Special topic: Almost-fine-grained Complexity

From Gap-ETH to FPT Inapproximability: Clique, Dominating Set and More

Danupon Nanongkai
KTH, Sweden

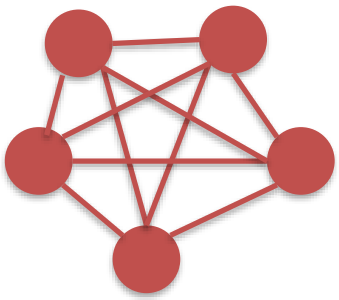
Joint work with with Parinya Chalermsook, Marek Cygan, Guy Kortsarz, Bundit Laekhanukit, Pasin Manurangsi, Luca Trevisan

Big thanks to Parinya Chalermsook for slides!

The problems

The k-clique problem

- **Input**: **n**-vertex undirected graph $G = (V, E)$
- **Output**: A clique of size **k**



The k-clique problem

- **Input**: **n**-vertex undirected graph $G = (V, E)$
- **Output**: A clique of size **k**

A trivial algorithm:

n^k

Enumerate all k-subsets of vertices and check whether it's a clique

$n^{\omega k/3}$

A not-so-trivial improvement:

Reduction to Boolean matrix multiplication

"Enumerative" running time: $n^{\Theta(k)}$ where
k = parameter

Beyond Enumerative Running Time?

- **Unlikely for k-Clique**
  Any $n^{o(k)}$ algorithm would imply $2^{o(n)}$ algorithm for solving 3SAT

Breaking Exponential Time Hypothesis (ETH)

What about approximation algorithms?

- **Input**: $G = (V, E)$
- **Promise**: There is a clique of size **q**
- **Output**: A clique of size **k**

(q,k)-Clique

A simple trick: If there is a clique of size ϵn, possible to beat $n^{o(k)}$

- **Partition** $V = V_1 \cup V_2 \cup \cdots \cup V_{\frac{n}{\log n}}$
- **Return** $\max_j clique(G[V_j])$

at least $\epsilon \log n$

Runtime: poly(n)

Open question: Is there a function F for which $(F(k), k)$-Clique is $n^{o(k)}$ solvable?

Think $F(k) = 2^{2^k}$

K-Dominating Set

- **Input**: $G = (V, E)$
- **Promise**: There is a dominating set of size **k**
- **Output**: A dominating set of size **q**

(k,q)-DomSet

Seems much harder than cliques

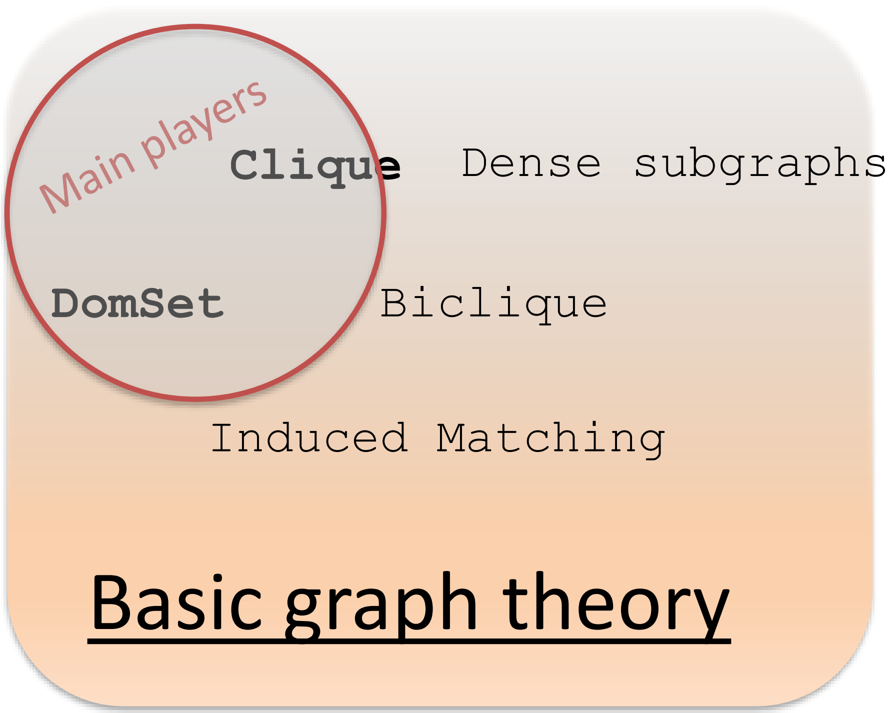
- Solvable exactly in $\boldsymbol{n^{(1+o(1))k}}$ time **[Patrascu-Williams 08]**
- In time $\boldsymbol{n^{k-\epsilon}}$, nothing beyond $(\boldsymbol{k}, \boldsymbol{k} \log \boldsymbol{n})$ −**DomSet**

Open question: Is there a function F for which $(k, F(k))$-DomSet is $n^{o(k)}$ solvable?

