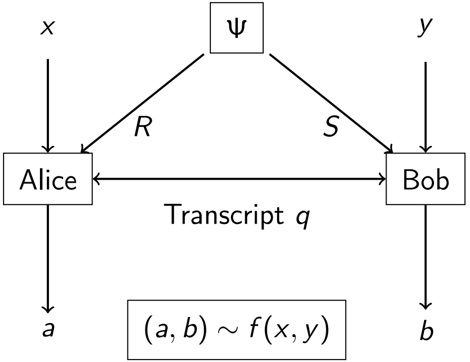
OT complexity



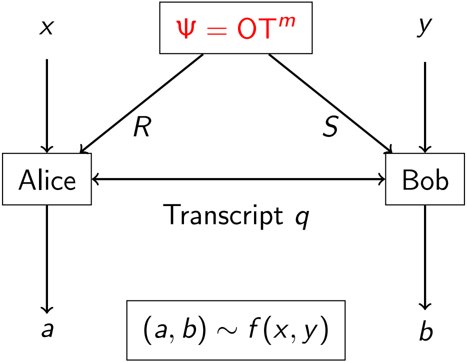
OT complexity



OT complexity

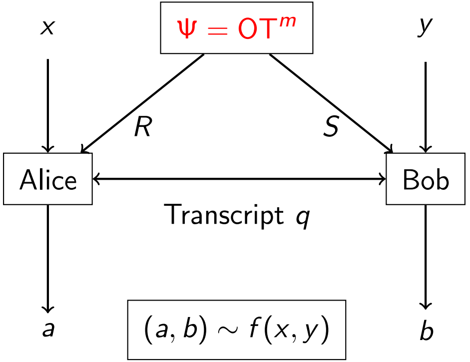


OT complexity



$\mathrm{OT} = (\theta_0, \theta_1)$ to Alice, (b, θ_b) to Bob

OT complexity



$\mathrm{OT} = (\theta_0, \theta_1)$ to Alice, (b, θ_b) to Bob

$|f|_{\mathrm{OT}}$ = minimum number of OTs required by an information-theoretically (semi-honest, perfectly) secure protocol for f

OT complexity

$|f|_{\text{OT}}$ = minimum number of OTs required by an information-theoretically (semi-honest, perfectly) secure protocol for f

- Well-studied since [Kilian'88] showed OT is complete
- But formidable barriers to proving super-linear lower bounds
  - In particular, lower bound for $|f|_{\text{OT}}$ is a lower bound for circuit complexity of f
  - For deterministic functions

OT complexity

$|f|_{\mathrm{OT}}$ = minimum number of OTs required by an information-theoretically (semi-honest, perfectly) secure protocol for f

- Well-studied since [Kilian'88] showed OT is complete
- But formidable barriers to proving super-linear lower bounds
  - In particular, lower bound for $|f|_{\mathrm{OT}}$ is a lower bound for circuit complexity of f
  - For deterministic functions

Can we hope to evade these barriers for randomized functions?

OT complexity

$|f|_{\text{OT}}$ = minimum number of OTs required by an information-theoretically (semi-honest, perfectly) secure protocol for f

- Well-studied since [Kilian'88] showed OT is complete
- But formidable barriers to proving super-linear lower bounds
  - In particular, lower bound for $|f|_{\text{OT}}$ is a lower bound for circuit complexity of f
  - For deterministic functions

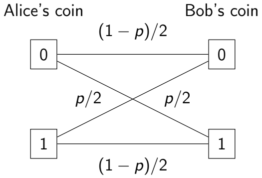
Can we hope to evade these barriers for randomized functions?

Yes!

Noisy coin-tossing

p-noisy coin toss is defined as follows:

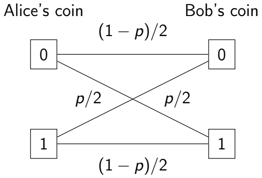
Noisy coin-tossing



p-noisy coin toss is defined as follows:

- Alice and Bob get a uniform bit each
- Which are unequal with probability p

Noisy coin-tossing



p-noisy coin toss is defined as follows:

- Alice and Bob get a uniform bit each
- Which are unequal with probability p

- Circuit complexity of sampling $f_{p\text{-coin}}$ is $\Theta(\log(1/p))$ (need that many bits as randomness)
- By basic GMW protocol [GMW87] $|f_{p\text{-coin}}|_{\mathsf{OT}} = O(\log 1/p)$
- Can we securely compute $f_{p\text{-coin}}$ with fewer OTs?

