FINE-GRAINED WORST-CASE TO AVERAGE-CASE REDUCTIONS (FGWCTAC)

Andrea Lincoln UC Berkeley

NEW TECHNIQUES FOR PROVING FINE-GRAINED AVERAGE-CASE HARDNESS

Mina Dalirrooyfard, **Andrea Lincoln**,
Virginia Vassilevska Williams

WHAT IS FINE-GRAINED COMPLEXITY?

- A concern over constants in the exponent
- E.G.
  - n^2 vs $n^{2-\epsilon}$
    - e.g. the 3-SUM hypothesis
  - 2^n vs $2^{(1-\epsilon)n}$
    - e.g. the Strong Exponential Time Hypothesis (SETH)

GOAL:
WORST CASE TO AVERAGE CASE

- We want to understand how hard our favorite problems are on average.
- We want to give explicit distributions over which we believe they are hard.

WHAT ARE WORST-CASE TO AVERAGE-CASE REDUCTIONS?

- Worst-case problem P
- Show P is equiv to (many) calls to avg case problem Q
  - Q's input is drawn from some distribution D
  - Q's success probability is over both randomness of algorithm and randomness of distribution

AVERAGE CASE ALGORITHMS VS RANDOMIZED ALGORITHMS

- Average-Case Algorithm with success probability $1 - \epsilon$: can be wrong *consistently* on a $1 - \epsilon$ fraction of inputs (no naïve boosting: re-run on same input same bad answer)
- Randomized Algorithm with success probability $1 - \epsilon$: must get at least $1 - \epsilon$ success on *all* inputs (naïve boosting: re-run on same input take majority rule)

A FUN EXAMPLE: MATRIX MULTIPLICATION

[BLUM, LUBY, RUBINFELD]

Worst-Case Problem:

$\boldsymbol{A} \times \boldsymbol{B}$ for $n \times n$ matrices in F_q

Average-Case Problems:

- Sample random matrices $\boldsymbol{X}, \boldsymbol{Y} \sim F_q^{n \times n}$
- Problems:
  - $(\boldsymbol{A} + \boldsymbol{X})(\boldsymbol{B} + \boldsymbol{Y})$
  - $(\boldsymbol{X})(\boldsymbol{B} + \boldsymbol{Y})$
  - $(\boldsymbol{A} + \boldsymbol{X})(\boldsymbol{Y})$
  - $(\boldsymbol{X})(\boldsymbol{Y})$

A FUN EXAMPLE: MATRIX MULTIPLICATION

[BLUM, LUBY, RUBINFELD]

- Problems:

$$
\begin{array}{l|l}
(A+X)(B+Y) & AB + XB + AY + XY \\
-(X)(B+Y) & \quad - XB \qquad - XY \\
-(A+X)(Y) & \qquad -AY - XY \\
(X)(Y) & \qquad XY \\
\hline
AB & AB
\end{array}
$$

WHAT TO GET FROM MM EXAMPLE

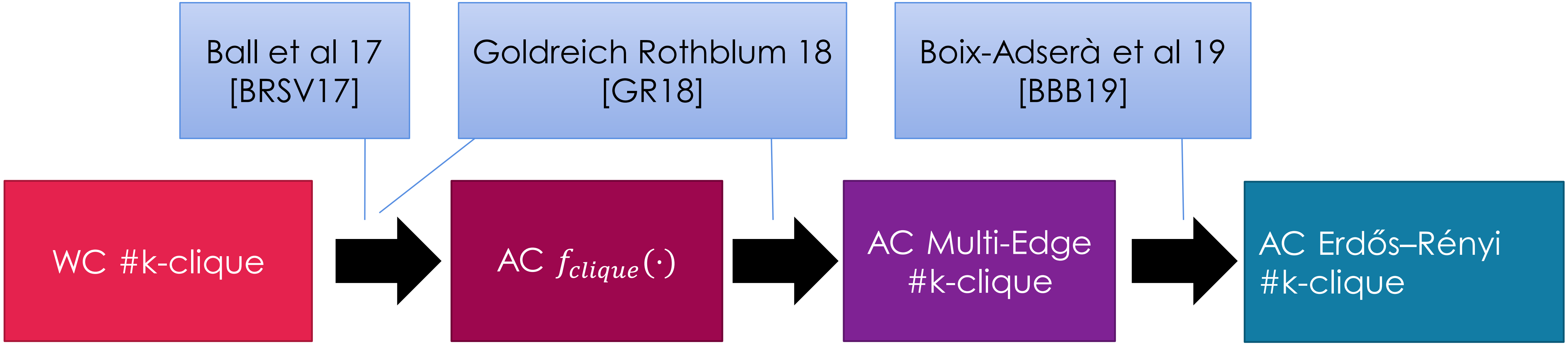
- Make multiple correlated calls
- Each call is indistinguishable from random MM
  - $(\boldsymbol{A} + \boldsymbol{X})(\boldsymbol{B} + \boldsymbol{Y})$
  - $(\boldsymbol{X})(\boldsymbol{B} + \boldsymbol{Y})$
  - $(\boldsymbol{A} + \boldsymbol{X})(\boldsymbol{Y})$
  - $(\boldsymbol{X})(\boldsymbol{Y})$
- With $\boldsymbol{\epsilon}$ error for random MM union bound to solve WC MM with probability $1 - \boldsymbol{4\epsilon}$

HIGHLIGHTED RESULTS

- A **framework** to give average case hardness for problems P with a "good low degree polynomial"
- **A new type of problem**, a "factored problem"
  - Factored-P is more expressive than P
  - #Factored-P hard on average

THE STORY I WANT TO TELL

- **Going from Ball et al 17**
- **To Boix-Adserà et al 19**
- To our paper



Previous work

CORE PROBLEMS AND HYPOTHESES

The core hypotheses of Fine-Grained Complexity (FGC) are:

- SETH [k-SAT requires $2^{n(1-o(1))}$ time]
- 3-SUM Hypothesis [3-SUM requires $n^{2-o(1)}$ time]
- All Pairs Shortest Paths (APSP) [APSP requires $n^{3-o(1)}$ time]

CORE PROBLEMS AND HYPOTHESES

The core hypotheses of Fine-Grained Complexity (FGC) are:

- SETH [k-SAT requires $2^{n(1-o(1))}$ time]
- 3-SUM Hypothesis [3-SUM requires $n^{2-o(1)}$ time]
- All Pairs Shortest Paths (APSP) [APSP requires $n^{3-o(1)}$ time]

k-SAT problem:
Given a Boolean formula in conjunctive normal form return true if there is an assignment that satisfies the formula, false otherwise.

