Deterministic counting Lovász local lemma beyond linear programming

Kun He* Chunyang Wang† Yitong Yin†

Abstract

We give a simple combinatorial algorithm to deterministically approximately count the number of satisfying assignments of general constraint satisfaction problems (CSPs). Suppose that the CSP has domain size $q = O(1)$, each constraint contains at most $k = O(1)$ variables, shares variables with at most $\Delta = O(1)$ constraints, and is violated with probability at most p by a uniform random assignment. The algorithm returns in polynomial time in an improved local lemma regime:

$$q^2 \cdot k \cdot p \cdot \Delta^5 \leq C_0 \quad \text{for a suitably small absolute constant } C_0.$$

Here the key term Δ^5 improves the previously best known Δ^7 for general CSPs [21] and $\Delta^{5.714}$ for the special case of k-CNF [20, 16].

Our deterministic counting algorithm is a derandomization of the very recent fast sampling algorithm in [17]. It departs substantially from all previous deterministic counting Lovász local lemma algorithms which relied on linear programming, and gives a deterministic approximate counting algorithm that straightforwardly derandomizes a fast sampling algorithm, hence unifying the fast sampling and deterministic approximate counting in the same algorithmic framework.

To obtain the improved regime, in our analysis we develop a refinement of the $\{2, 3\}$-trees that were used in the previous analyses of counting/sampling LLL. Similar techniques can be applied to the previous LP-based algorithms to obtain the same improved regime and may be of independent interests.

1 Introduction

Approximate counting and almost uniform sampling are two intimately related classes of computational problems that have been extensively studied in theoretical computer science. It was well-known that randomized approximate counting can be achieved by almost uniform sampling through the generic approaches of self-reduction [23] or annealing [6, 27].

On the other hand, *deterministic* approximate counting algorithms use different approaches such as decay of correlation [28], zero-freeness [3, 25], and cluster-expansion [18, 22], or in the case of counting constraint satisfaction solutions, the linear programming [24, 14, 21]. All these deterministic approximate counting methods have running times where the exponent over the input size depends on additional parameters such as degree of the underlying graph. And more fundamentally, all these deterministic counting algorithm work in quite different algorithmic frameworks that deviate far from those of the fast sampling algorithms where the exponents of the running times are universal constants. There is one exception very recently [19], where for matchings/independent sets with a given size, a *unified* algorithm based on a new technique called local central limit theorems was found to simultaneously resolve deterministic counting and fast randomized sampling within the same algorithmic framework.

We are focused on the problem of counting general constraint satisfaction solutions. Our goal is to give a unified approach for deterministic counting Lovász Local Lemma (LLL) [24, 14, 21] and fast sampling LLL [13, 9, 10, 20, 16, 17].

CSPs and Lovász Local Lemma. An instance of constraint satisfaction problem (CSP), called a *CSP formula*, denoted by $\Phi = (V, \mathcal{Q}, \mathcal{C})$, is defined as follows: V is a set of $n = |V|$ variables; $\mathcal{Q} \triangleq \bigotimes_{v \in V} Q_v$ is

*Institute of Computing Technology, Chinese Academy of Sciences, No.6 Kexueyuan South Road Zhongguancun, Haidian District, Beijing, China. E-mail: hekun@ict.ac.cn

†State Key Laboratory for Novel Software Technology, Nanjing University, 163 Xianlin Avenue, Nanjing, Jiangsu Province, China. E-mails: wcysai@smail.nju.edu.cn, yinyt@nju.edu.cn

a product space of all assignments of variables, where each Q_v is a finite domain of size $q_v \triangleq |Q_v| \geq 2$ over where the variable v ranges; and $\mathcal{C}$ is a collection of local constraints where each $c \in \mathcal{C}$ is a constraint function $c : \bigotimes_{v \in \mathsf{vbl}(c)} Q_v \to \{\mathtt{True}, \mathtt{False}\}$ defined on a subset of variables, denoted by $\mathsf{vbl}(c) \subseteq V$. An assignment $\boldsymbol{x} \in \mathcal{Q}$ is called *satisfying* for Φ if

$$\Phi(\boldsymbol{x}) \triangleq \bigwedge_{c \in \mathcal{C}} c\left(\boldsymbol{x}_{\mathsf{vbl}(c)}\right) = \mathtt{True}.$$

Some key parameters of a CSP formula $\Phi = (V, \mathcal{Q}, \mathcal{C})$ are listed in the following:

- *domain size* $q = q_\Phi \triangleq \max_{v \in V} |Q_v|$ and *width* $k = k_\Phi \triangleq \max_{e \in \mathcal{C}} |\mathsf{vbl}(c)|$;
- *constraint degree* $\Delta = \Delta_\Phi \triangleq \max_{c \in \mathcal{C}} |\{c' \in \mathcal{C} \mid \mathsf{vbl}(c) \cap \mathsf{vbl}(c') \neq \emptyset\}|$;
- *violation probability* $p = p_\Phi \triangleq \max_{c \in \mathcal{C}} \mathbb{P}[\neg c]$, where $\mathbb{P}$ denotes the law for the uniform assignment, in which each $v \in V$ draws its evaluation from Q_v uniformly and independently at random.

