

Sampling and Counting Solutions to Constraint Satisfaction Problems

约束满足问题解的采样与计数



国立情報学研究所 王淳扬
Chunyang Wang, National Institute of Informatics

2026年CCF全国理论计算机科学学术年会
NCTCS 2026



Constraint Satisfaction Problem

$$\Phi = (V, Q, \mathcal{C})$$

Variables: $V = \{v_1, v_2, \dots, v_n\}$ with **finite** domains Q_v for each $v \in V$

Constraints: $\mathcal{C} = \{c_1, c_2, \dots, c_m\}$ with each $c \in \mathcal{C}$ defined on $\text{vbl}(c) \subseteq V$

$$c : \bigotimes_{v \in \text{vbl}(c)} Q_v \rightarrow \{\text{satisfied}, \text{not satisfied}\}$$

CSP solution: assignment $X \in \bigotimes_{v \in V} Q_v$ s.t. all constraints are **satisfied**

Constraint Satisfaction Problem

$$\Phi = (V, Q, \mathcal{C})$$

Variables: $V = \{v_1, v_2, \dots, v_n\}$ with **finite** domains Q_v for each $v \in V$

Constraints: $\mathcal{C} = \{c_1, c_2, \dots, c_m\}$ with each $c \in \mathcal{C}$ defined on $\text{vbl}(c) \subseteq V$

$$c : \bigotimes_{v \in \text{vbl}(c)} Q_v \rightarrow \{\text{satisfied}, \text{not satisfied}\}$$

CSP solution: assignment $X \in \bigotimes_{v \in V} Q_v$ s.t. all constraints are **satisfied**

“

Constraint satisfaction is a subject of **central significance** in computer science, since a very large number of combinatorial problems, starting from Boolean Satisfiability and Graph Coloring, can be phrased as constraint satisfaction problems (CSP).

—2021 Gödel Prize official citation

Constraint Satisfaction Problem

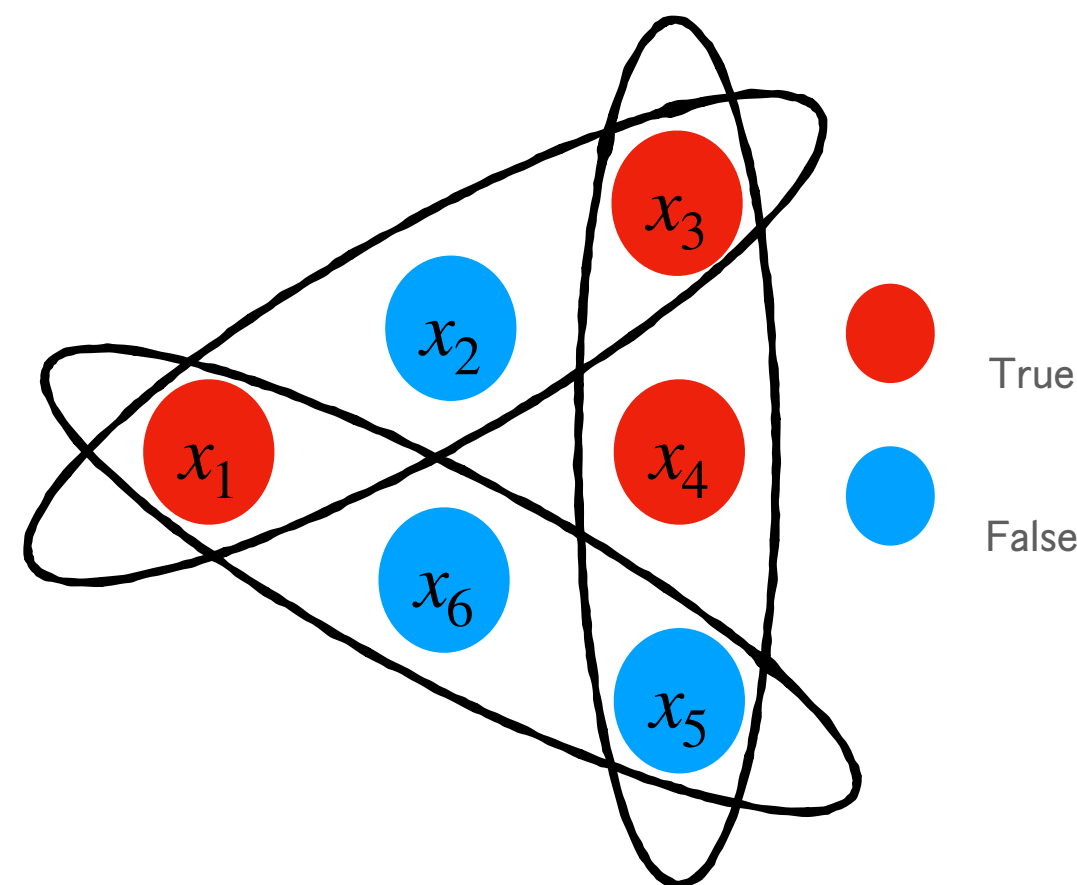
$$\Phi = (V, Q, \mathcal{C})$$

Variables: $V = \{v_1, v_2, \dots, v_n\}$ with **finite** domains Q_v for each $v \in V$

Constraints: $\mathcal{C} = \{c_1, c_2, \dots, c_m\}$ with each $c \in \mathcal{C}$ defined on $\text{vbl}(c) \subseteq V$

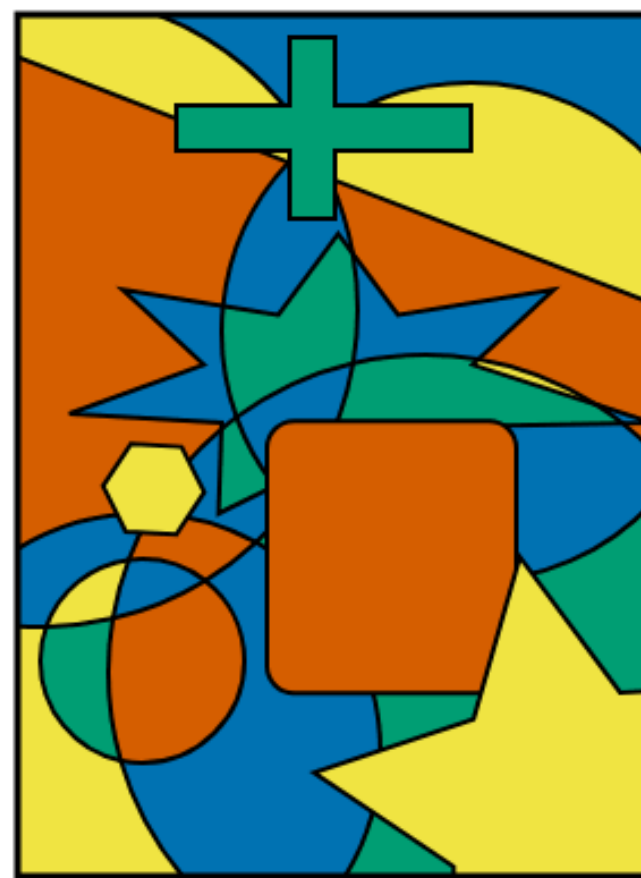
$$c : \bigotimes_{v \in \text{vbl}(c)} Q_v \rightarrow \{\text{satisfied}, \text{not satisfied}\}$$

CSP solution: assignment $X \in \bigotimes_{v \in V} Q_v$ s.t. all constraints are **satisfied**

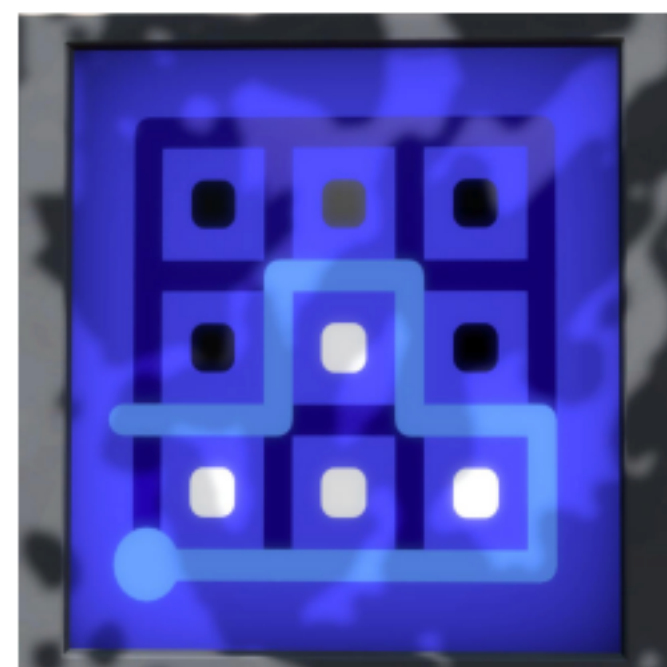
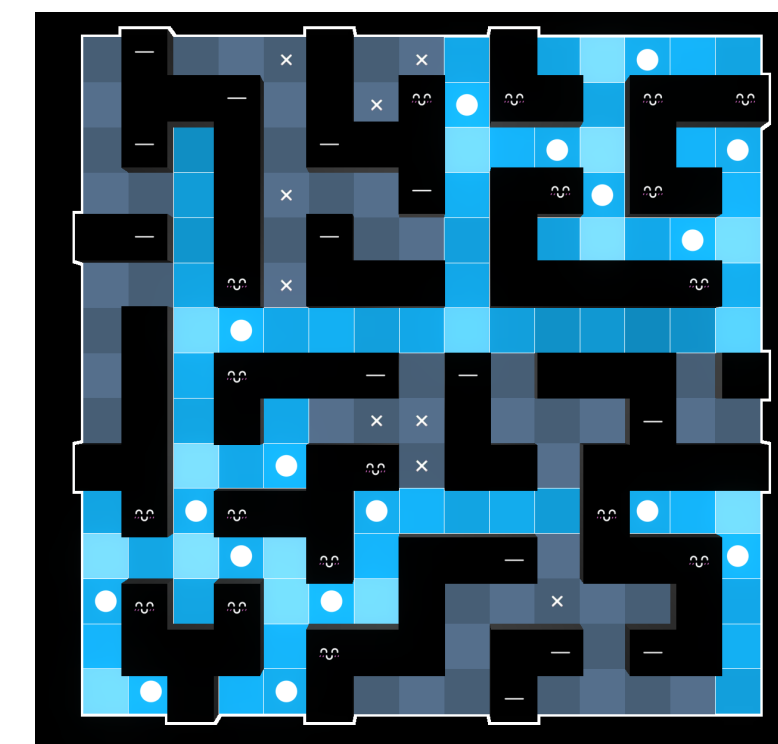
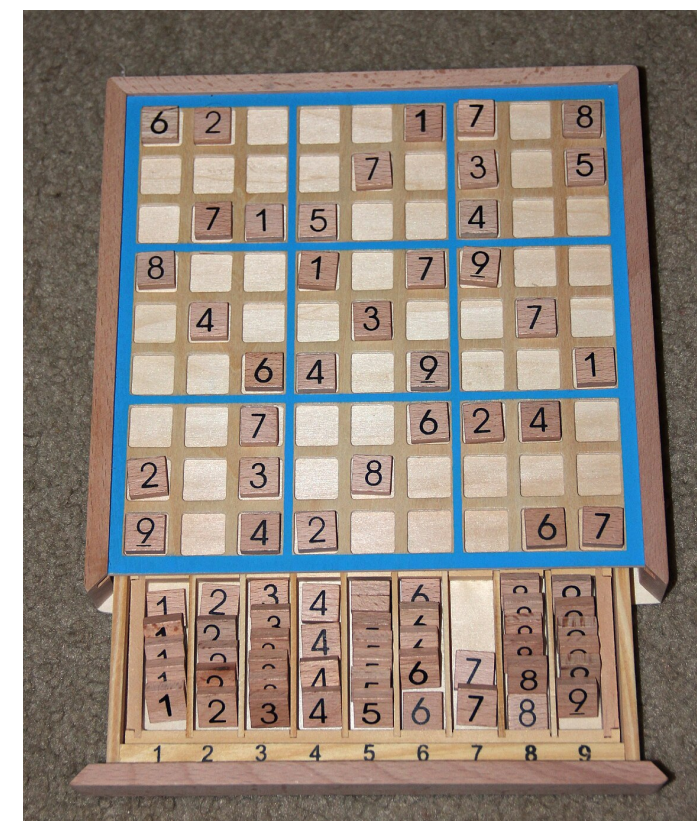


$$\Phi = (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_5 \vee \neg x_6) \wedge (x_3 \vee \neg x_4 \vee \neg x_5)$$

Boolean Satisfiability (SAT)



Graph Coloring



Logic Puzzles

CSP in TCS

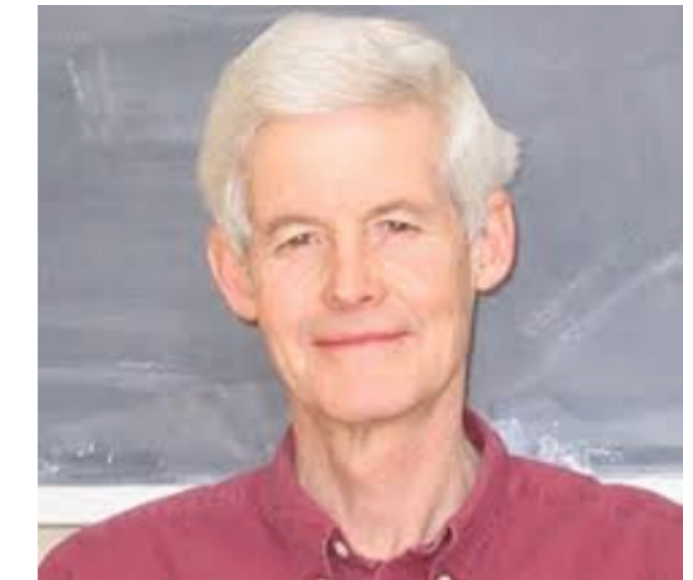
A **leitmotif** of TCS: What makes a problem computationally easy/hard?



Decision problem:
Can we efficiently decide
if CSP has a solution?

Appears in
Proceedings Third Annual
ACM Symposium
Theory of Computing
May, 1971

The Complexity of Theorem-Proving Procedures
Stephen A. Cook
University of Toronto



Cook-Levin Theorem: Boolean satisfiability is **NP**-complete
Turing Award 1982, foundation of computational complexity

SIAM J. COMPUT.
Vol. 28, No. 1, pp. 57–104

© 1998 Society for Industrial and Applied Mathematics

THE COMPUTATIONAL STRUCTURE OF MONOTONE MONADIC
SNP AND CONSTRAINT SATISFACTION: A STUDY THROUGH
DATALOG AND GROUP THEORY*

TOMÁS FEDER[†] AND MOSHE Y. VARDI[‡]

58th Annual IEEE Symposium on Foundations of Computer Science

A dichotomy theorem for nonuniform CSPs

Andrei A. Bulatov
School of Computing Science
Simon Fraser University

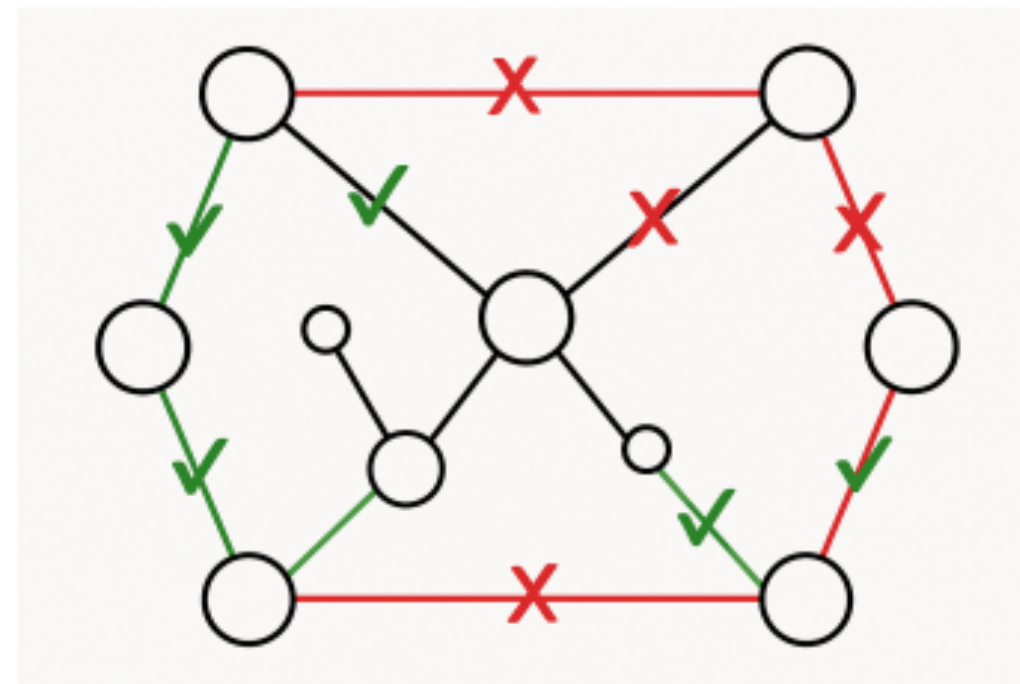
A Proof of the CSP Dichotomy Conjecture

DMITRIY ZHUK, Lomonosov Moscow State University, Russia

CSP Dichotomy Theorem: Every $\text{CSP}(\Gamma)$ is either in **P** or **NP**-complete
characterized through algebraic properties of **constraint languages** Γ

CSP in TCS

A **leitmotif** of TCS: What makes a problem computationally easy/hard?



Optimization problem:
Can we efficiently
approximately compute a solution?

Interactive Proofs and the Hardness of Approximating Cliques

URIEL FEIGE
The Weizmann Institute, Rehovot, Israel
SHAFI GOLDWASSER
Massachusetts Institute of Technology, Cambridge, Massachusetts
LASZLO LOVÁSZ
Yale University, New Haven, Connecticut
SHMUEL SAFRA
Tel-Aviv University, Tel-Aviv, Israel
AND
MARIO SZEGEDY
AT&T Bell Laboratories, Murray Hill, New Jersey

Probabilistic Checking of Proofs: A New Characterization of NP

SANJEEV ARORA
Princeton University, Princeton, New Jersey
AND
SHMUEL SAFRA
Tel-Aviv University, Tel-Aviv, Israel

The PCP Theorem by Gap Amplification

IRIT DINUR
The Hebrew University, Jerusalem, Israel

Proof Verification and the Hardness of Approximation Problems

SANJEEV ARORA
Princeton University, Princeton, New Jersey
CARSTEN LUND
AT&T Bell Laboratories, Murray Hill, New Jersey
RAJEEV MOTWANI
Stanford University, Stanford, California
MADHU SUDAN
Massachusetts Institute of Technology, Cambridge, Massachusetts
AND
MARIO SZEGEDY
AT&T Bell Laboratories, Murray Hill, New Jersey

PCP Theorem: foundation of hardness of approximation

Gödel Prize 2001, 2019

Optimal Algorithms and Inapproximability Results for Every CSP?

[Extended Abstract]

Prasad Raghavendra *†
Dept. of Computer Science and Eng.
University of Washington

Raghavendra's Theorem (informal): Assuming the **UGC**, a canonical SDP characterizes the optimal approximation ratio of every finite-domain CSP.