Think $F(k) = 2^{2^k}$

Breaking enumeration-type running time for optimization problems?

Generic Problems: [E.g. **Π=maximization problem**]
Find a size-**k** solution for problem Π given promise of size-**q** solution



- Solvable Exactly in $n^{\Theta(k)}$ time
- No non-trivial $(F(k), k) - \Pi$ algorithm in $n^{o(k)}$

Think $F(k) = 2^{2^{2^k}}$

OUR RESULTS

Many aforementioned problems are inherently enumerative

Consequence: FPT Inapproximability

(don't need to remember)

- k= parameter
- $\alpha(k)$-approximation Algorithm
- Running time $t(k)poly(n)$

Key open problems:

Non-trivial FPT approximation for **Clique** or **DomSet**?

- **Clique** is *FPT-inapproximable* if
  - No $o(k)$ approximation in time $t(k)poly(n)$

Our setting: $n^{o(k)}$

Fact:

No improvement over enumerative running time ⇒ No non-trivial FPT approximation

Our Complexity Assumption:

Gap-ETH

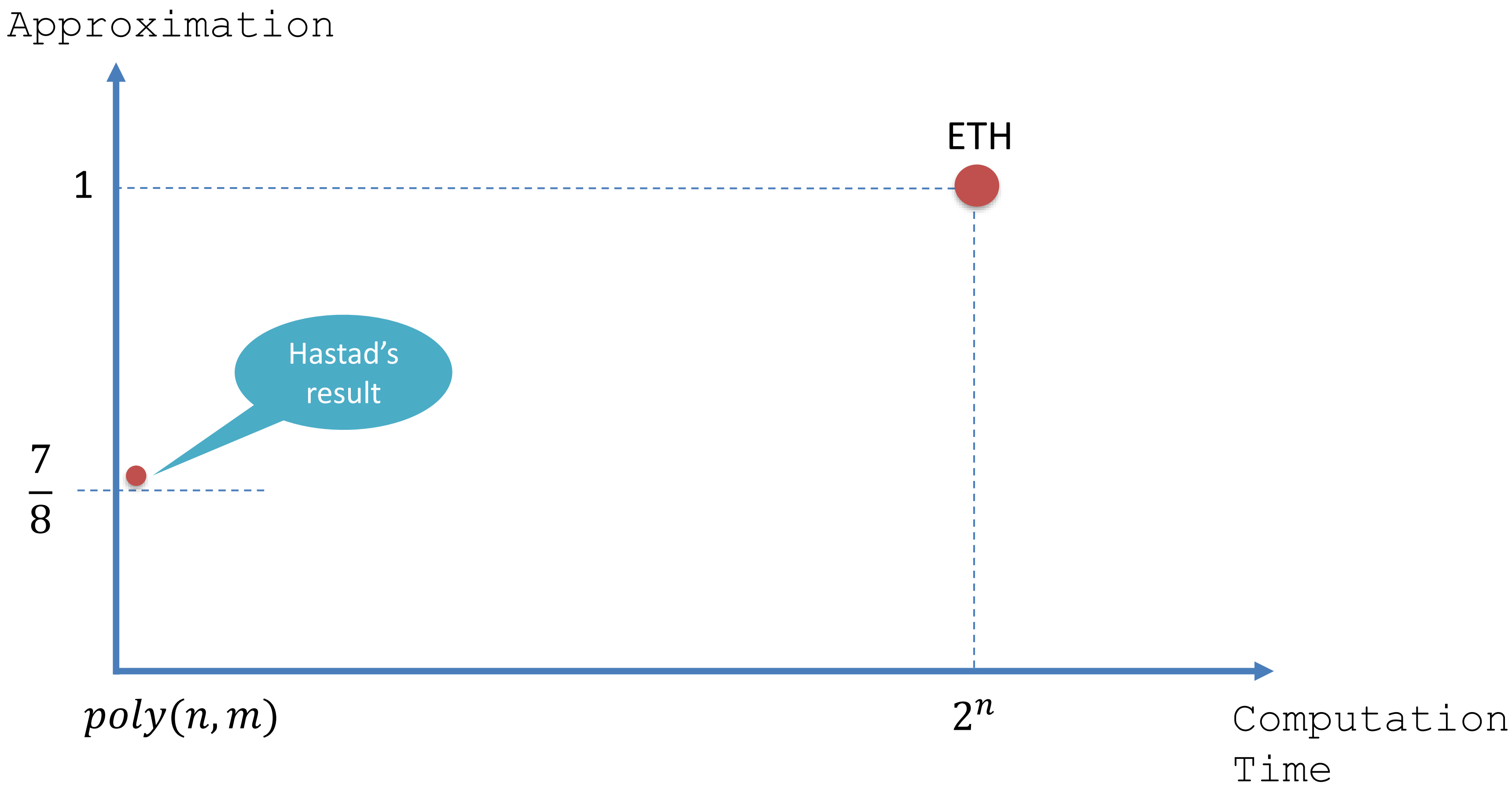
Exponential Time Hypothesis

ETH:
3SAT cannot be decided in time $2^{o(n)}$ or $2^{o(m)}$

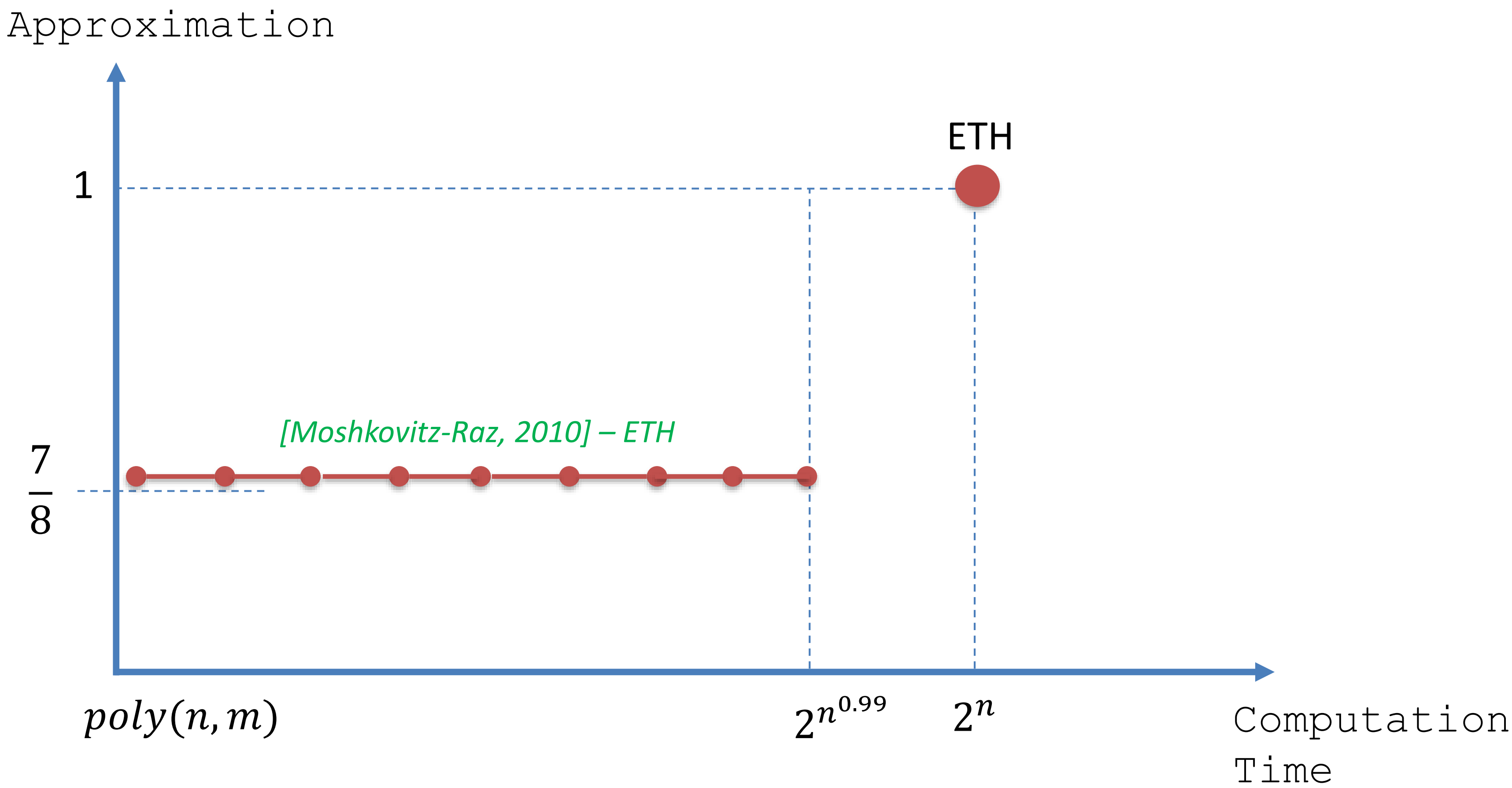
Used for ruling out $n^{o(k)}$ for k-clique