A characterization for the existence of a satisfying solution to CSP is given by the celebrated *Lovász Local Lemma (LLL)* [7]. By interpreting the space of all possible assignments as a probability space and the violation of each constraint as a bad event, the local lemma provides a sufficient condition

$$\mathrm{e}p\Delta \leq 1. \tag{1.1}$$

for the existence of an assignment to avoid all the bad events, i.e., the existence of a solution to the CSP.

Counting/Sampling LLL. A counting/sampling variant of the Lovász Local Lemma, which seeks algorithms to efficiently (approximate) count and sample (almost-uniform) solutions to CSPs in the local lemma regime, has drawn lots of recent attention [13, 24, 14, 11, 9, 10, 20, 21, 16, 12, 8, 17, 26]. There are two separate lines of work on deterministic counting LLL and fast sampling, using very different approaches.

To this date, all existing deterministic counting algorithms for LLL are based on linear programming. The algorithm was first found in a major breakthrough [24]. The algorithm properly marked the variables using algorithmic LLL and then constructed a polynomial-time deterministic oracle for approximately computing the marginal probabilities of marked variables via linear programs of sizes $n^{\mathrm{poly}(\Delta,k)}$, which can be used to deterministically approximately count the number of satisfying solutions to k-CNF formulas in $n^{\mathrm{poly}(\Delta,k)}$ time when $p\Delta^{60} \lesssim 1$. This LP-based approach was later extended to work for hypergraph colorings [14] and random CNF formulas [11] and finally, for general CSP instances with a substantially improved LLL regime of $p\Delta^7 \lesssim 1$ [21].

Another line of work for the counting/sampling local lemma focuses on *fast* sampling an almost-uniform satisfying solution. In [9], an algorithm was given for approximate sampling uniform solutions to k-CNF formulas with a near-linear running time $\widetilde{O}(n^{1.001})$ when $p\Delta^{20} \lesssim 1$. Their approach was based on a Markov chain on a projected space constructed using the mark/unmark strategy invented in [24]. This projected Markov chain approach was later refined in [10, 20, 16] for fast sampling nearly-atomic CSP solutions, where by atomic we mean each constraint is violated by one forbidden configuration, which achieved the state-of-the-arts regime $p\Delta^{5.714} \lesssim 1$. Very recently in [17], a new approach based on the recursive marginal sampler in [2] was given for sampling general CSP solutions in the local lemma regime within near-linear time. This new sampling algorithm was very different from all Markov chain based sampling algorithms.

1.1 Our results We give a new deterministic algorithm for approximately counting the number of satisfying solutions for general CSPs in an improved local lemma regime. This new deterministic approximate counting algorithm is a combinatorial one, which does not rely on linear programming, and hence is considerably simpler and more intuitive than all previous deterministic algorithms for counting LLL [24, 14, 11, 21] that were LP-based.

This new algorithm is in fact a derandomization of the very recent fast sampling algorithm in [17]. Furthermore, we obtain an improved regime with a much refined analysis, as stated in the following theorem.

THEOREM 1.1. (INFORMAL) *There is an algorithm such that given as input any $\varepsilon \in (0,1)$ and any CSP formula $\Phi = (V, \mathcal{Q}, \mathcal{C})$ with n variables satisfying*

$$q^2 \cdot k \cdot p \cdot \Delta^5 \leq \frac{1}{256\mathrm{e}^3}, \tag{1.2}$$

The algorithm terminates within $\left(\frac{n}{\varepsilon}\right)^{\mathrm{poly}(\log q,\Delta,k)}$ time and deterministically outputs an ε-approximation of Z, the number of satisfying solutions to Φ.

This improves the current state-of-the-arts $p\Delta^7 \lesssim 1$ for general CSP [21], and $p\Delta^{5.714} \lesssim 1$ for nearly-atomic CSP [16] including k-SAT. The $O\left(n^{\mathrm{poly}(\log q,\Delta,k)}\right)$ running time aligns with previous LP-based algorithms. The formal statement of the theorem is in Theorem 4.1.

We also show that our analysis can be used to improve the bound of the algorithm in [21] to the same regime stated as in Theorem 1.1. This is described in Section 5.

1.2 Technique Overview Our method follows the adaptive mark/unmark framework of counting LLL [14, 21]. We briefly describe the previous approach before introducing our modifications.

Given a CSP instance Φ, it has been observed by [15] that the marginal distribution of every variable is close to uniform within a local lemma regime. This is referred to as the "local uniformity" property.