Previous work

CORE PROBLEMS AND HYPOTHESES

The core hypotheses of Fine-Grained Complexity (FGC) are:

- SETH [k-SAT requires $2^{n(1-o(1))}$ time]
- 3-SUM Hypothesis [3-SUM requires $n^{2-o(1)}$ time]
- All Pairs Shortest Paths (APSP) [APSP requires $n^{3-o(1)}$ time]

3-SUM problem:
Given a lists of numbers L return true if there are three numbers $a, b, c \in L$ such that $a + b + c = 0$.

Previous work

CORE PROBLEMS AND HYPOTHESES

The core hypotheses of Fine-Grained Complexity (FGC) are:

- SETH [k-SAT requires $2^{n(1-o(1))}$ time]
- 3-SUM Hypothesis [3-SUM requires $n^{2-o(1)}$ time]
- All Pairs Shortest Paths (APSP) [APSP requires $n^{3-o(1)}$ time]

APSP problem:

Given a graph with n nodes and weighted edges give the shortest path length between all pairs of nodes in the graph.

Previous work

PROBLEMS AND HYPOTHESES FOR THIS TALK

First: we need to define the core hypotheses + problems of FGC

SETH:
K-SAT requires $2^{n(1-o(1))}$ time

PROBLEMS AND HYPOTHESES FOR THIS TALK

First: we need to define the core hypotheses + problems of FGC

OV problem:
Given a list of n vectors tell me if $\exists\, \vec{u}, \vec{v}$ s.t. $\vec{u} \cdot \vec{v} = 0$

SETH:
K-SAT requires $2^{n(1-o(1))}$ time

Previous work

PROBLEMS AND HYPOTHESES FOR THIS TALK

First: we need to define the core hypotheses + problems of FGC

OV hyp: OV requires $n^{2-o(1)}$ time

OV problem: Given a list of n vectors tell me if $\exists\ \vec{u}, \vec{v}$ s.t. $\vec{u} \cdot \vec{v} = 0$

SETH: K-SAT requires $2^{n(1-o(1))}$ time

Previous work

PROBLEMS AND HYPOTHESES FOR THIS TALK

First: we need to define the core hypotheses + problems of FGC

OV hyp:
OV requires $n^{2-o(1)}$ time

Fun fact:
OV hypothesis is implied by SETH! [W07]

Previous work

PROBLEMS AND HYPOTHESES FOR THIS TALK

First: we need to define the core hypotheses + problems of FGC

OV hyp:
OV requires $n^{2-o(1)}$ time

ZKC problem:
Given a dense graph with weighted edges return true if there is a k-clique whose edges sum to zero.

Previous work

PROBLEMS AND HYPOTHESES FOR THIS TALK

First: we need to define the core hypotheses + problems of FGC

OV hyp:
OV requires $n^{2-o(1)}$ time

ZKC problem:
Given a dense graph with weighted edges return true if there is a k-clique whose edges sum to zero.

ZkC Hyp.:
ZKC requires $n^{k-o(1)}$ time

PROBLEMS AND HYPOTHESES FOR THIS TALK

First: we need to define the core hypotheses + problems of fine-grained complexity

OV hyp:
OV requires $n^{2-o(1)}$ time

Fun fact:
Z3C hypothesis is implied by both **3-SUM** and **APSP**!
[VW09] [VW10]

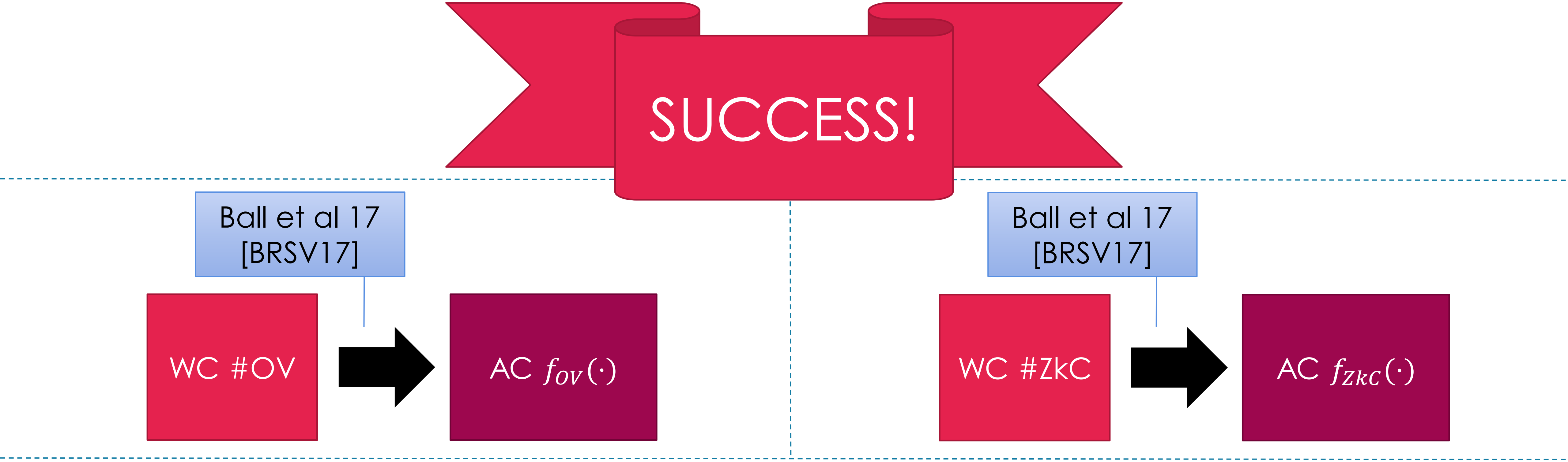
ZkC Hyp.:
ZKC requires $n^{k-o(1)}$ time

PREVIOUS WORK:[BRSV17] (BALL ET AL 17)

BRSV17 Goal: give a WC to AC reduction from the core FGC problems.