CSP in TCS

A **leitmotif** of TCS: What makes a problem computationally easy/hard?

SIAM J. COMPUT.
Vol. 20, No. 5, pp. 865–877, October 1991

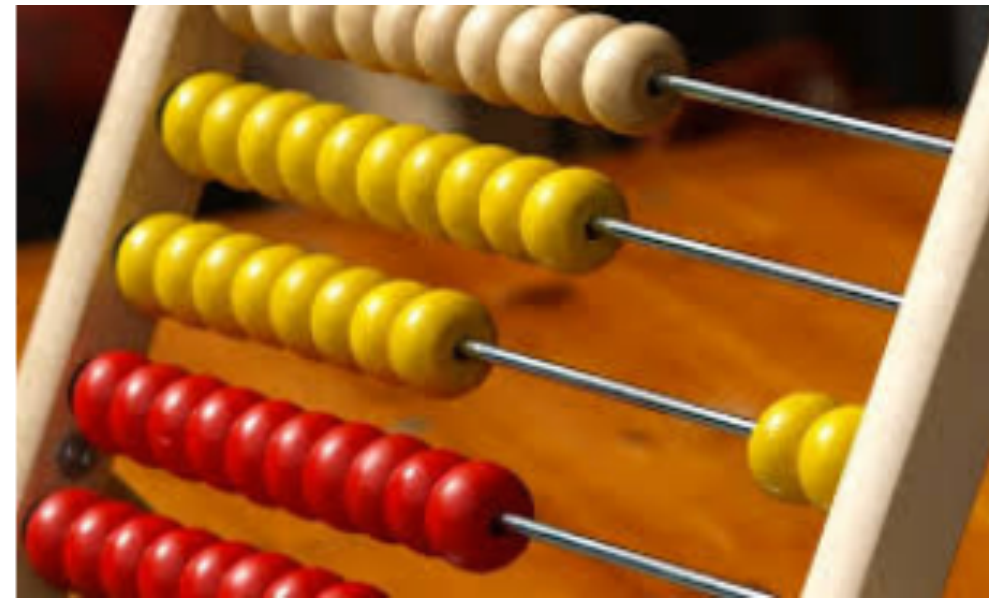
© 1991 Society for Industrial and Applied Mathematics
004

PP IS AS HARD AS THE POLYNOMIAL-TIME HIERARCHY*

SEINOSUKE TODA†

Toda's Theorem: $\text{PH} \subseteq \text{P}^{\#P}$

Gödel Prize 1998



Exact Counting problem:
Can we efficiently count
the number of solutions?

The Complexity of the Counting Constraint Satisfaction Problem

ANDREI A. BULATOV, Simon Fraser University

An Effective Dichotomy for the
Counting Constraint Satisfaction Problem

Martin Dyer*

David Richerby†

Complexity of Counting CSP with Complex Weights

Jin-Yi Cai
University of Wisconsin – Madison
jyc@cs.wisc.edu

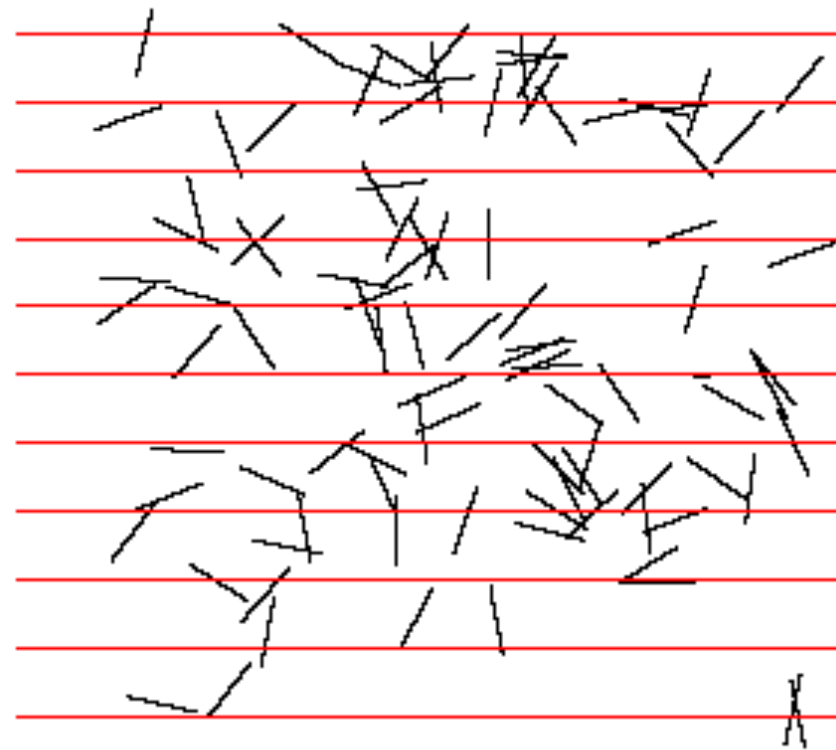
Xi Chen
Columbia University
xichen@cs.columbia.edu

Counting Dichotomy Theorem: Every $\#CSP(\Gamma)$ is either in **FP** or **#P**-complete.

Gödel Prize 2021

CSP in TCS

A **leitmotif** of TCS: What makes a problem computationally easy/hard?

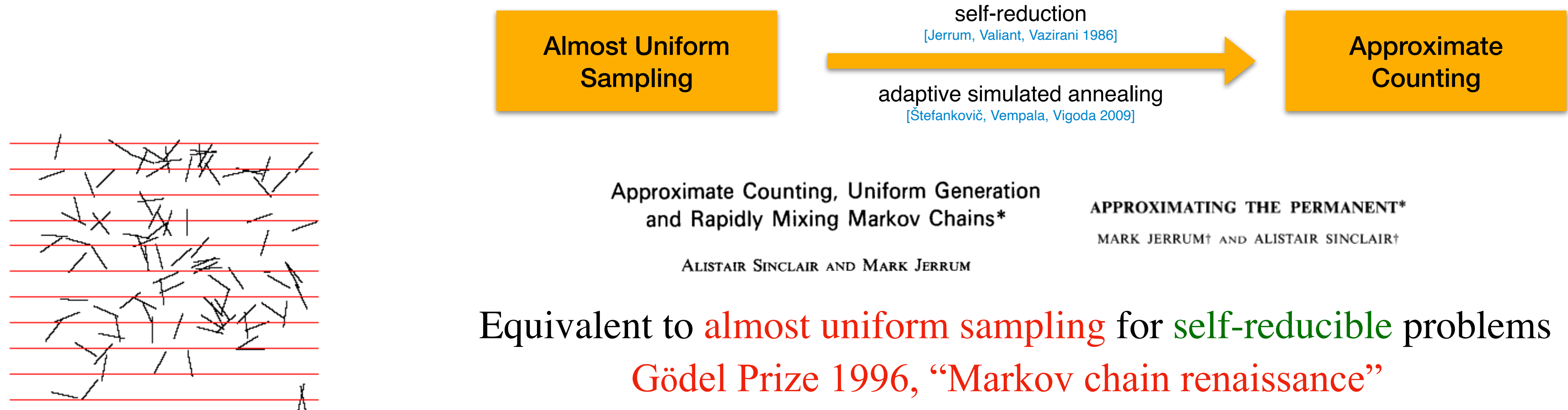


Approximate Counting problem:
Can we efficiently estimate
the number of solutions?

Our focus!

CSP in TCS

A **leitmotif** of TCS: What makes a problem computationally easy/hard?



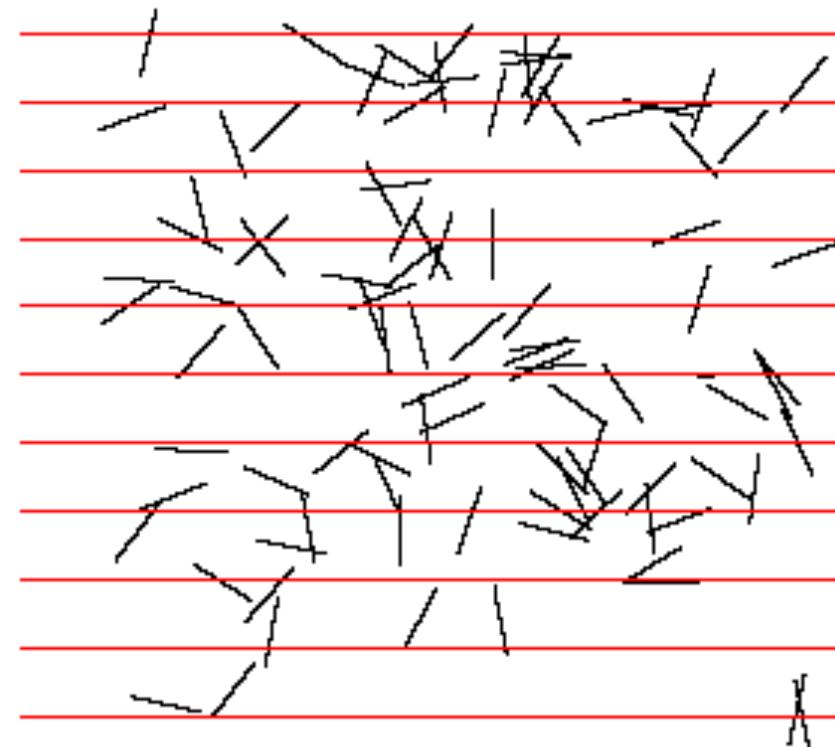
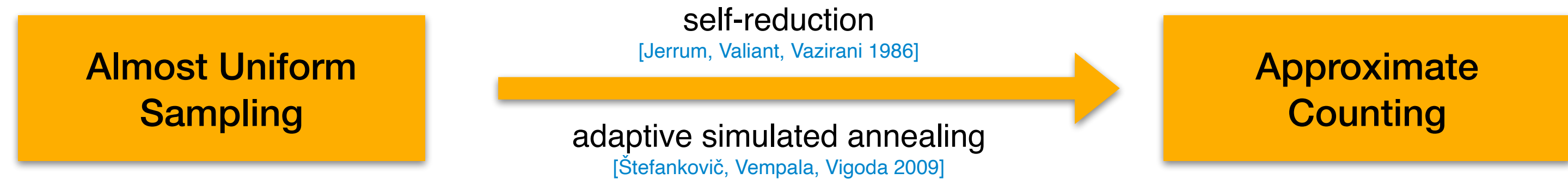
Equivalent to **almost uniform sampling** for **self-reducible** problems
Gödel Prize 1996, “Markov chain renaissance”

Approximate Counting problem:
Can we efficiently estimate
the number of solutions?

Our focus!

CSP in TCS

A **leitmotif** of TCS: What makes a problem computationally easy/hard?



Approximate Counting, Uniform Generation
and Rapidly Mixing Markov Chains*

ALISTAIR SINCLAIR AND MARK JERRUM

APPROXIMATING THE PERMANENT*

MARK JERRUM† AND ALISTAIR SINCLAIR†

Equivalent to **almost uniform sampling** for **self-reducible** problems
Gödel Prize 1996, “Markov chain renaissance”

Approximate Counting problem:
Can we efficiently estimate
the number of solutions?

Our focus!

An approximation trichotomy for Boolean #CSP[☆]

Martin Dyer^a, Leslie Ann Goldberg^b, Mark Jerrum^{c,*}

Boolean Approximate Counting **Trichotomy** Theorem:
Every Boolean #CSP(Γ) is: 1. in **FP** 2. **#BIS**-equivalent 3. **#SAT**-equivalent

**#BIS: approximately counting # of
independent sets in a bipartite graph**

Lovász Local Lemma

$$\Phi = (V, Q, \mathcal{C}) \quad \text{Parameters}$$

Variable framework

- each $v \in V$ draws from Q_v **uniformly** and **independently** at random
- product distribution $\mathcal{P}$

- **violation probability** $p = \max_{c \in \mathcal{C}} \Pr_{\mathcal{P}}[\neg c]$

- **dependency degree**

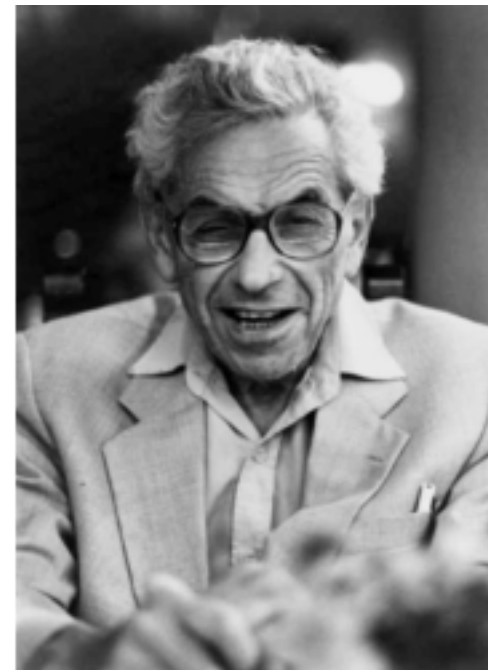
$$D = \max_{c \in \mathcal{C}} |\{c' \in \mathcal{C} \setminus \{c\} \mid \text{vbl}(c) \cap \text{vbl}(c') \neq \emptyset\}|$$

Lovász Local Lemma

$\Phi = (V, Q, \mathcal{C})$ Parameters

Variable framework

- each $v \in V$ draws from Q_v **uniformly** and **independently** at random
- product distribution $\mathcal{P}$



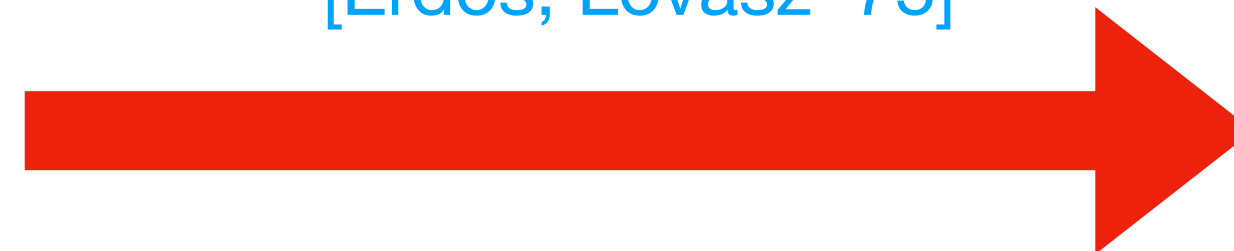
- **violation probability** $p = \max_{c \in \mathcal{C}} \Pr_{\mathcal{P}}[\neg c]$

- **dependency degree**

$$D = \max_{c \in \mathcal{C}} |\{c' \in \mathcal{C} \setminus \{c\} \mid \text{vbl}(c) \cap \text{vbl}(c') \neq \emptyset\}|$$

$$ep(D + 1) \leq 1$$

Lovász Local Lemma
[Erdős, Lovász '75]



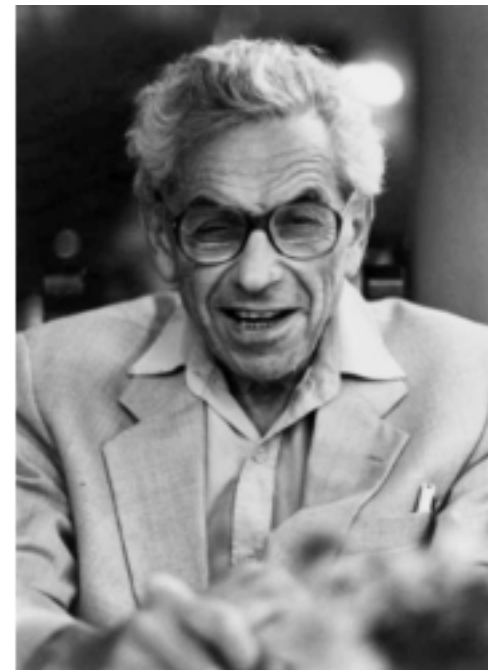
A CSP solution exists!

Lovász Local Lemma

$\Phi = (V, Q, \mathcal{C})$ Parameters

Variable framework

- each $v \in V$ draws from Q_v **uniformly** and **independently** at random
- product distribution $\mathcal{P}$



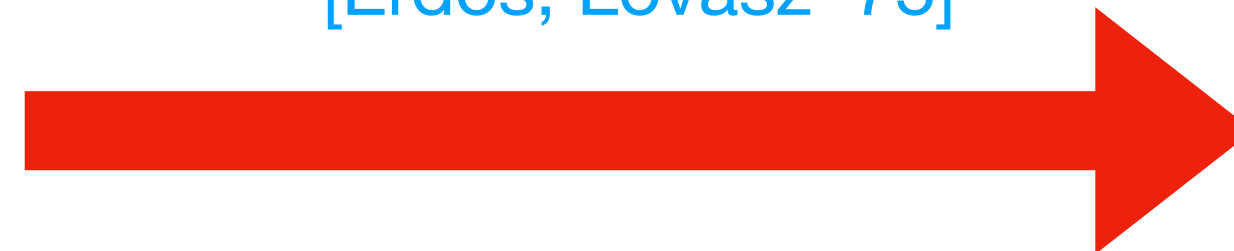
- **violation probability** $p = \max_{c \in \mathcal{C}} \Pr_{\mathcal{P}}[\neg c]$

- **dependency degree**

$$D = \max_{c \in \mathcal{C}} |\{c' \in \mathcal{C} \setminus \{c\} \mid \text{vbl}(c) \cap \text{vbl}(c') \neq \emptyset\}|$$

$$ep(D + 1) \leq 1$$

Lovász Local Lemma
[Erdős, Lovász '75]



A CSP solution exists!

Proof by **probabilistic method**...
Solution can be **exponentially sparse**!



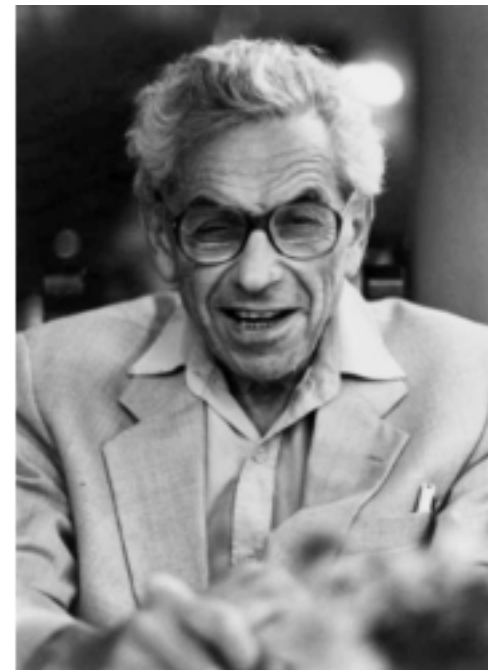
Finding needle in a haystack?