Therefore, we need at least an assumption as strong as ETH to study our question

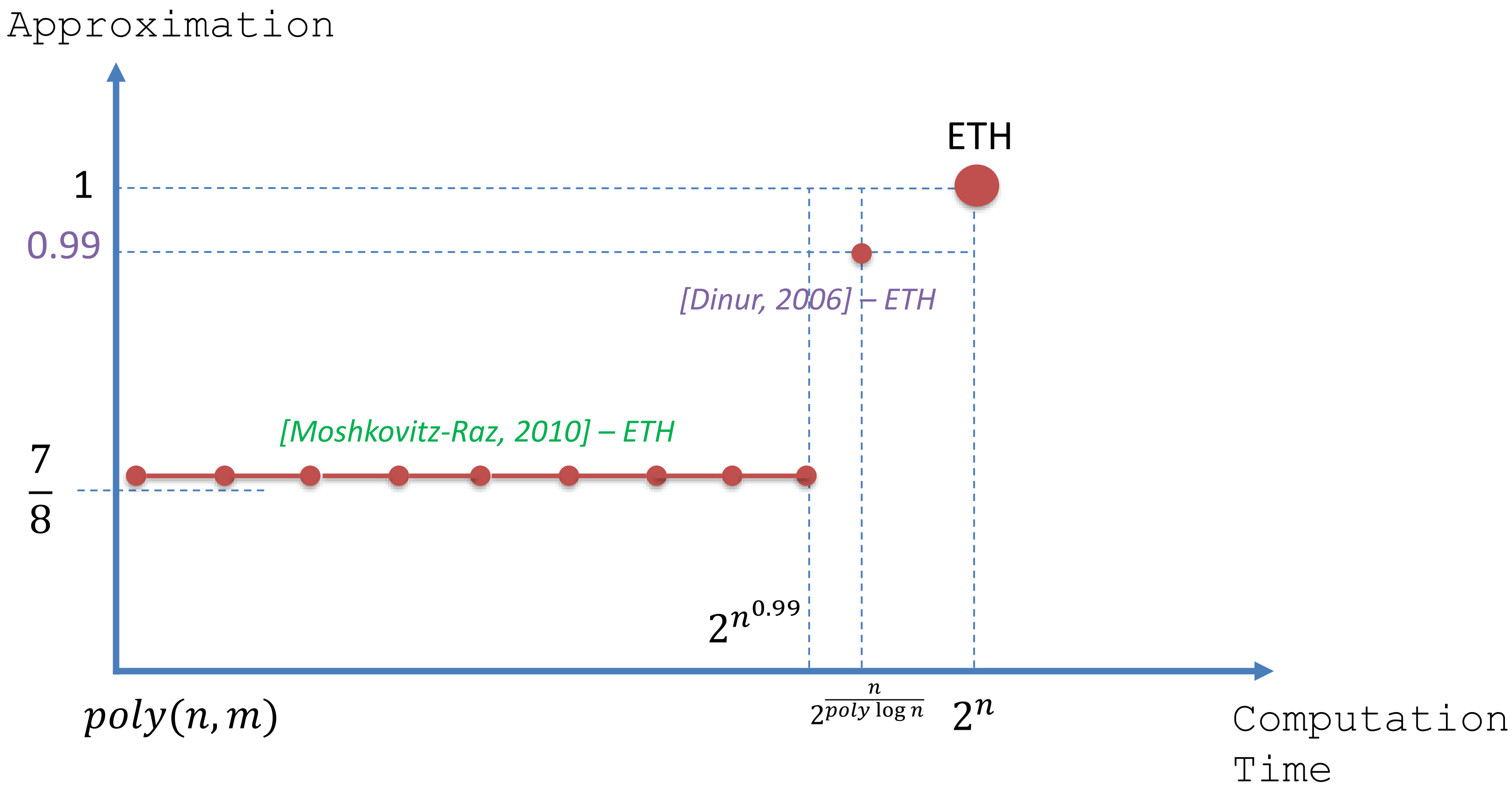
A fine-grained complexity of 3SAT



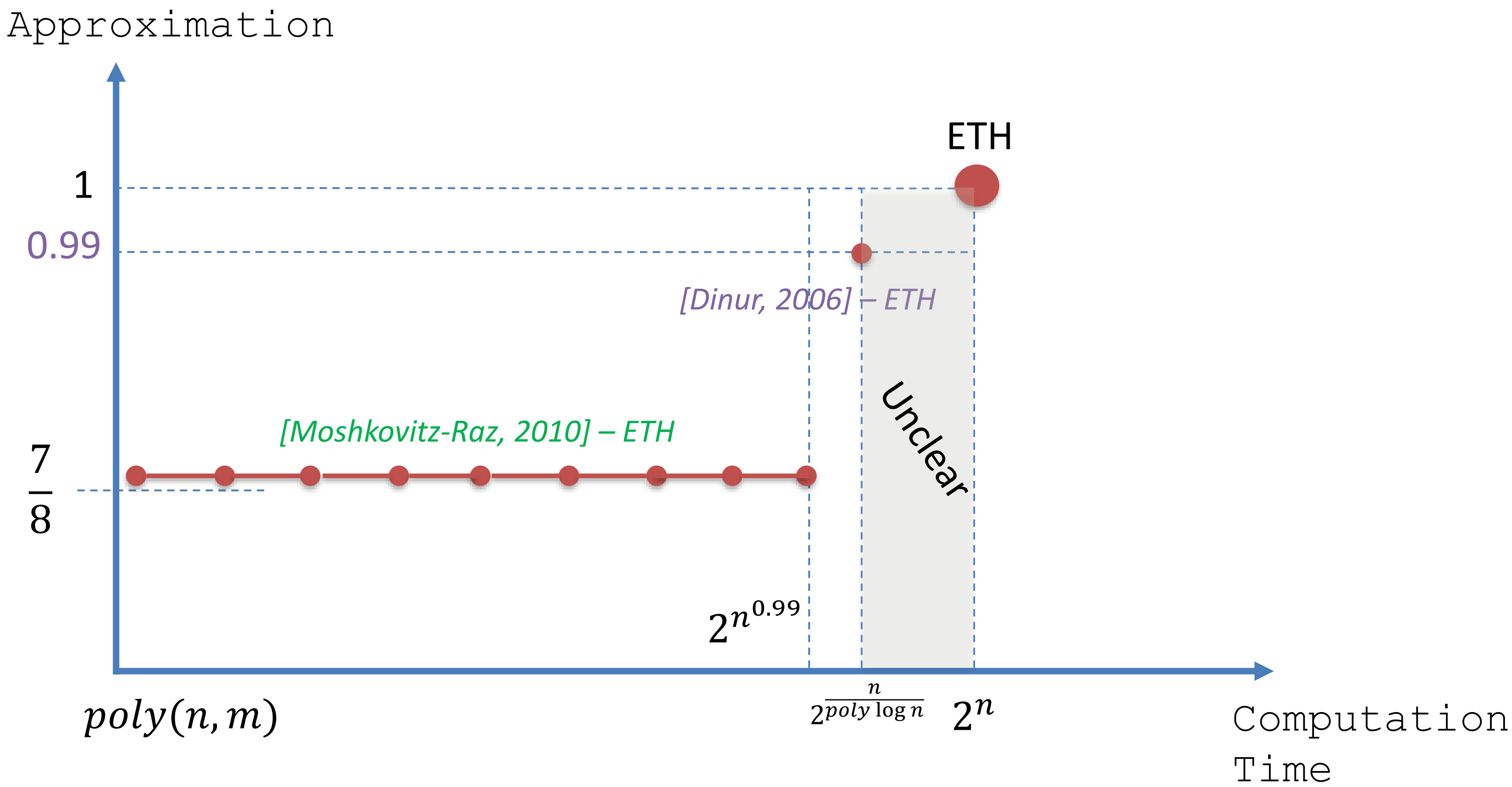
A fine-grained complexity of 3SAT



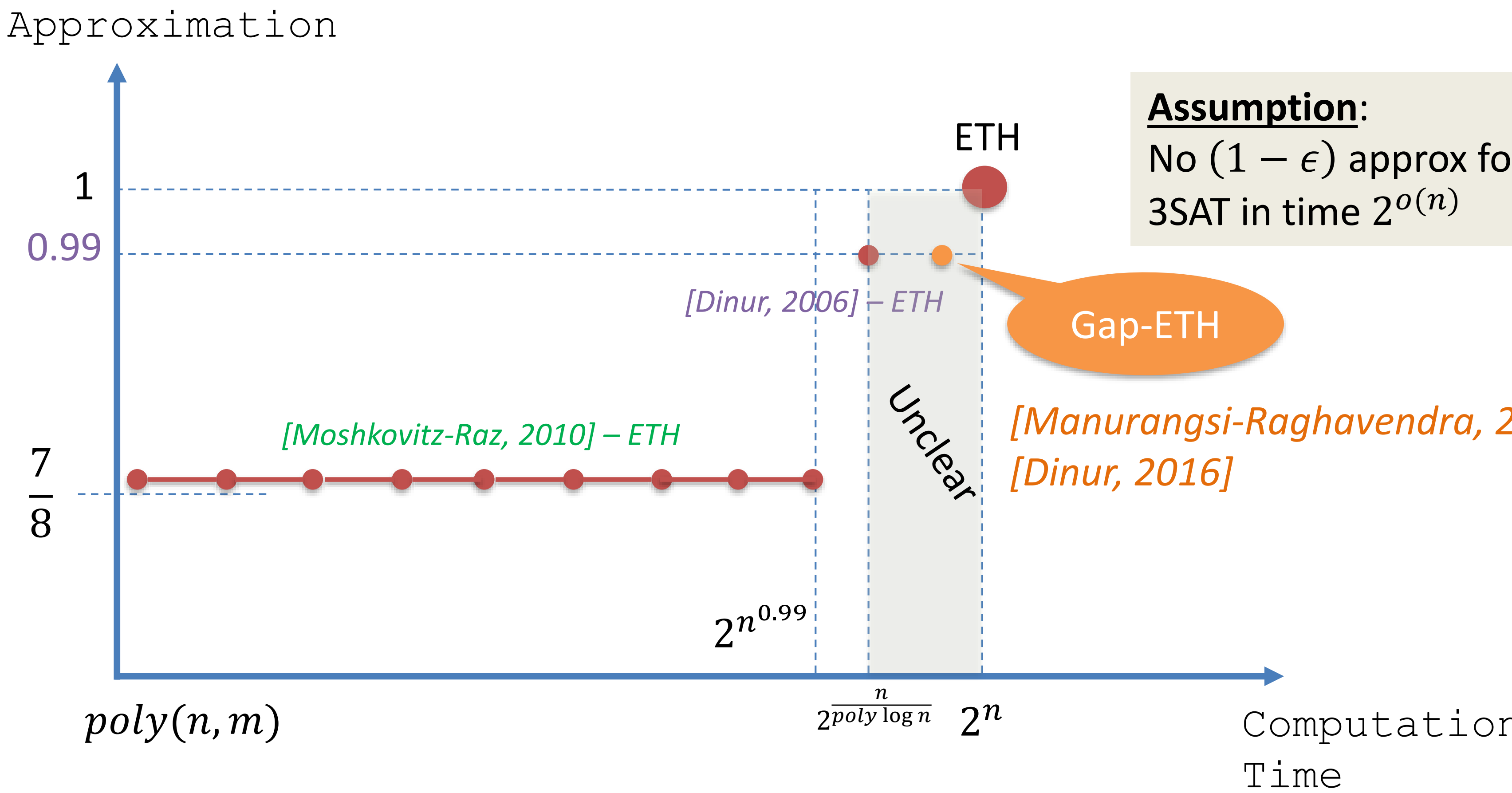
A fine-grained complexity of 3SAT



A fine-grained complexity of 3SAT



A fine-grained complexity of 3SAT



Assumption:
No $(1-\epsilon)$ approx for 3SAT in time $2^{o(n)}$

[Manurangsi-Raghavendra, 2016]
[Dinur, 2016]

Gap-ETH Lower Bounds

- **Inherently enumerative problems**
  - Clique
  - Dominating Set
  - Bipartite Induced Matching
  - Biclique

Knowing existence of $2^{2^{2^k}}$-clique does not help finding k-clique

- **Weakly inherently enumerative problems**
  - Max Induced Subgraph with hereditary properties

Hey, if you don't know what it is, ignore it!

