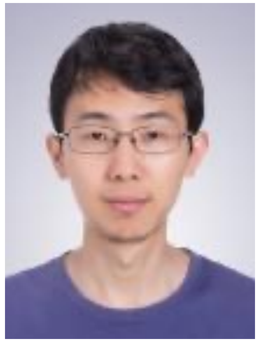
A Markov chain approach to the sampling Lovász local lemma

Weiming Feng

University of Edinburgh

Base on joint works with:

Heng Guo
Edin

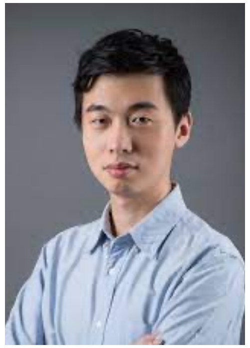


Ku He
ICT

Chunyang Wang
NJU

Jiaheng Wang
Edin

Yitong Yin
NJU



Chihao Zhang
SJTU

DIMAP Seminar, University of Warwick, UK

23rd Nov 2022

Conjunctive normal form (CNF)

- **Instance**: a formula $\Phi = (V, C)$, for example

$$\Phi = (x_1 \vee \neg x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee x_4) \wedge \boxed{(x_3 \vee \neg x_4 \vee \neg x_5)}$$ clause

$V = \{x_1, x_2, x_3, x_4, x_5\}$: set of Boolean variables; C: set of clauses.

- **SAT solutions**: an assignment of variables in V s.t. $\mathbf{\Phi = true}$.

- **Computational tasks**:
  - **Decision**: Does SAT solution exist?

    NP-Complete problem [Cook 1971, Levin 1973]
  - **Counting**: How many SAT solutions?

    #P-Complete problem [Valiant 1979].

(k,d)-CNF formula $\Phi = (V, C)$

$$\Phi = (x_1 \vee \neg x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee x_4) \wedge \underbrace{(x_3 \vee \neg x_4 \vee \neg x_5)}_{\text{clause size } k}$$

variable degree d

Lovász local lemma (LLL) [Erdős and Lovász, 1975]

SAT solution exists if $k \gtrsim \log d$ ($k \geq \log d + \log k + C$)

Algorithmic Lovász local lemma (ALLL) [Moser and Tardos, 2009]

Find a SAT solution when $k \gtrsim \log d$

Sampling Lovász local lemma (SLLL)

Sample a SAT solution uniformly at random if $k = \Omega(\log d)$

Counting Lovász local lemma (CLLL)

Approximately count number of SAT solutions if $k = \Omega(\log d)$

Sampling & counting k-SAT solutions

- **Input:** a (k, d)-CNF formula $\Phi = (V, C)$ with $|V| = n$, an error bound $\epsilon > 0$.

- **Almost uniform sampling:** random solution $\boldsymbol{X} \in \{\text{true}, \text{false}\}^V$ s.t.

  the *total variation distance* $d_{TV}(\boldsymbol{X}, \mu) \le \epsilon$,

  μ: the uniform distribution of all SAT solutions

- **Approximate counting:** estimate the number of SAT solutions, output

$$(1-\epsilon)Z \le \hat{\boldsymbol{Z}} \le (1+\epsilon)Z,$$

  Z = the number of SAT solutions

almost uniform sampling

self-reduction

[Jerrum, Valiant, Vazirani 1986]

$\Longleftrightarrow$

approximate counting

Work	**Regime**	**Running time or lower bound**	**Technique**
HSZ19	Monotone CNF[1] $k \gtrsim 2\log d$	$\mathrm{poly}(dk)n\log n$	Markov chain Monte Carlo (MCMC)

[1] *Monotone CNF: all variables appear* ***positively*** $\Phi = (x_1 \vee x_2 \vee x_3) \wedge (x_2 \vee x_4 \vee x_5) \wedge (x_3 \vee x_4 \vee x_6)$.

Work	Regime	Running time or lower bound	Technique
HSZ19	Monotone CNF[1] $k \gtrsim 2\log d$	$\mathrm{poly}(dk)n\log n$	Markov chain Monte Carlo (MCMC)
GJL17	$s \geq \min(\log dk, k/2)$ [2] $k \gtrsim 2\log d$	$\mathrm{poly}(dk)n$	partial rejection sampling

[1] *Monotone CNF: all variables appear* ***positively*** $\Phi = (x_1 \vee x_2 \vee x_3) \wedge (x_2 \vee x_4 \vee x_5) \wedge (x_3 \vee x_4 \vee x_6)$.
[2] *s: two dependent clauses share* ***at least*** *s variables.*

Work	Regime	Running time or lower bound	Technique
HSZ19	Monotone CNF[1] $k \gtrsim 2\log d$	$\mathrm{poly}(dk)n\log n$	Markov chain Monte Carlo (MCMC)
GJL17	$s \geq \min(\log dk, k/2)$ [2] $k \gtrsim 2\log d$	$\mathrm{poly}(dk)n$	partial rejection sampling
Moitra'17	$k \gtrsim 60\log d$	$n^{\mathrm{poly}(dk)}$	linear programming

[1] *Monotone CNF: all variables appear* ***positively*** $\Phi = (x_1 \vee x_2 \vee x_3) \wedge (x_2 \vee x_4 \vee x_5) \wedge (x_3 \vee x_4 \vee x_6)$.
[2] *s: two dependent clauses share* ***at least*** *s variables.*

Work	Regime	Running time or lower bound	Technique
HSZ19	Monotone CNF[1] $k \gtrsim 2\log d$	$\mathrm{poly}(dk)n\log n$	Markov chain Monte Carlo (MCMC)
GJL17	$s \geq \min(\log dk, k/2)$ [2] $k \gtrsim 2\log d$	$\mathrm{poly}(dk)n$	partial rejection sampling
Moitra'17	$k \gtrsim 60\log d$	$n^{\mathrm{poly}(dk)}$	linear programming
BGGGŠ15	$k \leq 2\log d - C$	**NP-hard**	-

[1] *Monotone CNF: all variables appear* ***positively*** $\Phi = (x_1 \vee x_2 \vee x_3) \wedge (x_2 \vee x_4 \vee x_5) \wedge (x_3 \vee x_4 \vee x_6)$.
[2] *s: two dependent clauses share* ***at least*** *s variables.*

Open Problem

fast sampling algorithm when $k \gtrsim 2\log d$

Our Results [F., Guo, Yin and Zhang, 2020 & F., He and Yin, 2021]

For any (k,d)-CNF formula satisfying $k \gtrsim 13 \log d$,

- ***sampling algorithm (main algorithm)***
  draw almost uniform random SAT solution in time $\tilde{O}(d^2 k^3 n^{1.001})$;

- ***counting algorithm (by reduction)***
  count SAT solutions approximately in time $\tilde{O}(d^3 k^3 n^{2.001})$.