In previous works of counting LLL [24, 14, 11, 21], a key ingredient is a marginal approximator, which approximates the marginal distribution of some variable conditioning on the current partial assignment. This approximator was built on a novel coupling procedure, first proposed by Moitra [24].

In the procedure, two copies of the Gibbs distribution (which in our context is the uniform distribution over all satisfying assignments) conditioning on partial assignments where only one variable is assigned different values are maximally coupled in a sequential and variable-wise fashion. In addition, the variables are picked in a manner such that all the variables assigned in the coupling procedure have the local uniformity property. Initially presented as mark/unmark framework by Moitra [24], the rule for picking variables was later refined to become adaptive [14, 21]. An observation is that after sufficiently many steps of the idealized coupling procedure, there is a good chance that the component containing v in the resulting formula is of logarithmic size, from where one can efficiently calculate the ratio of the number of satisfying assignments extending two partial assignments using exhaustive enumeration. This observation allows one to truncate the procedure up to some certain threshold so that there remains a large probability that the two distributions are successfully coupled.

Then a linear program is set up to mimic the transition probabilities in the (truncated) coupling procedure, so one can use a binary search to approximate the marginal distribution of v. The coupling procedure and the linear program are employed for marginal approximating by all the algorithms of counting LLL [24, 14, 11, 21]. It is worth noting that this linear program is of size $n^{\mathrm{poly}(k,\Delta,\log q)}$ and requires a polynomial-time algorithm for solving linear programs to achieve the desired running time.

In this paper, we propose a new combinatorial approximator for approximating the marginal distribution. Rather than dealing with the coupling of two Gibbs distributions, we decompose a *single* Gibbs distribution directly. Given a variable v with domain Q_v, if v satisfies the local uniformity property, there exists $\theta_v < \frac{1}{q_v}$ close enough to $\frac{1}{q_v}$, such that for each $i \in Q_v$ the probability that v is set as i is no less than θ_v. Thus, there are $q_v + 1$ branches for the possibilities of v: for each $i \in Q_v$, there is a branch of assignment i with probability θ_v, and the last branch is with probability $1 - q_v\theta_v$ and its assignment follows a "overflow" distribution $\mathcal{D}_v$. For the last branch, we repeatedly find a variable u whose successful pinning might help factorize the formula with respect to v, and calculate the marginal by recursively applying the marginal approximator using the chain rule. During the process, a similar rule in the adaptive mark/unmark framework by [21] is taken to guarantee that the local uniformity property persists throughout the algorithm for each chosen variable. A similar observation as in the LP approach that, with some appropriately chosen truncation condition, under a large fraction of the partial assignments generated from the recursive procedure, the component containing v in the resulting formula is of logarithmic size, from where one can efficiently calculate $\mathcal{D}_v$ using exhaustive enumeration.

Our marginal approximator is a derandomization of the marginal sampler of the recent sampling algorithm in [17]. Given a variable v, the marginal sampler samples an assignment of v from its marginal distribution, while our marginal approximator calculates the marginal probability that v is assigned as i for each $i \in Q_v$. Moreover, equipped with the marginal approximator, we use the same method as in [21] to find a "guiding assignment", which can be viewed as a method of conditional expectation for derandomization, to achieve a complete algorithm for estimating the number of satisfying solutions.

To bound the error and running time of our marginal approximator, we design a new combinatorial structure named generalized $\{2,3\}$-tree, which leads to the improved bound $p\Delta^5 \lesssim 1$. In most works on counting/sampling LLL, two types of bad events are considered: one is that the assignment of a marked variable does not fall into the zone of local uniformity; the other is that a constraint is still not satisfied after that a large proportion of its variables are assigned [24, 10, 14, 21, 17]. In previous work, these two bad events are treated similarly and bounded using a combinatorial structure named $\{2,3\}$-tree [1]. A crucial observation is that the densities of these two types of bad events are different, which inspires our design of this new combinatorial structure to take advantage of this property and push the bounds beyond state-of-the-arts. We remark that the generalized $\{2,3\}$-tree can also be applied to improve the bounds in [21], and may be of independent interests.

2 Notation and preliminaries

2.1 Notations for CSP Recall the definition of CSP formula $\Phi = (V, \mathcal{Q}, \mathcal{C})$ in Section 1. We further define the following notations. Let $\Omega = \Omega_\Phi$ be the set of all satisfying assignments of Φ, $Z = Z_\Phi$ be the size of Ω, and $\mu = \mu_\Phi$ be the uniform distribution over Ω. Recall that $\mathbb{P}$ denotes the law for the uniform product distribution over $\mathcal{Q}$. For $C \subseteq \mathcal{C}$, denote $\mathsf{vbl}(C) \triangleq \bigcup_{c\in C} \mathsf{vbl}(c)$; and for $\Lambda \subseteq V$, denote $\mathcal{Q}_\Lambda \triangleq \bigotimes_{v\in\Lambda} Q_v$.

