Fast sampling and counting k-SAT solutions in the local lemma regime

Weiming Feng
Nanjing University

Joint work with: Heng Guo (University of Edinburgh)
Yitong Yin (Nanjing University)
Chihao Zhang (Shanghai Jiao Tong University)

Online Seminar
Institute of Computing Technology, Chinese Academy of Sciences

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. $\mathbf{\Phi = true}$.
- Fundamental computational tasks for CNF formula:
  - **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 $\Phi = (V, C)$

- Each clause contains k Boolean variables.
- Each variable belongs to at most d clauses, e.g. **max degree** $\boldsymbol{\leq d}$.

Example: (3,2)-CNF formula $(x_1 \vee \neg x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee x_4) \wedge (x_3 \vee \neg x_4 \vee \neg x_5)$

Lovász Local Lemma (LLL)

Suppose a (k, d)-CNF formula satisfies $\boldsymbol{k \gtrsim \log d}$ $(k \geq \log d + \log k + C)$.

- **Existence** [Erdős, Lovász, 1975]

  If each variable takes a value in {true,false} *uniformly and independently*

$$\Pr[\text{all clauses are satisfied}] \geq \left(1 - \frac{1}{2dk}\right)^{dn} > 0,$$

  which implies the k-SAT solution ***must exist***;

- **Construction** [Moser, Tardos, 2010]

  a k-SAT solution can be ***constructed*** *in expected time* $O(ndk)$*.*

Sampling & counting k-SAT solutions

- **Input:** a (k,d)-CNF formula $\Phi = (V, C)$ with $|V| = n$**,** and error bound $\epsilon > 0$.
- **Almost uniform sampling:** generate a random SAT solution $X \in \{\text{true}, \text{false}\}^V$ s.t. the *total variation distance is at most* ϵ*,*

$$d_{TV}(X,\mu) = \frac{1}{2} \sum_{\sigma \in \{\text{true,false}\}^V} |\Pr[X = \sigma] - \mu(\sigma)| \leq \epsilon$$

μ: the uniform distribution of all k-SAT solutions.

Sampling & counting k-SAT solutions

- **Input:** a (k, d)-CNF formula $\Phi = (V, C)$ with $|V| = n$, and error bound $\epsilon > 0$.
- **Almost uniform sampling:** generate a k-SAT solution $X \in \{\text{true}, \text{false}\}^V$ s.t.

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

  μ: the uniform distribution of all k-SAT solutions.

- **Approximate counting:** estimate the number of k-SAT solutions, e.g. output

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

  Z = the number of k-SAT solutions.

Work	Regime	Running time/lower bound	Technique
Hermon et al.'19	Monotone CNF[1] $k \gtrsim 2\log d$	$\text{poly}(dk) n \log n$	Markov chain Monte Carlo (MCMC)
Guo et al.'17	$s \geq \min(\log dk, k/2)$ [2] $k \gtrsim 2\log d$	$\text{poly}(dk) n$	Partial rejection sampling
Moitra'17	$k \gtrsim 60 \log d$	$n^{\text{poly}(dk)}$	Linear programming
Bezáková et al.'15	$k \leq 2\log d - C$	NP-hard	-

Table: previous results for sampling SAT solutions of (k, d)-CNF formulas

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

Open Problem: Can we sample *general* (k, d)-CNF solutions such that

- the threshold down to $k \gtrsim 2\log d$;
- the running time $\text{poly}(dk)\tilde{O}(n)$.

Our result

Work	Regime	Running time/lower bound	Technique
Hermon et al.'19	Monotone CNF $k \gtrsim 2\log d$	$\mathrm{poly}(dk) n \log n$	MCMC
Guo et al.'17	$s \geq \min(\log dk, k/2)$ $k \gtrsim 2\log d$	$\mathrm{poly}(dk) n$	Partial rejection sampling
Moitra'17	$k \gtrsim 60 \log d$	$n^{\mathrm{poly}(dk)}$	Linear programming
Bezáková et al.'15	$k \lesssim 2\log d - C$	NP-hard	-
This work	$\boldsymbol{k \gtrsim 20 \log d}$	$\boldsymbol{\tilde{O}(d^2 k^3 n^{1.000001})}$	**MCMC**

Table: results for sampling SAT solutions of (k, d)-CNF formulas

Main theorem (this work)

For any sufficiently small $\zeta < 2^{-20}$, any (k, d)-CNF formula satisfying

$$k \geq 20 \log d + 20 \log k + 3 \log \frac{1}{\zeta},$$

- ***sampling algorithm (main algorithm)***
  draw almost uniform random k-SAT solution in time $\tilde{O}\left(d^2 k^3 n^{1+\zeta}\right)$;

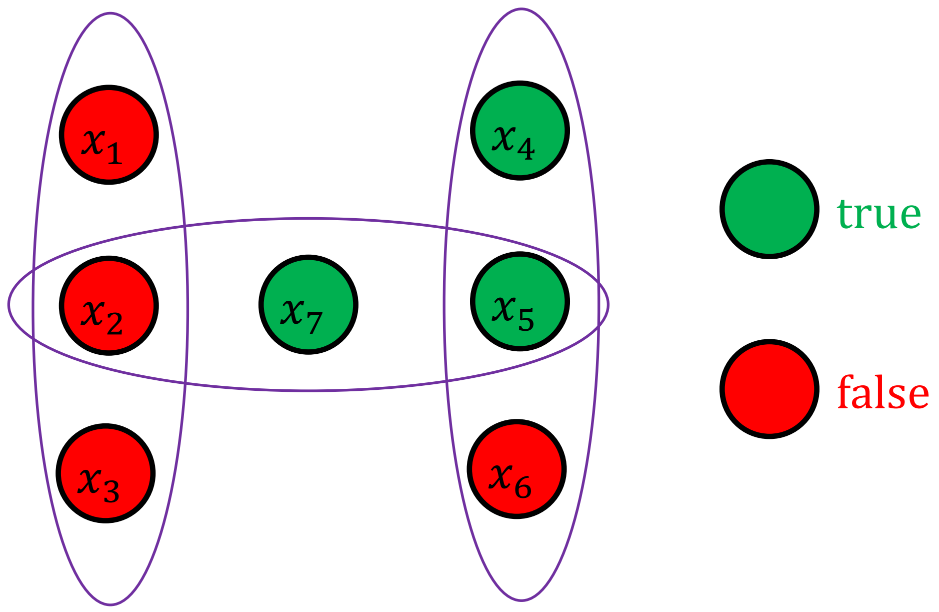
- ***counting algorithm (by simulated annealing reduction)***
  count #k-SAT solutions approximately in time $\tilde{O}\left(d^3 k^3 n^{2+\zeta}\right)$;

Classic 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 $Y_v \sim (\cdot \mid Y_{V\setminus v})$;



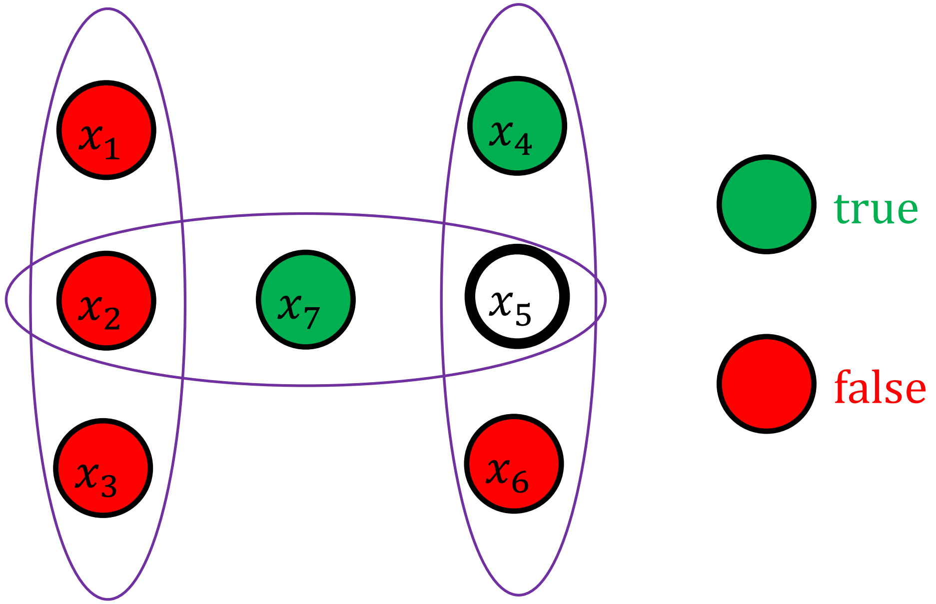
$$(x_1 \vee \neg x_2 \vee x_3) \wedge (x_2 \vee x_7 \vee x_5) \wedge (x_4 \vee \neg x_5 \vee x_6)$$