PREVIOUS WORK:[BRSV17] (BALL ET AL 17)

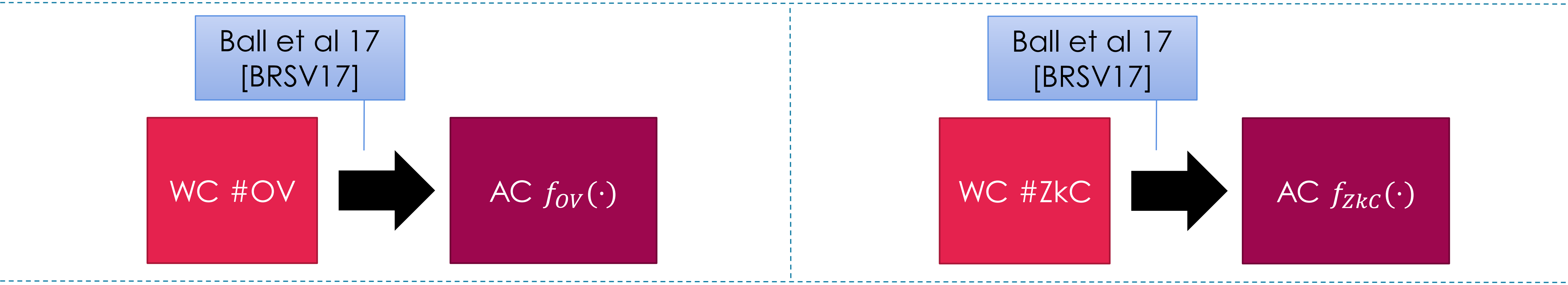
BRSV17 Goal: give a WC to AC reduction from the core FGC problems.



PREVIOUS WORK:[BRSV17] (BALL ET AL 17)

What are these problems?

They are based on polynomials over finite-fields.



[BRSV17] POLYNOMIALS

Given a problem P we want to generate a polynomial $f_P(\cdot)$ over Z_p such that:

1. $f_P(\cdot)$ has degree $d = n^{o(1)}$ (subpolynomial)
2. You can compute $P(I)$ from $f_P(I)$
   1. Treat the input I as an n bit vector

Previous work

[BRSV17] POLYNOMIALS

Given a problem P we want to generate a polynomial $f_P(\cdot)$ over Z_q such that:

1. $f_P(\cdot)$ has degree $d = n^{o(1)}$ (subpolynomial)
2. You can compute $P(I)$ from $f_P(I)$
   1. Treat the input I as an n bit vector

Then let $\hat{I} \sim (Z_q)^n$.

Computing $f_P(\hat{I})$ with probability $> 2/3$ in $O(T(n))$ implies a $T(n)n^{o(1)}$ algorithm for P in WC

Previous work

[BRSV17] POLYNOMIALS

Given a problem P w
polynomial $f_P(\cdot)$ over

1. $f_P(\cdot)$ has degree
2. You can compu
   1. Treat the input

$f_P(\hat{I})$ is **average-case**
from
P in the **worst-case**

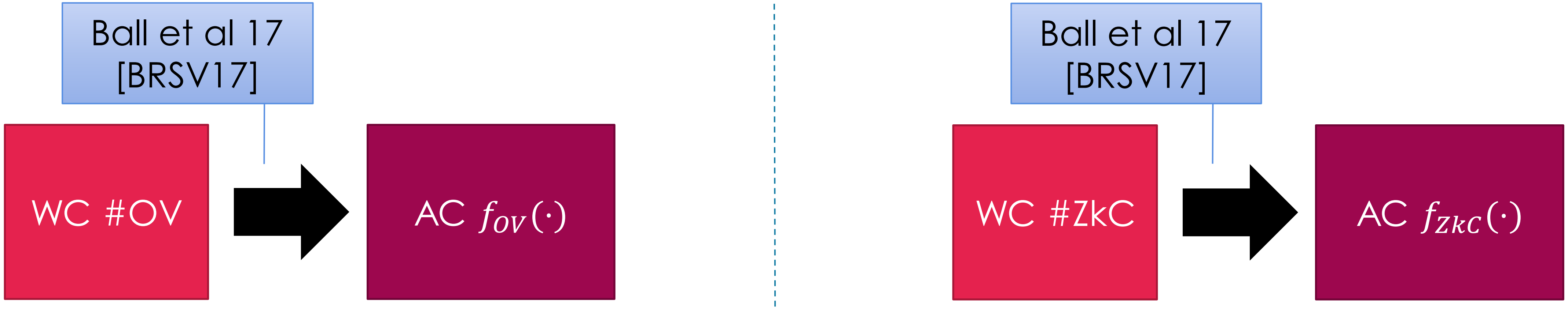
Then let $\hat{I} \sim (Z_q)^n$.
Computing $f_P(\hat{I})$ with probability $> 2/3$ in $O(T(n))$ implies a $T(n)n^{o(1)}$ algorithm for P in WC

PREVIOUS WORK:[BRSV17] (BALL ET AL 17)

Problem:

$f_{ZkC}(\cdot)$, for example, corresponds nicely to ZKC when inputs are zero and one.

But the average case **inputs are over large finite-fields**



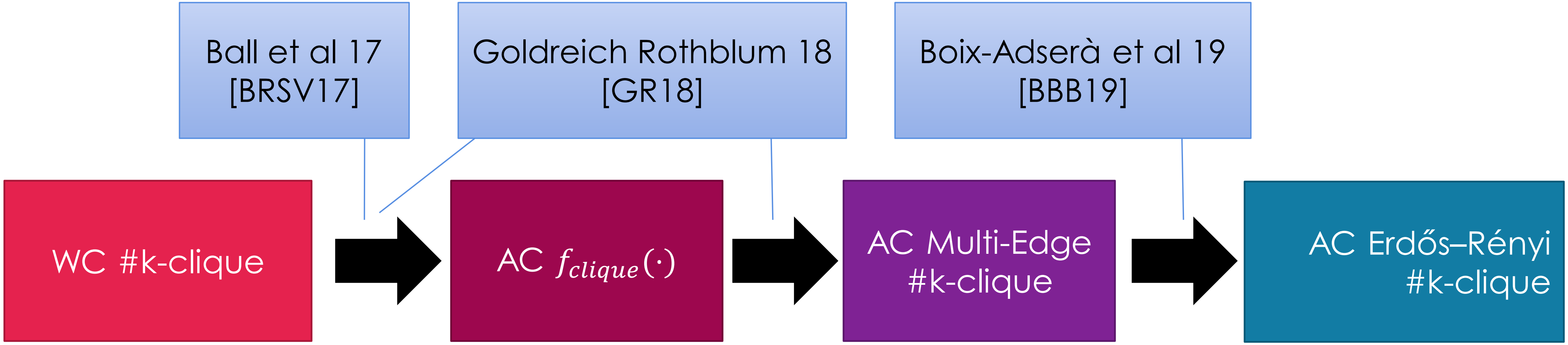
Previous work

HOW CAN WE GET BACK TO $\{0,1\}^n$?