Further Improvements on our algorithm

- Better analysis [Jain, Pham and Vuong, 2021]: improve to $k \gtrsim 5.741 \log d$.
- Better accuracy [He, Sun and Wu, 2021]: perfect sampling via CTFP

State-of-the-art [He, Wang and Yin, 2022]

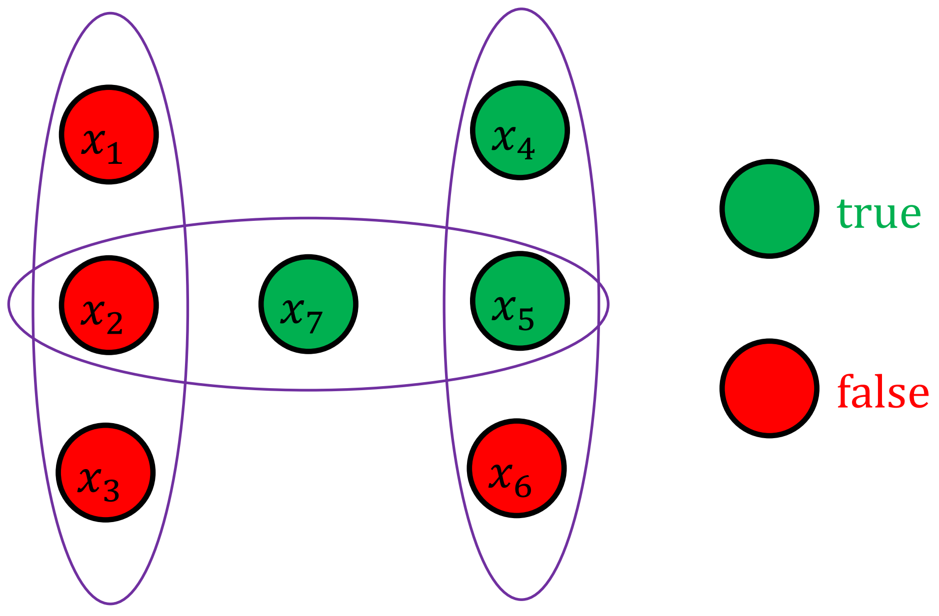
- $k \gtrsim 5 \log d$ (non-MCMC sampling algorithm)

Glauber dynamics (Gibbs sampling)

Start from an arbitrary solution $Y \in \{T, F\}^V$;

For each t from 1 to T **do**

- Pick $v \in V$ uniformly at random;
- Resample $Y_v \sim (\cdot \mid Y_{V \setminus v})$;



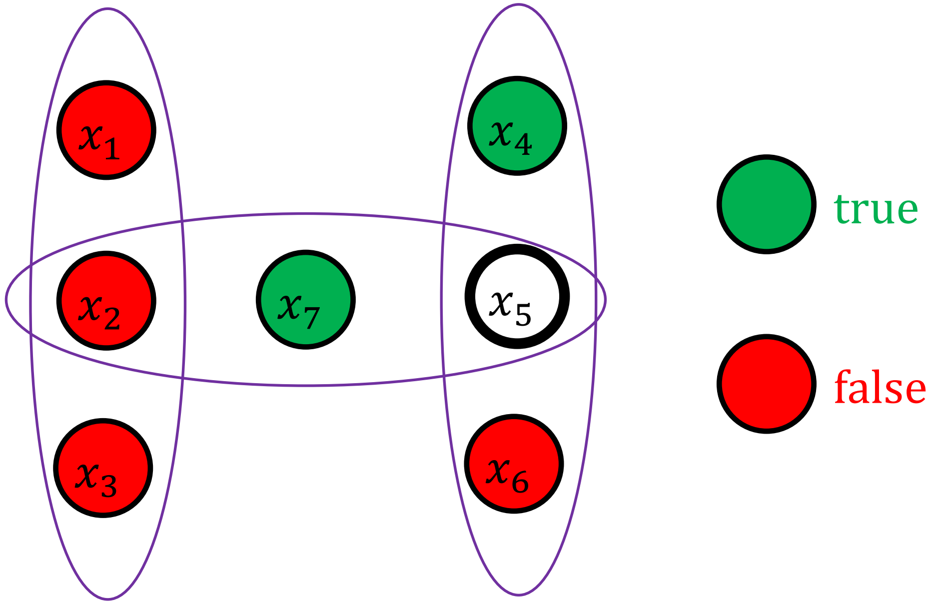
$(x_1 \vee \neg x_2 \vee x_3) \wedge (x_2 \vee x_7 \vee x_5) \wedge (x_4 \vee \neg x_5 \vee x_6)$

Glauber dynamics (Gibbs sampling)

Start from an arbitrary solution $Y \in \{T, F\}^V$;

For each t from 1 to T **do**

- Pick $v \in V$ uniformly at random;
- Resample $Y_v \sim (\cdot \mid Y_{V \setminus v})$;



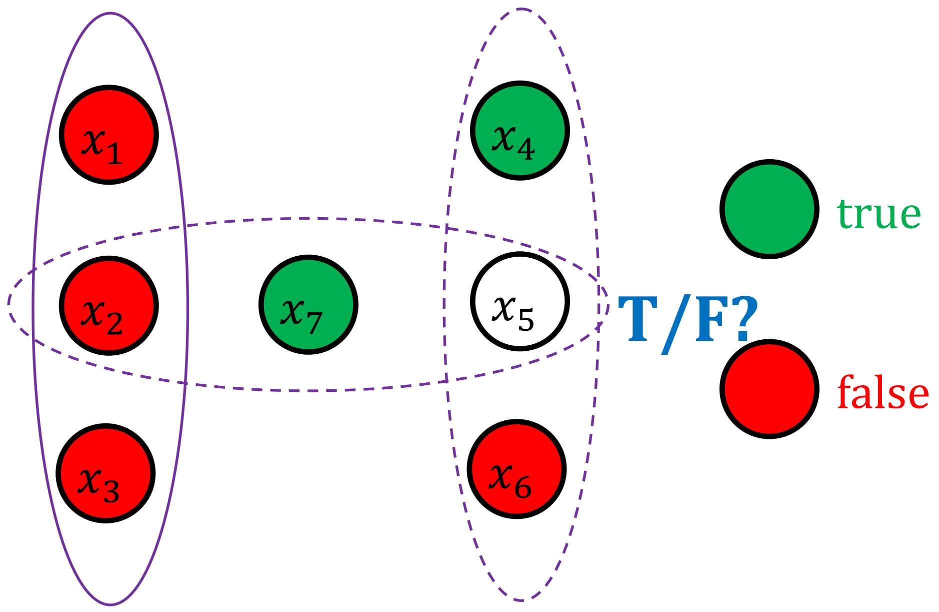
$(x_1 \vee \neg x_2 \vee x_3) \wedge (x_2 \vee x_7 \vee x_5) \wedge (x_4 \vee \neg x_5 \vee x_6)$

Glauber dynamics (Gibbs sampling)

Start from an arbitrary solution $Y \in \{T, F\}^V$;

For each t from 1 to T **do**

- Pick $v \in V$ uniformly at random;
- Resample $Y_v \sim \mu_v(\cdot \mid Y_{V \setminus v})$;