For each $v \in V$, we use an extra symbol $\star \notin Q_v$ to denote that v is *unassigned* with any value. Given a CSP formula $\Phi = (V, \mathcal{Q}, \mathcal{C})$ and a partial assignment $\sigma \in \bigotimes_{v\in V}(Q_v \cup \{\star\})$, let $\Lambda(\sigma)$ denote $\{v \in V : \sigma(v) \in Q_v\}$. The simplification of Φ under σ is a new CSP formula $\Phi^\sigma = (V^\sigma, \mathcal{Q}^\sigma, \mathcal{C}^\sigma)$, where $V^\sigma = V \setminus \Lambda(\sigma)$, $\mathcal{Q}^\sigma = \mathcal{Q}_{V\setminus\Lambda(\sigma)}$, and the $\mathcal{C}^\sigma$ is obtained from $\mathcal{C}$ by:

1. removing all the constraints that have already been satisfied by σ;
2. for the remaining constraints, replacing the variables $v \in \Lambda(\sigma)$ with their values $\sigma(v)$.

It is easy to see that the μ_{Φ^σ} is the same as the marginal distribution induced by μ on $V \setminus \Lambda(\sigma)$, conditional on the assignment over $\Lambda(\sigma)$ is σ.

A CSP formula $\Phi = (V, \mathcal{Q}, \mathcal{C})$ can be naturally represented as a (multi-)hypergraph H_Φ, where each variable $v \in V$ corresponds to a vertex in H_Φ and each constraint $c \in \mathcal{C}$ corresponds to a hyperedge in H_Φ which joins the vertices corresponding to $\mathsf{vbl}(c)$. We slightly abuse the notation and write $H_\Phi = (V, \mathcal{C})$.

Let $H_i = (V_i, \mathcal{C}_i)$ for $1 \le i \le K$ denote all $K \ge 1$ connected components in H_Φ, and $\Phi_i = (V_i, \mathcal{Q}_{V_i}, \mathcal{C}_i)$ their formulas. Obviously $\Phi = \Phi_1 \wedge \Phi_2 \wedge \cdots \wedge \Phi_K$ with disjoint Φ_i, and Z_Φ is the product of Z_{Φ_i}.

2.2 Lovász Local Lemma In the context of CSP, the celebrated asymmetric Lovász Local Lemma is as follows.

THEOREM 2.1. (ERDÖS AND LOVÁSZ [7]) *Given a CSP formula $\Phi = (V, \mathcal{Q}, \mathcal{C})$, if the following holds*

$$\exists x \in (0,1)^{\mathcal{C}} \quad s.t. \quad \forall c \in \mathcal{C}: \quad \mathbb{P}[\neg c] \le x(c) \prod_{\substack{c'\in\mathcal{C}\\ \mathsf{vbl}(c)\cap\mathsf{vbl}(c')\neq\emptyset}} (1 - x(c')), \tag{2.3}$$

then

$$\mathbb{P}\left[\bigwedge_{c\in\mathcal{C}} c\right] \ge \prod_{c\in\mathcal{C}} (1 - x(c)) > 0,$$

The following result shows that when the condition (2.3) is satisfied, the probability of any event in the uniform distribution μ over all satisfying assignments can be well approximated by the probability of the event in the product distribution. This was observed in [15]:

THEOREM 2.2. (HAEUPLER, SAHA, AND SRINIVASAN [15]) *Given a CSP formula $\Phi = (V, \mathcal{Q}, \mathcal{C})$, if (2.3) holds, then for any event A that is determined by the assignment on a subset of variables $\mathsf{vbl}(A) \subseteq V$,*

$$\Pr_\mu[A] = \mathbb{P}\left[A \mid \bigwedge_{c\in\mathcal{C}} c\right] \le \mathbb{P}[A] \prod_{\substack{c\in\mathcal{C}\\ \mathsf{vbl}(c)\cap\mathsf{vbl}(A)\neq\emptyset}} (1 - x(c))^{-1},$$

where μ denotes the uniform distribution over all satisfying assignments of Φ and $\mathbb{P}$ denotes the law of the uniform product distribution over $\mathcal{Q}$.

By setting $x(c) = \mathrm{e}p$ for every $c \in \mathcal{C}$ in Theorem 2.2, it is straightforward to prove the following "local uniformity" property, where the lower bound is calculated by $\mu_v(x) = 1 - \sum_{y\in Q_v\setminus\{x\}} \mu_v(y)$ for each $x \in Q_v$.