We will use k-clique as the example to work though

Previous work

THE CASE OF CLIQUE

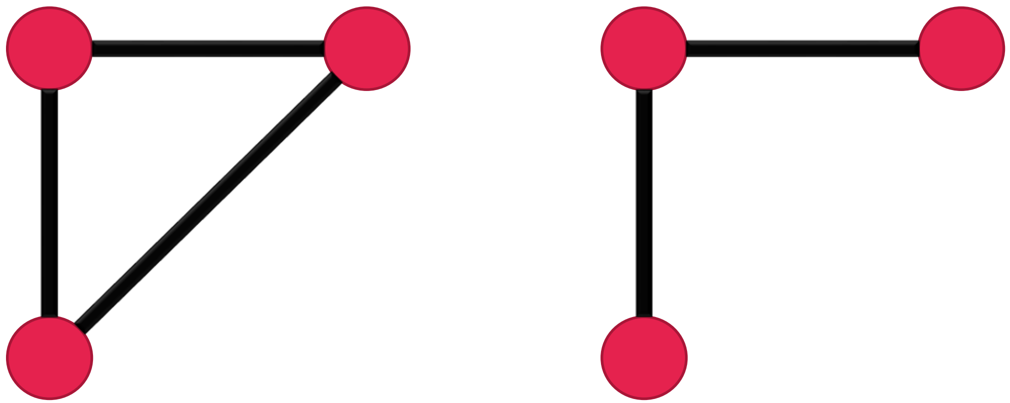


THE CASE OF CLIQUE

Ball et al 17 [BRSV17]

Goldreich Rothblum 18 [GR18]

WC #k-clique → AC $f_{clique}(\cdot)$



$$f_{clique}(G) = \sum_{v_1,\ldots,v_k} \left(\prod_{\substack{i,j \in [1,k] \\ i<j}} e(v_i, v_j) \right)$$

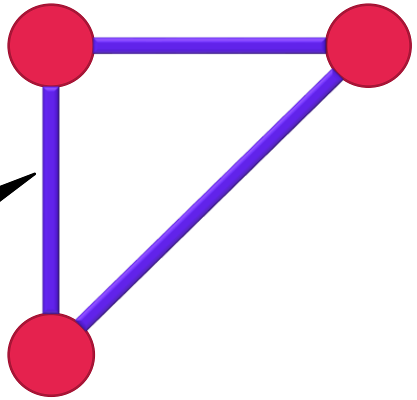
Previous work

THE CASE OF CLIQUE

Ball et al 17 [BRSV17]

Goldreich Rothblum 18 [GR18]

WC #k-clique → AC $f_{clique}(\cdot)$ → AC Multi-Edge #k-clique



Basically now all the edges have weights (uniformly random in $[0, \lg(n)]$).

$$f_{clique}(G) = \sum_{v_1,\ldots,v_k} \left(\prod_{\substack{i,j \in [1,k] \\ i<j}} e(v_i, v_j) \right)$$

Previous work

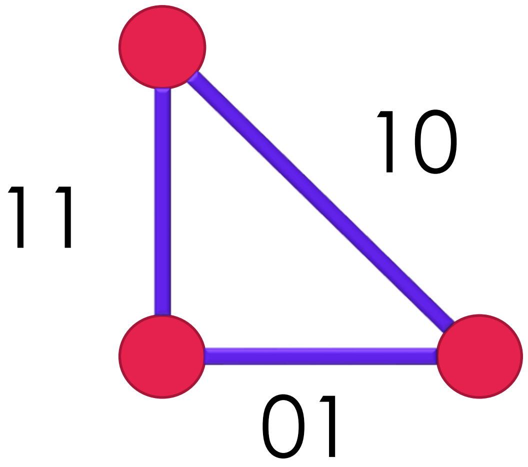
THE CASE OF CLIQUE

Ball et al 17 [BRSV17]

Goldreich Rothblum 18 [GR18]

Boix-Adserà et al 19 [BBB19]

WC #k-clique → AC $f_{clique}(\cdot)$ → AC Multi-Edge #k-clique → AC Erdős–Rényi #k-clique



Previous work

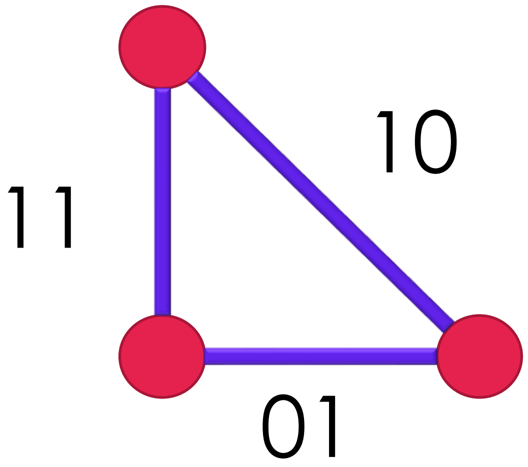
THE CASE OF CLIQUE

Ball et al 17 [BRSV17]

Goldreich Rothblum 18 [GR18]

Boix-Adserà et al 19 [BBB19]