Lovász Local Lemma

$\Phi = (V, Q, \mathcal{C})$ Parameters

Variable framework

- each $v \in V$ draws from Q_v **uniformly** and **independently** at random
- product distribution $\mathcal{P}$



- violation probability** $p = \max_{c \in \mathcal{C}} \Pr[\neg c]$

- dependency degree**

$$D = \max_{c \in \mathcal{C}} |\{c' \in \mathcal{C} \setminus \{c\} \mid \text{vbl}(c) \cap \text{vbl}(c') \neq \emptyset\}|$$

$$ep(D + 1) \leq 1$$

Lovász Local Lemma
[Erdős, Lovász '75]

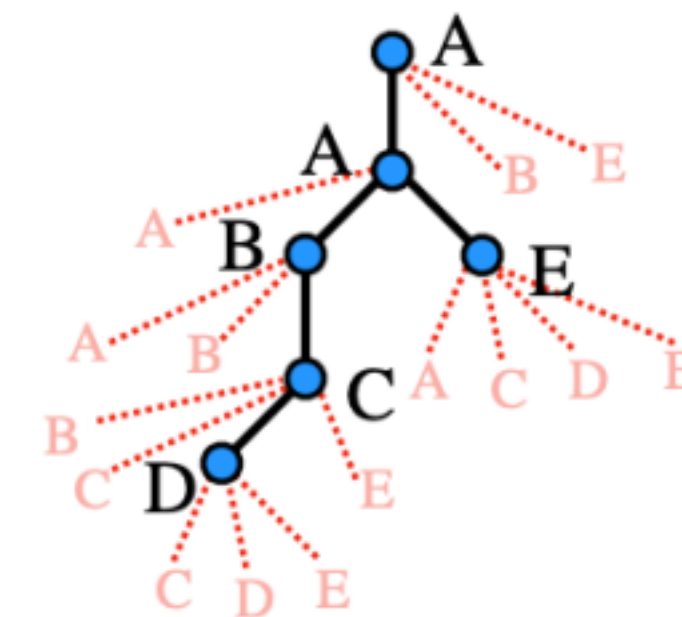


Algorithmic Lovász Local Lemma
[Moser, Tardos '10]
Gödel Prize 2020

A CSP solution exists
and can be efficiently found!

Moser-Tardos Algorithm:

draw independent samples of $X_1, \dots, X_n$;
while $\exists$ a bad event A_i that occurs:
resample all $X_j \in \text{vbl}(A_i)$;

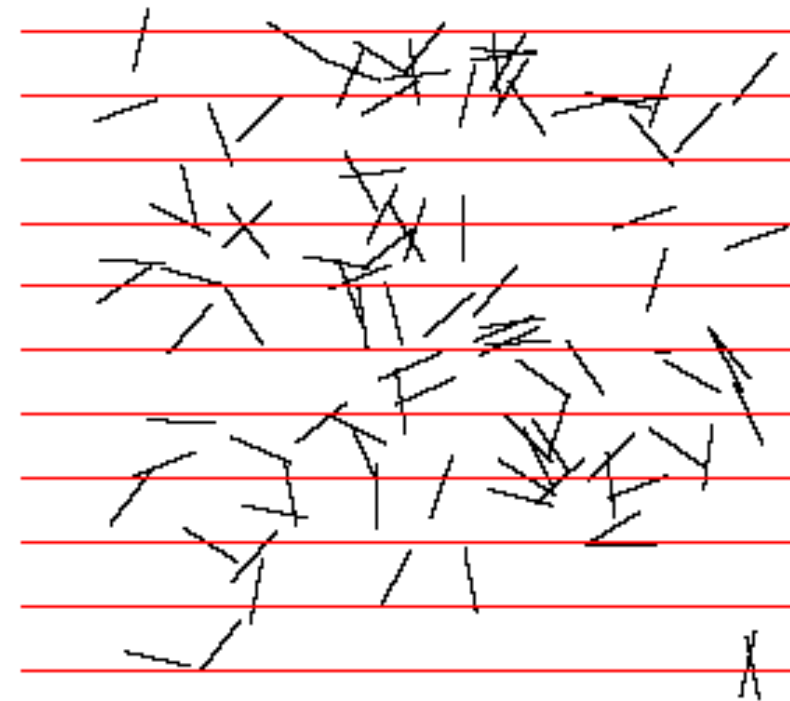


Counting/Sampling Lovász Local Lemma

Counting LLL: approximately count the number of CSP solutions

Sampling LLL: almost uniformly sample a random CSP solution

under LLL-like conditions



$$pD^c \lesssim 1$$

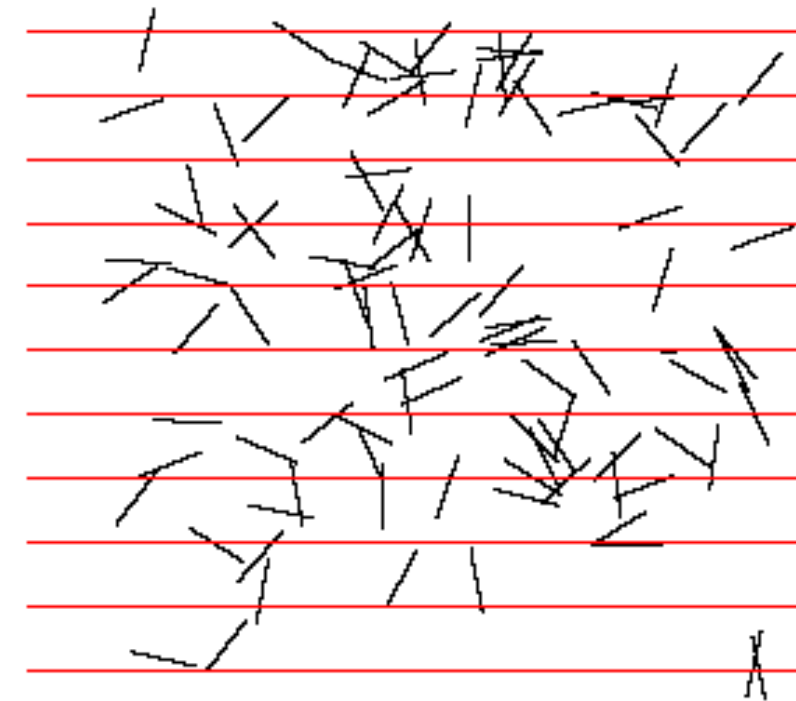


Counting/Sampling Lovász Local Lemma

Counting LLL: approximately count the number of CSP solutions

Sampling LLL: almost uniformly sample a random CSP solution

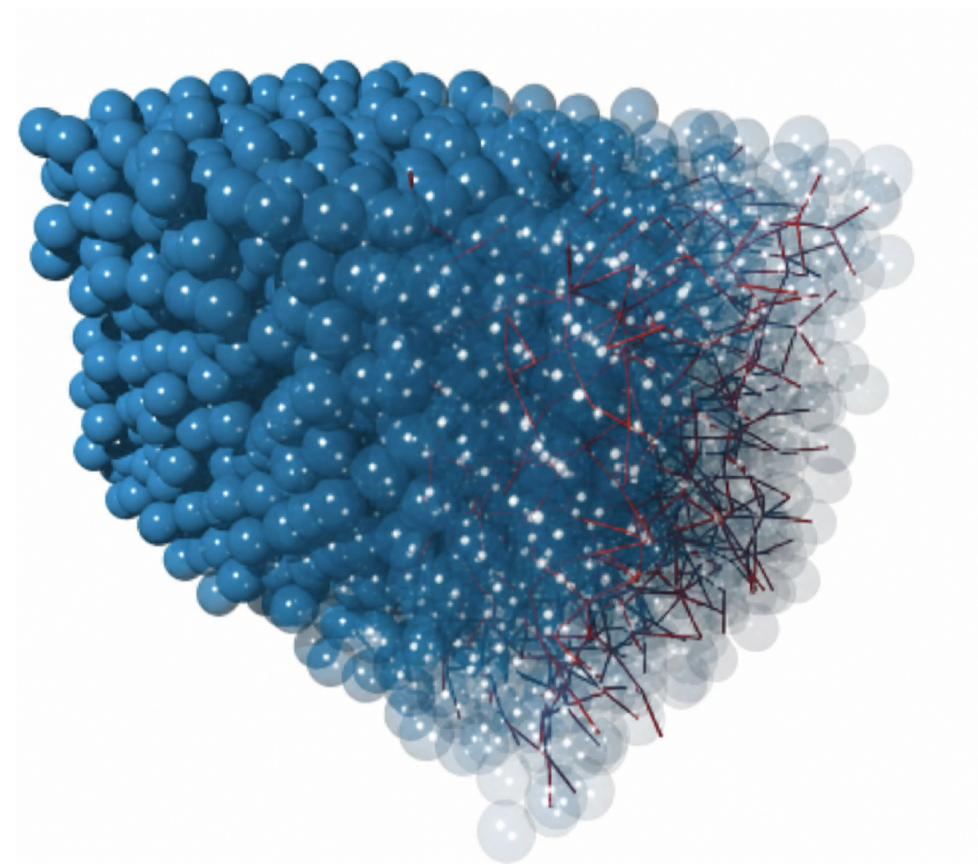
under LLL-like conditions



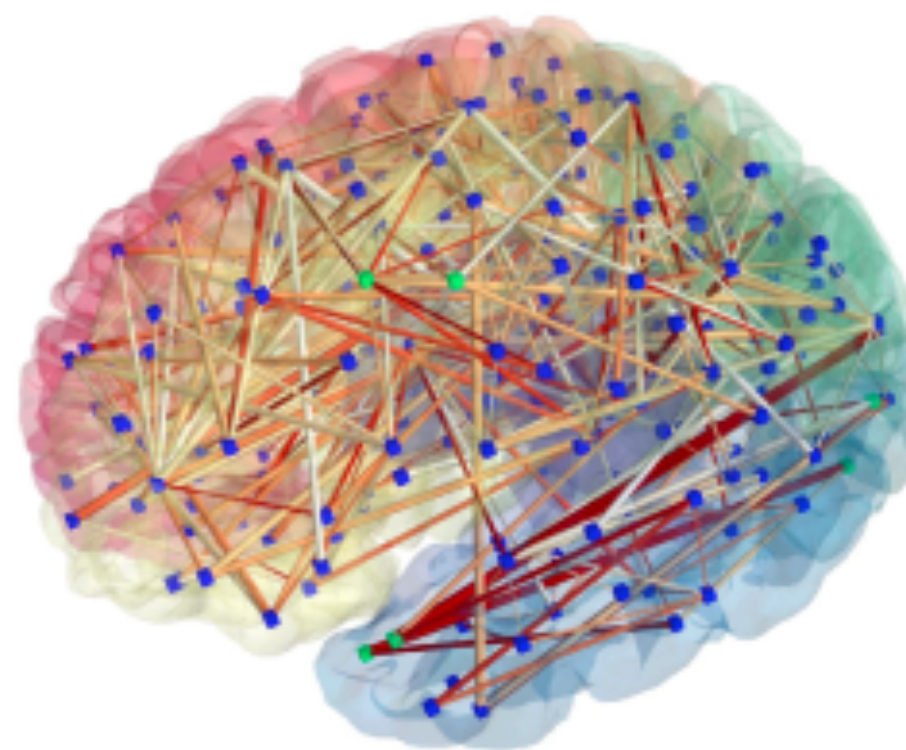
$$pD^c \lesssim 1$$



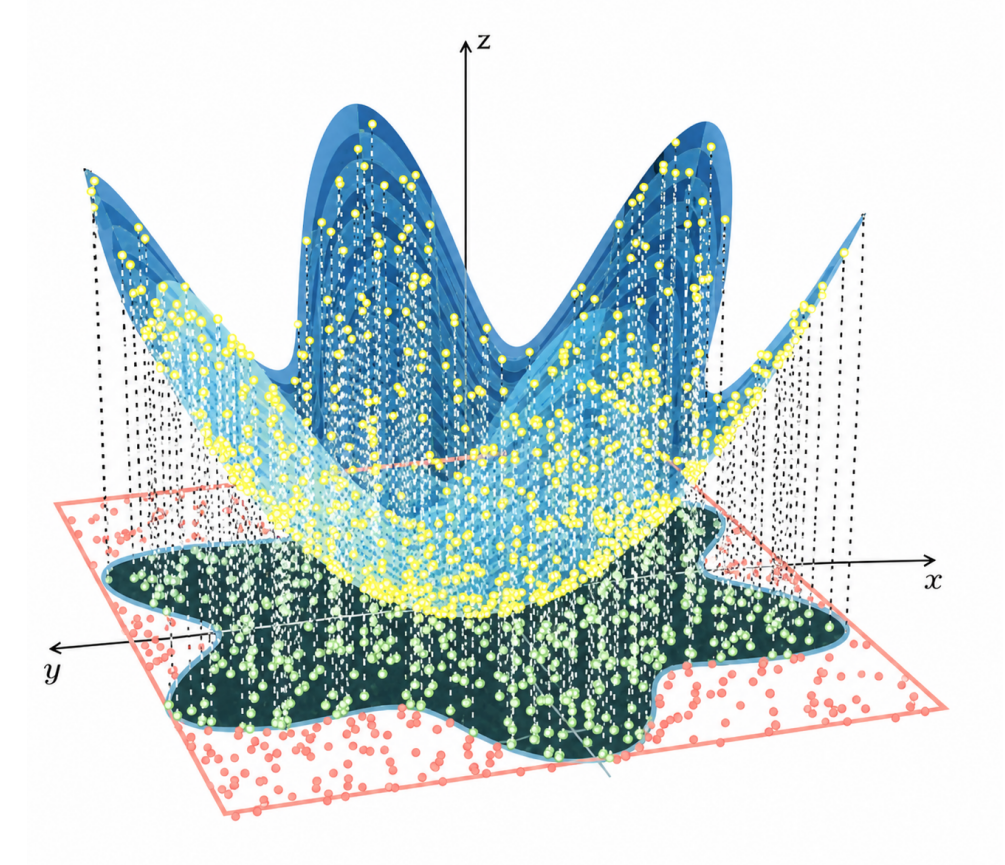
Applications:



statistical physics



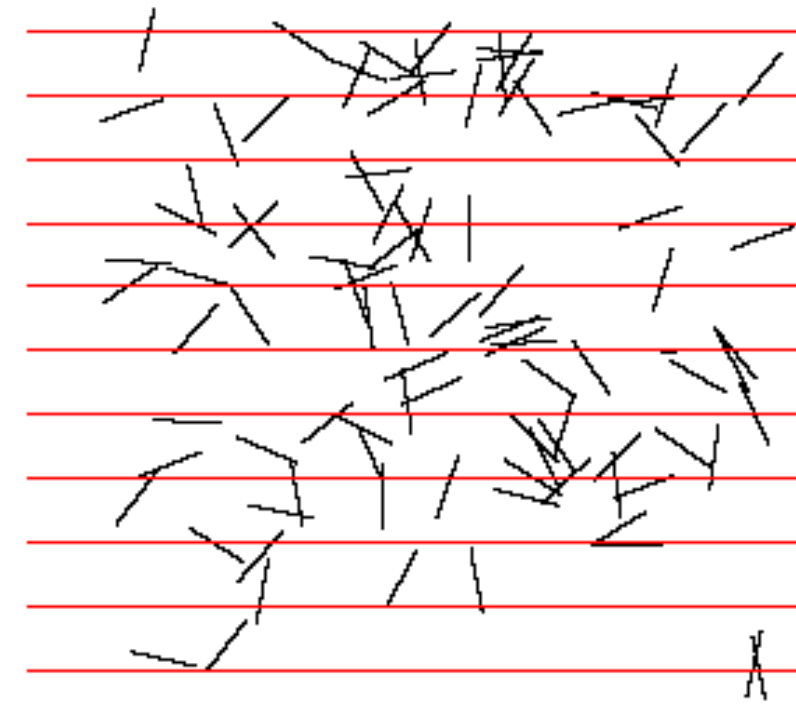
probabilistic inference



high-dimensional integration

Counting/Sampling Lovász Local Lemma

Counting LLL: approximately count the number of CSP solutions
Sampling LLL: almost uniformly sample a random CSP solution **under LLL-like conditions**



$$pD^c \lesssim 1$$

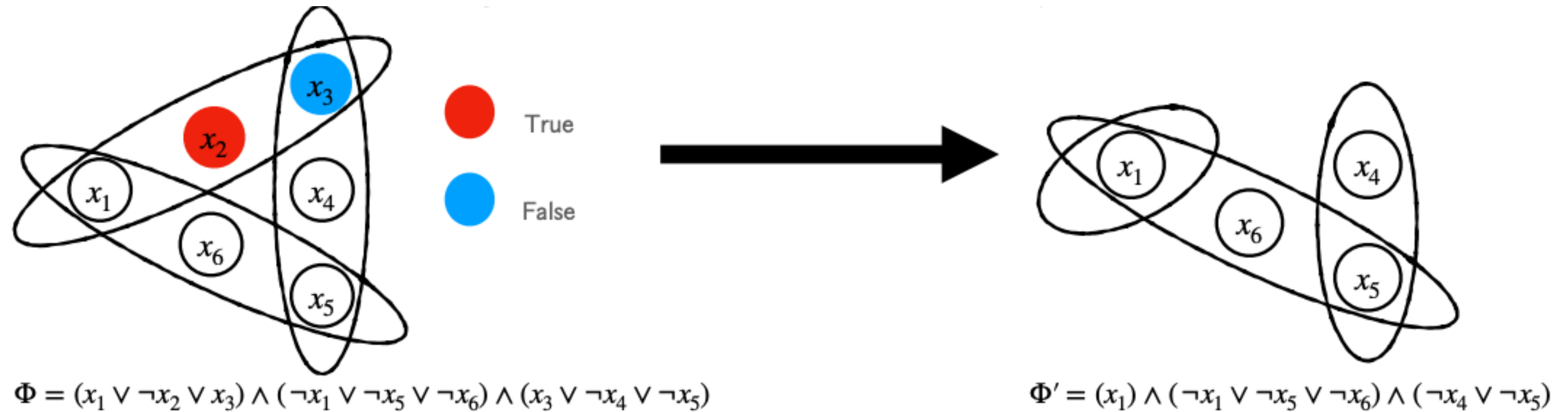


[Bezáková, Galanis, Goldberg, Guo, Štefankovič '16]: $pD^2 \gtrsim 1$ for monotone CNF

[Galanis, Guo, Wang '23]: $pD^2 \gtrsim 1$ for hypergraph colorings

Counting/Sampling is inherently harder!