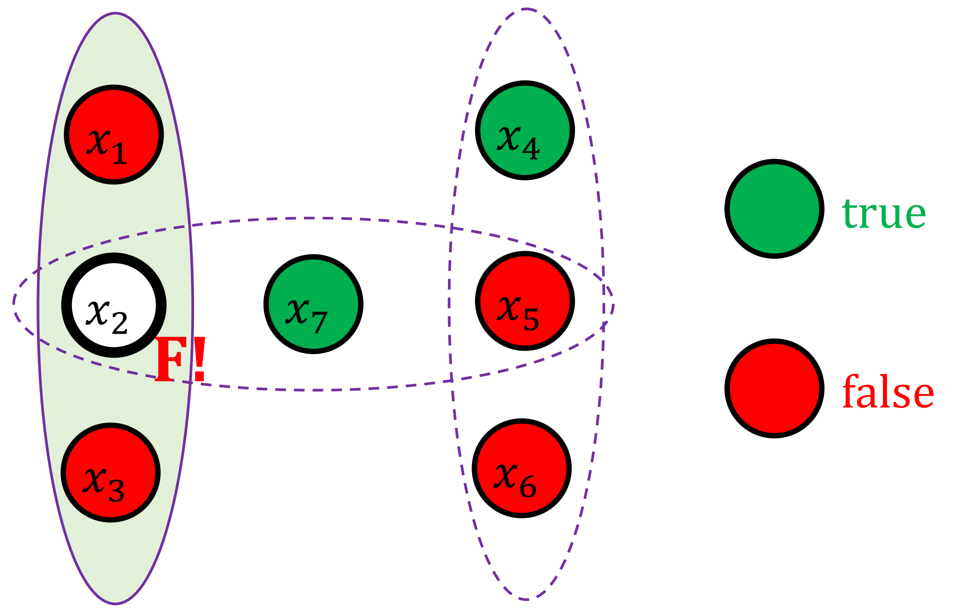
$(x_1 \vee \neg x_2 \vee x_3) \wedge (x_2 \vee x_7 \vee x_5) \wedge (x_4 \vee \neg x_5 \vee x_6)$

Glauber dynamics (Gibbs sampling)

Start from an arbitrary solution $Y \in \{T, F\}^V$;

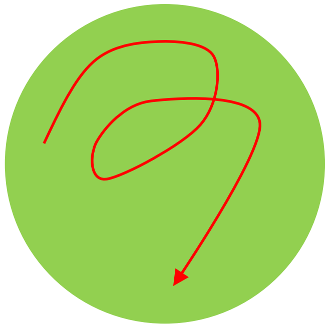
For each t from 1 to T **do**

- Pick $v \in V$ uniformly at random;
- Resample $Y_v \sim \mu_v(\cdot \mid Y_{V \setminus v})$;

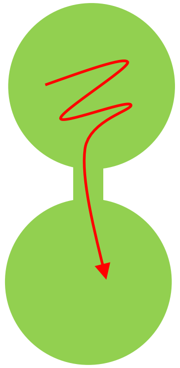


$(x_1 \vee \neg x_2 \vee x_3) \wedge (x_2 \vee x_7 \vee x_5) \wedge (x_4 \vee \neg x_5 \vee x_6)$

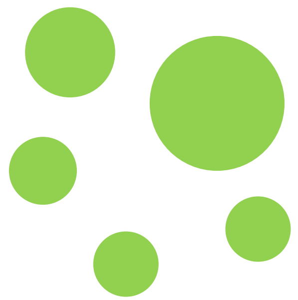
Glauber dynamics: *random walk* over solution space via *local update.*



rapid mixing



slow mixing



not mixing

Connectivity barrier (toy example)

- (k,d)-CNF formula $\Phi = (V, C)$ with $V = \{x_1, x_2, \dots x_k\}$:

$$\Phi = C_1 \wedge C_2 \wedge \cdots \wedge C_k.$$

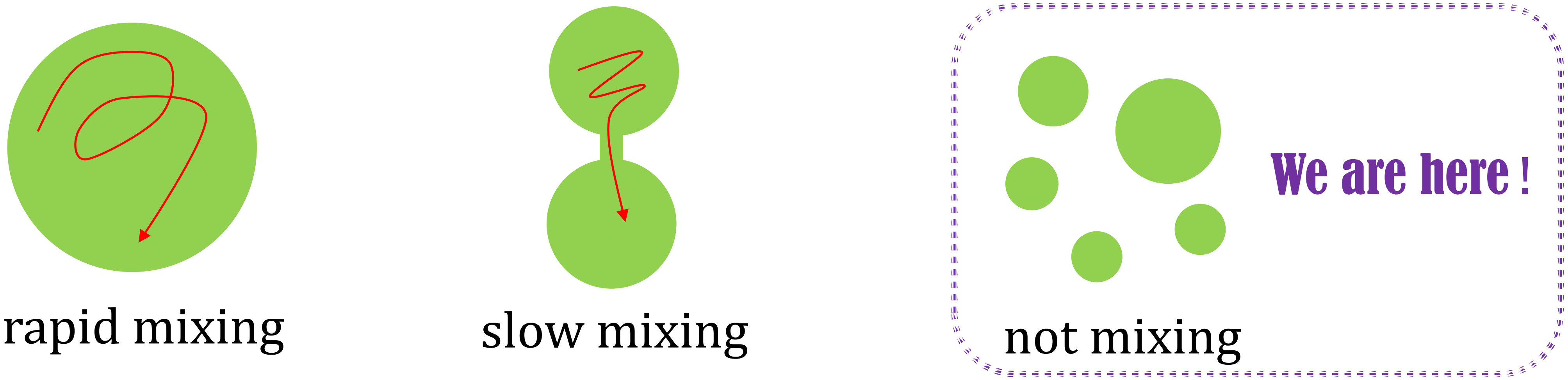
$$C_1 = (\neg x_1 \vee x_2 \vee x_3 \vee \cdots \vee x_k) \text{ forbids } 100\dots0$$

$$C_2 = (x_1 \vee \neg x_2 \vee x_3 \vee \cdots \vee x_k) \text{ forbids } 010\dots0$$

$$C_k = (x_1 \vee x_2 \vee x_3 \vee \cdots \vee \neg x_k) \text{ forbids } 000\dots1$$

- Any assignment $X \in \{0,1\}^V$ with $\|X\|_1 = 1$ is infeasible.
- All false solution **0** is **disconnected** with others.

Glauber dynamics: *random walk* over solution space via *local update.*



Question for SLLL

Can we obtain ***fast*** sampling algorithm
when the solution space is ***disconnected***?