Classic 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 $Y_v \sim (\cdot \mid Y_{V \setminus v})$;



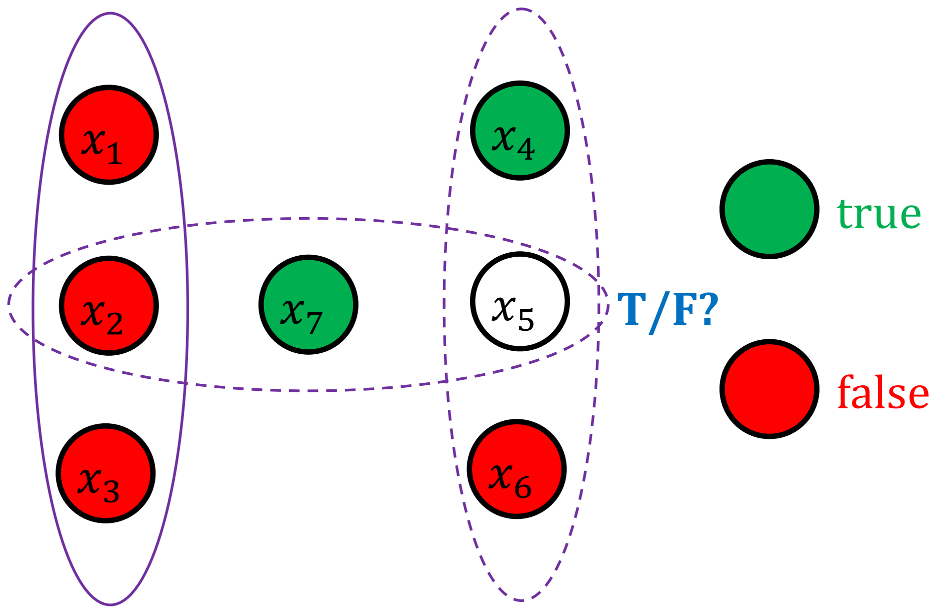
$(x_1 \vee \neg x_2 \vee x_3) \wedge (x_2 \vee x_7 \vee x_5) \wedge (x_4 \vee \neg x_5 \vee x_6)$

Classic 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 $Y_v \sim \mu_v(\cdot \mid Y_{V \setminus v})$;



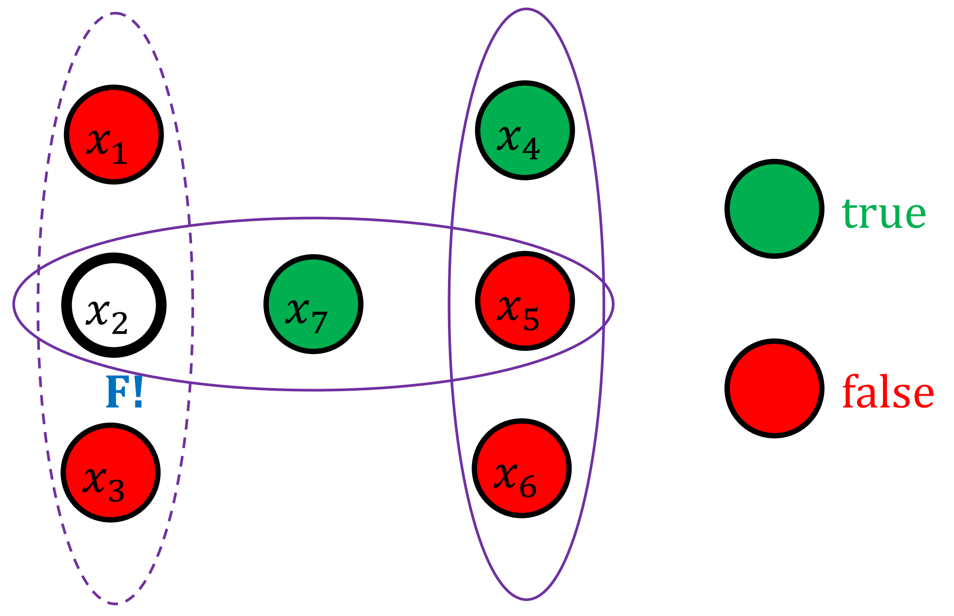
$(x_1 \vee \neg x_2 \vee x_3) \wedge (x_2 \vee x_7 \vee x_5) \wedge (x_4 \vee \neg x_5 \vee x_6)$

Classic 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 $Y_v \sim \mu_v(\cdot \mid Y_{V \setminus v})$;



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

Connectivity barrier (toy example)

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

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

$$C_1 = (\neg x_1 \lor x_2 \lor x_3 \lor \cdots \lor x_k) \text{ forbids } 100 \dots 0$$

$$C_2 = (x_1 \lor \neg x_2 \lor x_3 \lor \cdots \lor x_k) \text{ forbids } 010 \dots 0$$

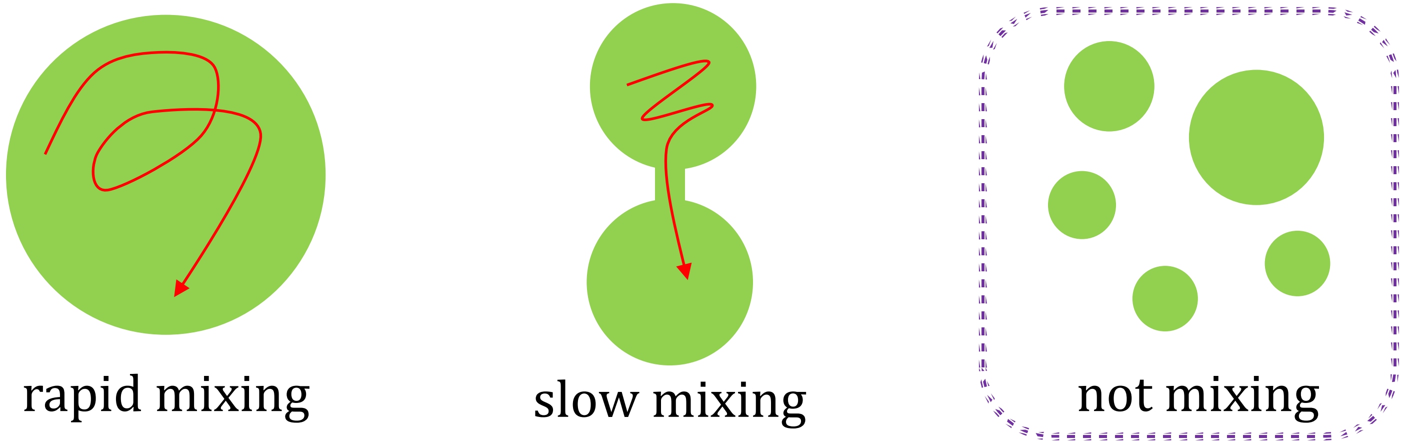
$$C_k = (x_1 \lor x_2 \lor x_3 \lor \cdots \lor \neg x_k) \text{ forbids } 000 \dots 1$$

- 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.*
- **Local Markov chain**: one of the ***most fundamental*** *approach* for sampling:

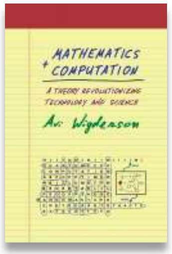
uniform graph coloring ✓ weighted matching/independent set ✓

Ising/spin system ✓ bases of a matroid ✓



We are here !

For sampling CNF solutions, the ***MCMC approach*** meets the ***connectivity barrier***.