COROLLARY 2.1. (LOCAL UNIFORMITY) *Given a CSP formula* $\Phi = (V, \mathcal{Q}, \mathcal{C})$, *if* $\mathrm{ep}\Delta < 1$, *then for any variable* $v \in V$ *and any value* $x \in Q_v$, *it holds that*

$$\frac{1}{q_v} - \left((1-\mathrm{ep})^{-\Delta} - 1\right) \le \mu_v(x) \le \frac{1}{q_v} + \left((1-\mathrm{ep})^{-\Delta} - 1\right).$$

3 The counting algorithm

We now present our algorithm for deterministically approximately counting CSP solutions.

3.1 The main counting algorithm The main counting algorithm takes as input a CSP formula $\Phi = (V, \mathcal{Q}, \mathcal{C})$ with domain size $q = q_\Phi$, width $k = k_\Phi$, constraint degree $\Delta = \Delta_\Phi$, and violation probability $p = p_\Phi$, where the meaning of these parameters are as defined in Section 1.

We assume that the $n = |V|$ variables are enumerated as $V = \{v_1, v_2, \ldots, v_n\}$ in an arbitrary order. For the CSP formula $\Phi = (V, \mathcal{Q}, \mathcal{C})$ presented to the algorithm, we assume that given any constraint $c \in \mathcal{C}$ (or any variable $v \in V$), the $\mathsf{vbl}(c)$ (or $\{c \in \mathcal{C} \mid v \in \mathsf{vbl}(c)\}$) can be retrieved in $\mathrm{poly}(k, \Delta)$ time, and furthermore, given any assignment $\sigma \in \mathcal{Q}_{\mathsf{vbl}(c)}$, it can be determined in $\mathrm{poly}(q, k)$ time if c is already satisfied by σ. It is also safe to assume $\Delta \ge 2$ as otherwise the problem would be trivial.

The main counting algorithm incorporates the idea of "guiding assignment" proposed in [21]. We will construct a sequence of partial assignments $P_0, P_1, \ldots, P_s \in \bigotimes_{v \in V}(Q_v \cup \{\star\})$, where P_0 is the empty assignment, and for each $i \in [s]$, P_i extends P_{i-1} by assigning value to some unassigned variable v_i^*. For any partial assignment X we use $\mathcal{S}_X$ to denote the set of satisfying assignments that agree with X on the assigned variables. Then we will use the following telescopic product to estimate Z_Φ:

$$Z_\Phi = \frac{|\mathcal{S}_{P_0}|}{|\mathcal{S}_{P_1}|} \cdot \frac{|\mathcal{S}_{P_1}|}{|\mathcal{S}_{P_2}|} \cdots \frac{|\mathcal{S}_{P_{s-1}}|}{|\mathcal{S}_{P_s}|} \cdot |\mathcal{S}_{P_s}| = |\mathcal{S}_{P_s}| \cdot \prod_{i \in [s]} \frac{|\mathcal{S}_{P_{i-1}}|}{|\mathcal{S}_{P_i}|} = |\mathcal{S}_{P_s}| \cdot \prod_{i \in [s]} \left(\mu_{v_i^*}^{P_{i-1}}\right)^{-1}. \tag{3.4}$$

We will then calculate the number $|\mathcal{S}_{P_s}|$ and approximate the marginal probability $\mu_{v_i^*}^{P_{i-1}}$ conditional on P_{i-1} for each i respectively. We will calculate the former using a subroutine that exhaustively enumerates all possible satisfying assignment and approximate the latter using a "marginal approximator" subroutine.

Intuitively, we need to carefully construct such "guiding assignment" to meet the following two requirements:

- For each $i \in [s]$, the marginal probability $\mu_{v_i^*}^{P_{i-1}}$ is efficiently approximable with enough accuracy.
- The number of satisfying assignments $|\mathcal{S}_{P_s}|$ is efficiently enumerable.

The precise construction of such guiding assignment is a bit technical and involved. We then present the main framework of the algorithm, leaving some details to be specified later. One of the key steps is to "freeze" the constraints with high violation probability to ensure no constraint becomes too easy to violate, so that a "local uniformity" property is maintained throughout. The same idea has been used in [21, 17] and dated back to [4].

A key threshold α for the violation probability is chosen, for now to satisfy:

$$p < \alpha < (\mathrm{e}q\Delta)^{-1} \tag{3.5}$$

We will fix the specific choice of α later.