Our technique: projection

Source: https://www.shadowmatic.com/presskit/images/IMG_0650.png

Projecting from a high dimension to a lower dimension to improve connectivity

Construct a *good subset* of variables $S \subseteq V$

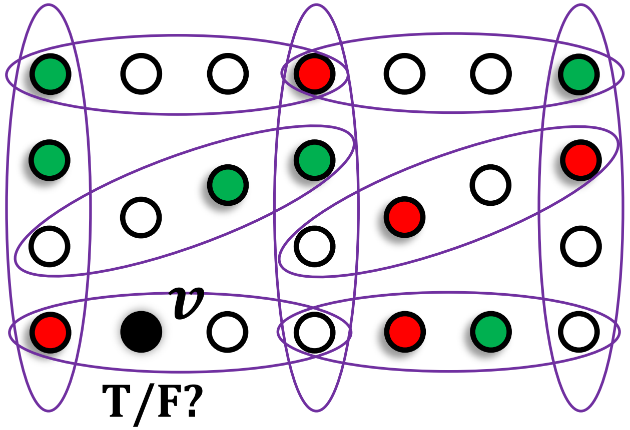
Run *Glauber dynamics* on *projected distribution* μ_S to draw sample $X \sim \mu_S$

Start from a uniform random $X \in \{\text{true,false}\}^S$;

For each t from 1 to T

- Pick a variable $v \in S$ uniformly at random;
- Resample $X_v \sim \mu_v(\cdot \mid X_{S \setminus v})$;

Return X;



Draw sample $Y \sim \mu_{V \setminus S}(\cdot \mid X)$ from the *conditional distribution*

There exists an ***efficiently constructible subset*** $S \subseteq V$ such that:

- the Glauber dynamics on μ_S is ***rapidly mixing***,
- the Glauber dynamics on μ_S can be ***implemented efficiently*** (draw $X_v \sim \boxed{\mu_v(\cdot \mid X_{S\setminus v})}$),
- sampling assignment for $V \setminus S$ can be ***implemented efficiently*** (draw $Y \sim \boxed{\mu_{V\setminus S}(\cdot \mid X)}$).

computing exact distr. can be #P-hard

Construct a *good subset* of variables $S \subseteq V$

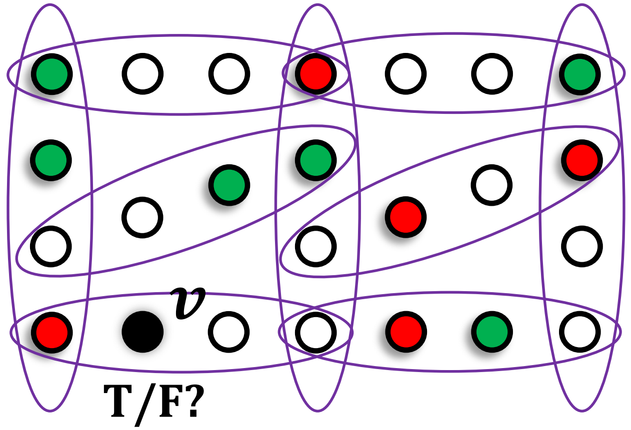
Run *Glauber dynamics* on *projected distribution* μ_S to draw sample $X \sim \mu_S$

Start from a uniform random $X \in \{\text{true,false}\}^S$;

For each t from 1 to T

- Pick a variable $v \in S$ uniformly at random;
- Resample $X_v \sim \mu_v(\cdot \mid X_{S \setminus v})$;

Return X;



Draw sample $Y \sim \mu_{V \setminus S}(\cdot \mid X)$ from the *conditional distribution*

Our Tasks:

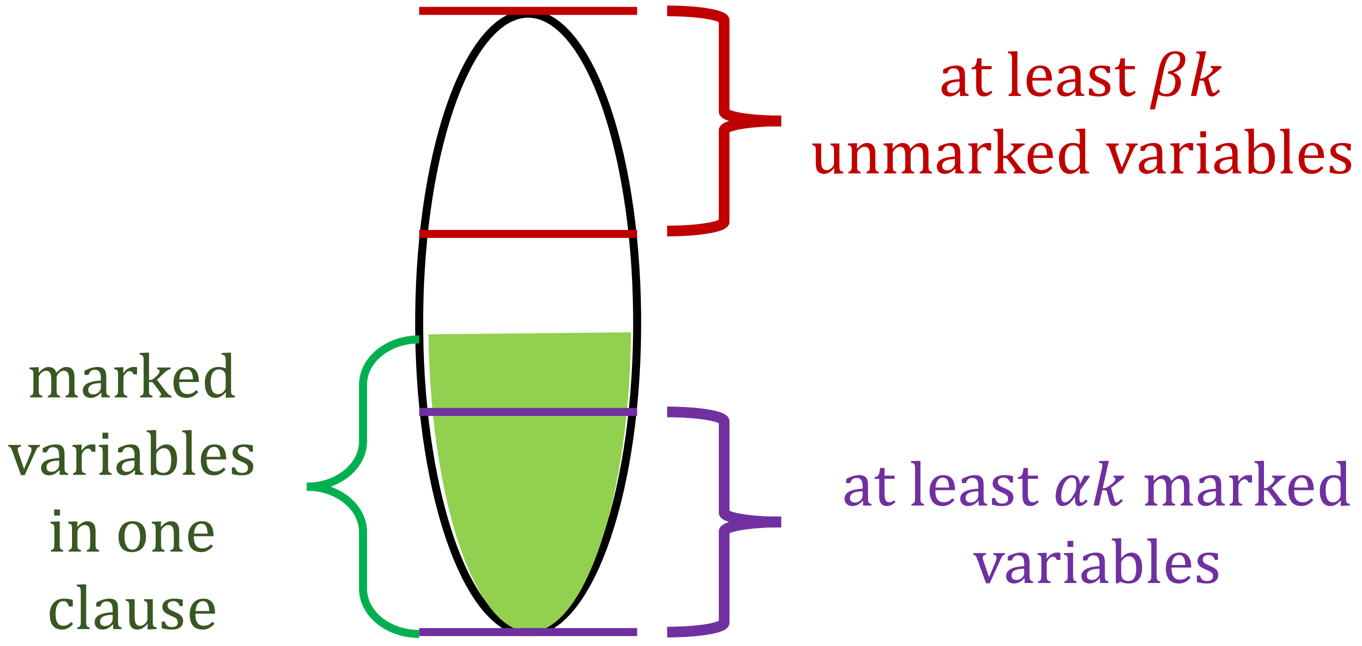
- Construct such a ***good subset*** $S \subseteq V$.
- Show that the Glauber dynamics on μ_S is ***rapidly mixing***.
- Given assignment on S, draw samples ***efficiently*** from the conditional distribution.