Challenges for LLL Distribution

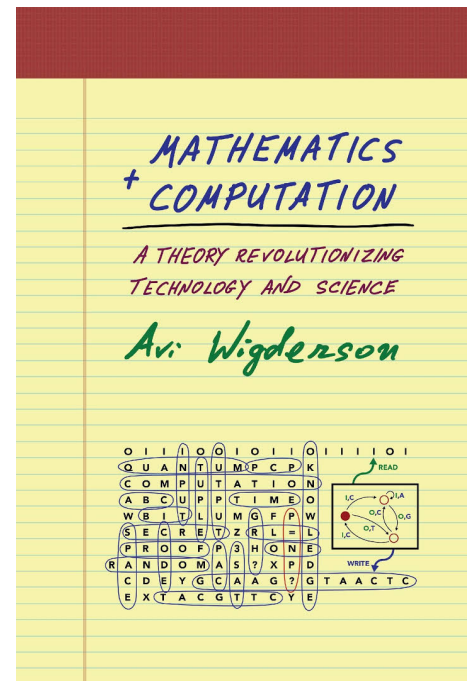


Non-self-reducibility: LLL condition may degrade after pinning

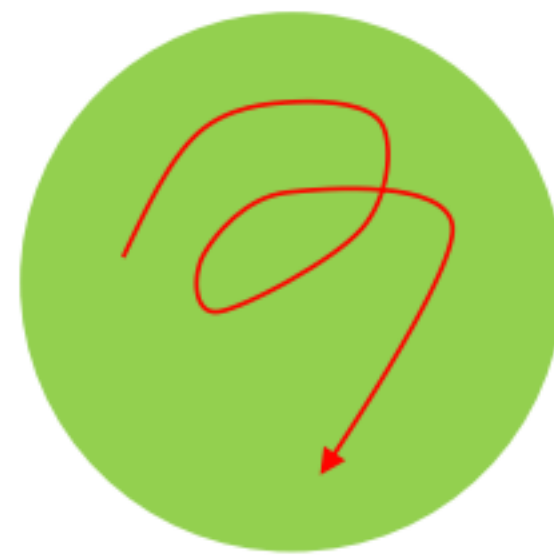
Challenges for LLL Distribution

“

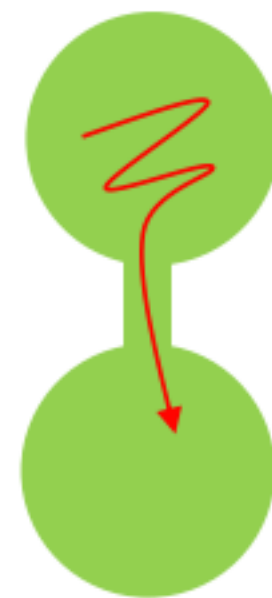
The solution space (and hence the natural Markov chain) is **not connected**.



—Mathematics and Computation [Wigderson '20]



rapid mixing



slow mixing



not mixing

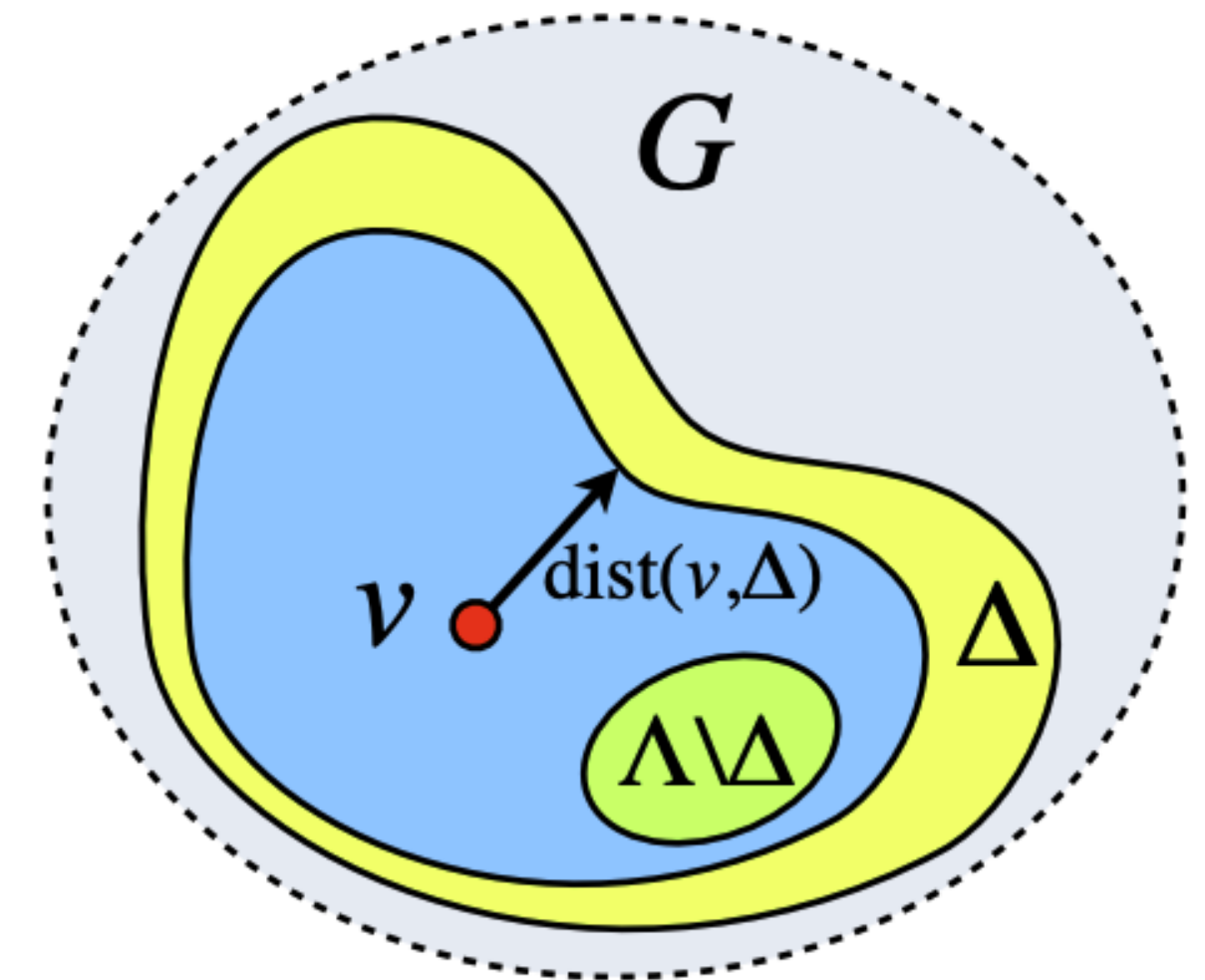
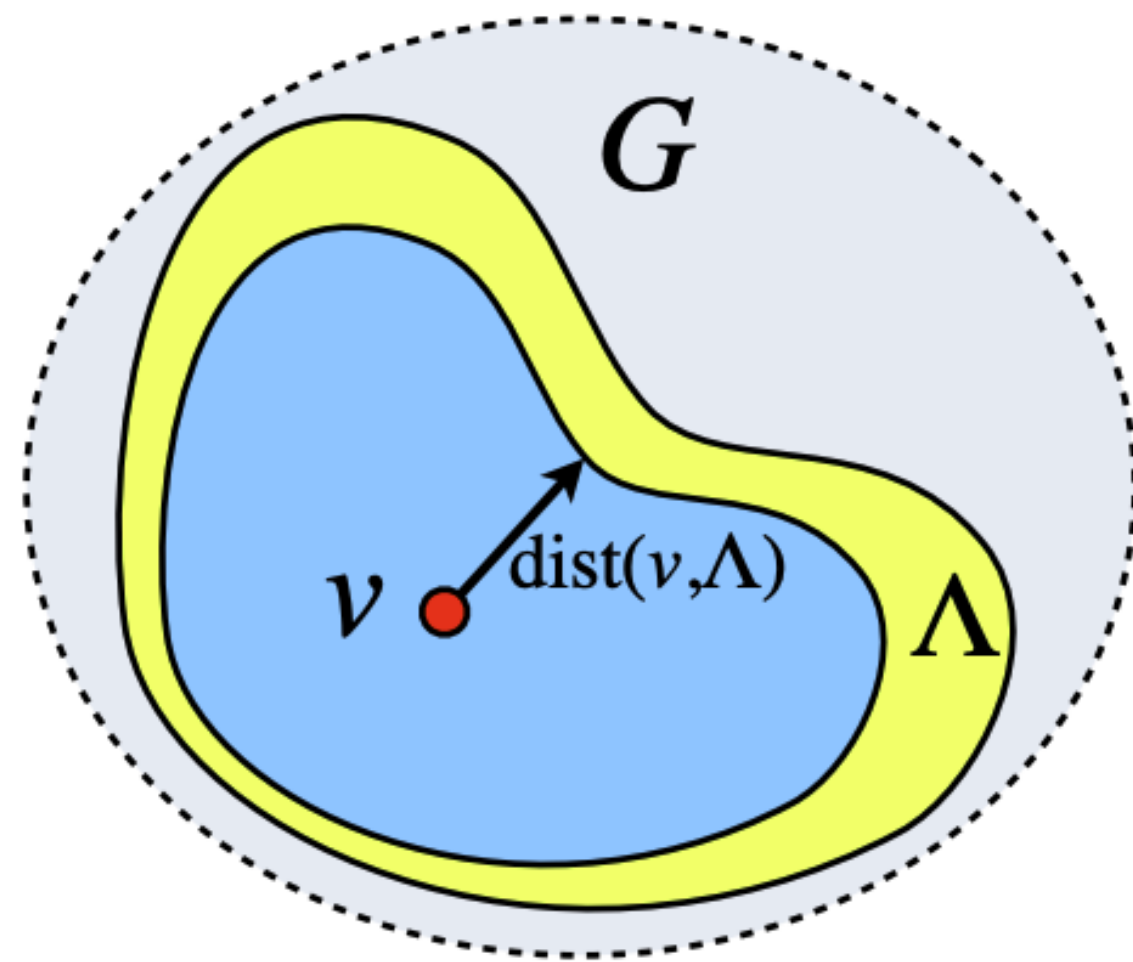
Connectivity Barrier: The state space is not connected

Challenges for LLL Distribution

μ_v^σ : marginal probability of v conditioning

Weak Spatial Mixing (WSM): $\forall \sigma, \tau \in \mathcal{Q}_\Lambda : |\mu_v^\sigma - \mu_v^\tau|_{\text{TV}} \leq \delta(\text{dist}_G(v, \Lambda))$

Strong Spatial Mixing (SSM): $\forall \sigma, \tau \in \mathcal{Q}_\Lambda$ that differ on $\Delta : |\mu_v^\sigma - \mu_v^\tau|_{\text{TV}} \leq \delta(\text{dist}_G(v, \Delta))$

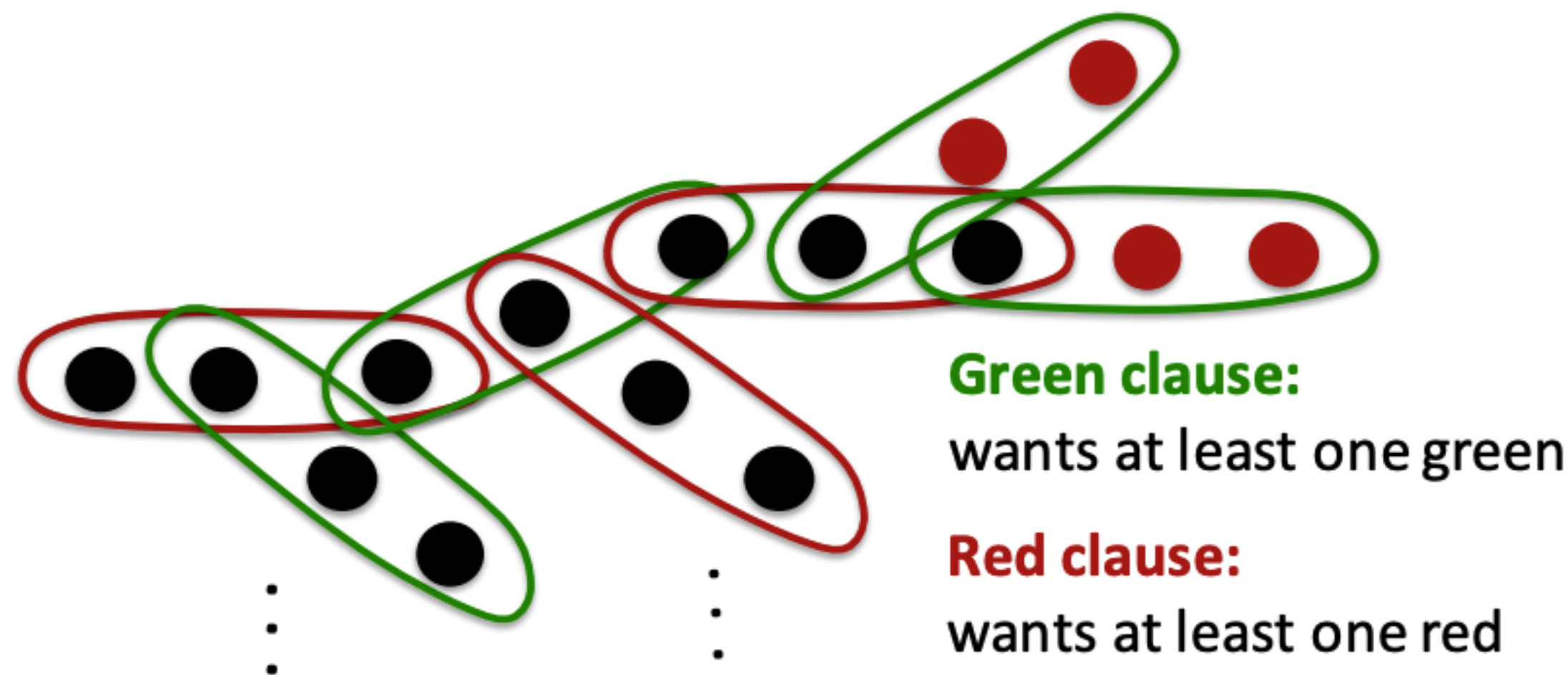


Challenges for LLL Distribution

μ_v^σ : marginal probability of v conditioning

Weak Spatial Mixing (WSM): $\forall \sigma, \tau \in \mathcal{Q}_\Lambda : |\mu_v^\sigma - \mu_v^\tau|_{\text{TV}} \leq \delta(\text{dist}_G(v, \Lambda))$

Strong Spatial Mixing (SSM): $\forall \sigma, \tau \in \mathcal{Q}_\Lambda$ that differ on $\Delta : |\mu_v^\sigma - \mu_v^\tau|_{\text{TV}} \leq \delta(\text{dist}_G(v, \Delta))$

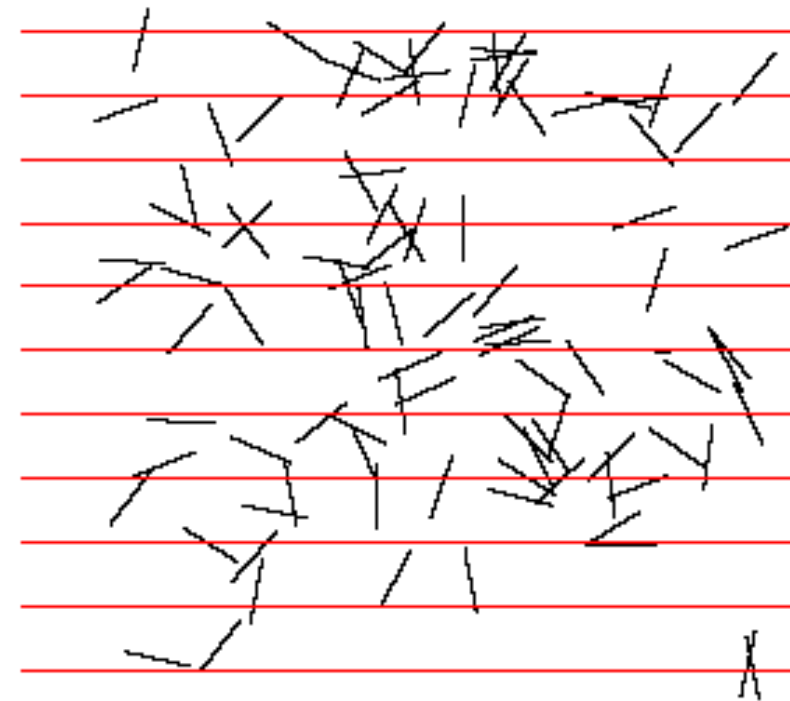


long-range dependencies exist
when $D = O(k)$

Failure of Spatial Mixing: (old notions) of correlation decay does not hold

Counting/Sampling Lovász Local Lemma

Counting LLL: approximately count the number of CSP solutions
Sampling LLL: almost uniformly sample a random CSP solution **under LLL-like conditions**



$$pD^c \lesssim 1$$



[Moitra '17]: $pD^{60} \lesssim 1$ for k -CNF

[Guo, Liao, Lu, Zhang '18]: $pD^{14} \lesssim 1$ for hypergraph coloring

[Hermon, Sly, Zhang '19]: $pD^2 \lesssim 1$ for monotone k -CNF, **fast** sampler

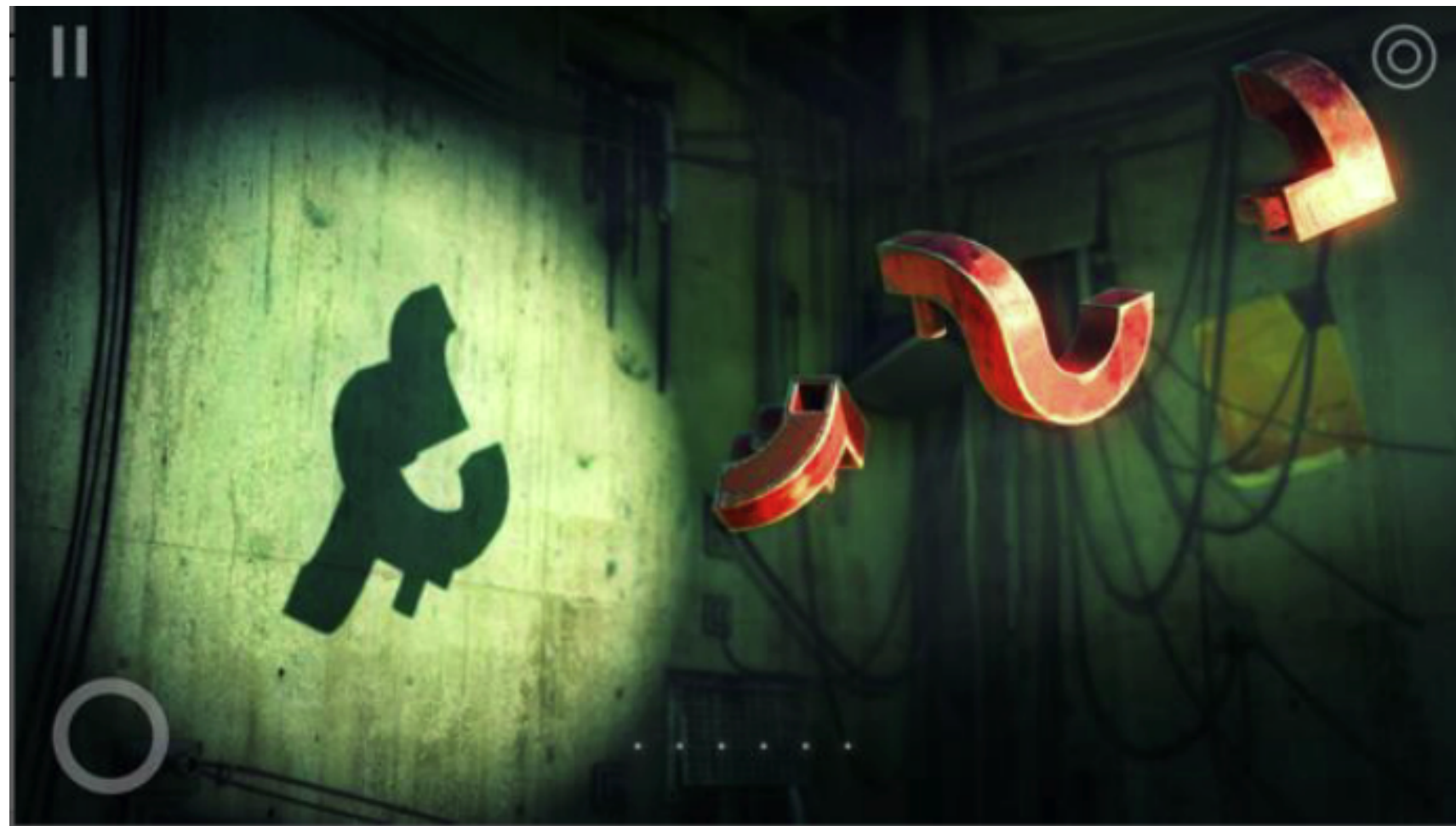
[Feng, Guo, Yin, Zhang '20]: $pD^{20} \lesssim 1$ for k -CNF, **fast** sampler