Contributions

- Main Result: $|f_{p\text{-coin}}|_{\mathsf{OT}} = \Theta(\log 1/p)$

Contributions

- Main Result: $|f_{p\text{-coin}}|_{\mathsf{OT}} = \Theta(\log 1/p)$
  - Not limited by input/output size!

Contributions

- Main Result: $|f_{p\text{-coin}}|_{\mathsf{OT}} = \Theta(\log 1/p)$
  - Not limited by input/output size!
  - Beyond the reach of state-of-the-art Information-theoretic techniques
    - Limited by input/output size: a lower bound of at most 1
    - As p approaches 0, their lower bound degrades

Contributions

- Main Result: $|f_{p\text{-coin}}|_{\mathsf{OT}} = \Theta(\log 1/p)$
  - Not limited by input/output size!
  - Beyond the reach of state-of-the-art Information-theoretic techniques
    - Limited by input/output size: a lower bound of at most 1
    - As p approaches 0, their lower bound degrades
- Tools:
  - Relation between **Secure Zero-Communication Reductions (SZCR)** [NPP20] and OT complexity for randomised functions
  - Define a **Balanced Embedding complexity** to simplify analysis

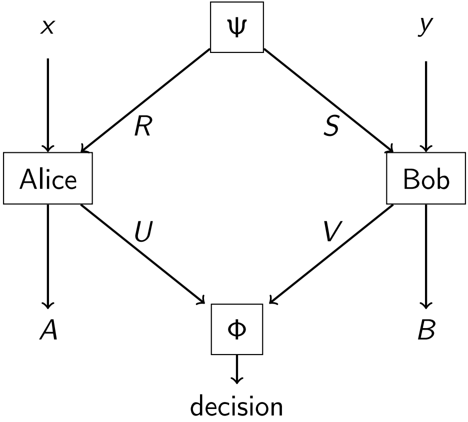
Contributions

- Main Result: $|f_{p\text{-coin}}|_{\mathsf{OT}} = \Theta(\log 1/p)$
  - Not limited by input/output size!
  - Beyond the reach of state-of-the-art Information-theoretic techniques
    - Limited by input/output size: a lower bound of at most 1
    - As p approaches 0, their lower bound degrades
- Tools:
  - Relation between **Secure Zero-Communication Reductions (SZCR)** [NPP20] and OT complexity for randomised functions
  - Define a **Balanced Embedding complexity** to simplify analysis

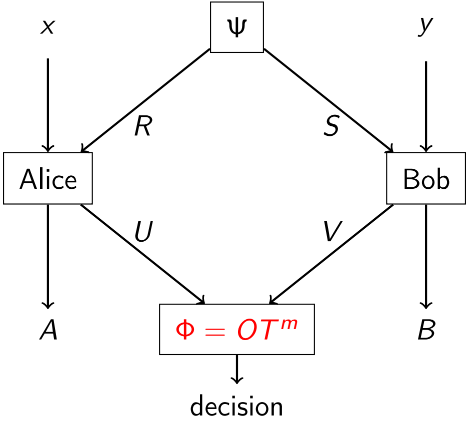
$$|f|_{\mathsf{emb}} \leq |f|_{\mathsf{szcr}} \lesssim |f|_{\mathsf{OT}}$$

Table of Contents

SZCR



SZCR



$|f|_{\text{szcr}}$ = minimum number of OTs needed by an SZCR of f

Table of Contents

SZCR and OT complexity

Using our construction for SZCR given an MPC protocol, we will show:

$$|f|_{\text{szcr}} \lesssim |f|_{\text{OT}}$$

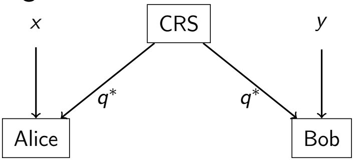
Same result was shown in [NPP20] for deterministic f

Deterministic functions [NPP20]

Given an MPC protocol using m OTs for computing f, an SZCR for f using OT^m:

Deterministic functions [NPP20]

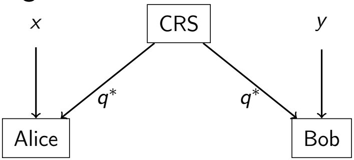
Given an MPC protocol using m OTs for computing f, an SZCR for f using OT^m:



1. Receive transcript q^* from CRS

Deterministic functions [NPP20]