"the solution space (and hence the natural Markov chain) is not connected"

Mathematics and Computation [Wigderson'19]

Bypass the connectivity barrier

Work	Regime	Running time	Technique
Hermon et al.'19	Monotone CNF $k \gtrsim 2 \log d$	$\mathrm{poly}(dk) n \log n$	MCMC
Guo, Jerrum, Liu'17	$s \geq \min(\log dk, k/2)$ $k \gtrsim 2 \log d$	$\mathrm{poly}(dk) n$	Partial rejection sampling
Moitra'17	$k \gtrsim 60 \log d$	$n^{\mathrm{poly}(dk)}$	Linear programming

monotone CNF

heavy intersection

constant d and k

Non-MCMC approach

Technique Motivation:

*Can **MCMC approach** bypass the **connectivity barrier**?*

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 $M \subseteq V$

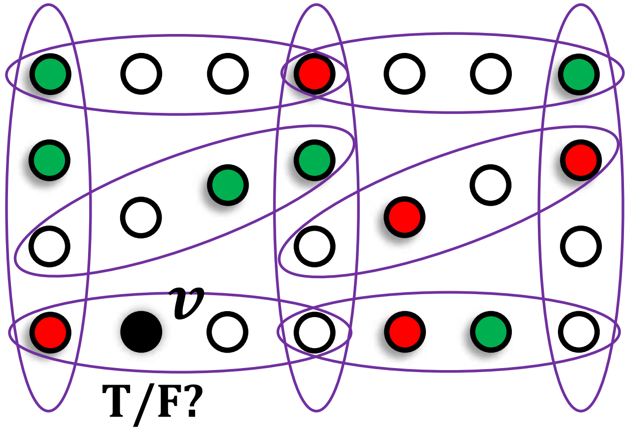
Run *Glauber dynamics* on *projected distribution* μ_M to draw sample $X \sim \mu_M$

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

For each t from 1 to T

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

Return $X \in \{\text{true,false}\}^M$.



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

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

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

computing exact distr. can be #P-hard

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

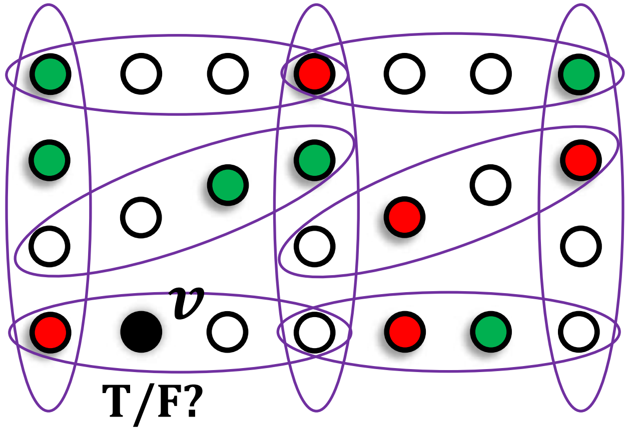
Run *Glauber dynamics* on *projected distribution* μ_M to draw sample $X \sim \mu_M$

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

For each t from 1 to T

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

Return X;



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

Our Tasks:

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

Mark variables [Moitra' 17]

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

- each clause contains ***at least*** $\alpha k \approx 0.11k$ marked variables;
- each clause contains ***at least*** $\beta k \approx 0.51k$ unmarked variables;

Mark each $v \in V$ independently w.p. $P = \frac{1+\alpha-\beta}{2}$ to construct a random set $\mathcal{M} \subseteq V$

by ***LLL***, $\Pr[\mathcal{M} \text{ satisfies above property}] > 0$

Lemma: marking (prove via LLL)

If $k \geq 20 \log d + 20 \log k + 3 \log \frac{1}{\zeta}$, then

$$\Pr\left[\text{Moser–Tardos alg } \textbf{constructs } M \text{ in time } O\left(ndk \log \frac{1}{\epsilon}\right)\right] \geq 1 - \frac{\epsilon}{3}.$$

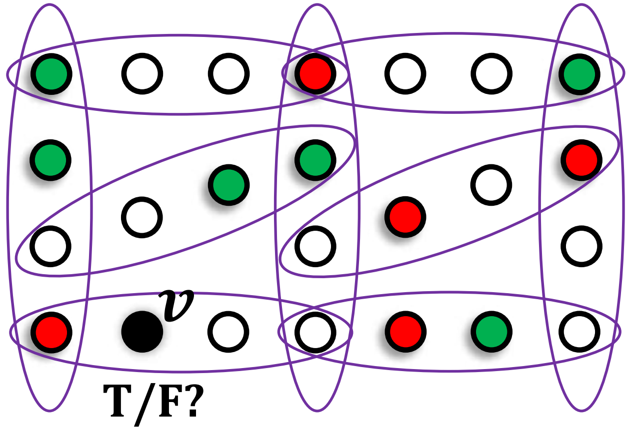
The rapid mixing of Glauber dynamics on μ_M

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

For each t from 1 to T

- Pick a marked variable $v \in M$ u.a.r.;
- Resample $X_v \sim \mu_v(\cdot \mid X_{M \setminus v})$;

Return X;



Each clause has $\geq \beta k$ *unmarked variables*, by **LLL** [Haeupler, Saha, Srinivasan' 11]:

Property: local uniformity (proved via LLL [Haeupler, Saha, Srinivasan' 11])

For any assignment $X_{M \setminus v}$, the distribution $\mu_v(\cdot \mid X_{M \setminus v})$ is ***close to uniform***:

$$\forall c \in \{\text{true,false}\}, \qquad \mu_v(c \mid X_{M \setminus v}) = \frac{1}{2} \pm \frac{1}{\text{poly}(dk)}.$$

- After each transition, $\Pr[X_v = \text{true}] \approx \frac{1}{2} > 0$ and $\Pr[X_v = \text{false}] \approx \frac{1}{2} > 0$.
- **Local uniformity** ⟹ **Glauber dynamics on μ_M is connected!**

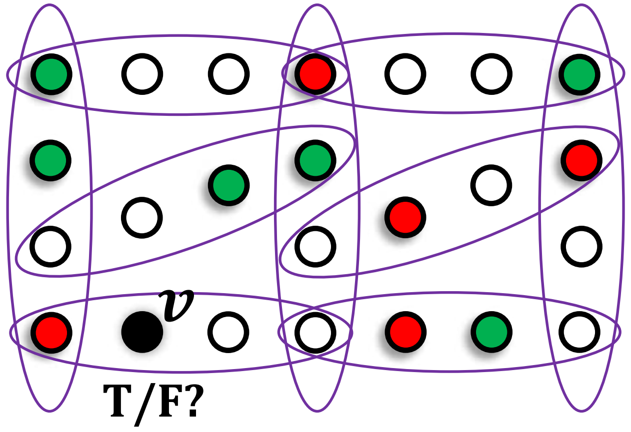
The rapid mixing of Glauber dynamics on μ_M

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

For each t from 1 to $T = 2n \log \frac{4n}{\epsilon}$

- Pick a marked variable $v \in M$ u.a.r.;
- Resample $X_v \sim \mu_v(\cdot \mid X_{M \setminus v})$;

Return X;



Lemma: rapid mixing

If $T = 2n \log \frac{4n}{\epsilon}$, then the returned random assignment X satisfies

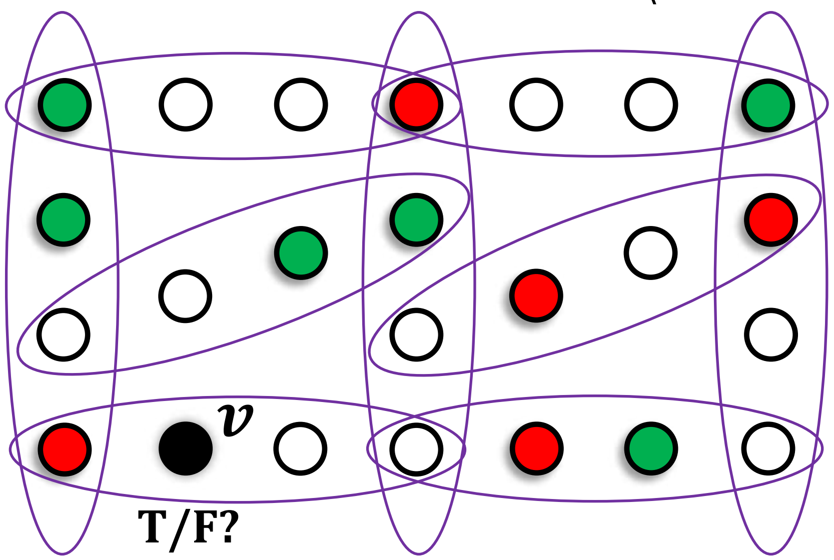
$$d_{TV}(X, \mu_M) \le \frac{\epsilon}{3}.$$

- Use ***path coupling*** [Bubley, Dyer'97] to bound the mixing time.
- Use "***disagreement coupling***" [Moitra'17, Guo et al.' 18] to bound the discrepancy of path coupling.
- Use ***local uniformity property (LLL)*** to show the small discrepancy of "disagreement coupling".