[Feng, He, Yin '21]: $pD^{350} \lesssim 1$ for **atomic** CSPs, **fast** sampler

[Jain, Pham, Vuong '21b]: $pD^7 \lesssim 1$ for general CSPs

[Jain, Pham, Vuong '21a, He, Sun, Wu '21]: $pD^{5.71} \lesssim 1$ for **atomic** CSPs, **fast** sampler

Previous efficient algorithms: **require additional assumptions**



Technique: Projection + MCMC

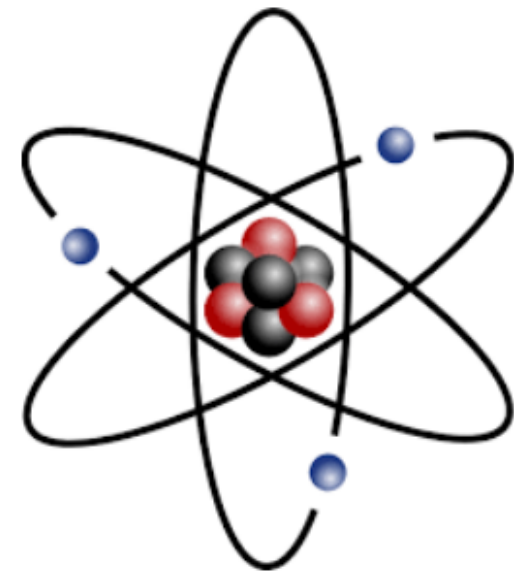
[Feng, Guo, Yin, Zhang '20]

[Feng, He, Yin '21]

[Jain, Pham, Vuong '21a]

[He, Sun, Wu '21]

works for **atomic** CSPs:
 $|c^{-1}(\text{False})| = \text{Constant}$



$$\text{for every } c \in [q], \underline{p}_{x,y}^x = \sum_{c' \in [q]} \underline{p}_{x^{u \leftarrow c}, y^{u \leftarrow c'}}^{x^{u \leftarrow c}};$$

$$\text{for every } c \in [q], \underline{p}_{x,y}^y = \sum_{c' \in [q]} \underline{p}_{x^{u \leftarrow c'}, y^{u \leftarrow c}}^{y^{u \leftarrow c}};$$

$$0 \leq \underline{p}_{x,y}^x, \underline{p}_{x,y}^y \leq 1.$$

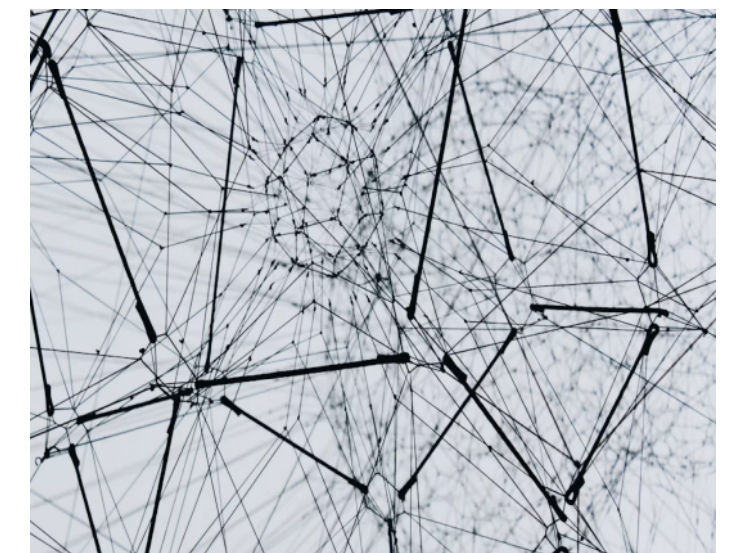
Technique: Coupling + LP

[Moitra '17]

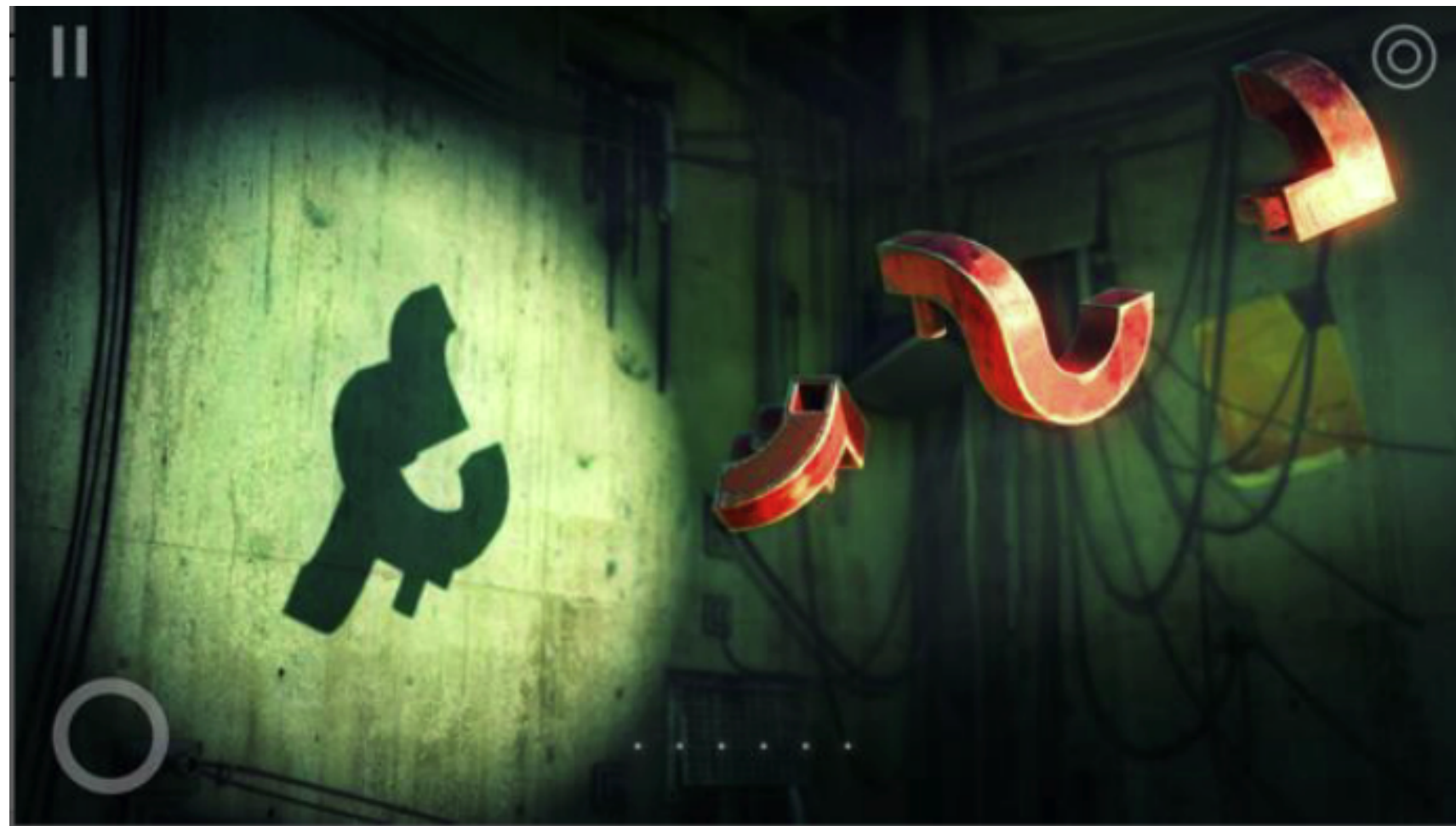
[Guo, Liao, Lu, Zhang '18]

[Jain, Pham, Vuong '21b]

works for **sparse** CSPs:
 $D = \text{Constant}$



Previous efficient algorithms: **require additional assumptions**



Technique: Projection + MCMC

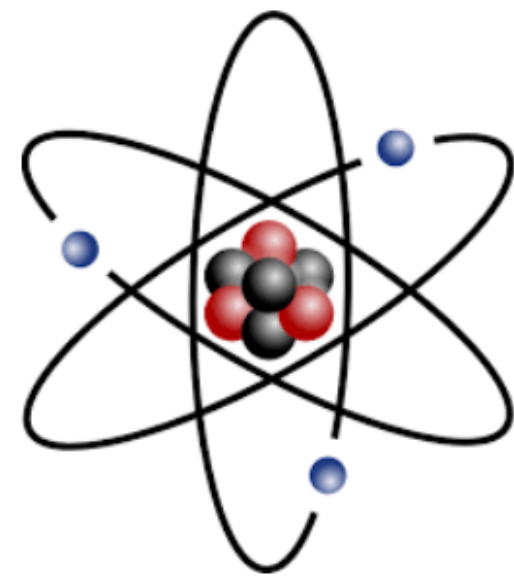
[Feng, Guo, Yin, Zhang '20]

[Feng, He, Yin '21]

[Jain, Pham, Vuong '21a]

[He, Sun, Wu '21]

works for **atomic** CSPs:
 $|c^{-1}(\text{False})| = \text{Constant}$



$$\text{for every } c \in [q], \underline{p}_{x,y}^x = \sum_{c' \in [q]} \underline{p}_{x^{u \leftarrow c}, y^{u \leftarrow c'}}^{x^{u \leftarrow c}};$$

$$\text{for every } c \in [q], \underline{p}_{x,y}^y = \sum_{c' \in [q]} \underline{p}_{x^{u \leftarrow c'}, y^{u \leftarrow c}}^{y^{u \leftarrow c}};$$

$$0 \leq \underline{p}_{x,y}^x, \underline{p}_{x,y}^y \leq 1.$$

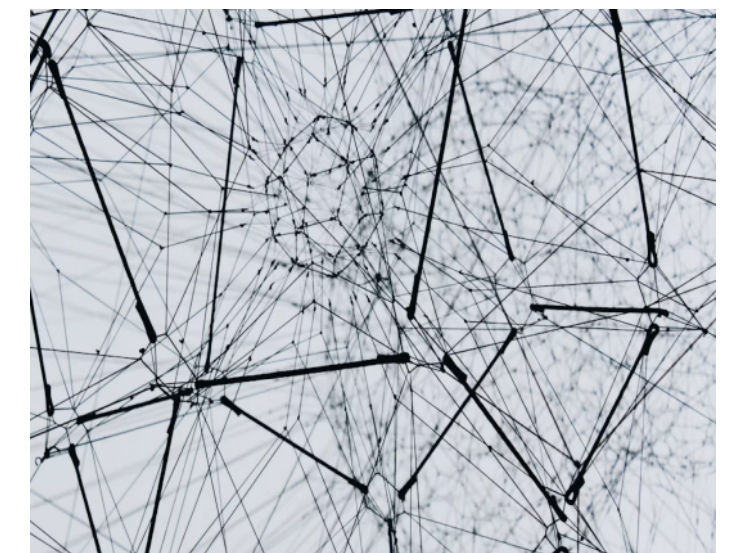
Technique: Coupling + LP

[Moitra '17]

[Guo, Liao, Lu, Zhang '18]

[Jain, Pham, Vuong '21b]

works for **sparse** CSPs:
 $D = \text{Constant}$



Can we efficiently count/sample **general** CSP solutions?

Our Results: **Fast** Sampling for **General** CSPs

We obtain the first efficient sampling algorithm for **general** CSPs under LLL regime with **improved** condition [He, W., Yin, '22, 23]

Improved condition

$$150e^3 \cdot q^2 \cdot k \cdot (D + 1)^5 \leq 1$$

q : domain size k : constraint width

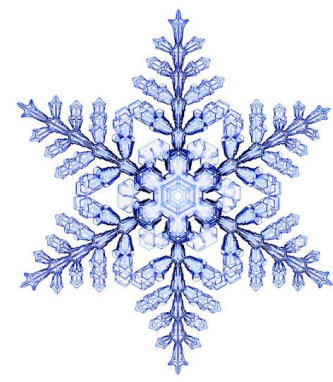
Near-linear running time

$$\text{poly}(q, k, D) \cdot n \log n \text{ expected}$$

$n = |V|$: number of variables

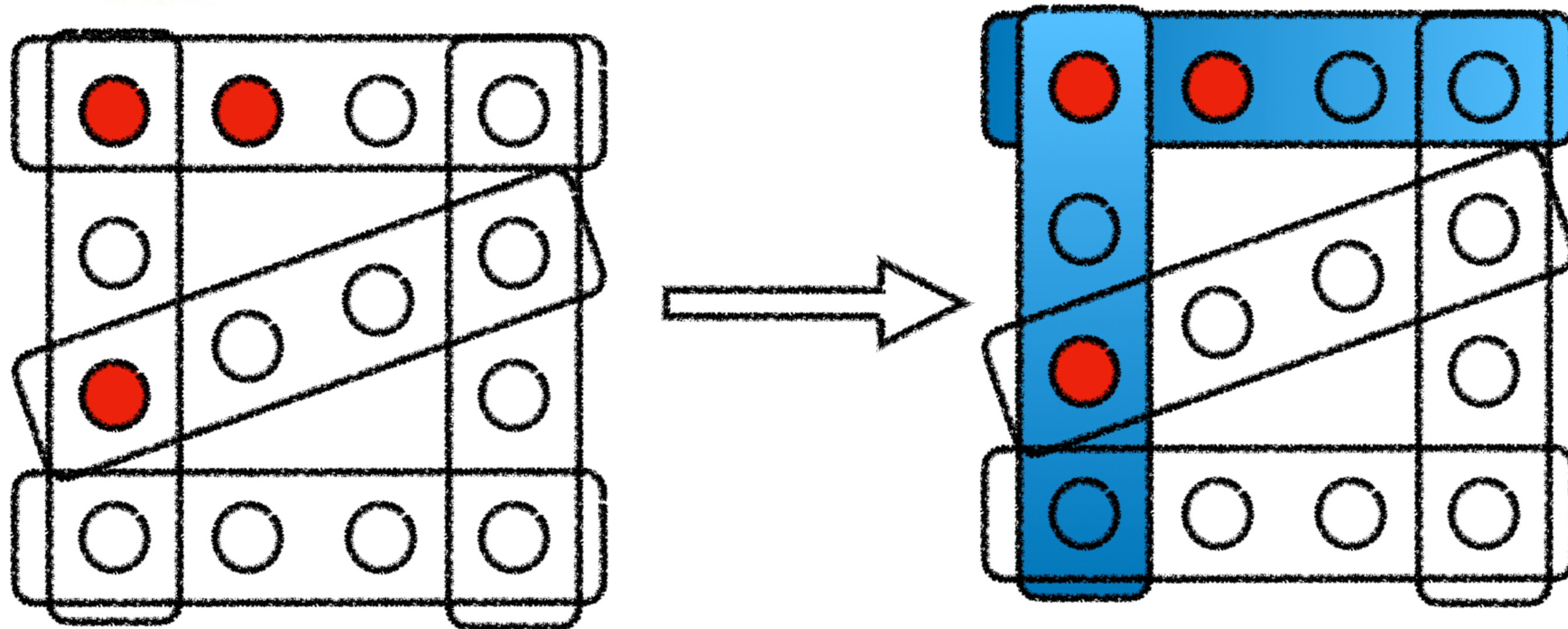
Remains **the only** efficient sampling algorithm for general CSPs up to date.

Based on a non-Markov chain based new **recursive sampling** technique [Anand, Jerrum '23]



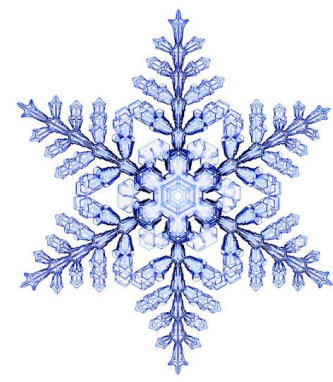
Freezing Constraints

Freeze (lock all its variables) a constraint once its violation prob. exceeds $p' > p$



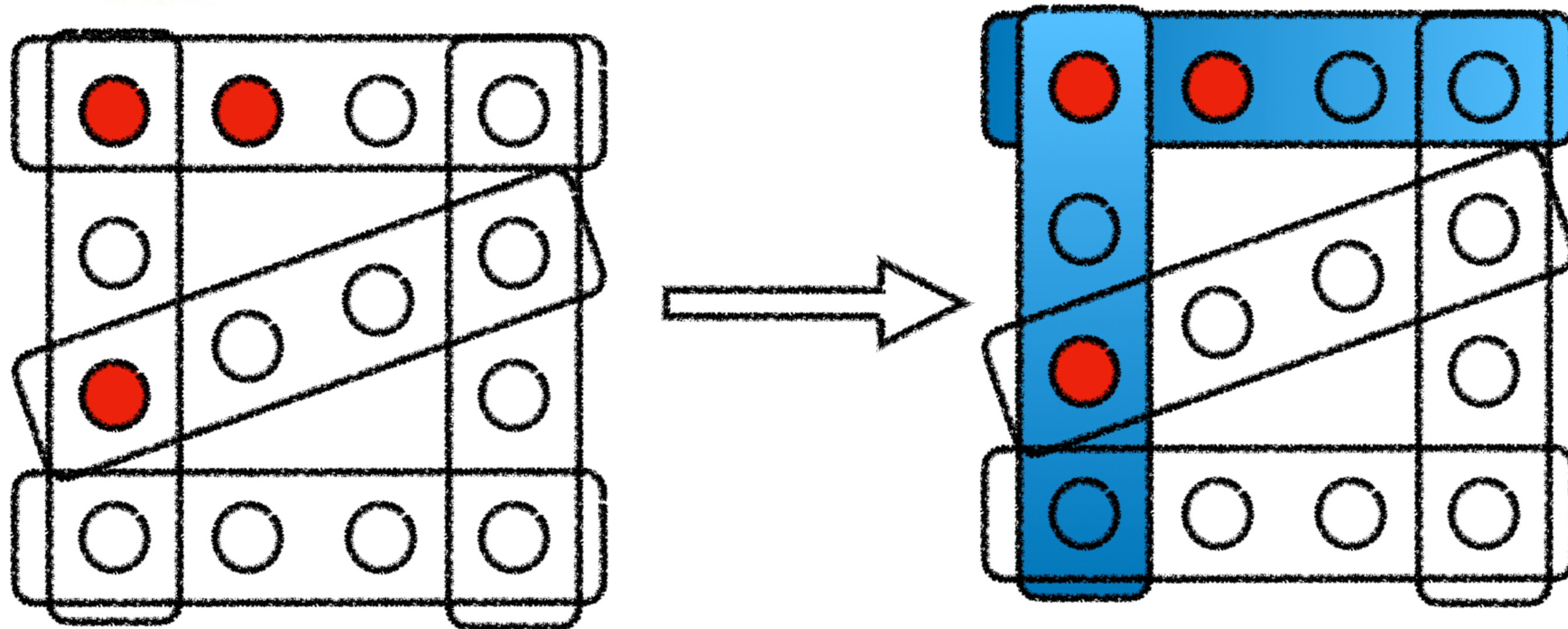
Ensures **self-reducibility** for a weaker LLL parameter p'