Mark variables [Moitra' 17]

Mark a set of variables $S \subseteq V$ such that

- each clause contains ***at least*** $\alpha k = \Theta(k)$ marked variables;
- each clause contains ***at least*** $\beta k = \Theta(k)$ unmarked variables;

where $0 < \alpha, \beta < 1$ are two constant parameters with $\alpha + \beta < 1$



Marked set $S \subseteq V$
is constructed by
algorithmic LLL
(Moser-Tardos)

The rapid mixing of Glauber dynamics on μ_S

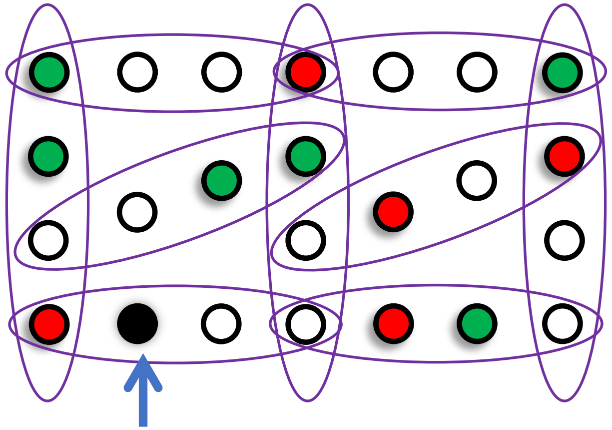
Each clause has **at least** βk **unmarked variables**

by LLL

Key Property: local uniformity

For any $v \in S$, any $X_{S \setminus v} \in \{T, F\}^{S \setminus v}$,

$$\forall c \in \{T, F\}, \qquad \mu_v(\, c \mid X_{S \setminus v}\,) = \frac{1}{2}\left(1 + \frac{1}{\mathrm{poly}(dk)}\right)$$



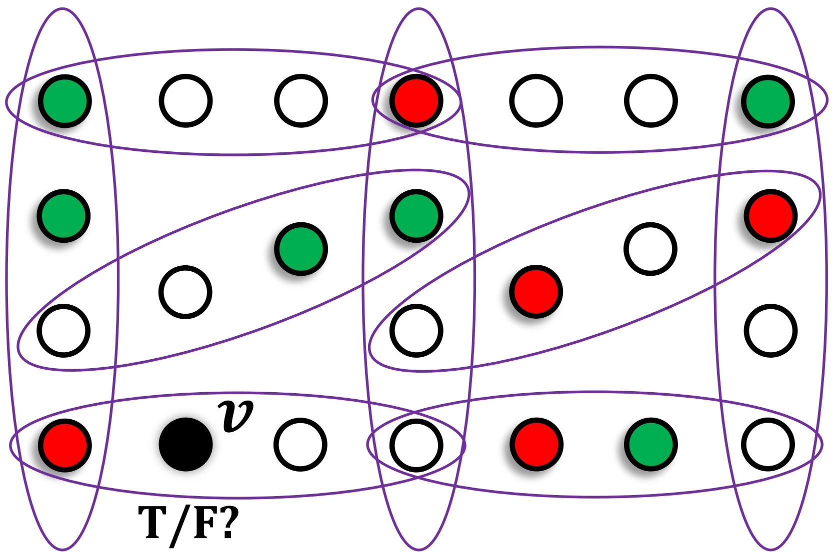
Pr[T]≈ 0.5 and Pr[F]≈ 0.5

- The Glauber dynamics is **connected** ($\mu_S(X_S) > 0$ for all $X_S \in \{T, F\}^S$)
- The Glauber dynamics **mixes** in $O(n \log n)$ steps
  - path coupling analysis [F., Guo, Yin and Zhang, 2020] [F., He and Yin, 2021]
  - information percolation analysis [Jain, Pham and Vuong, 2021]

Implementation of the algorithm

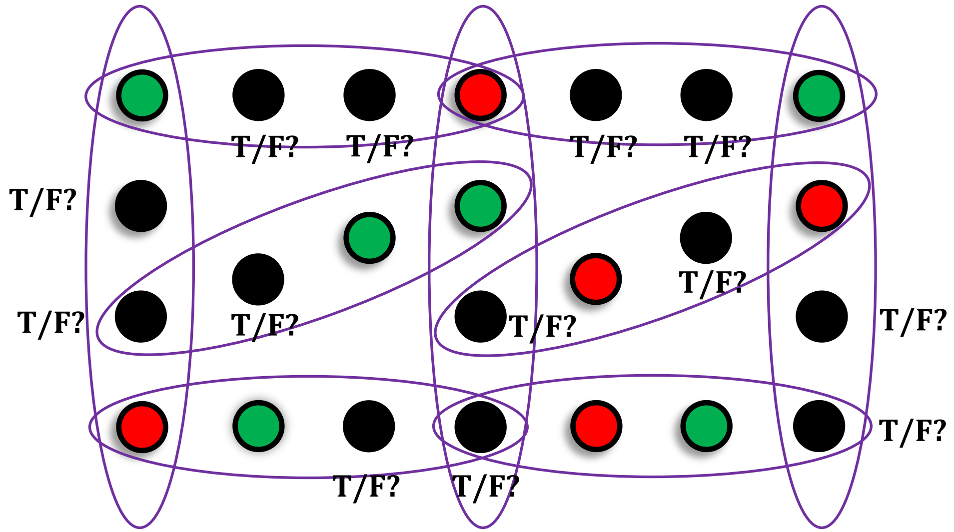
Transition of Glauber dynamics

resample $X_v \sim \mu_v(\cdot \mid X_{S\setminus v})$



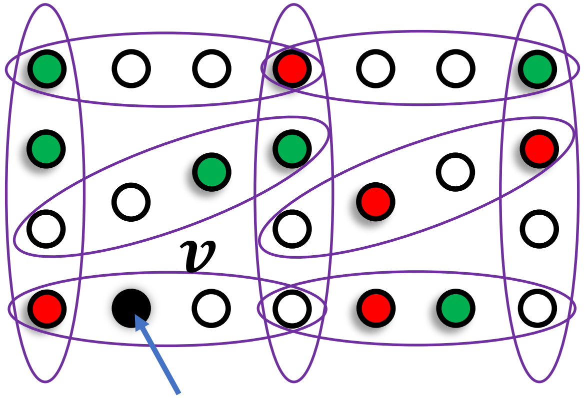
Sample unmarked variable

sample $Y \sim \mu_{V\setminus S}(\cdot \mid X)$



Challenge: computing the ***exact*** conditional distributions can be **#P-hard**.