Given $\sigma \in \bigotimes_{v \in V}(Q_v \cup \{\star\})$ and $c \in \mathcal{C}$, we say c is σ-frozen if $\mathbb{P}[\neg c \mid \sigma] > \alpha$. Let $\digamma$ be some potential function defined over partial assignments which will be specified later. The main counting algorithm then follows the procedure below, where the guiding assignment X is constructed on the fly:

<u>Main counting algorithm</u>

1. Initialize X as the empty assignment and Z as 1.
2. For $i = 1, \ldots, n$, if v_i is not involved in any X-frozen constraint, do the followings:
   (a) estimate the marginal distribution $\mu_{v_i}^X$ using Algorithm 1 and let $\hat{\mu}_{v_i}$ be the estimator;
   (b) $X(v_i) \leftarrow \arg\min_{a \in Q_v} \digamma(X_{v_i \leftarrow a})$ and $Z \leftarrow Z / \hat{\mu}_{v_i}(X(v_i))$.
3. Use exhaustive enumeration for each connected component in H_{Φ^X} to compute $|\mathcal{S}_X|$, the number of ways to extend X to a full satisfying assignment, and return $Z \cdot |\mathcal{S}_X|$.

REMARK 3.1. (UPPER BOUND FUNCTION $F(\cdot)$) *The upper bound function $F(\cdot)$ plays a key role in the main counting algorithm. It is chosen to be in conformity with our analysis as in Definition 3.7, and thus not explicitly defined here. In Definition 3.7, $F(\cdot)$ is well designed such that both the upper bounds on the error and the time cost of the main counting algorithm can be derived from it. Concretely, we have that*

- *for each partial assignment σ, $F(\sigma)$ is always (for any $v \in V$) an upper bound for the total variation distance between the output of Algorithm 1 and the marginal distribution μ_v^σ;*
- *if $F(X)$ is small, then we can obtain a good upper bound on the running time of the exhaustive enumeration part for calculating $|\mathcal{S}_X|$.*

3.2 A marginal approximator The main tool of the main counting algorithm is a subroutine which returns a probability vector approximating the (conditional) marginal distribution μ_v^σ of a variable v. Before presenting our subroutine, we need to formally define the notion of partial assignments, which is the same as the one defined in [17].

DEFINITION 3.1. (PARTIAL ASSIGNMENT) *Given a CSP formula $\Phi = (V, \mathcal{Q}, \mathcal{C})$, let $\star$ and $\star$ be two special symbols not in any Q_v. Define:*

$$\mathcal{Q}^* \triangleq \bigotimes_{v \in V} (Q_v \cup \{\star, \star\}).$$

Each $\sigma \in \mathcal{Q}^*$ is called a partial assignment.

Given a partial assignment $\sigma \in \mathcal{Q}^*$, each variable $v \in V$ has three possibilities:

- $\sigma(v) \in Q_v$. That is, v is *accessed* by the algorithm and *assigned* with the value $\sigma(v) \in Q_v$;
- $\sigma(v) = \star$. That is, v is just *accessed* by the algorithm but *unassigned* yet with a value in Q_v;
- $\sigma(v) = \star$. That is, v is *unaccessed* by the algorithm and hence *unassigned* with any value.

Recall the notation $\Lambda(\sigma) \triangleq \{v \in V \mid \sigma(v) \in Q_v\}$. Given a partial assignment $\sigma \in \mathcal{Q}^*$, we further define $\Lambda^+(\sigma) \triangleq \{v \in V \mid \sigma(v) \neq \star\}$ to be the sets of accessed variables σ. For any variable $v \in V$, let $\sigma_{v \leftarrow x}$ be the partial assignment modified from σ by replacing $\sigma(v)$ with $x \in Q_v \cup \{\star, \star\}$.

Given any two partial assignments $\sigma, \tau \in \mathcal{Q}^*$, if $\Lambda(\sigma) \subseteq \Lambda(\tau)$, $\Lambda^+(\sigma) \subseteq \Lambda^+(\tau)$, and σ, τ agree with each other over all variables in $\Lambda(\sigma)$, τ is said to *extend* σ. A partial assignment σ is said to satisfy a constraint $c \in \mathcal{C}$, if c is satisfied by all full assignments extending σ. And σ is said to be *feasible*, if there is a satisfying assignment extending σ.

For each variable $v \in V$, we always assume an arbitrary order over all values in Q_v in the paper. Let $q_v \triangleq |Q_v|$. The following parameters are used in our subroutine:

$$\theta_v \triangleq \frac{1}{q_v} - \eta \quad \text{and} \quad \theta \triangleq \frac{1}{q} - \eta \quad \text{where} \quad \eta = (1 - \mathrm{e}\alpha q)^{-\Delta} - 1 \tag{3.6}$$

Assuming the LLL condition in (1.2), we always have $\eta < \frac{1}{q_v}$, and hence $\theta_v, \theta > 0$.