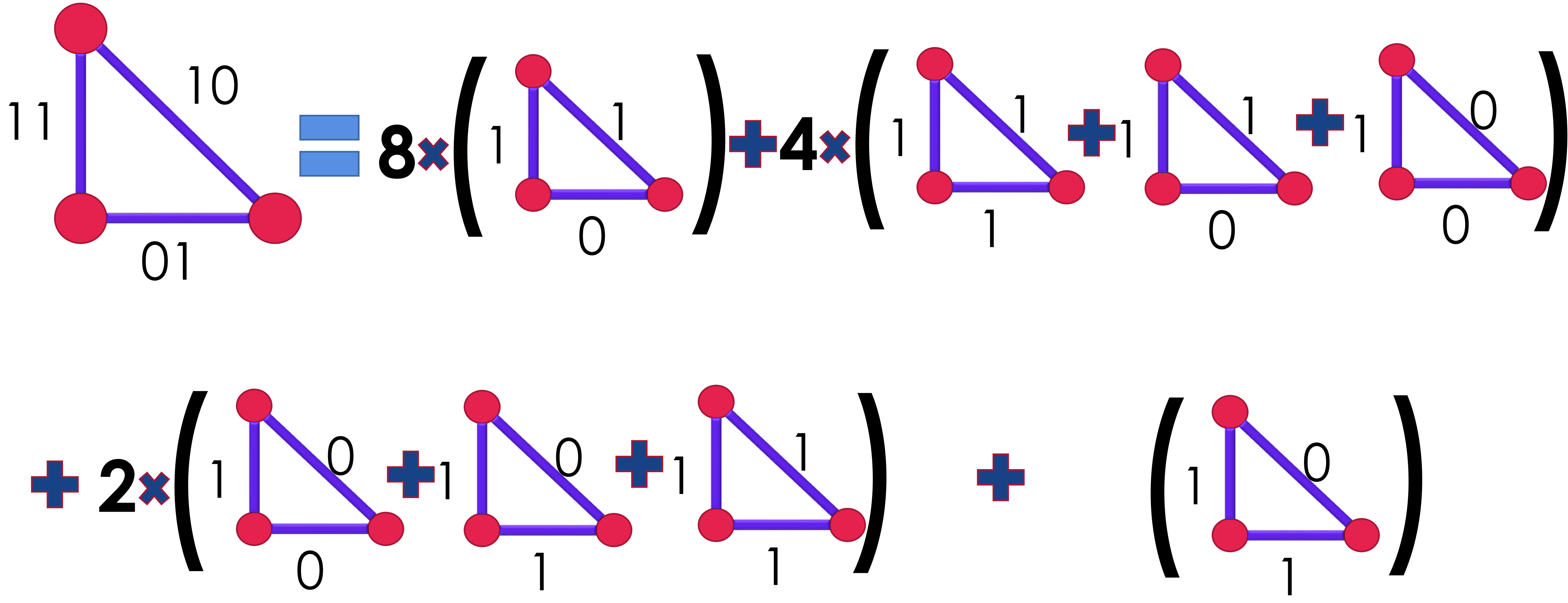
WC #k-clique → AC $f_{clique}(\cdot)$ → AC Multi-Edge #k-clique → AC Erdős–Rényi #k-clique



To get back to zero-one they can look at each bitwise multiplication.

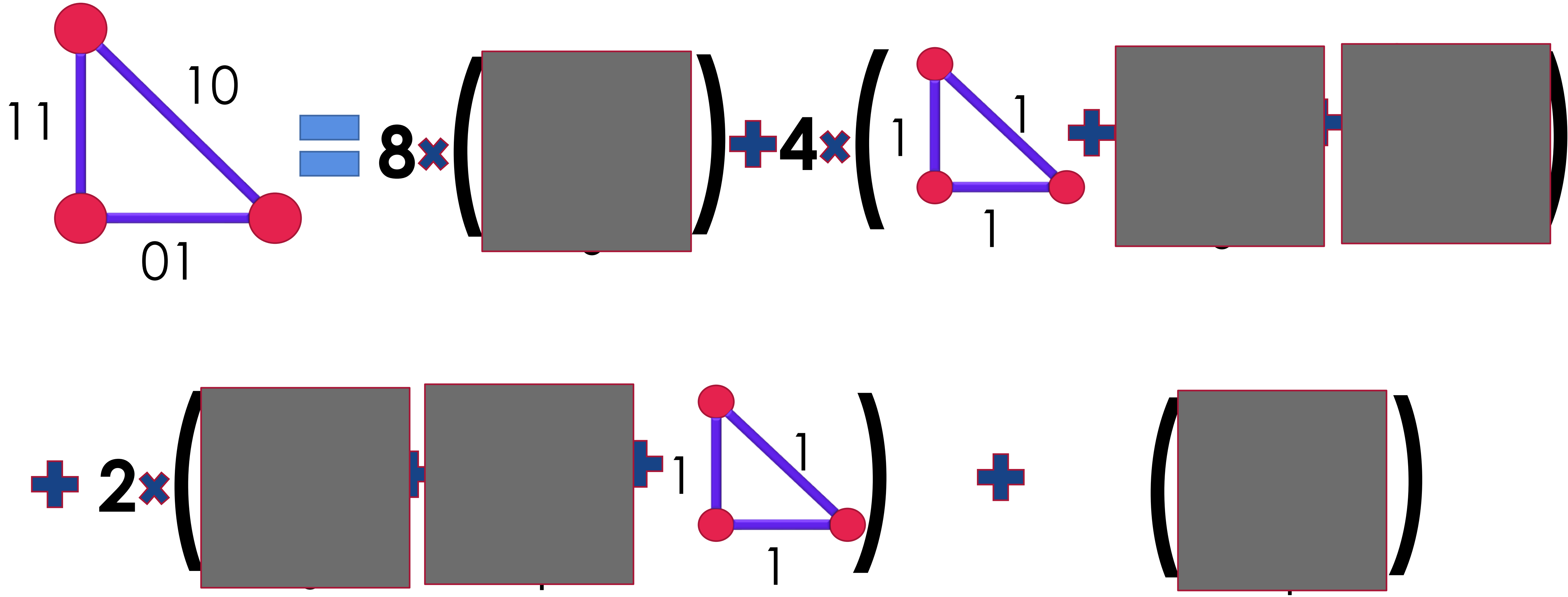
Previous work

THE CASE OF CLIQUE



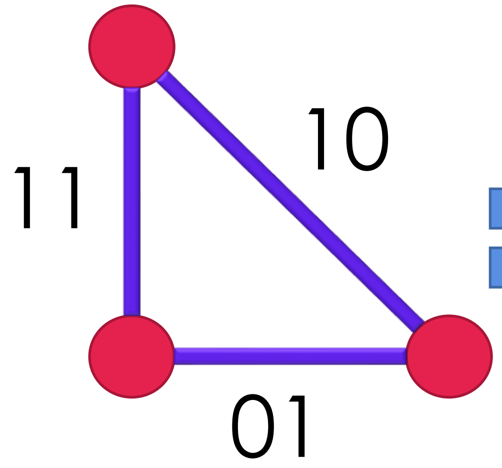
Previous work

THE CASE OF CLIQUE

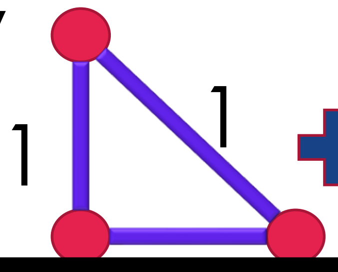


Previous work

THE CASE OF CLIQUE



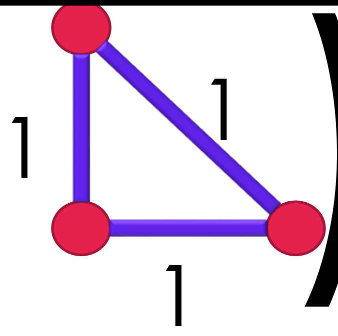
= 8×( ) + 4×(



+ +)