First used in [Bec '91] for algorithmic LLL (finally leads to $pD^4 \lesssim 1$)



Freezing Constraints

Freeze (lock all its variables) a constraint once its violation prob. exceeds $p' > p$

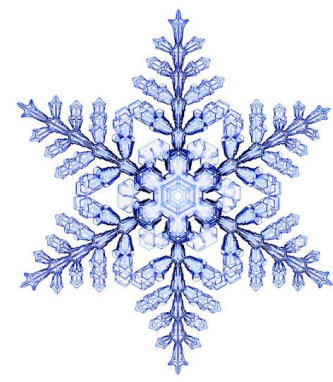


Witness argument

algorithm running too long $\implies$ a large **witness** occurs (happens with very small probability)

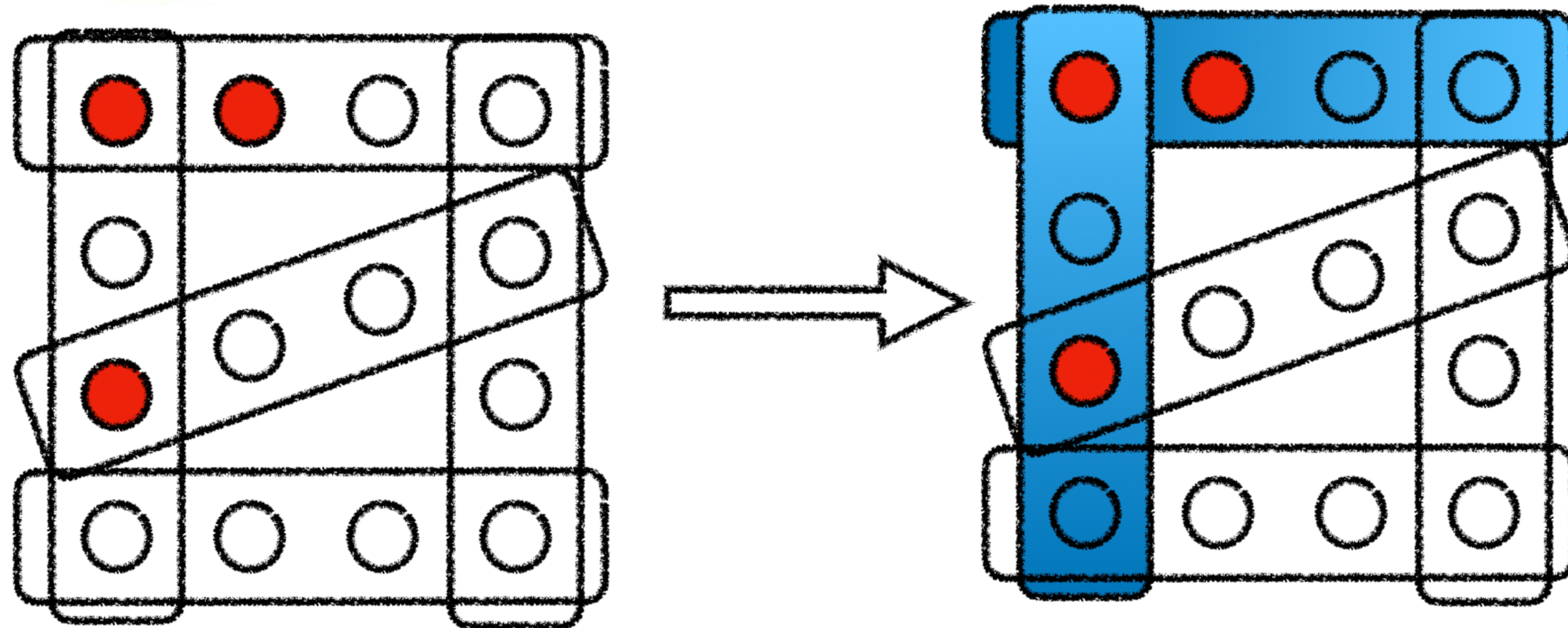
previous witness: $\{2,3\}$ -tree [Jain, Pham, Vuong '21] $\implies pD^7 \lesssim 1$

denser witness: generalized $\{2,3\}$ -tree [He, W., Yin, '23] $\implies pD^5 \lesssim 1$



Freezing Constraints

Freeze (lock all its variables) a constraint once its violation prob. exceeds $p' > p$

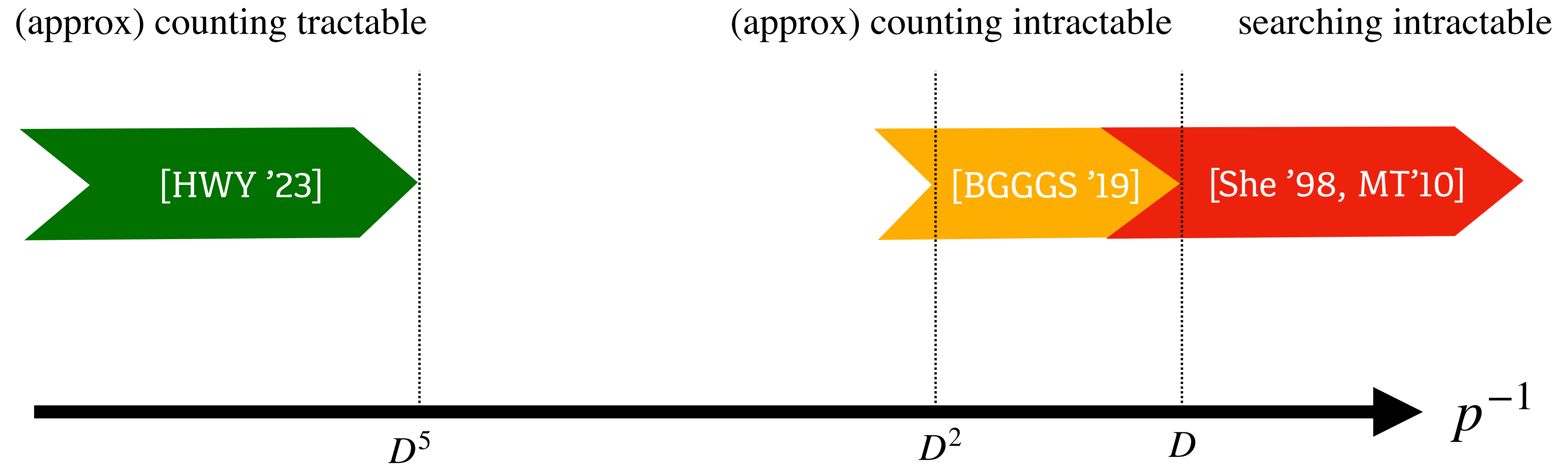


“

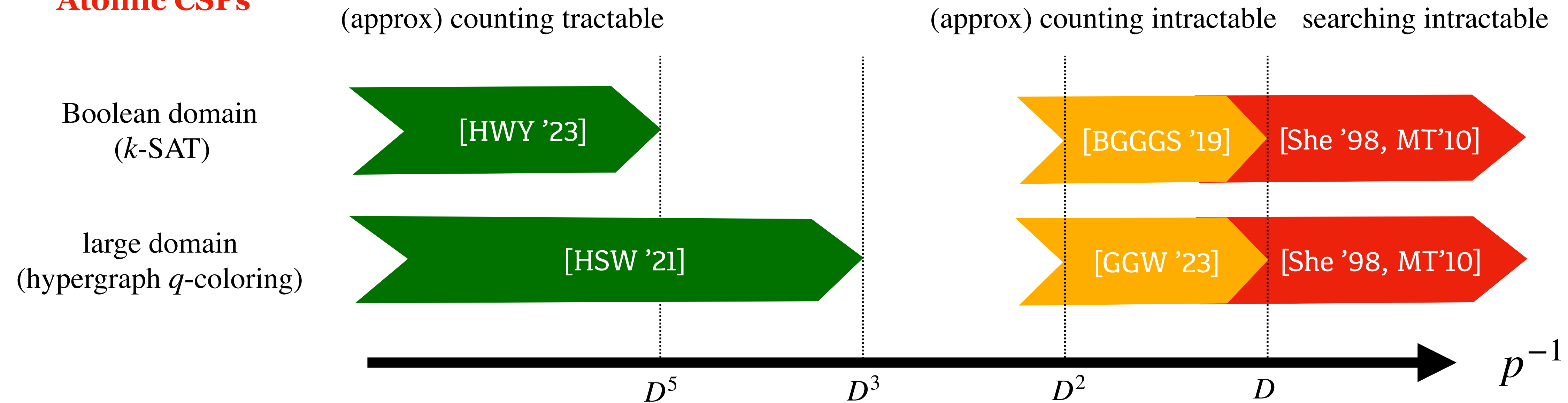
We believe that our analysis **lays bare the limits** of this (freezing) approach ... even for the algorithmic LLL, $\{2,3\}$ -trees only lead to D^4 instead of D , and being able to use “denser witness trees” was the major innovation in the works of Moser [Mos ’09] and Moser and Tardos [Moser, Tardos ’10].

—Jain, Pham and Vuong. *Towards the sampling Lovász Local Lemma*, STOC ’21

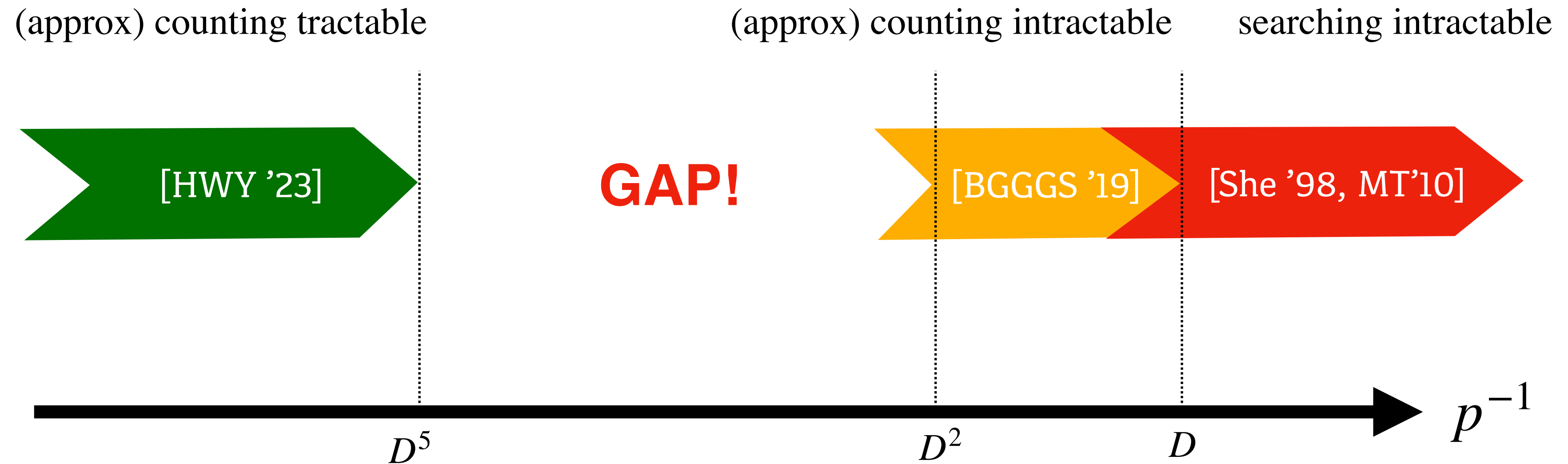
General CSPs



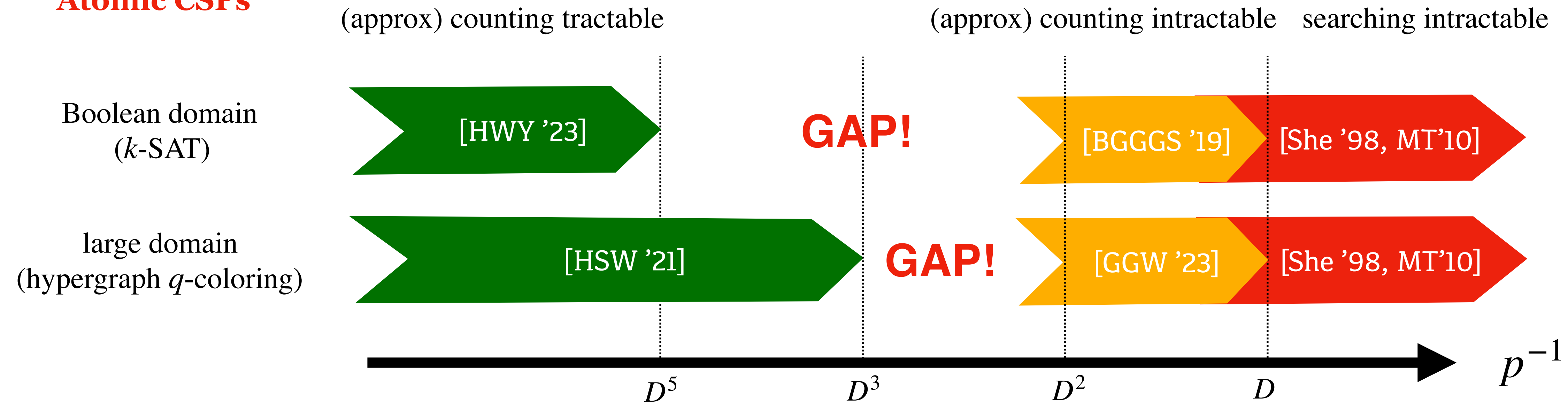
Atomic CSPs



General CSPs

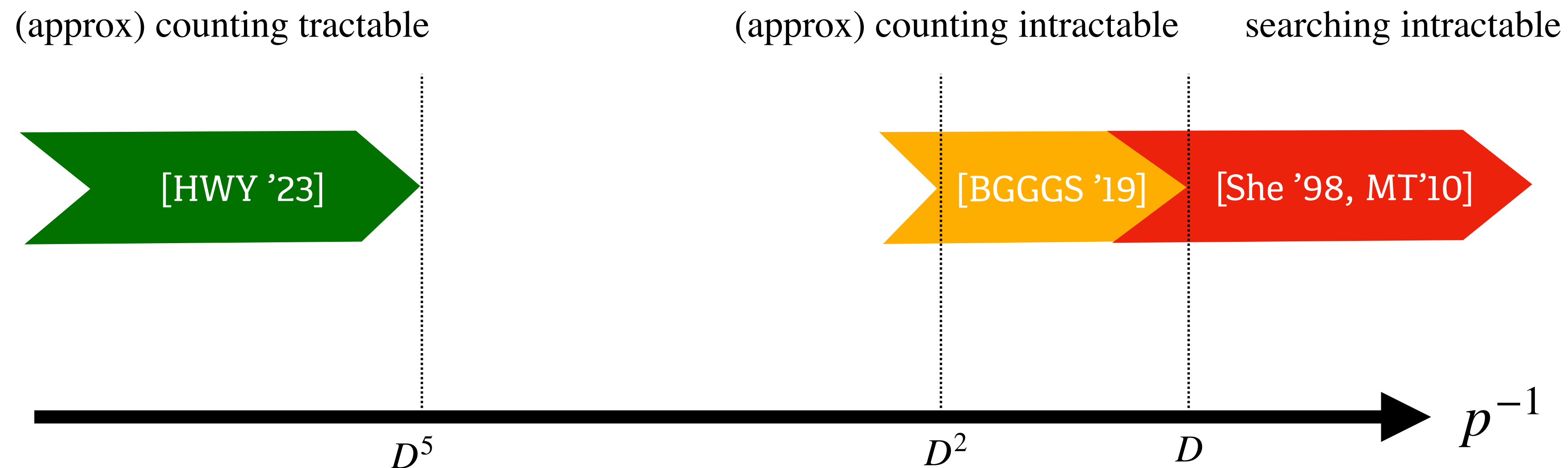


Atomic CSPs

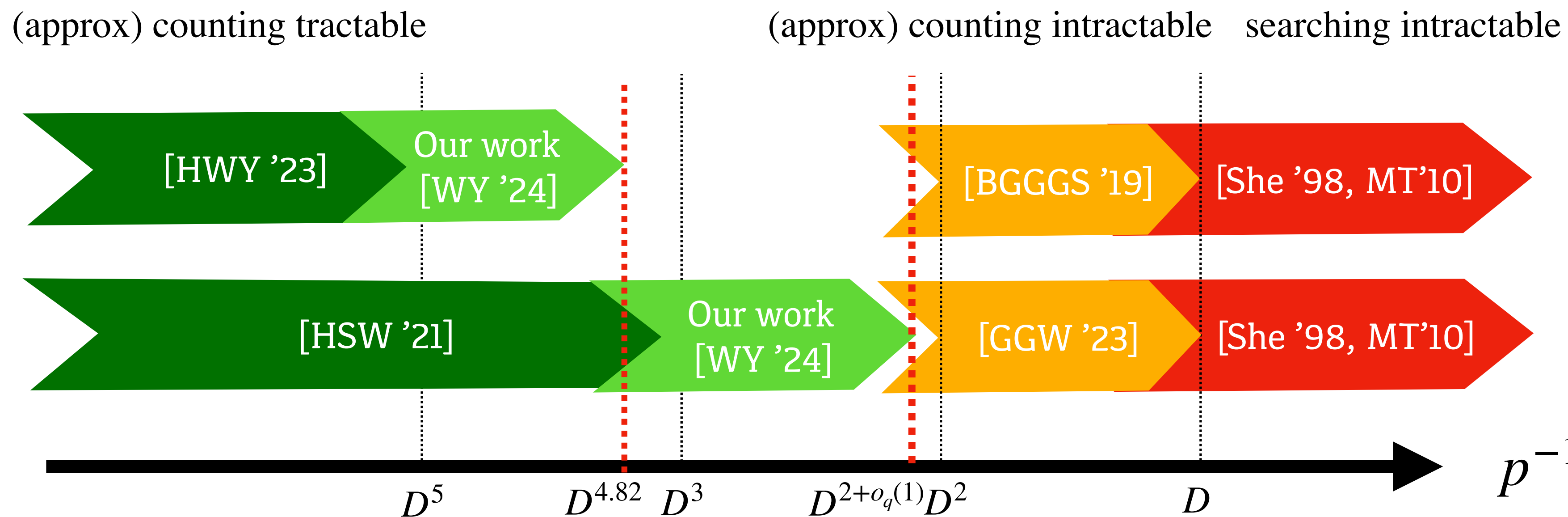


What is the **critical** threshold?