Technique: A reduction from optimization problems on Label Cover instance

A showcase: Clique

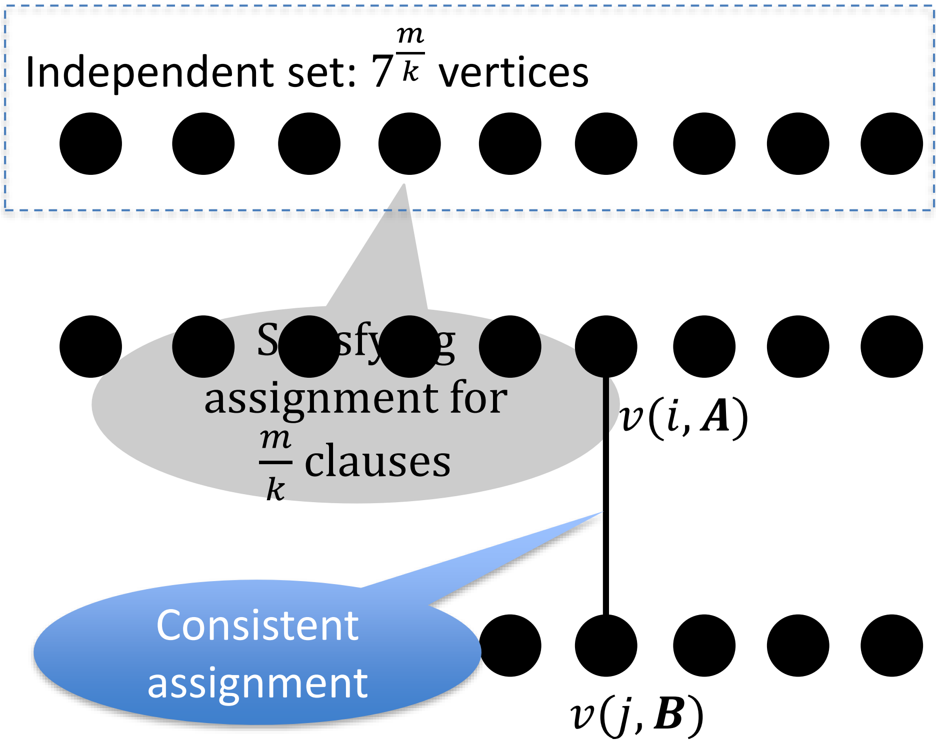
A warm-up: ETH-hardness of k-clique

$$\bigwedge_{i=1\ldots m} C_i \Rightarrow \bigwedge_{i=1\ldots\frac{m}{k}} C_i$$

$\Rightarrow$

(k groups)

$$\Rightarrow \bigwedge_{i=(m-\frac{m}{k})\ldots m} C_i$$



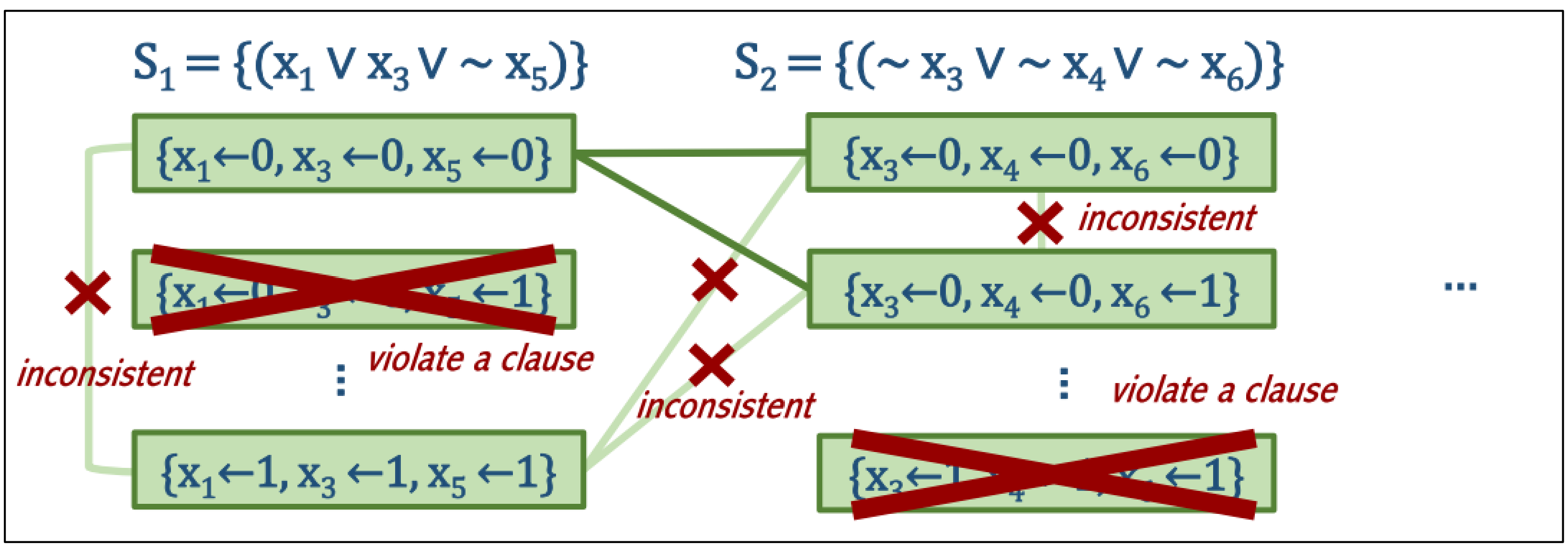
Step 1: Partitioning 3SAT formula into k groups

Step 2: Create a graph

$$|V(G)| = k7^{m/k}$$

Example (with $|I_j| = 1$)

$(x_1 \lor x_3 \lor \sim x_5)$
$\land\ (\sim x_3 \lor \sim x_4 \lor \sim x_6)$
$\land\ (x_2 \lor \sim x_4 \lor x_6)$
$\land\ \ldots$