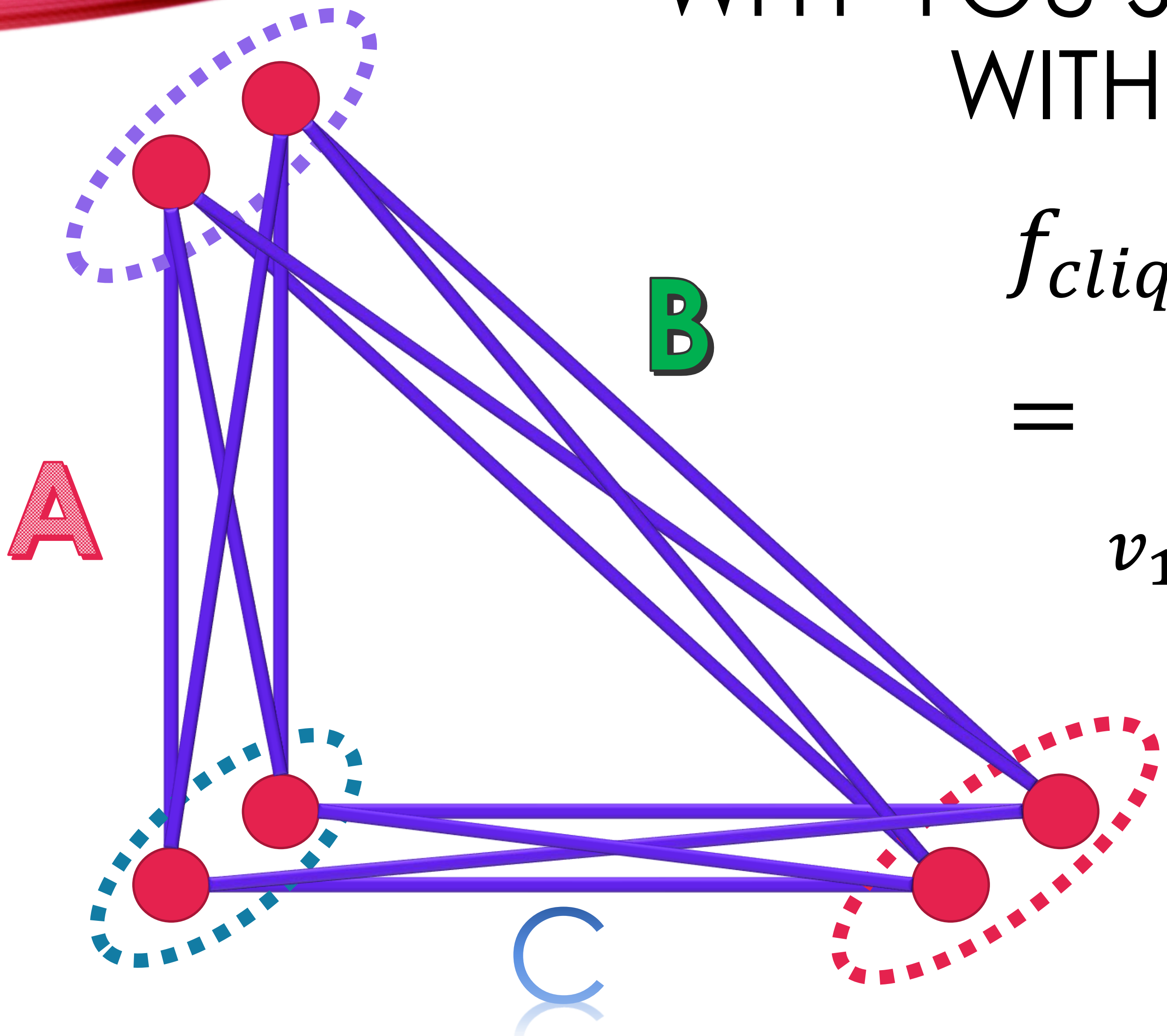
3x2x1 = 4+2

+ 2×(+ +



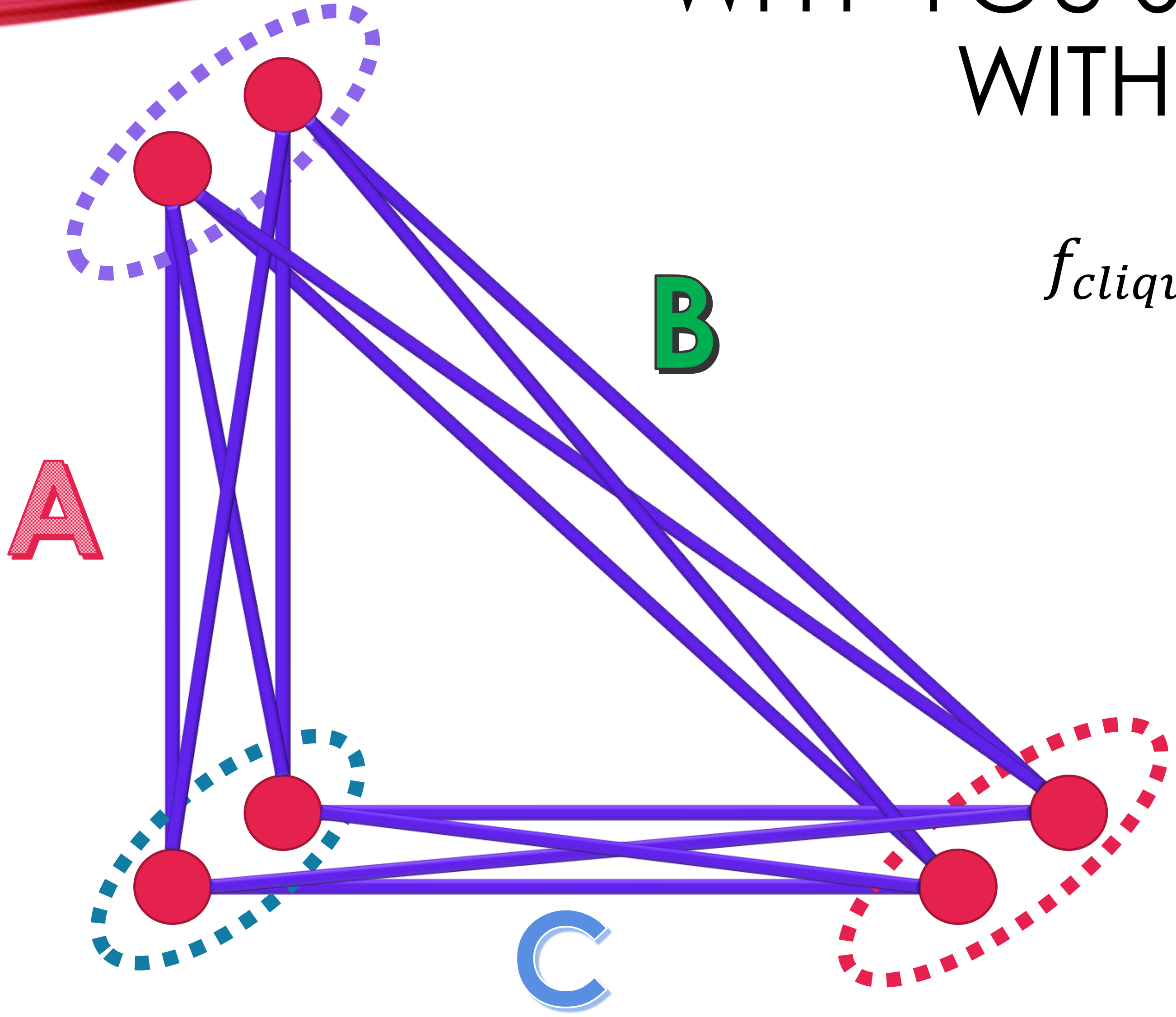
) + ( )