General CSPs



Atomic CSPs



Our Results: Sampling LLL for Large Domain

Condition. (Condition for sampling/counting **atomic** CSPs)

$\Phi = (V, \mathcal{Q}, \mathcal{C})$ is an atomic CSP satisfying

$$(8e)^3 \cdot p \cdot (D + 1)^{2+\zeta} \leq 1,$$

where $\zeta = \frac{2 \ln(2 - 1/q)}{\ln q - \ln(2 - 1/q)} = o_q(1)$

Theorem. (Sampling/Counting **atomic** CSPs for large domain sizes)

We give poly-time sampling/counting algorithms for **atomic** CSPs satisfying condition above.

$$\text{Running time: } n^{\text{poly}(k, D, \log q)}$$

(A new form of) Decay of Correlation

Condition. (Condition for sampling/counting atomic CSPs for large domain sizes)

$\Phi = (V, \mathcal{Q}, \mathcal{C})$ is an atomic CSP satisfying

$$(8e)^3 \cdot p \cdot (D + 1)^{2+\zeta} \leq 1,$$

where $\zeta = \frac{2 \ln(2 - 1/q)}{\ln q - \ln(2 - 1/q)} = o_q(1)$

Theorem. (Decay of correlation, informal)

For two CSPs $(V, \mathcal{Q}, \mathcal{C})$ and $(V, \mathcal{Q}, \mathcal{C} \setminus \{c_0\})$ (**differ in one constraint**) under condition above, there exists a **coupling** (X, Y) of $\mu_{\mathcal{C} \setminus \{c_0\}}$ and $\mu_{\mathcal{C}}$ such that

$$\Pr[d_{\text{Ham}}(X, Y) \geq K] \leq \exp(-O(K)).$$

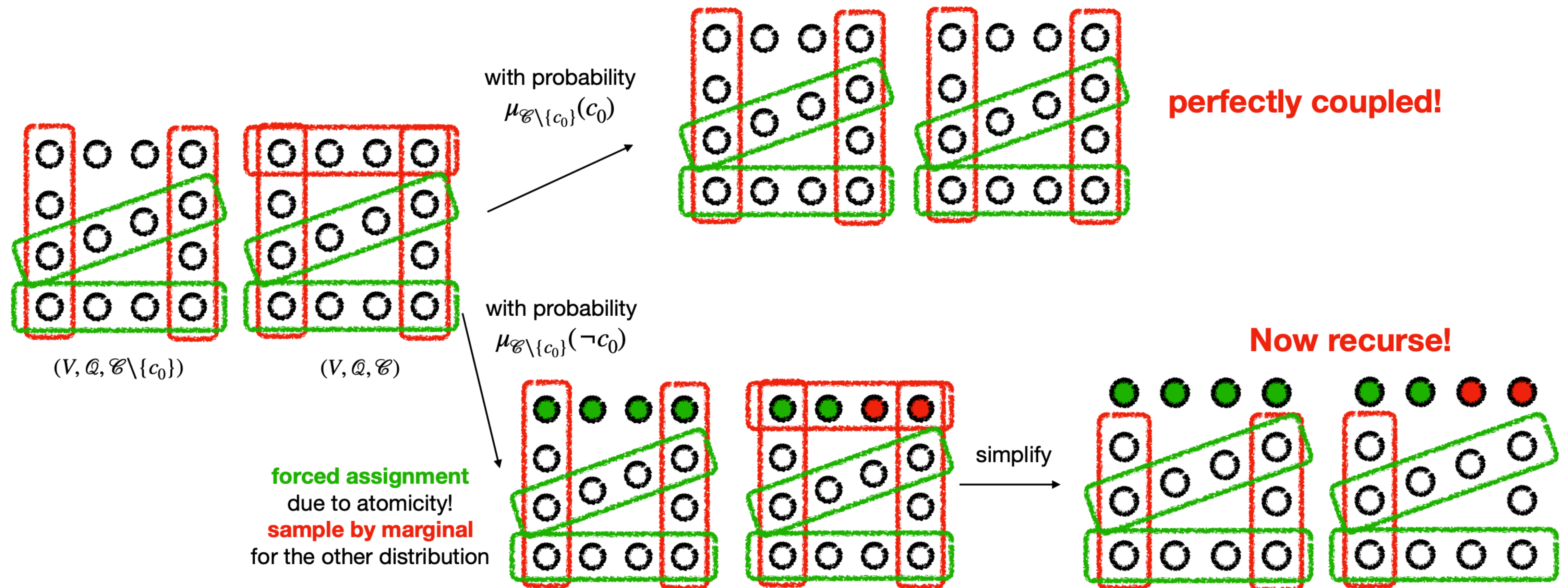
$\mu_{\mathcal{C} \setminus \{c_0\}}$: uniform distribution over solutions of $(V, \mathcal{Q}, \mathcal{C})$

$\mu_{\mathcal{C}}$: uniform distribution over solutions of $(V, \mathcal{Q}, \mathcal{C} \setminus \{c_0\})$

A Constraint-wise Coupling

Coupling between $\mu_{\mathcal{E}}$ and $\mu_{\mathcal{E} \setminus \{c_0\}}$: recursively fixing discrepancies

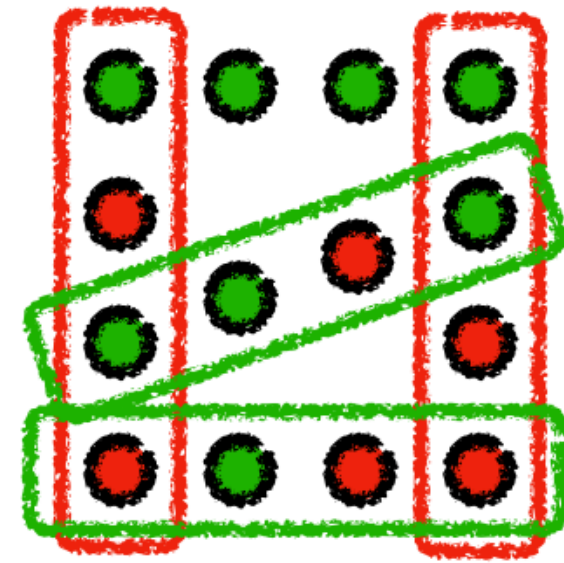
Decomposition using **law of total probability**: $\mu_{\mathcal{E} \setminus \{c_0\}} = \mu_{\mathcal{E} \setminus \{c_0\}}(c_0) \cdot \mu_{\mathcal{E}} + \mu_{\mathcal{E} \setminus \{c_0\}}(\neg c_0) \cdot \mu_{\mathcal{E} \setminus \{c_0\}}(\cdot \mid \neg c_0)$



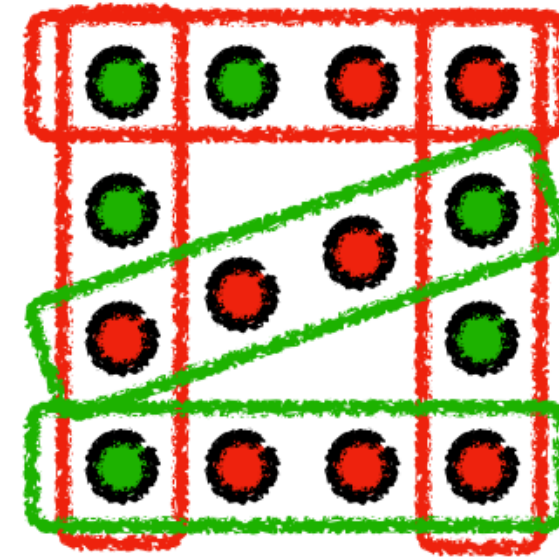
A Constraint-wise Coupling

Principle of deferred decisions: overcoming non-self-reducibility

All randomness by the procedure can be identified by two independent samples:



$$\mathfrak{X} \sim \mu_{\mathcal{C} \setminus \{c_0\}}$$



$$\mathfrak{Y} \sim \mu_{\mathcal{C}}$$

[HSS '14]: under local lemma regimes, $\mathfrak{X}$ and $\mathfrak{Y}$ behave close to uniform !

Sampling by marginal distribution = Revealing local information of $\mathfrak{X}$ and $\mathfrak{Y}$



Witness argument

coupling running too long $\implies$ a large **witness** occurs (happens with very small probability)

witness for constraint-wise coupling: 2-tree [Alon '91] $\implies pD^{2+o_q(1)} \lesssim 1$

Constraint-wise Self-Reducibility

$$\begin{array}{c}
 \text{Diagram 1: } Z(\text{set of } x_1, x_2, x_3, x_4, x_5, x_6 \text{ with two overlapping ellipses}) \\
 \\
 Z(\text{Diagram 1}) = \frac{Z(\text{Diagram 2}) \cdot Z(\text{Diagram 3}) \cdot Z(\text{Diagram 4})}{Z(\text{Diagram 5}) \cdot Z(\text{Diagram 6}) \cdot Z(\text{Diagram 7})}
 \end{array}$$

The diagram illustrates the constraint-wise self-reducibility of a set of points $x_1, x_2, x_3, x_4, x_5, x_6$. The set is partitioned into three constraints (ellipses):

- Constraint 1 (Top-left):** Contains x_1, x_2, x_3 .
- Constraint 2 (Top-right):** Contains x_4, x_5, x_6 .
- Constraint 3 (Bottom):** Contains $x_1, x_2, x_3, x_4, x_5, x_6$.

The equation shows that the complexity of the full set is equal to the product of the complexities of the three constraints, divided by the complexities of the three constraints taken individually.

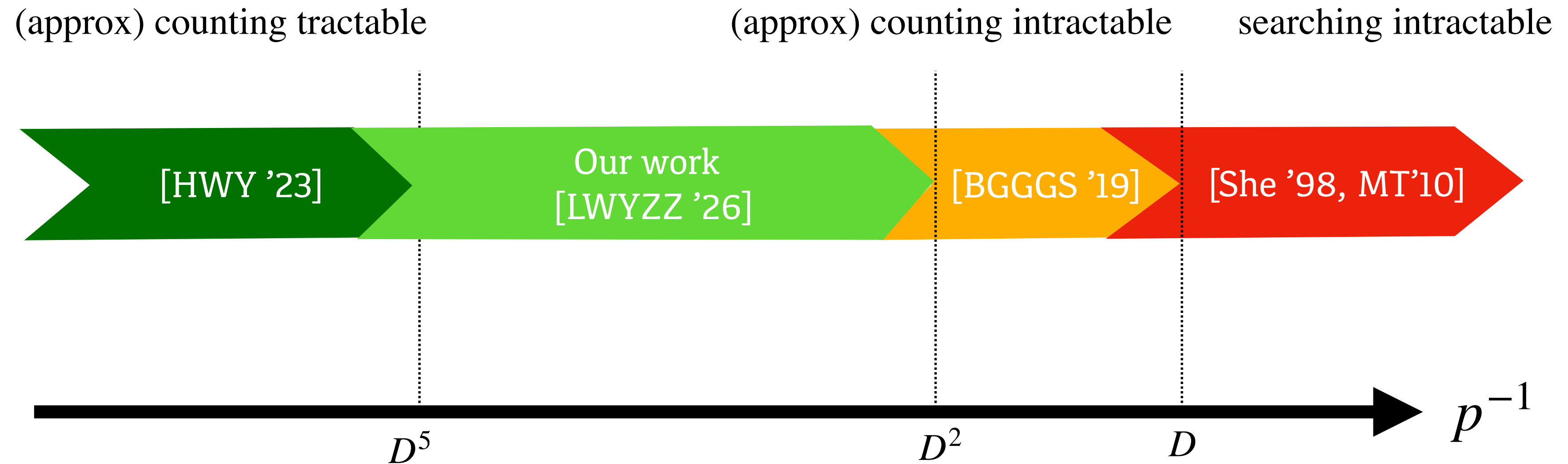
Constraint-wise Self-Reducibility

$$\begin{array}{c}
 \text{Diagram 1: } Z(\text{set of } x_1, x_2, x_3, x_4, x_5, x_6 \text{ with overlapping constraints}) \\
 = \frac{
 \begin{array}{c}
 \text{Diagram 2: } Z(\text{set of } x_1, x_2, x_3, x_4, x_5, x_6 \text{ with } x_1, x_2, x_3 \text{ in red constraint}) \\
 \text{Diagram 3: } Z(\text{set of } x_1, x_2, x_3, x_4, x_5, x_6 \text{ with } x_3, x_4, x_5 \text{ in red constraint}) \\
 \text{Diagram 4: } Z(\text{set of } x_1, x_2, x_3, x_4, x_5, x_6 \text{ with } x_1, x_4, x_5 \text{ in red constraint})
 \end{array}
 }{
 \begin{array}{c}
 \text{Diagram 5: } Z(\text{set of } x_1, x_2, x_3, x_4, x_5, x_6 \text{ with } x_1, x_2, x_3 \text{ in black constraint}) \\
 \text{Diagram 6: } Z(\text{set of } x_1, x_2, x_3, x_4, x_5, x_6 \text{ with } x_1, x_4, x_5 \text{ in black constraint}) \\
 \text{Diagram 7: } Z(\text{set of } x_1, x_2, x_3, x_4, x_5, x_6 \text{ with } x_1, x_4, x_5 \text{ in black constraint})
 \end{array}
 }
 \end{array}$$

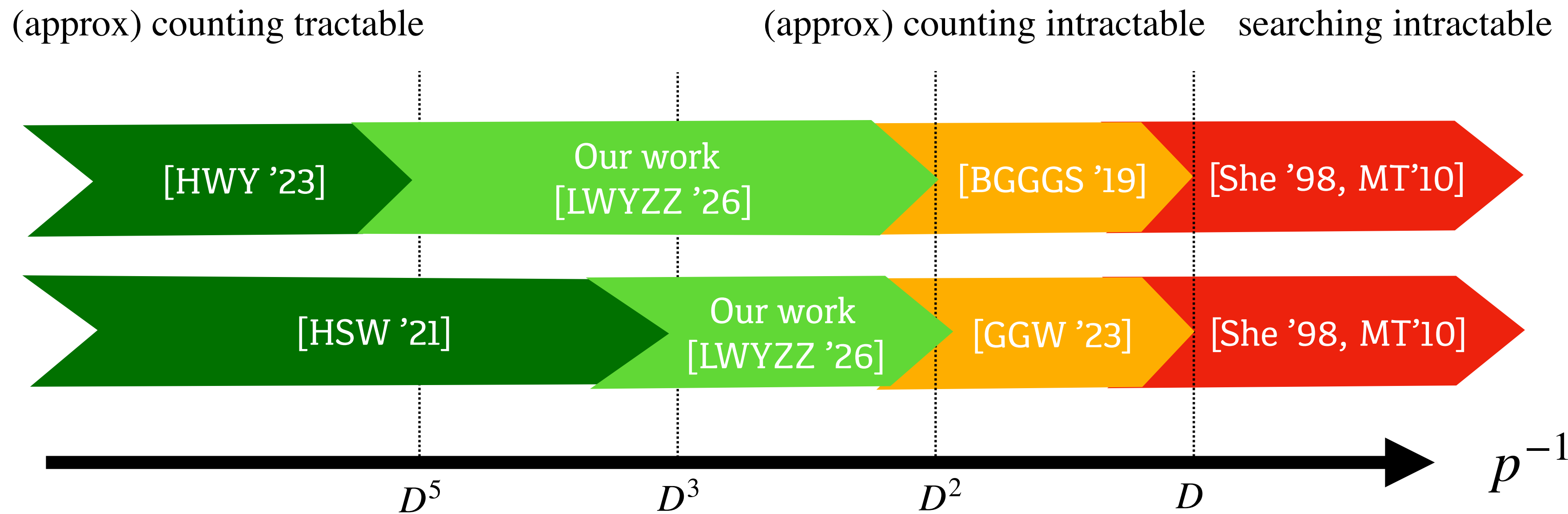
Marginal estimator for $\mu_{\mathcal{C} \setminus \{c_0\}}(c_0)$ $\longrightarrow$ Efficient counting

Dynamic sampler that updates $X \sim \mu_{\mathcal{C} \setminus \{c_0\}}$ to $Y \sim \mu_{\mathcal{C}}$ $\longrightarrow$ Efficient sampling

General CSPs



Atomic CSPs



Our Results: A Counting LLL

Condition. (Counting LLL condition)

$\Phi = (V, \mathcal{Q}, \mathcal{C})$ is an atomic CSP satisfying

$$4e \cdot p \cdot (D + 1)^2 \leq 1.$$

Theorem. (A Counting LLL)

We give poly-time counting algorithms for CSPs satisfying condition above.

Deterministic approximate counting: running time $n^{\text{poly}(k, D, \log q)}$;

Randomized approximate counting: running time $\text{poly}(k, D, q) \cdot n^2$.

Tight up to constants!