Implementation of the algorithm

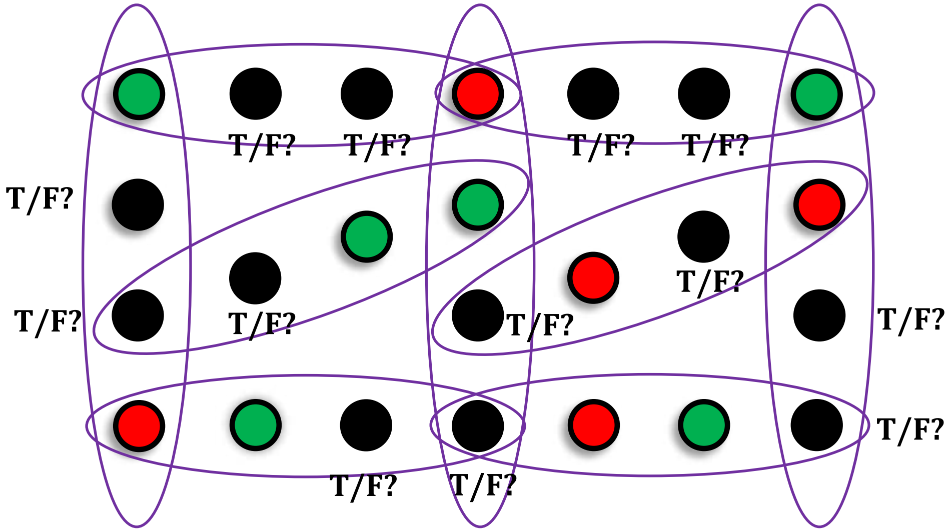
Transition of Glauber dynamics

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



Sample unmarked variable in last step

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



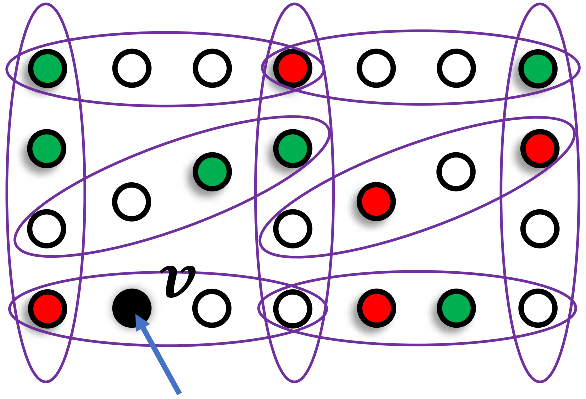
Challenge: computing the ***exact*** conditional distributions can be **#P-hard**.

$Z_T = \#\{Y \in \{T,F\}^V \text{ is a SAT solution} \mid Y_v = T, Y_{M\setminus v} = X_{M\setminus v}\}$

$Z_F = \#\{Y \in \{T,F\}^V \text{ is a SAT solution} \mid Y_v = F, Y_{M\setminus v} = X_{M\setminus v}\}$

$$\mu_v(T \mid X_{M\setminus v}) = \frac{Z_T}{Z_T + Z_F}$$

$$\mu_v(F \mid X_{M\setminus v}) = \frac{Z_F}{Z_T + Z_F}$$

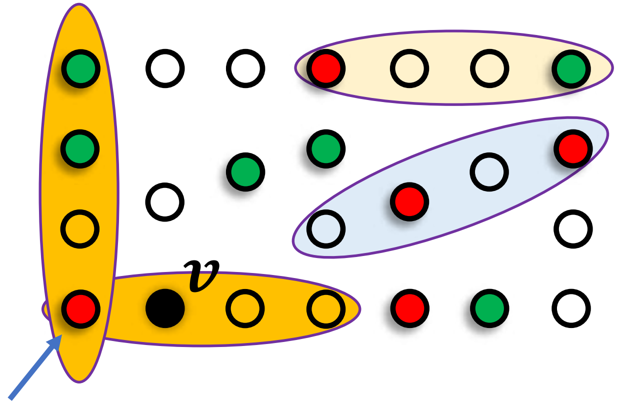


resample X_v from $\mu_v(\cdot \mid X_{M\setminus v})$

remove satisfied clauses

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

$x_1 = \text{true}$ or $x_4 = \text{false}$



C: connected component containing v

Key Property: w.h.p., the graph is deconstructed into ***small components*** of size $O\left(dk \log \frac{n}{\epsilon}\right)$

Start from a **uniform random** $X \in \{\text{true,false}\}^M$; $\forall u \in M,\ \Pr[X_u = T] = \frac{1}{2}, \Pr[X_u = F] = \frac{1}{2}$

For each t from 1 to T

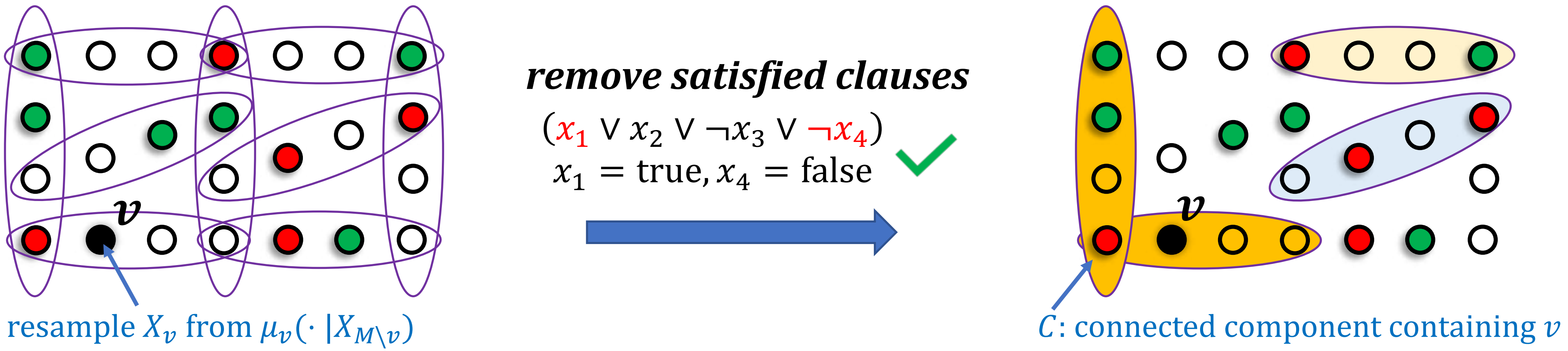
- Pick a marked variable $v \in M$ u.a.r.;
- Resample $X_v \sim \mu_v(\cdot \mid X_{M\setminus v})$; *by local uniformity* $\Pr[X_v = T] \approx \frac{1}{2}, \Pr[X_v = F] \approx \frac{1}{2}$

Return X;

at any time, any mark variable takes an almost uniform value.

each clause contains $\geq \alpha k$ marked variables;

$\Rightarrow \Pr[\text{each clause is removed}] \gtrsim 1 - \left(\frac{1}{2}\right)^{\alpha k}$



Key Property: w.h.p., the graph is deconstructed into ***small components*** of size $O\left(dk \log \frac{n}{\epsilon}\right)$