WHY YOU SHOULD BE FRIENDS WITH K-PARTITE GRAPHS



$$f_{clique}(G) = \sum_{v_1,v_2,v_3} \left(a_{v_1 v_2}\, b_{v_2 v_3}\, c_{v_3 v_1} \right)$$

WHY YOU SHOULD BE FRIENDS WITH K-PARTITE GRAPHS

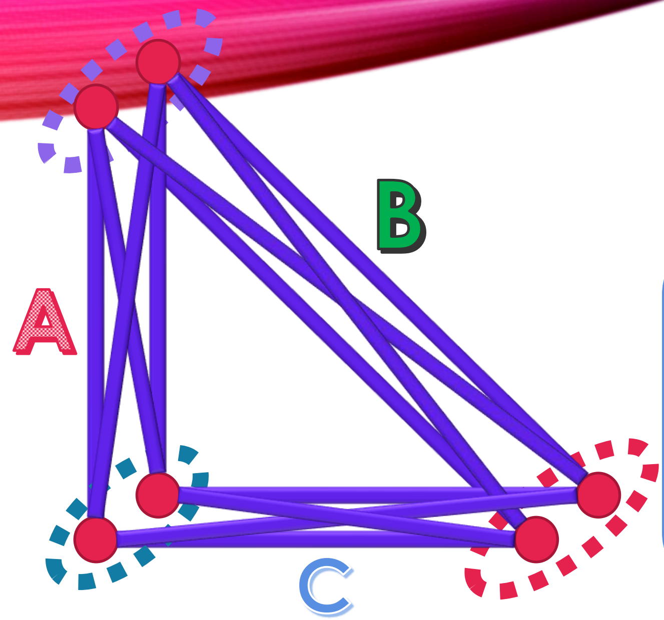


$$f_{clique}(G) = \sum_{v_1,v_2,v_3} \left(a_{v_1v_2}\, b_{v_2v_3}\, c_{v_3v_1}\right)$$

Grab the:
i^{th} bit weights in A,
j^{th} bit weights in B,
k^{th} bit weights in C
to form an instance
weight output by 2^{i+j+k}

Previous work

WHY YOU SHOULD BE FRIENDS WITH K-PARTITE GRAPHS



Grab the:
i^{th} bit weights in A,
j^{th} bit weights in B,
k^{th} bit weights in C
to form an instance

weight output by 2^{i+j+k}

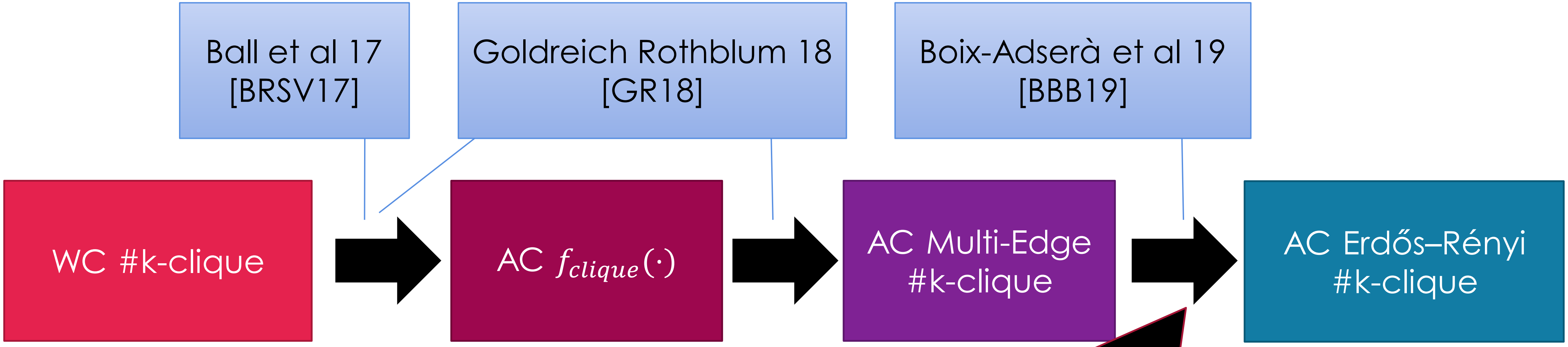
$$f_{clique}(G) = \sum_{v_1,v_2,v_3} \left(a_{v_1 v_2}\, b_{v_2 v_3}\, c_{v_3 v_1} \right)$$

But wait! Are those bits iid {0,1}?

With a bit of work yes.
BBB19 make bigger numbers with same value mod p. The bits in the big numbers are random.

Previous work

THE CASE OF CLIQUE



They got back to $\{0,1\}^n$,
and it **wasn't really about clique**!

…IT WASN’T REALLY ABOUT CLIQUE!

So what was it about?

- f was a sum of monomials all of degree d
- f was “d-partite”
- d was not too big (overhead exp in d)
- The output of f and P are the same

…IT WASN'T REALLY ABOUT CLIQUE!