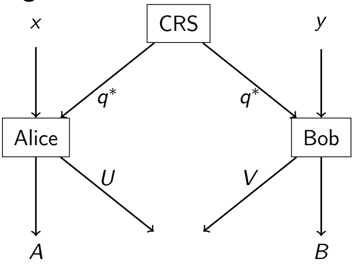
Given an MPC protocol using m OTs for computing f, an SZCR for f using OT^m:



1. Receive transcript q^* from CRS
2. If q^* compatible with input:

Deterministic functions [NPP20]

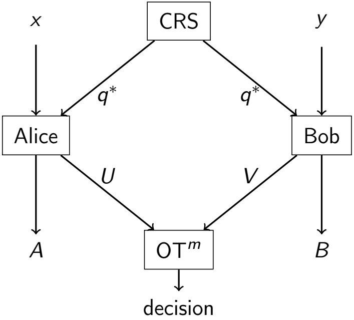
Given an MPC protocol using m OTs for computing f, an SZCR for f using OT^m:



1. Receive transcript q^* from CRS
2. If q^* compatible with input:
   - Compute output
   - Sample OT view

Deterministic functions [NPP20]

Given an MPC protocol using m OTs for computing f, an SZCR for f using OT^m:

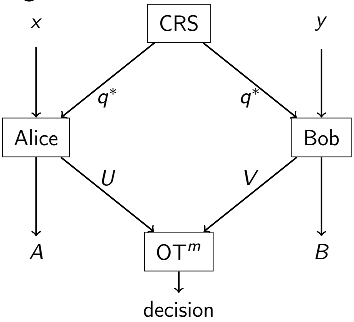


1. Receive transcript q^* from CRS
2. If q^* compatible with input:
   - Compute output
   - Sample OT view

- Output will be correct if OT^m predicate accepts (U, V)

Randomised functions

Given an MPC protocol using m OTs for computing f, an SZCR for f using OT^m:



1. Sample output
2. Receive transcript q^* from CRS
3. If q^* compatible with ~~input~~ input-output pair
   - ~~Compute output~~
   - Sample OT view

- Output will be correct if OT^m predicate accepts (U, V)

SZCR and OT complexity

Using our construction for SZCR given an MPC protocol, we are able to conclude:

$$|f|_{\mathsf{szcr}} \lesssim |f|_{\mathsf{OT}}$$

SZCR and OT complexity

Using our construction for SZCR given an MPC protocol, we are able to conclude:

$$|f|_{\mathsf{szcr}} \lesssim |f|_{\mathsf{OT}}$$

Remarks on the construction:

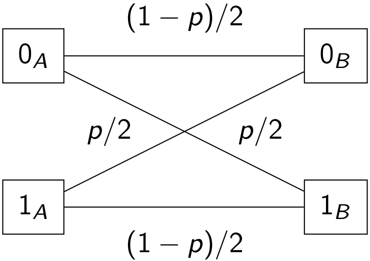
- Comes with a lower bound on the acceptance probability
- Generalizes to correlations beyond OT
- Can avoid the need for common randomness

Table of Contents

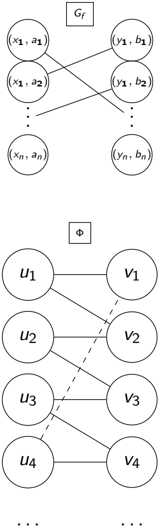
Evaluation Graph

For a function $f : X \times Y \to A \times B$, we define the evaluation graph G_f as:

- Bipartite graph on $(X \times A), (Y \times B)$
- Weight of edge $((x, a), (y, b)) = Pr[f(x, y) = (a, b)]$

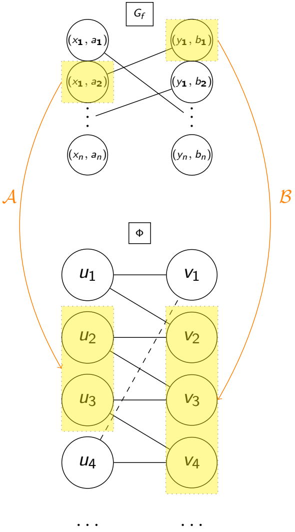


SZCR as a graph embedding



Consider the evaluation graph of f and the graph of the predicate Φ. An SZCR:

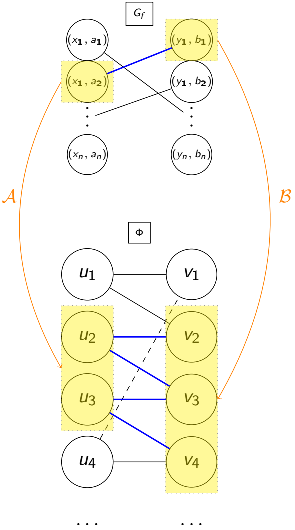
SZCR as a graph embedding



Consider the evaluation graph of f and the graph of the predicate Φ. An SZCR:

- Defines a probabilistic map from nodes in G_f to the graph of Φ

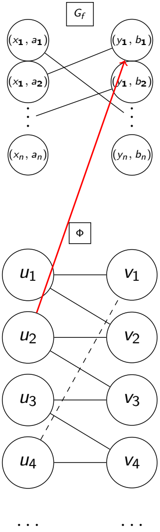
SZCR as a graph embedding



Consider the evaluation graph of f and the graph of the predicate Φ. An SZCR:

- Defines a probabilistic map from nodes in G_f to the graph of Φ
- Results in "embedding" the function graph into predicate

SZCR as a graph embedding

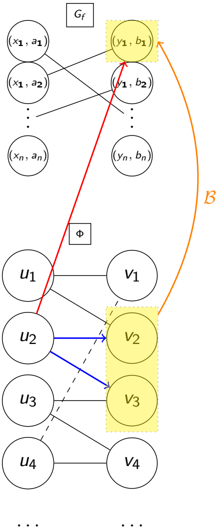


Consider the evaluation graph of f and the graph of the predicate Φ. An SZCR:

- Defines a probabilistic map from nodes in G_f to the graph of Φ
- Results in "embedding" the function graph into predicate

What can a semi-honest adversary infer about (y_1, b_1) from u_2?:

SZCR as a graph embedding



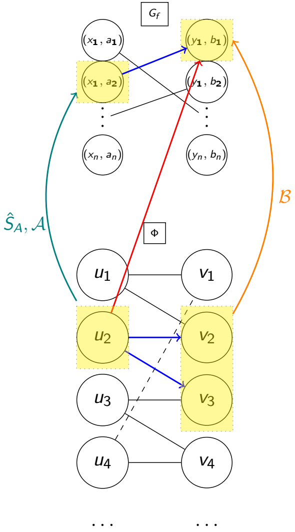
Consider the evaluation graph of f and the graph of the predicate Φ. An SZCR:

- Defines a probabilistic map from nodes in G_f to the graph of Φ
- Results in "embedding" the function graph into predicate

What can a semi-honest adversary infer about (y_1, b_1) from u_2?:

- Using the protocol

SZCR as a graph embedding



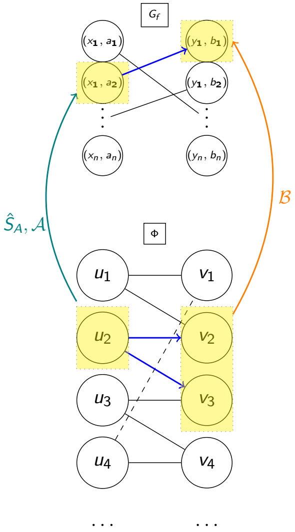
Consider the evaluation graph of f and the graph of the predicate Φ.
An SZCR:

- Defines a probabilistic map from nodes in G_f to the graph of Φ
- Results in "embedding" the function graph into predicate

What can a semi-honest adversary infer about (y_1, b_1) from u_2?:

- Using the protocol
- Using the simulator

SZCR as a graph embedding



Consider the evaluation graph of f and the graph of the predicate Φ.
An SZCR:

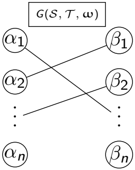
- Defines a probabilistic map from nodes in G_f to the graph of Φ
- Results in "embedding" the function graph into predicate

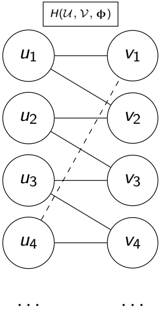
What can a semi-honest adversary infer about (y_1, b_1) from u_2?:

- Using the protocol
- Using the simulator

For security, we need both to be equivalent

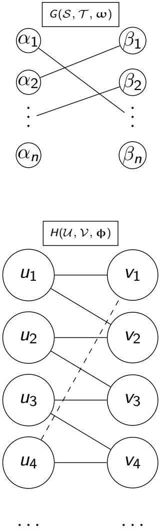
Simplifying to a Balanced Embedding





Balanced embedding of weighted bipartite graph $G(\mathcal{S}, \mathcal{T}, \omega)$ into $H(\mathcal{U}, \mathcal{V}, \phi)$ where weight function ω is non-negative:

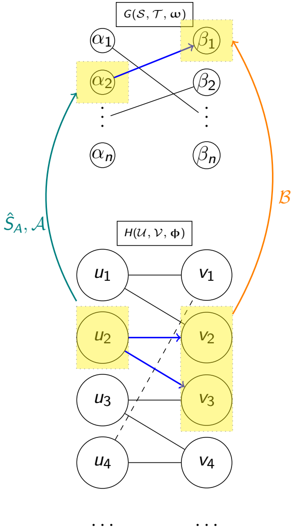
Simplifying to a Balanced Embedding



Balanced embedding of weighted bipartite graph $G(\mathcal{S}, \mathcal{T}, \omega)$ into $H(\mathcal{U}, \mathcal{V}, \phi)$ where weight function ω is non-negative:

- (π, θ), two embeddings that balance each other

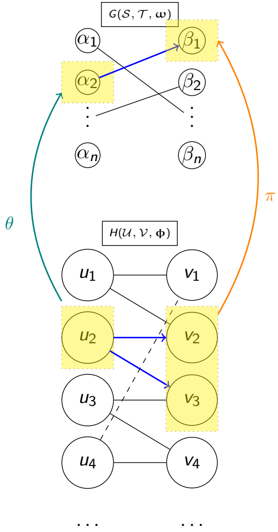
Simplifying to a Balanced Embedding



Balanced embedding of weighted bipartite graph $G(\mathcal{S}, \mathcal{T}, \omega)$ into $H(\mathcal{U}, \mathcal{V}, \phi)$ where weight function ω is non-negative:

- (π, θ), two embeddings that balance each other

Simplifying to a Balanced Embedding



Balanced embedding of weighted bipartite graph $G(\mathcal{S}, \mathcal{T}, \omega)$ into $H(\mathcal{U}, \mathcal{V}, \phi)$ where weight function ω is non-negative:

- (π, θ), two embeddings that balance each other

Balancing (similar to security):

$$\sum_{v \in \mathcal{V}} \pi(v, \beta) \cdot \phi(u, v) = \theta(u, \alpha) \cdot \omega(\alpha, \beta) \forall u \in \mathcal{U}$$

Balanced Embedding complexity

Balanced Embedding

For $\pi, \theta : (\mathcal{U} \times \mathcal{S}) \cup (\mathcal{V} \times \mathcal{T}) \to \mathbb{R}_{\geq 0}$, (π, θ) is a balanced embedding of $G(\mathcal{S}, \mathcal{T}, \boldsymbol{\omega})$ into $H(\mathcal{U}, \mathcal{V}, \boldsymbol{\phi})$ if for all $(\alpha, \beta) \in \mathcal{S} \times \mathcal{T}$:

$$\sum_{v \in \mathcal{V}} \pi(v, \beta) \cdot \boldsymbol{\phi}(u, v) = \theta(u, \alpha) \cdot \boldsymbol{\omega}(\alpha, \beta) \qquad \forall u \in \mathcal{U}$$

$$\sum_{u \in \mathcal{U}} \pi(u, \alpha) \cdot \boldsymbol{\phi}(u, v) = \theta(v, \beta) \cdot \boldsymbol{\omega}(\alpha, \beta) \qquad \forall v \in \mathcal{V}$$

$$\sum_{u \in \mathcal{U}} \pi(u, \alpha) \cdot \theta(u, \alpha) = 1 \qquad \sum_{v \in \mathcal{V}} \pi(v, \beta) \cdot \theta(v, \beta) = 1 \qquad \text{if } \boldsymbol{\omega}(\alpha, \beta) > 0$$

Balanced Embedding complexity

Balanced Embedding

For $\pi, \theta : (\mathcal{U} \times \mathcal{S}) \cup (\mathcal{V} \times \mathcal{T}) \to \mathbb{R}_{\geq 0}$, (π, θ) is a balanced embedding of $G(\mathcal{S}, \mathcal{T}, \boldsymbol{\omega})$ into $H(\mathcal{U}, \mathcal{V}, \boldsymbol{\phi})$ if for all $(\alpha, \beta) \in \mathcal{S} \times \mathcal{T}$:

$$\sum_{v \in \mathcal{V}} \pi(v, \beta) \cdot \boldsymbol{\phi}(u, v) = \theta(u, \alpha) \cdot \boldsymbol{\omega}(\alpha, \beta) \qquad \forall u \in \mathcal{U}$$

$$\sum_{u \in \mathcal{U}} \pi(u, \alpha) \cdot \boldsymbol{\phi}(u, v) = \theta(v, \beta) \cdot \boldsymbol{\omega}(\alpha, \beta) \qquad \forall v \in \mathcal{V}$$