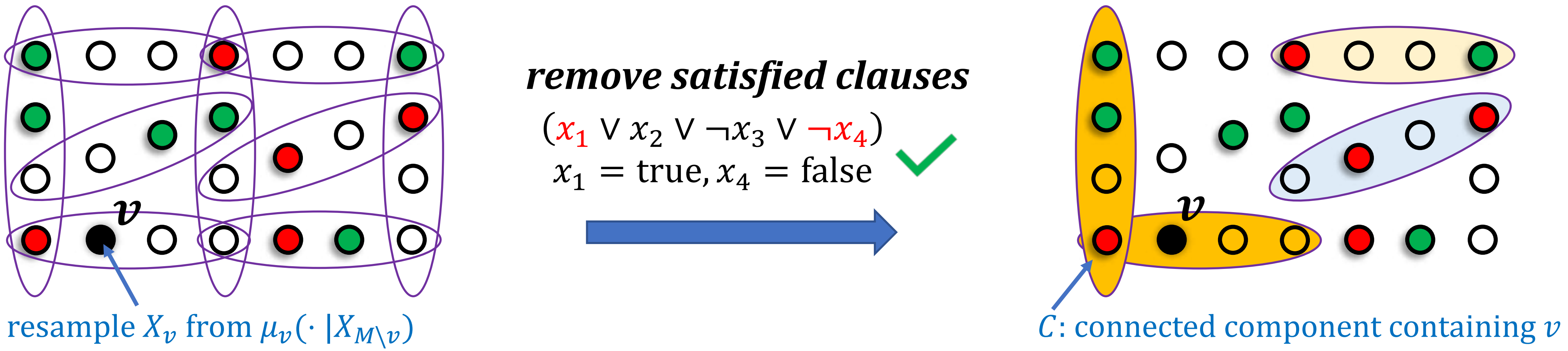
Our solution: try ***rejection sampling*** on v and other unmarked variables in component C

by ***LLL***, if $k \geq 20 \log d + 20 \log k + 3 \log \frac{1}{\zeta}$, then

$$\Pr\left[\text{all clauses in } C \text{ are satisfied} \mid \#C = O\left(dk \log \frac{n}{\epsilon}\right)\right] \geq \left(\frac{\epsilon}{n}\right)^{\zeta};$$

try rejection sampling for $R = \tilde{O}\left((n/\epsilon)^{\zeta}\right)$ times, then we can draw $X_v \sim \mu_v(\cdot \mid X_{M\setminus v})$ w.h.p.

$$\Pr\left[\text{rejection sampling draw } X_v \sim \mu_v(\cdot \mid X_{M\setminus v}), \text{namely} \begin{cases} |C| = O\left(dk \log \frac{n}{\epsilon}\right) \\ \text{one of } R \text{ tires succeeds} \end{cases}\right] \geq 1 - 2\left(\frac{\epsilon}{n}\right)^3$$



Key Property: w.h.p., the graph is deconstructed into ***small components*** of size $O\left(dk \log \frac{n}{\epsilon}\right)$

Our solution: try ***rejection sampling*** on v and other unmarked variables in component C

by ***LLL***, if $k \geq 20 \log d + 20 \log k + 3 \log \frac{1}{\zeta}$, then

$$\Pr\left[\text{all clauses in } C \text{ are satisfied} \mid \#C = O\left(dk \log \frac{n}{\epsilon}\right)\right] \geq \left(\frac{\epsilon}{n}\right)^{\zeta};$$

try rejection sampling for $R = \tilde{O}\left((n/\epsilon)^{\zeta}\right)$ times, then we can draw $X_v \sim \mu_v(\cdot \mid X_{M \setminus v})$ w.h.p.

Lemma: Each *transition step* of the Glauber dynamics and the *last step (i.e. sampling unmarked variables)* can be implemented using rejection sampling

$$\Pr\left[\text{all } T + 1 = O\left(n \log \frac{n}{\epsilon}\right) \text{ rejection samplings succeed}\right] \geq 1 - \frac{\epsilon}{3}.$$

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. $d_{TV}(\sigma, \mu) \le \epsilon$.

1. Run ***Moser-Tardos*** algorithm to construct a set of marked variables $M \subseteq V$;
2. Run ***Glauber dynamics*** on projected distribution μ_M for $O\left(n \log \frac{n}{\epsilon}\right)$ steps to draw approximate sample $X \sim \mu_M$; *(implemented using rejection sampling)*
3. Run ***rejection sampling*** to draw $Y \sim \mu_{V \setminus M}(\cdot \mid X)$;
4. Return $X \cup Y$.

- ***Marking lemma***: $\Pr\left[\text{MT}-\text{alg fails to find } M \text{ in time } O\left(ndk\log\frac{1}{\epsilon}\right)\right] \le \frac{\epsilon}{3}$.
- **Rapid mixing lemma:** The X returned by Glauber dynamics satisfies $d_{TV}(X,\mu_M) \le \frac{\epsilon}{3}$.
- **Rej. Sampling lemma:** $\Pr[\text{one of the } (T+1) \text{ rejection samplings fails}] \le \frac{\epsilon}{3}$

Correctness of the algorithm: $d_{TV}(\text{output},\mu) \le \epsilon$.

- The running time is dominated by simulating Glauber dynamics for $T = O\left(n\log\frac{n}{\epsilon}\right)$ steps;
- Each step is implemented using rejection sampling for $R = \tilde{O}\left(\left(\frac{n}{\epsilon}\right)^{\zeta}\right)$ times.

Efficiency of the algorithm: running time $= \tilde{O}\left(d^2k^3\epsilon^{-\zeta}n^{1+\zeta}\right)$.

Simulated annealing counting [Štefankovič et al. 2009]

Randomized approximate counting

- **Input**: a (k, d) −CNF instance $\Phi = (V, E)$, an error bound $\epsilon > 0$.
- **Output**: a random number $\hat{Z}$, such that

$$\Pr\left[(1-\epsilon)Z \le \hat{Z} \le (1+\epsilon)Z\right] \ge \frac{3}{4}$$

$Z =$ the number of k-SAT solutions.

Weighted CNF a CNF-formula $\Phi = (V, C)$ and *parameter* $\theta > 0$.

- for any $X \in \{T, F\}^V$, define the **weight**

$$w_\theta(X) = \exp(-\theta F(X)),$$

where $F(X)$ is the **number** of clauses ***NOT*** satisfied by X.

- induced **Gibbs distribution**

$$\forall X \in \{T, F\}^V: \quad \mu_\theta(X) = \frac{w_\theta(X)}{Z(\theta)}, \qquad Z(\theta) = \sum_{X \in \{T,F\}^V} w_\theta(X).$$

$$Z(\theta) = \sum_{X\in\{T,F\}^V} w_\theta(X) = \sum_{X\in\{T,F\}^V} \exp(-\theta F(X))$$

Properties:

- $\theta = 0$: $Z(0) = 2^n$ (***easy to compute***);
- $\theta \to \infty$: $\lim_{\theta\to\infty} Z(\theta) = Z = \#k$-SAT solutions. (***target of counting***)

Lemma: counting (proved by LLL[Haeupler, Saha, Srinivasan' 11])

If $k \geq \log d + C$, *it holds that*

$$Z(\theta) \in \left(1 \pm \frac{\epsilon}{2}\right) Z, \qquad where\ \theta = O\left(\log \frac{nd}{\epsilon}\right).$$

- **Non-adaptive cooling schedule**: define $\ell = O\left(nd \log \frac{nd}{\epsilon}\right)$ parameters
$$0 = \theta_0 < \theta_1 < \cdots < \theta_\ell = O\left(\log \frac{nd}{\epsilon}\right),$$
where the adjacent parameters satisfies $\theta_i - \theta_{i-1} = \frac{1}{dn}$.

- **Telescoping product**: approximate $Z = \#k\text{-SAT}$ solutions using
$$Z \approx Z(\theta_\ell) = \frac{Z(\theta_\ell)}{Z(\theta_{\ell-1})} \times \frac{Z(\theta_{\ell-1})}{Z(\theta_{\ell-2})} \times \cdots \times \frac{Z(\theta_1)}{Z(\theta_0)} \times 2^n$$

- **Estimate ratios**: let $X \sim \mu_{\theta_{i-1}}$, define the random variable W_i as
$$W_i = \frac{w_{\theta_i}(X)}{w_{\theta_{i-1}}(X)}, \qquad \text{then } E[W_i] = \frac{Z(\theta_i)}{Z(\theta_{i-1})}.$$
draw samples from $\mu_{\theta_0}, \mu_{\theta_1}, \ldots, \mu_{\theta_{\ell-1}}$ to estimate each ratio.