The **Good Low-Degree Polynomial (GDLP)** $f_P(\cdot)$:

- Degree $d = o\left(\frac{\lg(n)}{lglg(n)}\right)$
- Strongly d -partite
- $f_P(I) = P(I)$

…IT WASN'T REALLY ABOUT CLIQUE!

The **Good Low-Degree Polynomial (GDLP)** $f_P(\cdot)$:

- Degree $d = o\left(\frac{\lg(n)}{lglg(n)}\right)$
- Strongly d -partite
- $f_P(I) = P(I)$

If P has a GLDP and WC P requires $T(n)$ time then
Uniform AC P requires $T(n)/\lg(n)^d$ time
if it succeeds with probability $1 - \frac{1}{\lg(n)^d}$

…IT WASN'T REALLY ABOUT CLIQUE!

The **Good Low-Degree Polynomial (GDLP)** $f_P(\cdot)$:

- Degree $d = o\left(\frac{\lg(n)}{lglg(n)}\right)$
- Strongly d -partite
- $f_P(I) = P(I)$

If P has a GLDP and WC P requires $T(n)$ time then
Uniform AC P requires $T(n) \cdot n^{-o(1)}$ time
if it succeeds with probability $1 - \frac{1}{n^{\epsilon}}$

This is the probability throughout the rest of the talk

…IT WASN'T REALLY ABOUT CLIQUE!

The **Good Low-Degree Polynomial (GDLP)** $f_P(\cdot)$:

- Degree $d = o\left(\frac{\lg(n)}{lglg(n)}\right)$
- Strongly d -partite
- $f_P(I) = P(I)$

WC P → AC $f_P(\cdot)$ → Uniform AC P

Content not in talk

RESULTS IN THIS PAPER

- A **framework** built on BBB19
- A new type of problem: **“factored” problems**
- Using the framework, factored problems, and **reductions**:
  - Avg. Case hardness for various string or graph problems
  - Avg. Case hardness for a graph problem from APSP,3-SUM & SETH
  - New candidate “hard from everything” problem
- We show that #OV is easy on average
- Reduction from Counting to Detection for avg case ZKC
- Avg case hardness for counting any small subgraph

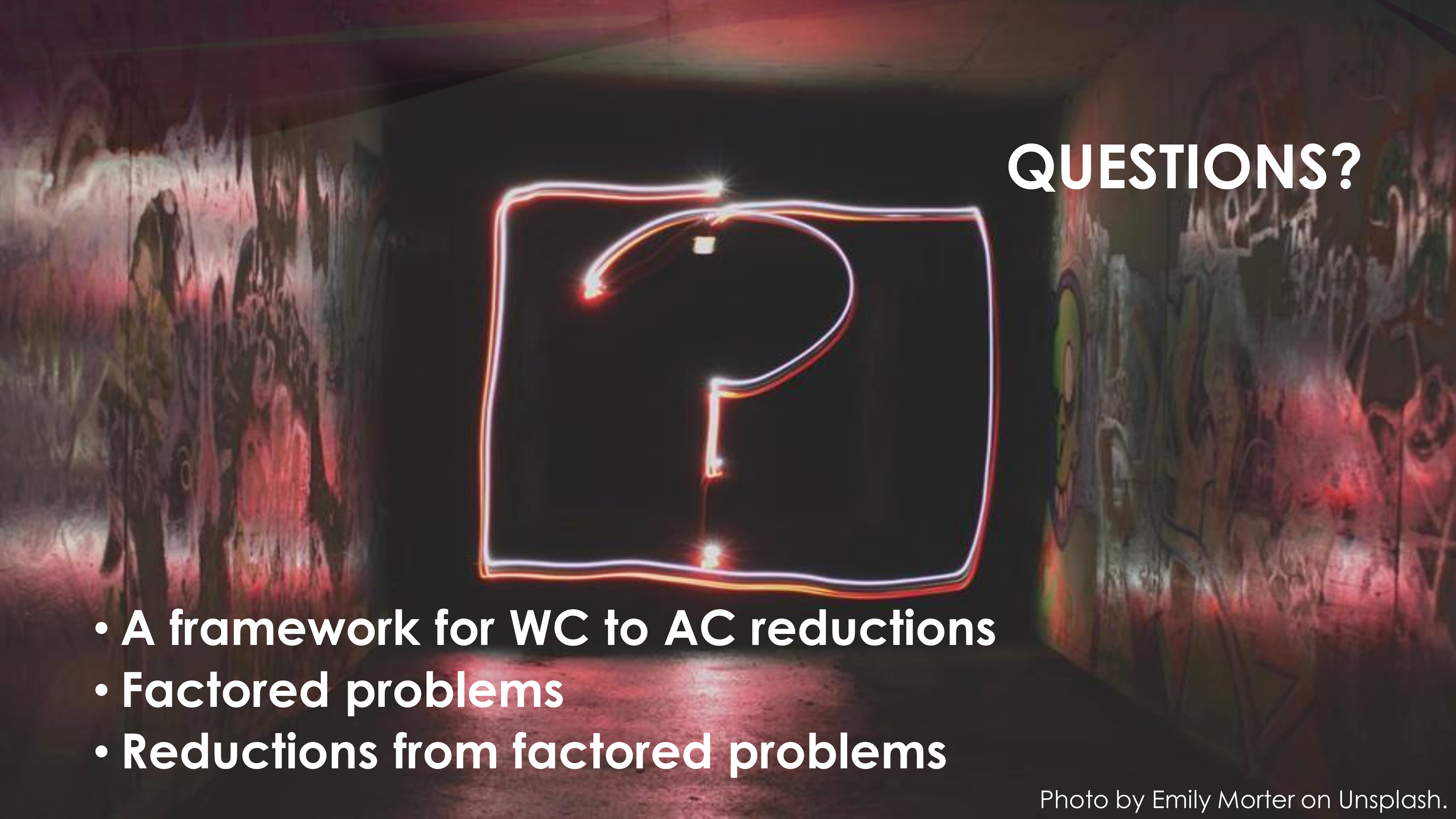
Content not in talk

RECENT WORK + PITCH

Shuichi Hirahara, Nobutaka Shimizu:
Nearly Optimal Average-Case Complexity of Counting Bicliques Under SETH
SODA 2020

Oded Goldreich:
On Counting t-Cliques Mod 2.

ECCC 2020



QUESTIONS?

- A framework for WC to AC reductions
- Factored problems
- Reductions from factored problems

Photo by Emily Morter on Unsplash.