Solution: draw ***approximate*** samples via rejection sampling

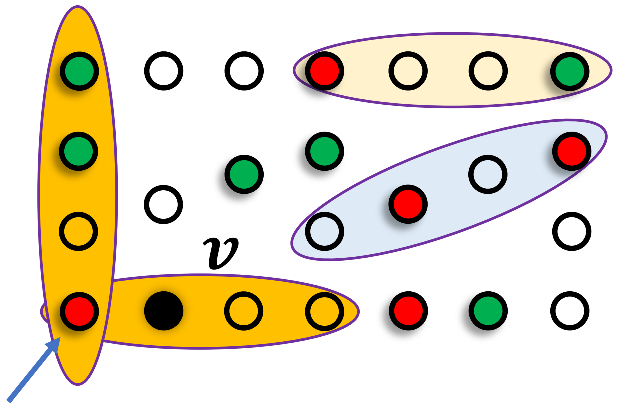


resample X_v from $\mu_v(\cdot \mid X_{S\setminus v})$

remove satisfied clauses

$(x_1 \vee x_2 \vee \neg x_3 \vee \neg x_4)$

$x_1 = \mathrm{T}$ or $x_4 = \mathrm{F}$ ✓



C: connected component containing v

each clause has
at least αk **marked variables**

local uniformity
each marked variables takes
an almost uniform random value

remove
each clause
with prob.
$\approx 1 - \left(\frac{1}{2}\right)^{\alpha k}$

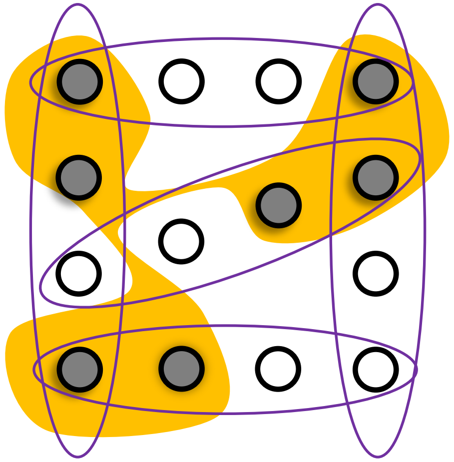
with high probability
component size is
$O(\log n)$

rejection sampling
on component

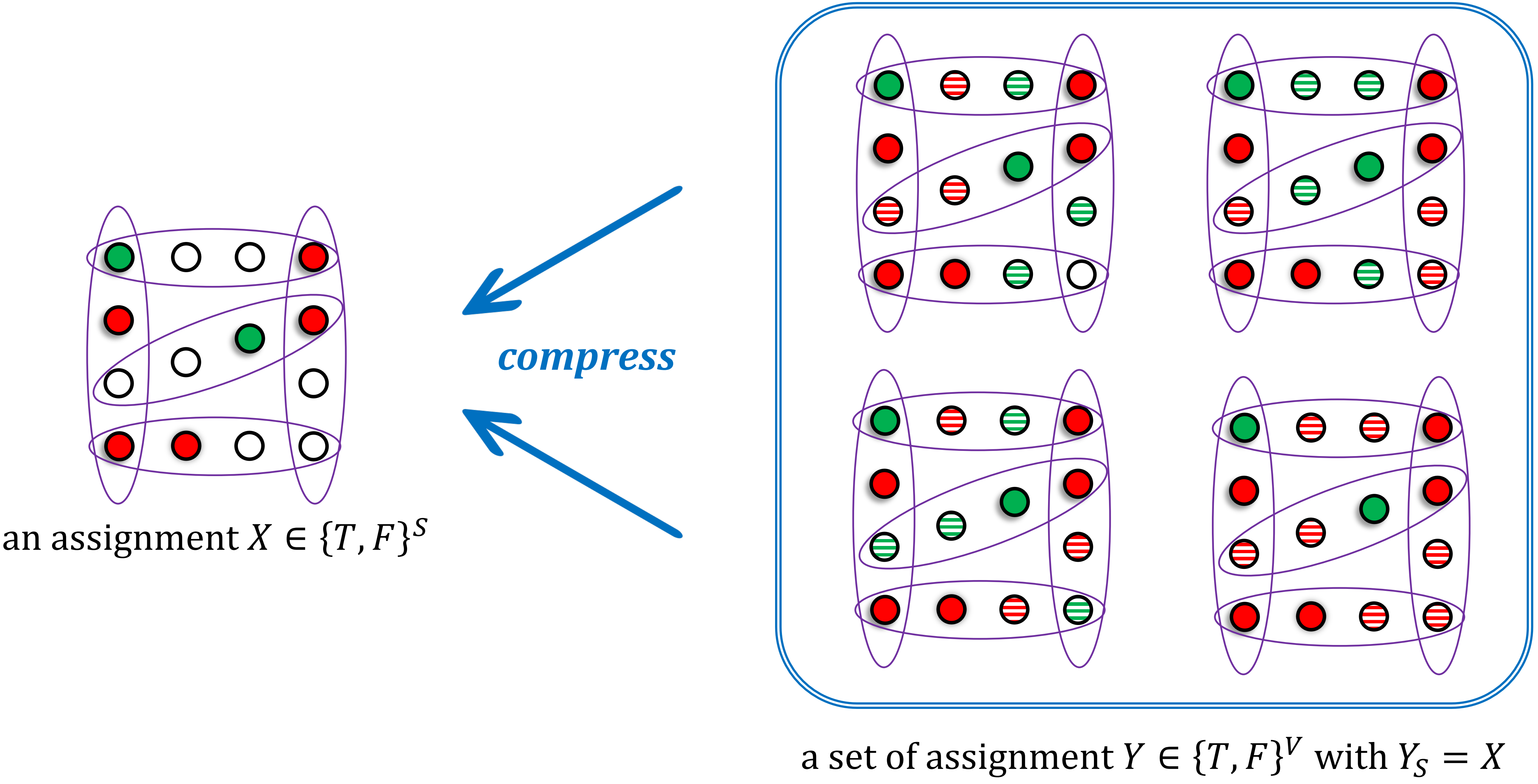
Input: a k-CNF formula $\Phi = (V, E)$ with maximum degree d, an error bound $\epsilon > 0$.

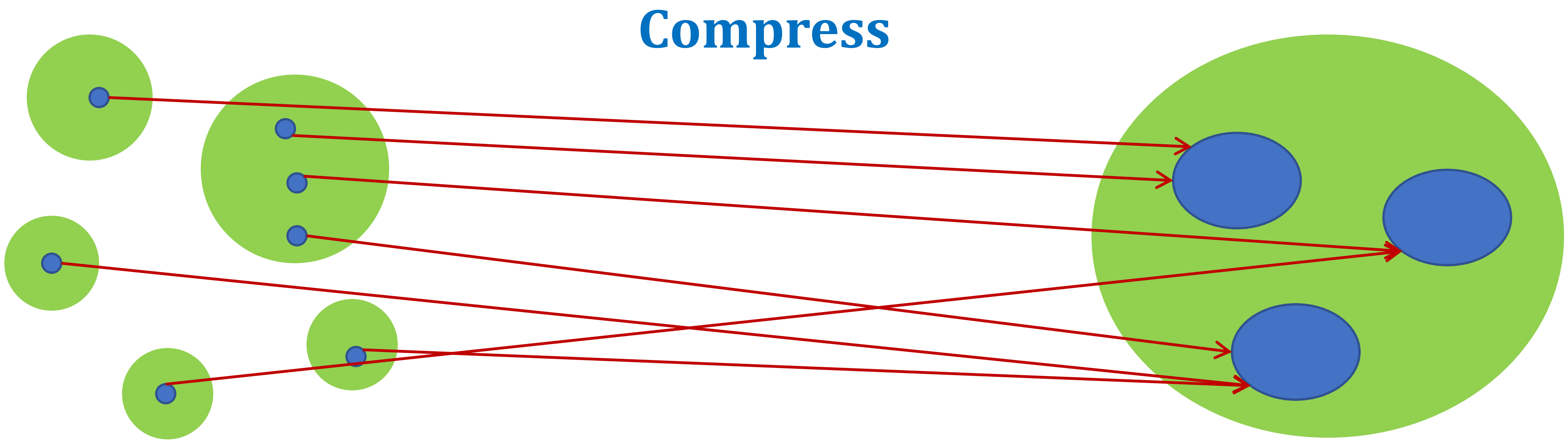
Output: a random sample $\sigma \in \{\text{true}, \text{false}\}^V$ s.t. $d_{TV}(\sigma, \mu) \leq \epsilon$.