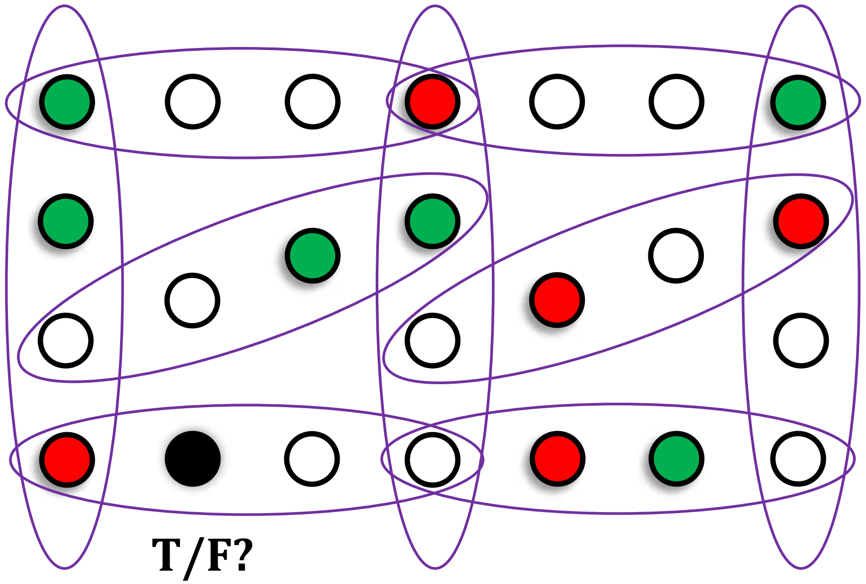
Proof of the rapid mixing

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

For each t from 1 to $T = 2n \log \frac{4n}{\epsilon}$

- Pick a marked variable $v \in M$ u.a.r.;
- Resample $Y_v \sim \mu_v(\cdot \mid Y_{M \setminus v})$;

Return Y;



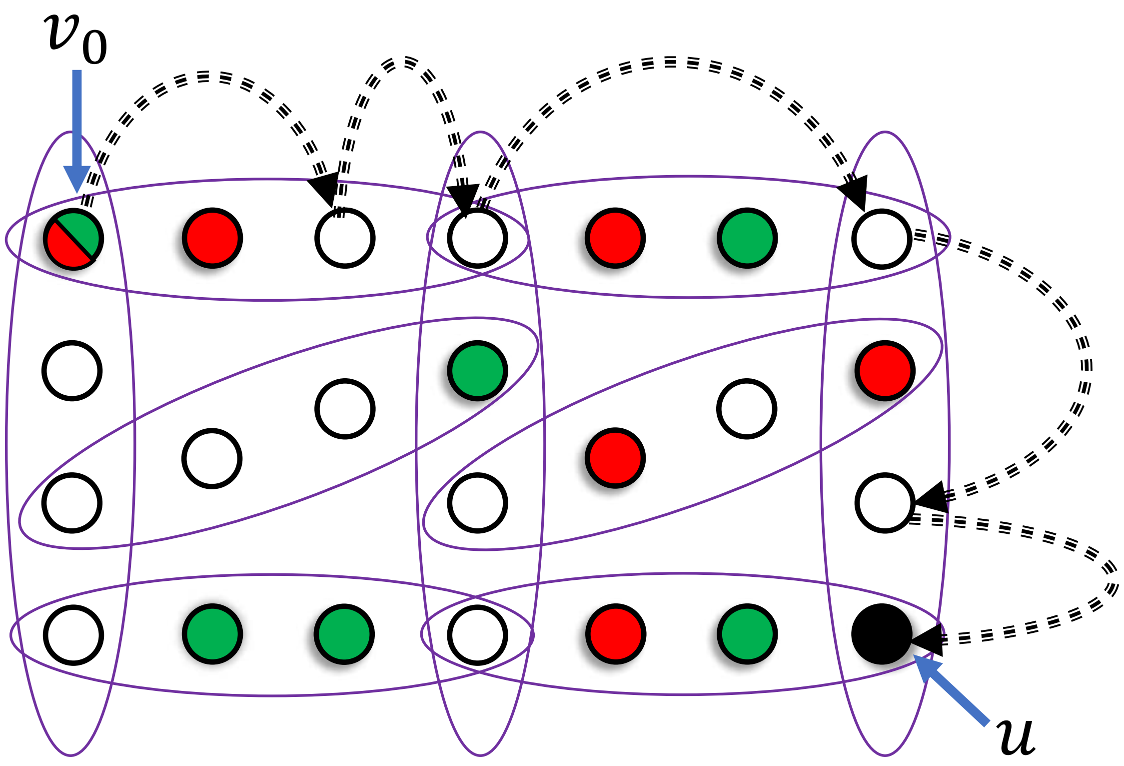
Lemma: mixing

The $Y = Y_T$ returned by Glauber dynamics satisfies

$$d_{TV}(Y, \mu_M) \leq \frac{\epsilon}{3}.$$

Path coupling [Bubley and Dyer' 97]

- Let $X, Y \in \{\text{true}, \text{false}\}^M$ be two assignments ***disagree only at*** v_0.
- For each $u \in M$, we bound the ***influence*** on u from v_0
  $I_u = d_{TV}\left(\mu_u(\cdot \mid X_{M\setminus u}), \mu_u(\cdot \mid Y_{M\setminus u})\right)$.
- Path Coupling: if
  $$\sum_{u \in M \setminus v_0} I_u \leq \frac{1}{2},$$
  then Glauber dynamics is rapid mixing.



Influence may percolate very far away through unmarked variables

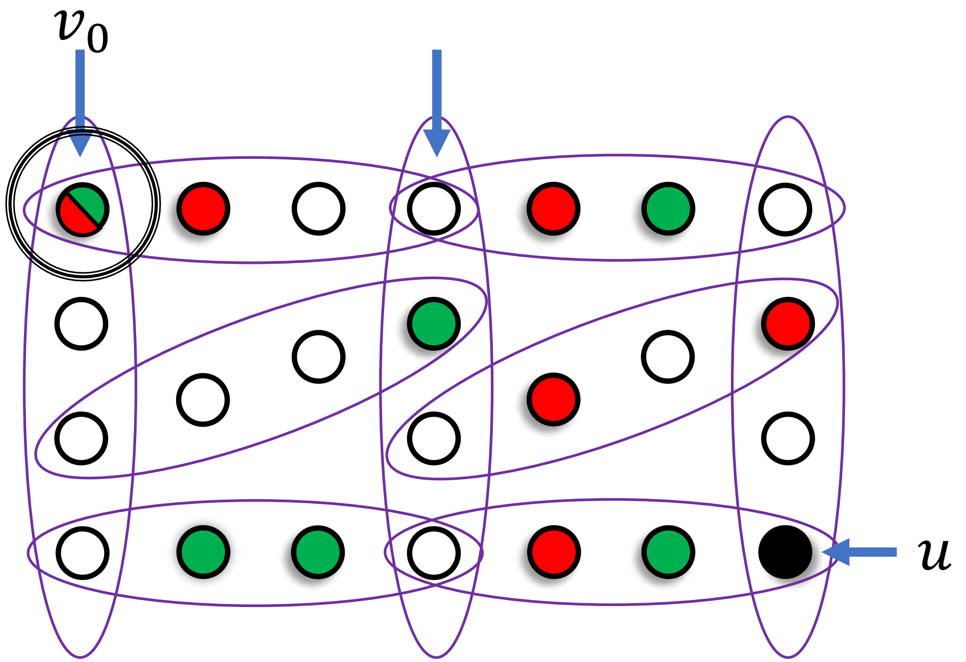
Disagreement percolation coupling [Moitra'17, Guo, et al.'18]

Couple ***unmarked variables*** and ***u*** to generate $X, Y \in \{T, F\}^V$ s.t. $X \sim \mu(\cdot \mid X_{M\setminus u}), Y \sim \mu(\cdot \mid Y_{M\setminus u})$

$$I_u = d_{TV}\left(\mu_u(\cdot \mid X_{M\setminus u}), \mu_u(\cdot \mid Y_{M\setminus u})\right) \le \Pr_{\text{Coupling}}[X_u \ne Y_u].$$

The coupling sketch

- Let D be the ***set of disagreements***, initially, $D = \{v_0\}$.
- Coupling variables in a **BFS** order.
- For each w, couple $X(w)$ and $Y(w)$ ***optimally***.
    - **If** $X(w) = Y(w)$, **then** remove all clauses satisfied by w;
    - **If** $X(w) \ne Y(w)$, **then** add u into D.
- Repeat until D and $\overline{D}$ are ***disconnected***.