Next, we define some distributions used in the algorithm. For any feasible $\sigma \in \mathcal{Q}^*$ and any $S \subseteq V$, we denote by μ_S^σ the marginal distribution induced by μ on S conditional on σ. Formally, for each $\tau \in \mathcal{Q}_S$, $\mu_S^\sigma(\tau) = \mathbf{Pr}_{X \sim \mu}[X(S) = \tau \mid \forall v \in \Lambda(\sigma), X(v) = \sigma(v)]$. We write $\mu_v^\sigma = \mu_{\{v\}}^\sigma$ for $v \in V$. Similarly, for any $\sigma \in \mathcal{Q}^*$ and any event $A \subseteq \mathcal{Q}$, denote that $\mathbb{P}[A \mid \sigma] = \mathbb{P}_{X \in \mathcal{Q}}[X \in A \mid \forall v \in \Lambda(\sigma), X(v) = \sigma(v)]$, recalling that $\mathbb{P}$ is the law for the uniform product distribution over $\mathcal{Q}$.

For any $\sigma \in \mathcal{Q}^*$ and $v \in V$, define:

$$\forall x \in Q_v, \qquad \mathcal{D}_v^\sigma(x) \triangleq \frac{\mu_v^\sigma(x) - \theta_v}{1 - q_v \cdot \theta_v}. \tag{3.7}$$

In our subroutine for approximately calculating μ_v^σ, it is guaranteed that θ_v always lower bounds the marginal probability (Proposition 4.1). Therefore, $\mathcal{D}_v^\sigma$ is a well-defined probability distribution over Q_v.

By (3.7) we have that $\mu_v^\sigma = \theta_v + (1 - q_v\theta_v)\mathcal{D}_v^\sigma$. The MarginalApproximator then simply uses this equation to approximate the distribution μ_v^σ, assuming another subroutine RecursiveApproximator$(\Phi, \sigma_{v \leftarrow \star}, v)$ for approximately calculating $\mathcal{D}_v^\sigma$. This is formally described in Algorithm 1.

Algorithm 1: MarginalApproximator(Φ, σ, v)

Input: a CSP formula $\Phi = (V, \mathcal{C})$, a partial assignment $\sigma \in \mathcal{Q}^*$ and a variable v
Output: a distribution approximating $\mu_v^\sigma(\cdot)$

```
D̂ ← RecursiveApproximator(Φ, σ_{v←⋆}, v);
μ̂(i) ← θ_v + (1 − q_v θ_v) · D̂(i) for each 1 ≤ i ≤ q_v;
return μ̂;
```

3.3 A recursive approximator The goal of the RecursiveApproximator subroutine is to approximate the distribution $\mathcal{D}_v^\sigma$ which is computed from the marginal distribution μ_v^σ as defined in (3.7). This subroutine is a derandomization of the recursive marginal sampler in [17].

Note that we can compute the exact distribution $\mathcal{D}_v^\sigma$ by exhaustively enumerating all assignments and checking if the assignment satisfies the formula. Still, such exhaustive enumeration can be inefficient as we must enumerate the assignment over too many variables.

Nevertheless, such exhaustive enumeration subroutine for computing $\mathcal{D}_v^\sigma$ may serve as the basis of the recursion. If sufficiently many variables are assigned during the recursion, the remaining CSP formula will be "factorized" with respect to v. In most cases, the connected component containing v in H_{Φ^σ} is small, in which case the exhaustive enumeration subroutine for computing $\mathcal{D}_v^\sigma$ becomes efficient. Therefore, our approximator will try to assign all possible values to some variable that can help "factorize" the formula and approximate the distribution $\mathcal{D}_v^\sigma$ recursively. However, this may still be inefficient as the number of recursive calls may grow at an exponential rate in the recursion depth. To resolve this issue, we will truncate the recursion when some suitable condition is reached. Later we will show that with some properly formulated condition for truncation, our approximator is both efficient and accurate enough.

Before presenting the subroutine, we formally define notions of frozen constraints and fixed variables. Note that this definition also apply in Line 2 of the main counting algorithm.

Definition 3.2. (frozen and fixed) *Let $\sigma \in \mathcal{Q}^*$ be a partial assignment.*

- *A constraint $c \in \mathcal{C}$ is called σ-*frozen *if $\mathbb{P}[\neg c \mid \sigma] > \alpha$. Let $\mathcal{C}^\sigma_{\mathsf{frozen}} \triangleq \{c \in \mathcal{C} \mid \mathbb{P}[\neg c \mid \sigma] > \alpha\}$ be the set of all σ-frozen constraints.*
- *A variable $v \in V$ is called σ-*fixed *if v is accessed in σ or is involved in some σ-frozen constraint. Let $V^\sigma_{\mathsf{fix}} \triangleq \Lambda^+(\sigma) \cup \bigcup_{c \in \mathcal{C}^\sigma_{\mathsf{frozen}}} \mathsf{vbl}(c)$ be the set of all σ-fixed variables.*

Given a partial assignment σ, the following definition specifies the next variable to assign according to σ, which has already appeared in [17].