1. Run ***Moser-Tardos*** algorithm to construct marked set $S \subseteq V$;
2. Run $\boxed{\textit{\textbf{Glauber dynamics}} \text{ on } \mu_S}$ for $O\left(n \log \frac{n}{\epsilon}\right)$ steps to sample $X \sim \mu_S$;
   (implemented using rejection sampling)
3. Run ***rejection sampling*** to draw $Y \sim \mu_{V \setminus S}(\cdot \mid X)$;
4. Return $X \cup Y$.



set of marked variables $S \subseteq V$


compress
an assignment $X \in \{T, F\}^S$
a set of assignment $Y \in \{T, F\}^V$ with $Y_S = X$

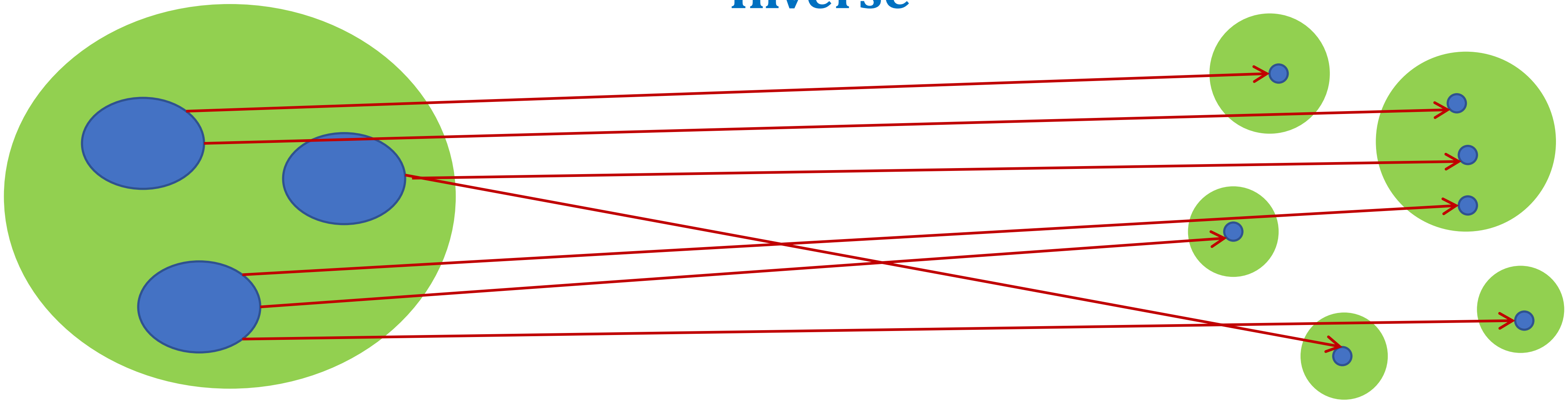


CNF solution space: disconnected

compressed space: connected

Rapid mixing of Glauber dynamics

Inverse



compressed state

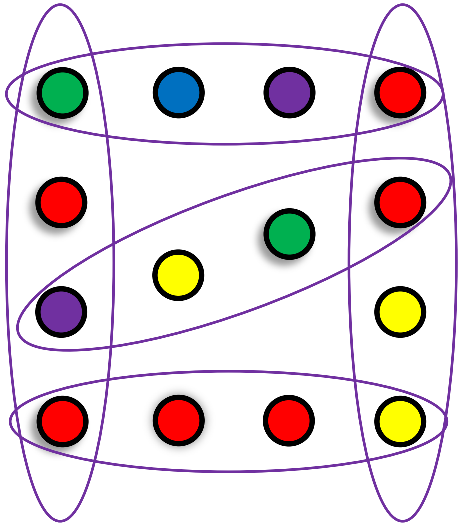
random CNF solution

Fast implementation of algorithm

Hypergraph colouring

Instance: a k-uniform hypergraph $H(V,E)$ with max degree d and q colours

- each hyperedge contains k vertices
- each vertex belongs to $\leq d$ hyperedges



Hypergraph colouring: $X \in [q]^V$ assign $v \in V$ a colour X_v

- no hyperedge is monochromatic

Lovász Local lemma and algorithmic LLL

- find a hypergraph colouring when $q \gtrsim d^{1/k}$ ($q \geq C_k d^{1/(k-1)}$)

Sampling Lovász Local lemma

- Sample a uniform hypergraph colouring in the local lemma regime

Work	Regime	Running time or lower bound	Technique
FA17	Linear hypergraph[1] $q \gtrsim \max\{\log n, d^{1/k}\}$	$O(n \log n)$	Markov chain Monte Carlo (MCMC)

[1] *Linear hypergraph: for all distinct hyperedge* $e_1, e_2 \in E, |e_1 \cap e_2| \leq 1$

Work	**Regime**	**Running time or lower bound**	**Technique**
FA17	Linear hypergraph[1] $q \gtrsim \max\{\log n, d^{1/k}\}$	$O(n \log n)$	Markov chain Monte Carlo (MCMC)
GLLZ17	$q \gtrsim d^{16/k}$	$n^{\mathrm{poly}(dk\log q)}$	linear programming

[1] *Linear hypergraph: for all distinct hyperedge* $e_1, e_2 \in E, |e_1 \cap e_2| \leq 1$

Work	Regime	Running time or lower bound	Technique
FA17	Linear hypergraph[1] $q \gtrsim \max\{\log n, d^{1/k}\}$	$O(n \log n)$	Markov chain Monte Carlo (MCMC)
GLLZ17	$q \gtrsim d^{16/k}$	$n^{\mathrm{poly}(dk\log q)}$	linear programming
GGW22	$q \lesssim d^{2/k}$	**NP-hard**	-
GGW22	Linear hypergraph $q \lesssim d^{1/k}$	**NP-hard**	

[1] *Linear hypergraph: for all distinct hyperedge* $e_1, e_2 \in E, |e_1 \cap e_2| \leq 1$

Open Problem

fast sampling algorithm when $q \gtrsim d^{2/k}$ (general) and $q \gtrsim d^{1/k}$ (linear)

Results obtained by MCMC with compression

MCMC with compression [F., He and Yin, 2021]

- $\tilde{O}(\text{poly}(dk) \cdot n^{1.001})$ running time if $q \gtrsim d^{9/k}$

Improved analysis on general hypergraph [Jain, Pham and Vuong, 2021]