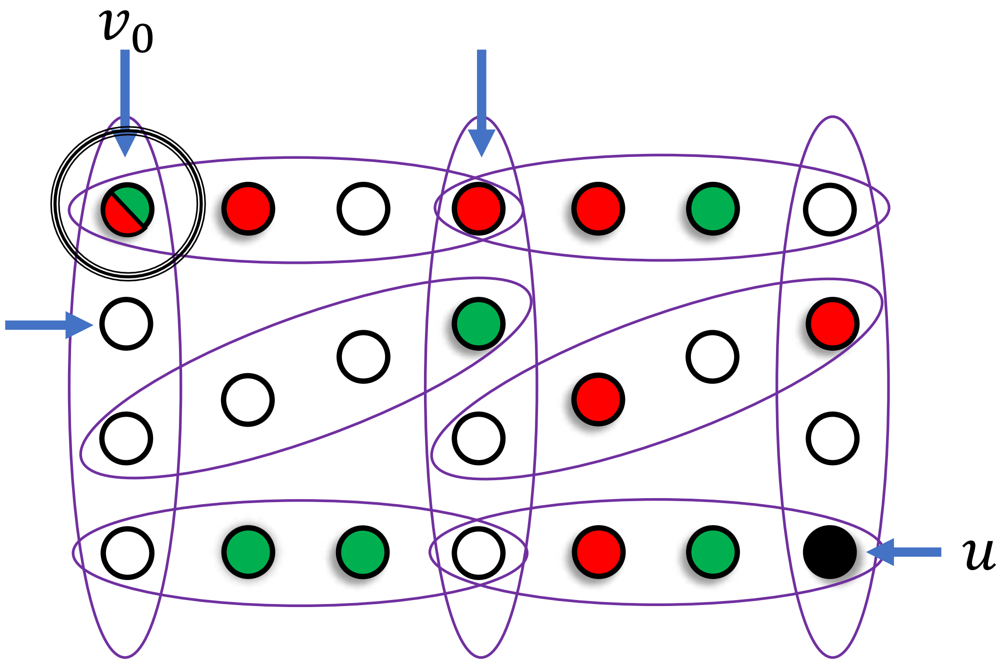
Disagreement percolation coupling [Moitra'17, Guo, et al.'18]

Couple ***unmarked variables*** and ***u*** to generate $X, Y \in \{T, F\}^V$ s.t. $X \sim \mu(\cdot \mid X_{M \setminus u}), Y \sim \mu(\cdot \mid Y_{M \setminus u})$

$$I_u = d_{TV}\left(\mu_u(\cdot \mid X_{M\setminus u}), \mu_u(\cdot \mid Y_{M\setminus u})\right) \le \Pr_{\text{Coupling}}[X_u \ne Y_u].$$

The coupling sketch

- Let D be the ***set of disagreements***, initially, $D = \{v_0\}$.
- Coupling variables in a BFS order.
- For each w, couple $X(w)$ and $Y(w)$ ***optimally***.
  - **If** $X(w) = Y(w)$, **then** remove all clauses satisfied by w;
  - **If** $X(w) \ne Y(w)$, **then** add u into D.
- Repeat until D and $\overline{D}$ are ***disconnected***.



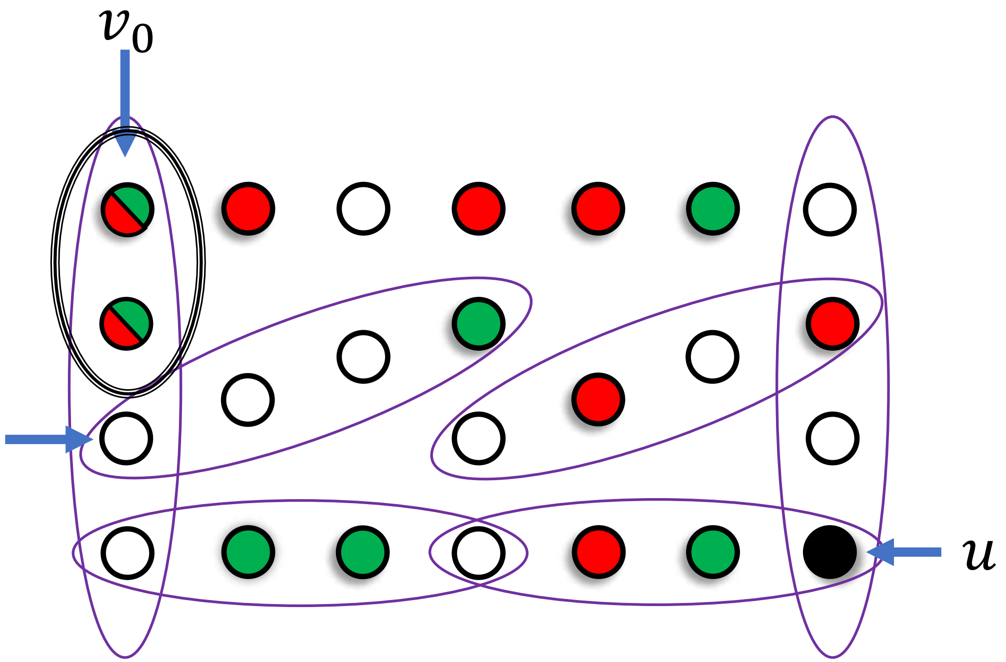
Disagreement percolation coupling [Moitra'17, Guo, et al.'18]

Couple ***unmarked variables*** and ***u*** to generate $X, Y \in \{T, F\}^V$ s.t. $X \sim \mu(\cdot \mid X_{M \setminus u}), Y \sim \mu(\cdot \mid Y_{M \setminus u})$

$$I_u = d_{TV}\left(\mu_u(\cdot \mid X_{M\setminus u}), \mu_u(\cdot \mid Y_{M\setminus u})\right) \le \Pr_{\text{Coupling}}[X_u \ne Y_u].$$

The coupling sketch

- Let D be the ***set of disagreements***, initially, $D = \{v_0\}$.
- Coupling variables in a BFS order.
- For each w, couple $X(w)$ and $Y(w)$ ***optimally***.
  - **If** $X(w) = Y(w)$, **then** remove all clauses satisfied by w;
  - **If** $X(w) \ne Y(w)$, **then** add u into D.
- Repeat until D and $\overline{D}$ are ***disconnected***.



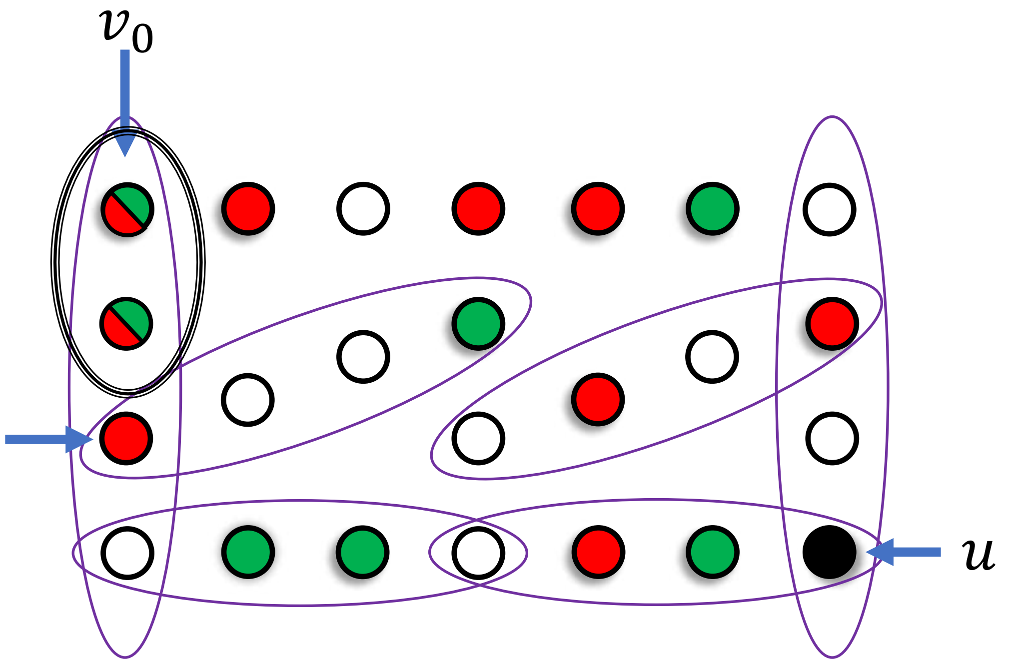
Disagreement percolation coupling [Moitra'17, Guo, et al.'18]

Couple *unmarked variables* and ***u*** to generate $X, Y \in \{T, F\}^V$ s.t. $X \sim \mu(\cdot \mid X_{M\setminus u}), Y \sim \mu(\cdot \mid Y_{M\setminus u})$

$$I_u = d_{TV}\left(\mu_u(\cdot \mid X_{M\setminus u}), \mu_u(\cdot \mid Y_{M\setminus u})\right) \le \Pr_{\text{Coupling}}[X_u \neq Y_u].$$

The coupling sketch

- Let D be the ***set of disagreements***, initially, $D = \{v_0\}$.
- Coupling variables in a BFS order.
- For each w, couple $X(w)$ and $Y(w)$ ***optimally***.
  - **If** $X(w) = Y(w)$, **then** remove all clauses satisfied by w;
  - **If** $X(w) \neq Y(w)$, **then** add u into D.
- Repeat until D and $\overline{D}$ are ***disconnected***.



each clause contains sufficiently many ***free*** variables

adaptive disagreement percolation coupling

LLL →

$$\Pr[X(w) = \text{true}] = \frac{1}{2} \pm \frac{1}{\text{poly}(dk)}$$
$$\Pr[Y(w) = \text{true}] = \frac{1}{2} \pm \frac{1}{\text{poly}(dk)}$$

local uniformity

→

$X(w) = Y(w)$ w.p. $1 - \frac{1}{\text{poly}(dk)}$

coupling succeeds w.h.p.