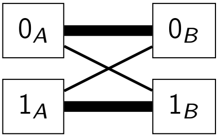
$$\sum_{u \in \mathcal{U}} \pi(u, \alpha) \cdot \theta(u, \alpha) = 1 \qquad \sum_{v \in \mathcal{V}} \pi(v, \beta) \cdot \theta(v, \beta) = 1 \qquad \text{if } \boldsymbol{\omega}(\alpha, \beta) > 0$$

Balanced Embedding Complexity

The *balanced embedding complexity* of f, $|f|_{\mathrm{emb}}$ is the smallest m such that G_f has a balanced embedding into $H_{\boldsymbol{\phi}^m_{\mathrm{OT}}}$. By construction, we conclude: $|f|_{\mathrm{emb}} \leq |f|_{\mathrm{szcr}}$

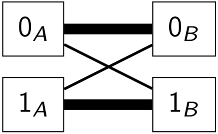
Table of Contents

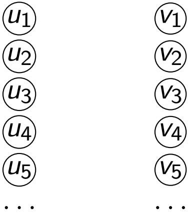
Bounding Embedding Complexity



- The noisy coin-toss looks like this for small p

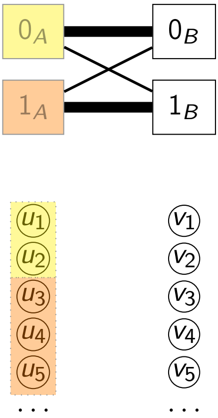
Bounding Embedding Complexity





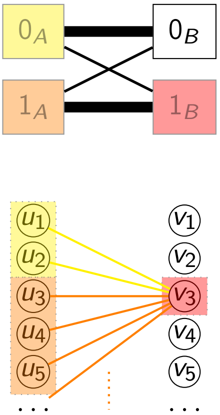
- The noisy coin-toss looks like this for small p
- We want to embed it into m instances of OT

Bounding Embedding Complexity



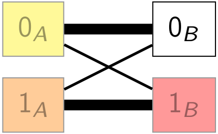
- The noisy coin-toss looks like this for small p
- We want to embed it into m instances of OT
- Assume that nodes are embedded as follows

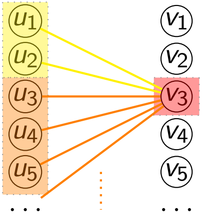
Bounding Embedding Complexity



- The noisy coin-toss looks like this for small p
- We want to embed it into m instances of OT
- Assume that nodes are embedded as follows
- We show that there is a v:
  - Mapped from 1_B
  - Number of edges to groups depends on thickness in function graph

Bounding Embedding Complexity





- The noisy coin-toss looks like this for small p
- We want to embed it into m instances of OT
- Assume that nodes are embedded as follows
- We show that there is a v:
  - Mapped from 1_B
  - Number of edges to groups depends on thickness in function graph
- We find that $2^m \geq \frac{1-p}{p}$

Table of Contents

Conclusion

- We showed:
  - $|f|_{\mathsf{emb}} \leq |f|_{\mathsf{szcr}} \lesssim |f|_{\mathsf{OT}}$ for all (possibly randomised) f
  - $|f_{p\text{-coin}}|_{\mathsf{emb}} = \omega(\log 1/p)$

Hence, $|f_{p\text{-coin}}|_{\mathsf{OT}} = \omega(\log 1/p)$

Conclusion

- We showed:
  - $|f|_{\mathsf{emb}} \leq |f|_{\mathsf{szcr}} \lesssim |f|_{\mathsf{OT}}$ for all (possibly randomised) f
  - $|f_{p\text{-coin}}|_{\mathsf{emb}} = \omega(\log 1/p)$

Hence, $|f_{p\text{-coin}}|_{\mathsf{OT}} = \omega(\log 1/p)$

Open: Super-linear balanced-embedding complexity for deterministic functions

- For explicit functions, this faces circuit complexity barriers
- Immediate question: Do such functions **exist**?