- $\tilde{O}(\text{poly}(dk) \cdot n^{1.001})$ running time if $q \gtrsim d^{3/k}$

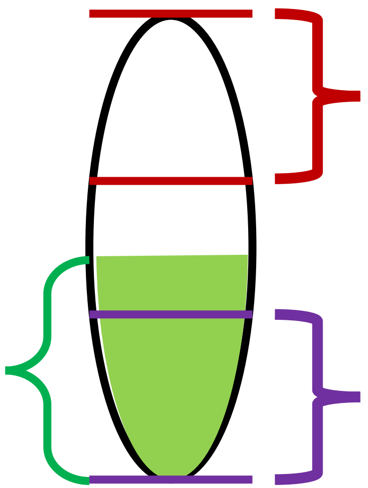
Improved analysis on linear hypergraph [F., Guo and Wang, 2022]

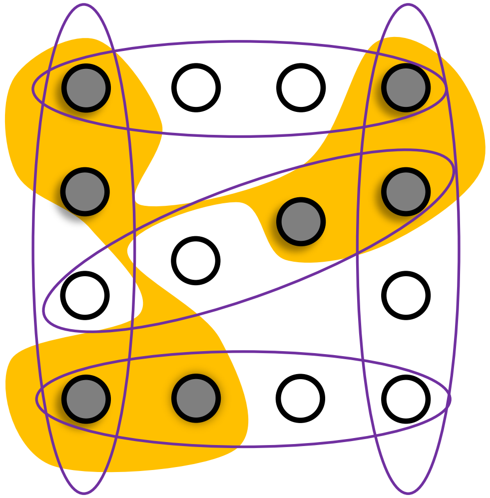
- $\tilde{O}(\text{poly}(dk) \cdot n^{1.001})$ running time if $q \gtrsim d^{(2+\delta)/k}$ for any constant $\delta > 0$

Perfect sampling via CFTP [He, Sun and Wu, 2021]

- $\tilde{O}(\text{poly}(dk) \cdot n)$ expected running time if $q \gtrsim d^{3/k}$

Mark/unmarked paradigm





set of marked variables

By **Lovász Local lemma,**

such marked set exists if

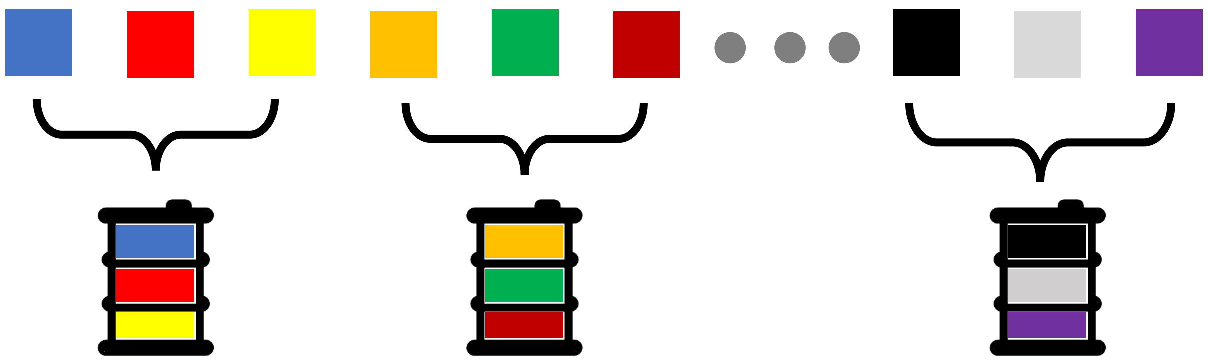
$k = \Omega(\log d)$

Local lemma regime:

- CNF: $k = \Omega(\log d)$
- Hypergraph colouring: $q = \Delta^{\Omega(1/k)}$

State compression for hypergraph colouring

the set of q colours



Bucket #1 Bucket #2 Bucket # s

the set of s buckets

Balanced projection scheme

$$h: [q] \to [s]$$

for any $j \in [s]$,

$$|h^{-1}(j)| = \frac{q}{s} \pm 1$$

$s = q^{\gamma}$ for constant $0 < \gamma < 1$

Projection of colouring

for any hypergraph colouring $X \in [q]^V$,

$$Y = h(X) \text{ s.t. } Y_v = h(X_v) \text{ for all } v \in V$$

Compression: ***different*** colouring $X \in [q]^V$ may be mapped to the ***same*** $Y \in [s]^V$

Distribution $\pi = \pi_h$ over the ***compressed space*** $[s]^V$

$$h(X) \sim \pi \text{ if } X \sim \mu,$$

μ is the ***uniform*** distribution over all hypergraph colourings

Sampling algorithm for hypergraph colourings

1. Choose a proper $s = q^\gamma$ to define the balanced projection scheme h;
2. Run ***Glauber dynamics*** on π_h for $O(n \log n)$ steps to sample $Y \sim \pi_h$; *(implemented using rejection sampling)*
3. Run ***rejection sampling*** to draw $X \in h^{-1}(Y)$ uniformly at random;
4. Return X.

Deterministic approximate counting

almost uniform sampling

standard reduction
[Bezáková, Štefankovič, Vazirani, Vigoda, 2008]
[Jerrum, Valiant, Vazirani 1986]

randomised approximate counting

Randomised counting: ***with probability at least*** $\mathbf{2/3}$, output $\hat{Z}$ satisfying

$$(1-\epsilon)Z \leq \hat{Z} \leq (1+\epsilon)Z,$$

Z total number of solutions (say total number of hypergraph colourings)

Deterministic counting: output $\hat{Z}$ satisfying

$$(1-\epsilon)Z \leq \hat{Z} \leq (1+\epsilon)Z$$

?

Deterministic approximate counting for hypergraph colourings

Work	Regime	Running time	Technique
GLLZ17	$q \gtrsim d^{14/k}$	$n^{\mathrm{poly}(dk \log q)}$	linear programming
JPV21	$q \gtrsim d^{7/k}$	$n^{\mathrm{poly}(dk \log q)}$	linear programming
HWY22	$q \gtrsim d^{5/k}$	$n^{\mathrm{poly}(dk \log q)}$	derandomisation

MCMC & Compression: sampling full colouring $X \in [q]^V$ in $O(n \log n)$ transition steps

⬇ Coupling towards the past [F., Guo, Wang, Wang and Yin, 2022]

Sample from marginal distribution μ_S for a small subset $S \subseteq V$ in $O(\log n)$ step

⬇ derandomisation

$n^{\mathrm{poly}(dk \log q)}$-time deterministic approximate counting if $q \gtrsim d^{3/k}$

Open problems

CNF formula

NP-Hard	?	Poly-Time Algorithm
$k \lesssim 2 \log d$		$k \gtrsim 5 \log d$

my guess

Hypergraph colouring

NP-Hard	?	Poly-Time Algorithm
$q \lesssim \Delta^{2/k}$		$q \lesssim \Delta^{3/k}$

my guess

Thank you! Q&A