The 2-Tree Structure

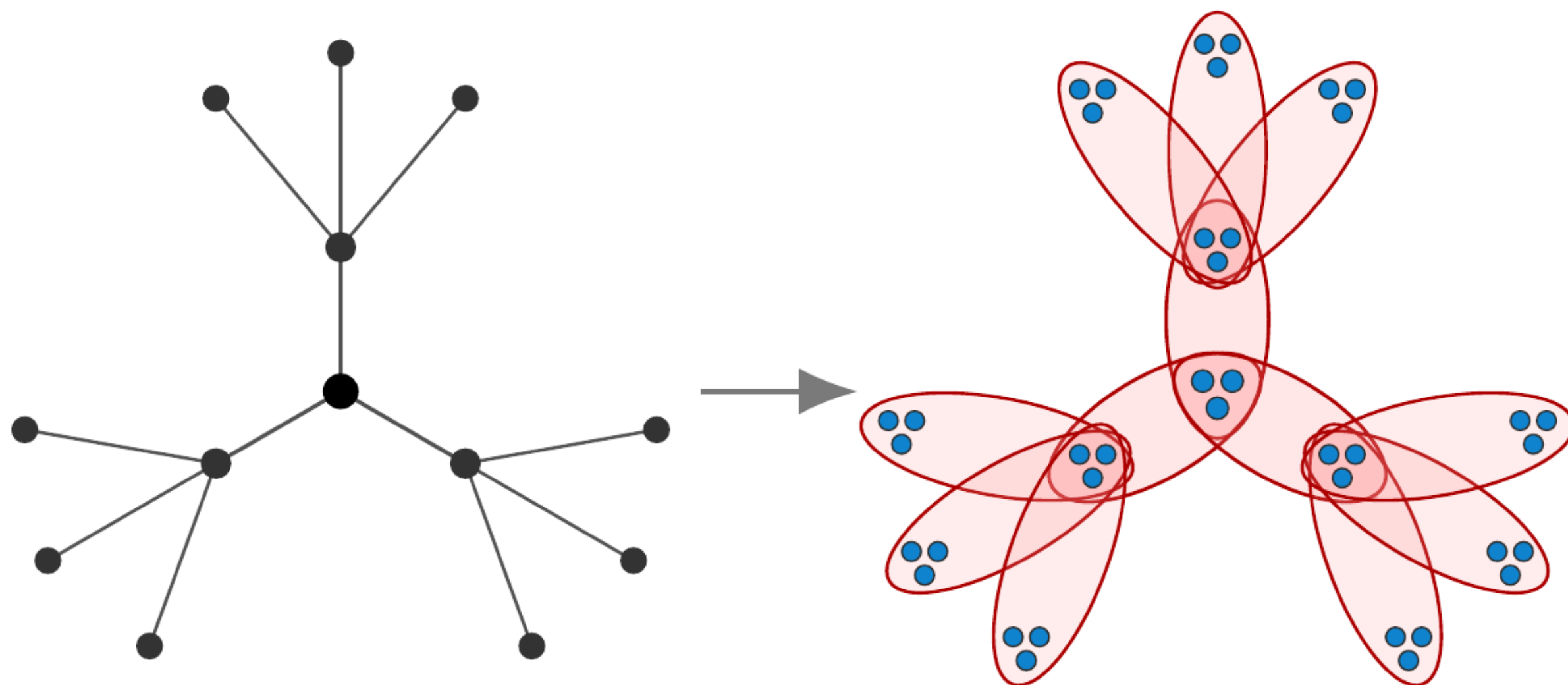
Definition. (2-Tree)

A 2-tree (in the dependency graph of CSP) is a set of constraints $T \subseteq \mathcal{C}$ that:

(**Independence**): T is an independent set in $G(\mathcal{C})$

(**Connectivity in square graph**): T is connected in $G^2(\mathcal{C})$;

$G(\mathcal{C})$: dependency graph with vertex set $\mathcal{C}$, two constraints connected by an edge if they share at least one variable



Hardness reduction in [\[BGGGS '19\]](#):

achievable by 2-trees

The 2-Tree Expansion

Marginal violation probability: $r_{\mathcal{C},c_0} \triangleq \Pr[\neg c_0 \mid \mathcal{C} \setminus \{c_0\}]$

$$\text{Inclusion-exclusion: } r_{\mathcal{C},c_0} = \frac{1}{\Pr[\mathcal{C} \setminus \{c_0\}]} \sum_{c_0 \in \mathcal{D} \subseteq \mathcal{C}} (-1)^{|\mathcal{D}|-1} \Pr \left[\bigwedge_{c \in \mathcal{D}} \neg c \right]$$

Idea: group each $\mathcal{D}$ through **greedily** constructing a **canonical 2-tree** $\text{Tree}(\mathcal{D})$

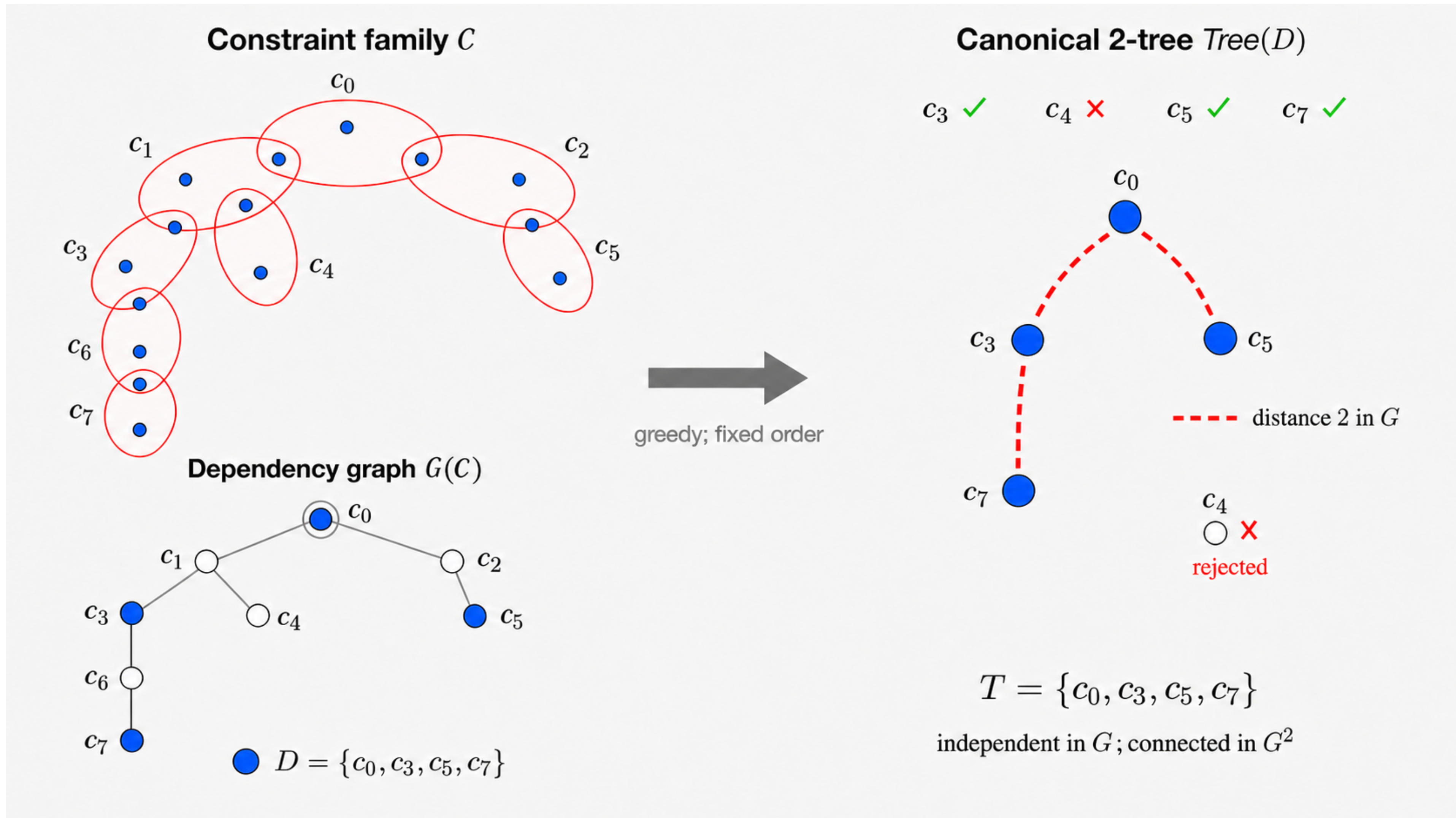
Algorithm 1: $\text{Tree}_{\mathcal{C},c_0}(\mathcal{D})$: Construction of a canonical 2-tree

Input : A constraint set $\mathcal{D} \subseteq \mathcal{C}$ containing c_0 .

Output: A 2-tree T in $G(\mathcal{C})$ containing c_0 and satisfying $T \subseteq \mathcal{D}$.

```
1 Initialize  $R \leftarrow \emptyset, T \leftarrow \{c_0\}$ ;  
2 while  $\exists c \in \mathcal{C} \setminus R$  such that  $\text{dist}_{G(\mathcal{C})}(c, T) = 2$  do  
3   Let  $c$  be the lowest-index constraint satisfying the above condition;  
4   if  $c \in \mathcal{D}$  then  
5      $T \leftarrow T \cup \{c\}$ ;                                     // accepted set  
6   else  
7      $R \leftarrow R \cup \{c\}$ ;                                     // rejected set  
8 return  $T$ ;
```

The 2-Tree Expansion



The 2-Tree Expansion

Marginal violation probability: $r_{\mathcal{C},c_0} \triangleq \Pr[\neg c_0 \mid \mathcal{C} \setminus \{c_0\}]$

Inclusion-exclusion:
$$r_{\mathcal{C},c_0} = \frac{1}{\Pr[\mathcal{C} \setminus \{c_0\}]} \sum_{c_0 \in \mathcal{D} \subseteq \mathcal{C}} (-1)^{|\mathcal{D}|-1} \Pr \left[\bigwedge_{c \in \mathcal{D}} \neg c \right]$$

Idea: group each $\mathcal{D}$ through **greedily** constructing a **canonical 2-tree** $\text{Tree}(\mathcal{D})$

2-tree expansion:
$$r_{\mathcal{C},c_0} = \frac{1}{\Pr[\mathcal{C} \setminus \{c_0\}]} \sum_{T \in \mathfrak{T}_{\mathcal{C},c_0}} \sum_{\substack{c_0 \in \mathcal{D} \subseteq \mathcal{C} \\ \text{Tree}(\mathcal{D}) = T}} (-1)^{|\mathcal{D}|-1} \Pr \left[\bigwedge_{c \in \mathcal{D}} \neg c \right]$$

The 2-Tree Expansion

$$\text{2-tree expansion: } r_{\mathcal{C}, c_0} = \frac{1}{\Pr[\mathcal{C} \setminus \{c_0\}]} \sum_{T \in \mathfrak{T}_{\mathcal{C}, c_0}} \sum_{\substack{c_0 \in \mathcal{D} \subseteq \mathcal{C} \\ \text{Tree}(\mathcal{D}) = T}} (-1)^{|\mathcal{D}|-1} \Pr \left[\bigwedge_{c \in \mathcal{D}} \neg c \right]$$

Lemma. (Canonical 2-tree inversion)

Let $\mathcal{C}$ be a constraint set and let $c_0 \in \mathcal{C}$. For every $T \in \mathfrak{T}_{\mathcal{C}, c_0}$,

$$\{\mathcal{D} \subseteq \mathcal{C} \mid c_0 \in \mathcal{D}, \text{Tree}(\mathcal{D}) = T\} = \{S \cup T \mid S \subseteq \text{free}(T)\}.$$

$$\text{recursion: } r_{\mathcal{C}, c_0} = \frac{1}{\Pr[\mathcal{C} \setminus \{c_0\}]} \sum_{T \in \mathfrak{T}_{\mathcal{C}, c_0}} (-1)^{|T|-1} \Pr \left[\bigwedge_{c \in \mathcal{T}} \neg c \mid \text{free}(T) \right] \prod_i \frac{1}{1 - r_{\mathcal{C}_i, c_i}}$$

The 2-Tree Expansion

$$\text{2-tree expansion: } r_{\mathcal{C}, c_0} = \frac{1}{\Pr[\mathcal{C} \setminus \{c_0\}]} \sum_{T \in \mathfrak{T}_{\mathcal{C}, c_0}} \sum_{\substack{c_0 \in \mathcal{D} \subseteq \mathcal{C} \\ \text{Tree}(\mathcal{D}) = T}} (-1)^{|\mathcal{D}|-1} \Pr \left[\bigwedge_{c \in \mathcal{D}} \neg c \right]$$

Lemma. (Canonical 2-tree inversion)

Let $\mathcal{C}$ be a constraint set and let $c_0 \in \mathcal{C}$. For every $T \in \mathfrak{T}_{\mathcal{C}, c_0}$,

$$\{\mathcal{D} \subseteq \mathcal{C} \mid c_0 \in \mathcal{D}, \text{Tree}(\mathcal{D}) = T\} = \{S \cup T \mid S \subseteq \text{free}(T)\}.$$

$$\text{recursion: } r_{\mathcal{C}, c_0} = \frac{1}{\Pr[\mathcal{C} \setminus \{c_0\}]} \sum_{T \in \mathfrak{T}_{\mathcal{C}, c_0}} (-1)^{|T|-1} \Pr \left[\bigwedge_{c \in T} \neg c \mid \text{free}(T) \right] \prod_i \frac{1}{1 - r_{\mathcal{C}_i, c_i}}$$

Correlation decay of the recursion $\implies pD^2 \lesssim 1$

Truncation $\implies$ deterministic counting

Recursive Estimator $\implies$ randomized counting

Summary

Our Results: **Fast** Sampling for **General** CSPs

We obtain the first efficient sampling algorithm for **general** CSPs under LLL regime with **improved** condition [He, W., Yin, '22, 23]

Improved condition

$$150e^3 \cdot q^2 \cdot k \cdot (D+1)^5 \leq 1$$

q : domain size k : constraint width

Near-linear running time

$$\text{poly}(q, k, D) \cdot n \log n \text{ expected}$$

$n = |V|$: number of variables

Our Results: Sampling LLL for Large Domain

Condition. (Condition for sampling/counting atomic CSPs)

$\Phi = (V, \mathcal{Q}, \mathcal{C})$ is an atomic CSP satisfying

$$(8e)^3 \cdot p \cdot (D+1)^{2+\zeta} \leq 1,$$

$$\text{where } \zeta = \frac{2 \ln(2 - 1/q)}{\ln q - \ln(2 - 1/q)} = o_q(1)$$

Theorem. (Sampling/Counting **atomic** CSPs for large domain sizes)

We give poly-time sampling/counting algorithms for **atomic** CSPs satisfying condition above.

Running time: $n^{\text{poly}(k, D, \log q)}$

Our Results: A Counting LLL

Condition. (Counting LLL condition)

$\Phi = (V, \mathcal{Q}, \mathcal{C})$ is an atomic CSP satisfying

$$4e \cdot p \cdot (D+1)^2 \leq 1.$$

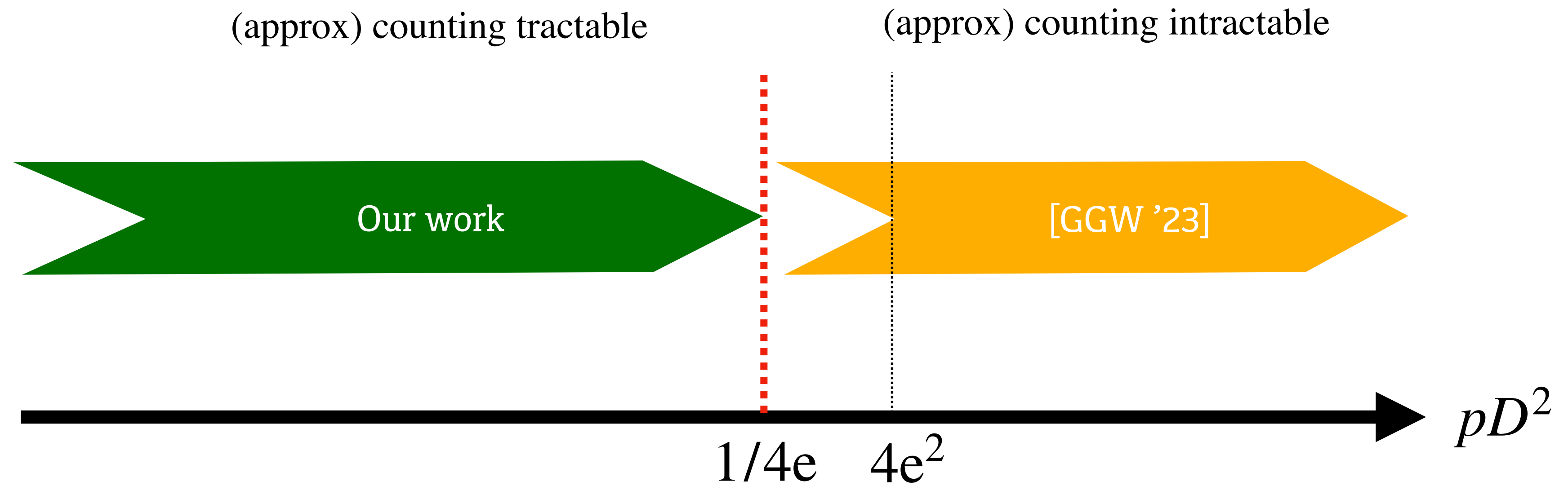
Theorem. (A Counting LLL)

We give poly-time counting algorithms for CSPs satisfying condition above.

Deterministic approximate counting: running time $n^{\text{poly}(k, D, \log q)}$;

Randomized approximate counting: running time $\text{poly}(k, D, q) \cdot n^2$

The Road Ahead



What is the critical threshold?

What happens **at** the critical threshold?

The Road Ahead

The Road Ahead

“

The asymptotic enumeration of objects of a given type frequently goes hand in hand with the problem of generating the same objects uniformly at random, and with finding some of the limiting probabilities in the resulting probability space (see Jerrum and Sinclair [58]). **However, the two problems are not equivalent.**

—N. C. Wormald. *Models of random regular graphs*

The Road Ahead

Decision: $\# > 0$?



(Approx) Counting:
Estimate $\#$



Search: Find a solution



Sampling: Find an
almost-uniform solution



The world diverges?

The Road Ahead

constraint languages



Lovász Local Lemma



“横看成岭侧成峰，远近高低各不同。”

More characterizations?

Thank you!



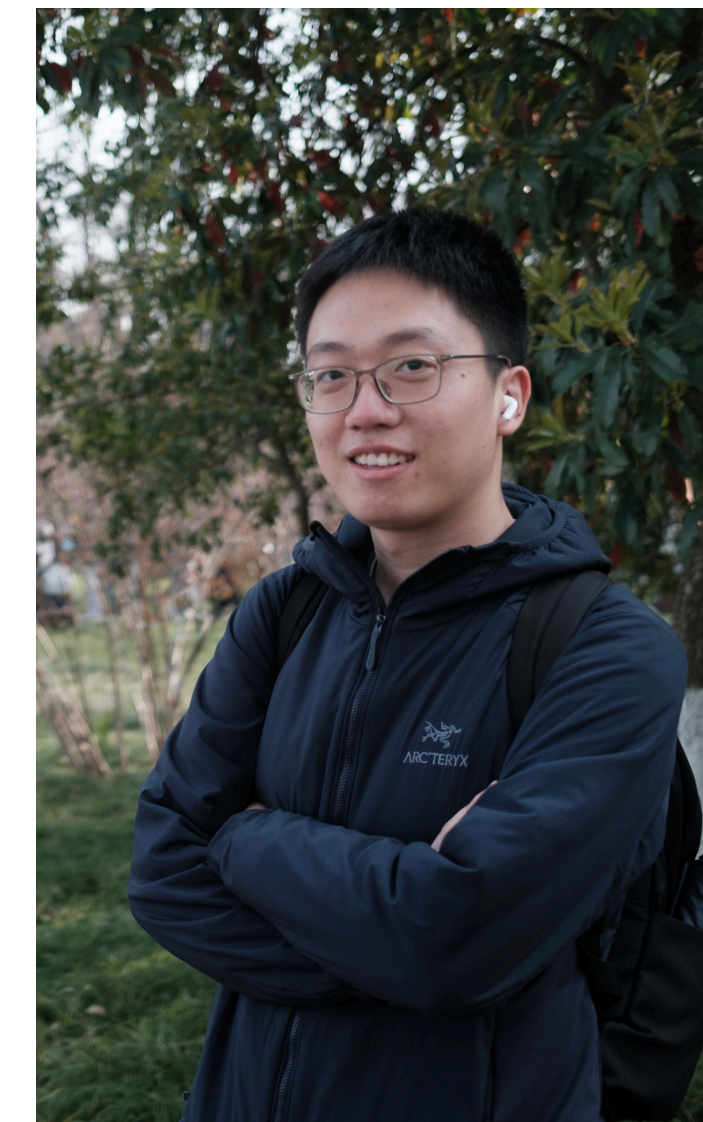
何昆
中国人民大学



刘弘洋
南京大学



尹一通
南京大学



张弈垚
南京大学



周灿
南京大学

[He-W.-Yin '22] Sampling Lovász Local Lemma for general constraint satisfaction solutions in near-linear time. **FOCS** '22

[He-W.-Yin '23] Deterministic counting Lovász local lemma beyond linear programming. **SODA** '23

[W.-Yin '24] A Sampling Lovász Local Lemma for Large Domain Sizes. **FOCS** '24

[Liu-W.-Yin-Zhang-Zhou '26] A Counting Lovász Local Lemma. arXiv preprint:2608.08616