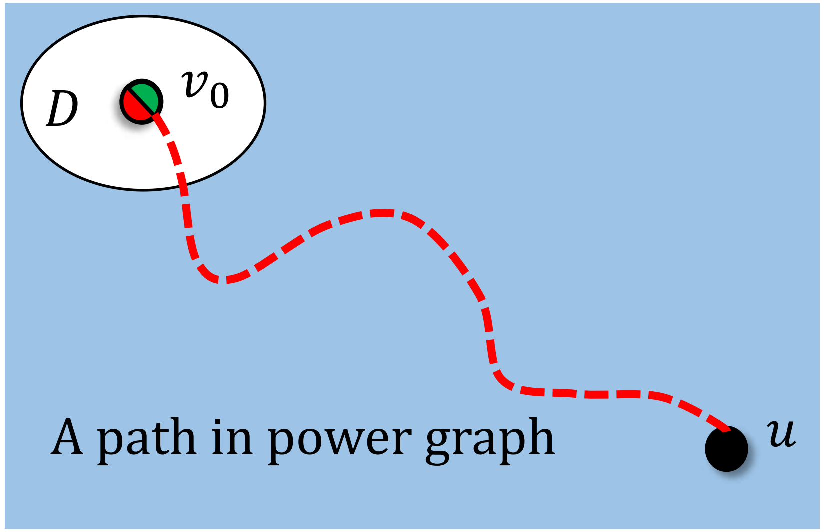
Disagreement percolation coupling [Moitra'17, Guo, et al.'18]

Couple ***unmarked variables*** and ***u*** to generate $X, Y \in \{T, F\}^V$ s.t. $X \sim \mu(\cdot \mid X_{M\setminus u}), Y \sim \mu(\cdot \mid Y_{M\setminus u})$

$$I_u = d_{TV}\left(\mu_u(\cdot \mid X_{M\setminus u}), \mu_u(\cdot \mid Y_{M\setminus u})\right) \le \Pr_{\text{Coupling}}[X_u \ne Y_u].$$

The coupling sketch

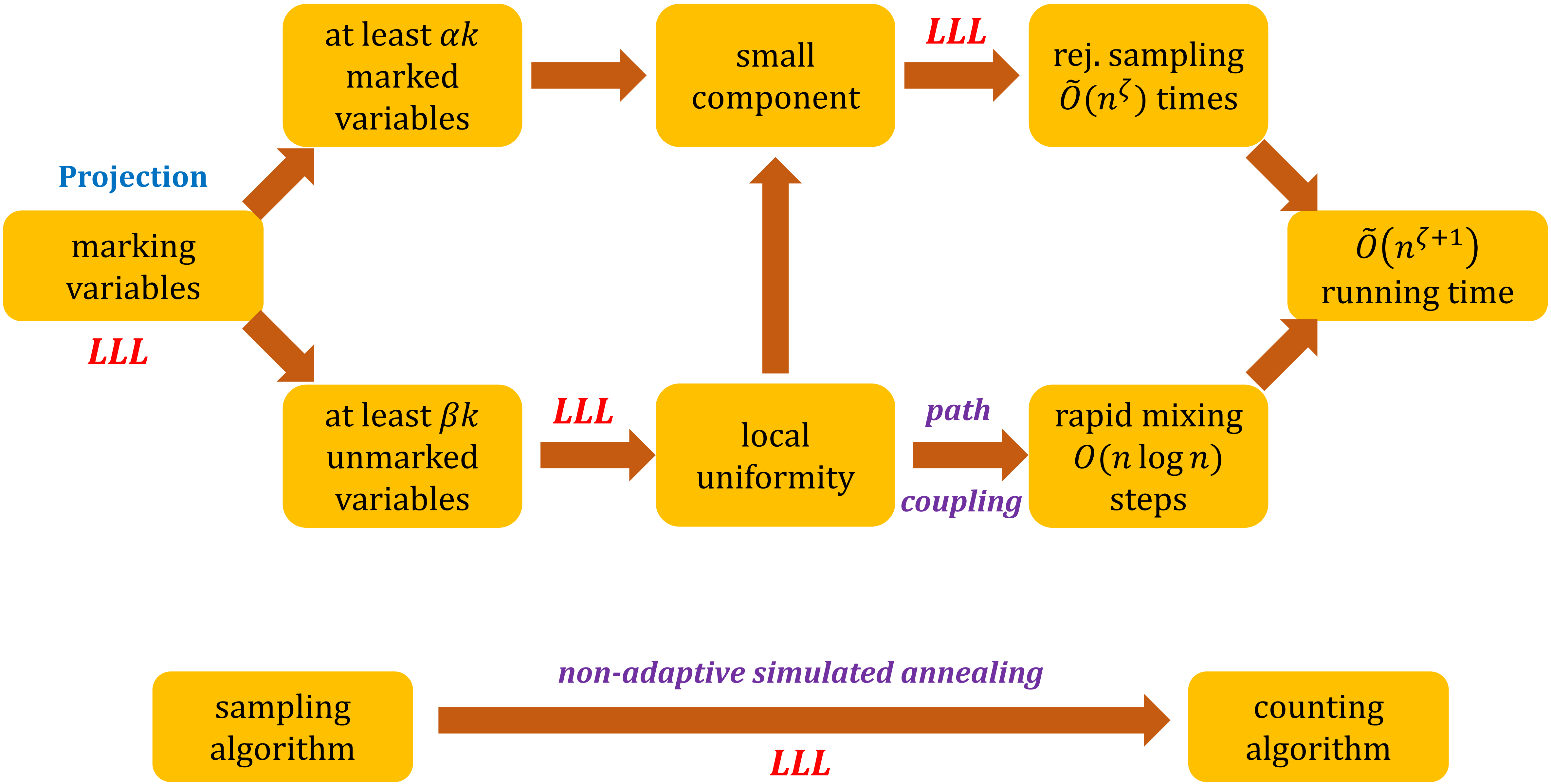
- Let D be the ***set of disagreements***, initially, $D = \{v_0\}$.
- Coupling variables in a BFS order.
- For each w, couple $X(w)$ and $Y(w)$ ***optimally***.
  - **If** $X(w) = Y(w)$, **then** remove all clauses satisfied by w;
  - **If** $X(w) \ne Y(w)$, **then** add u into D.
- Repeat until D and $\overline{D}$ are ***disconnected***.



with high probability, size of the disagreement set D is small

$$I_u \le \Pr_{\text{Couling}}(X_u \ne Y_u) \le \Pr_{\text{Coupling}}[u \notin D] \lesssim \left(\frac{1}{\text{poly}(dk)}\right)^{d_{pow}(v_0,u)}$$

$$\Rightarrow \sum_{u \in M\setminus v_0} I_u \le \frac{1}{2}$$



Projection
marking variables
LLL
at least αk marked variables
at least βk unmarked variables
LLL
small component
local uniformity
LLL
rej. sampling $\tilde{O}(n^{\zeta})$ times
path
coupling
rapid mixing $O(n \log n)$ steps
$\tilde{O}(n^{\zeta+1})$ running time
sampling algorithm
non-adaptive simulated annealing
LLL
counting algorithm

Summary

- A close to linear time algorithm for sampling k-SAT solutions in LLL regime.
- A close to quadratic time algorithm for counting k-SAT solutions in LLL regime.
- Projection + LLL technique to bypass the connectivity barrier of MCMC method.

Open problems

- Sampling & counting k-SAT solutions when $k \gtrsim 2 \log d$.
- Extend the technique to more general distributions, e.g. hyper-graph coloring.

Thank you!