Compression

- **Reduction from 3SAT to CLIQUE**
  - **<u>Size</u>**: $\mathrm{N} = \; k\, 2^{O\left(\frac{m}{k}\right)}$
  - **<u>Solutions</u>**: Clique of size **k** if and only if 3SAT formula is satisfiable

<u>ETH-hardness</u>:

Solving k-Clique in time $\left(\frac{N}{k}\right)^{o(k)}$ implies deciding 3SAT in time $2^{o(m)}$

Gap-ETH Hardness

$$\bigwedge_{i=1\ldots m} C_i$$

Random I_1

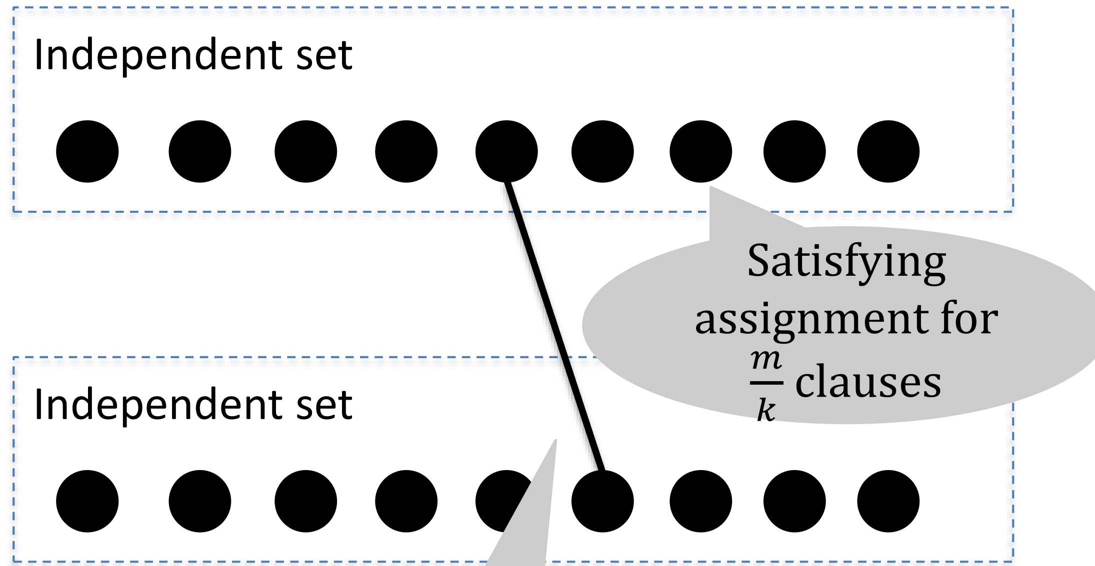
$|I_1| = \frac{m}{\epsilon k}$

$$\bigwedge_{j\in I_1} C_j$$

Random I_2

$$\bigwedge_{j\in I_2} C_j$$

Keep doing this for q rounds



Analysis

- **Completeness**: If 3SAT has satisfying assignment, then there is a **q**-clique

Soundness:

The probability of having a clique of size 100k is very low … (exercises)

Concluding remarks

Take-home message

ETH → Exact lower bound in time $n^{o(k)}$

Gap-ETH → Approximation lower bound in time $n^{o(k)}$

Our work!

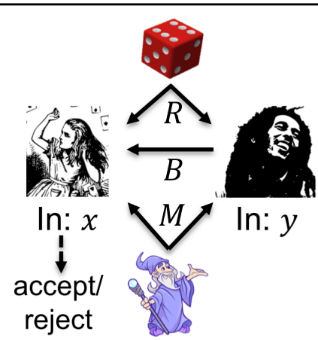
Follow-up: Dominating Set

[C.S., Manurangsi, Laekhanukit, 2018]

Hypothesis	Inapprox	Running Time	
W[1]≠FPT	$\Omega(\log^{1/poly(k)} n)$	FPT-Time (T(k) poly(n))	
ETH	$\Omega(\log^{1/poly(k)} n)$	$n^{o(k)}$	
SETH	$\Omega(\log^{1/f(k)} n)$	$n^{k-\varepsilon}$	for any $\varepsilon > 0$
k-SUM	$\Omega(\log^{1/poly(k)} n)$	$n^{k/2-\varepsilon}$	for any $\varepsilon > 0$

- SETH → No $(\log n)^{\frac{1}{poly(k)}}$ approximation for k-DomSet in time $n^{k-\epsilon}$
- ETH → Inherently enumerative
- $W[1] \neq FPT$ → FPT-inapproximability

Tool from fine-grained complexity:
"Distributed PCP" [Abboud, Rubinstein, Williams'17]
(See Karl's lecture)



Thank you!