Definition 3.3. ($\star$-influenced variables) *Given a partial assignment $\sigma \in \mathcal{Q}^*$, let $H^\sigma = H_{\Phi^\sigma} = (V^\sigma, \mathcal{C}^\sigma)$ be the hypergraph of the simplified formula Φ^σ and H^σ_{fix} be the sub-hypergraph of H^σ induced by $V^\sigma \cap V^\sigma_{\mathsf{fix}}$.*

- *Let $V^\sigma_{\star\text{-con}} \subseteq V^\sigma \cap V^\sigma_{\mathsf{fix}}$ be the set of vertices belong to the connected components in H^σ_{fix} that contain any v with $\sigma(v) = \star$.*
- *Let $V^\sigma_{\star\text{-inf}} \triangleq \{u \in V^\sigma \setminus V^\sigma_{\star\text{-con}} \mid \exists c \in \mathcal{C}^\sigma, v \in V^\sigma_{\star\text{-con}} : u, v \in \mathsf{vbl}(c)\}$ be the vertex boundary of $V^\sigma_{\star\text{-con}}$ in H^σ.*
- *Let $\mathsf{NextVar}(\sigma)$ be the next variable to assign under σ where*

$$\mathsf{NextVar}(\sigma) \triangleq \begin{cases} v_i \in V^\sigma_{\star\text{-inf}} \text{ with smallest } i & \text{if } V^\sigma_{\star\text{-inf}} \neq \emptyset, \\ \perp & \text{otherwise.} \end{cases} \tag{3.8}$$

Intuitively, given a partial assignment $\sigma \in \mathcal{Q}^*$, a variable u is a good candidate for assignment if it has enough "freedom" under σ (u is not σ-fixed) and can "influence" the variables that we are trying to approximate the marginal in the recursion (which are marked by $\star$) through a chain of constraints in the simplified formula Φ^σ. The first such variable is returned by $\mathsf{NextVar}(\sigma)$.

The RecursiveApproximator subroutine is given in Algorithm 2.

REMARK 3.2. (TRUNCATION CONDITION $f(\cdot)$) *Note that we haven't explicitly define the function $f(\cdot)$ in Line 1 of Algorithm 2. This is for the same reason we didn't explicitly define $F(\cdot)$ in the main counting algorithm. The function $f : \mathcal{Q}^* \to \{\texttt{True}, \texttt{False}\}$ is some kind of condition for "truncation" that decides when we should stop the recursion. An implementation of $f(\cdot)$ will be specified later in Definition 3.5, to be in conformity with the analysis.*

Algorithm 2: RecursiveApproximator(Φ, σ, v)

Input: a CSP formula $\Phi = (V, \mathcal{C})$, a feasible partial assignment $\sigma \in \mathcal{Q}^*$ and a variable v
Output: a distribution over Q_v that approximates the distribution $\mathcal{D} = \mathcal{D}_v^\sigma$ defined in (3.7)

```
1  if f(σ) = True then            // the condition for truncation is satisfied
2      return (1/q_v, 1/q_v, ··· , 1/q_v);
3  else
4      u ← NextVar(σ);
5      if u ≠ ⊥ then
6          D̂ ← (0, 0, ··· , 0);
7          D̂_u^σ ← RecursiveApproximator(Φ, σ_{u←⋆}, u);
8          μ̂_u^σ(i) ← θ_u + (1 − q_u θ_u) D̂_u^σ(i) for each 1 ≤ i ≤ q_u;
9          for 1 ≤ i ≤ q_u do                // approximate D_v^σ(·) by reduction
10             D̂_v^{σ_{u←i}} ← RecursiveApproximator(Φ, σ_{u←i}, v);
11             for 1 ≤ j ≤ q_v do
12                 D̂(j) ← D̂(j) + μ̂_u^σ(i) · D̂_v^{σ_{u←i}}(j);
13         return D̂;
14     else                              // the Factorization succeeds.
15         Calculate μ_v^σ by counting the number of satisfying assignments exhaustively for the connected
           component in H_{Φ^σ} containing v;
16         Calculate D_v^σ with μ_v^σ according to (3.7);
17         return D_v^σ;
```

3.4 The choice of the truncation condition and the upper bound function It remains to explicitly specify the upper bound function $F(\cdot)$ and the truncation condition $f(\cdot)$, stated respectively in Remark 3.1 and Remark 3.2, to complete the definition of our algorithm. For this purpose, we bring forward some definitions used in the analysis. In particular, we will introduce the notion of generalized $\{2, 3\}$-tree, which is crucial to our choice of the upper bound function and is also a main technical contribution.

3.4.1 The choice of the truncation condition To specify our choice of the truncation condition, we need to classify those "bad constraints" with respect to a partial assignment σ.

DEFINITION 3.4. *Let $\sigma \in \mathcal{Q}^*$ be a partial assignment.*

- *Define $\mathcal{C}^\sigma_{\star\text{-con}}$ to be the set of constraints $c \in \mathcal{C}$ such that $\mathsf{vbl}(c)$ intersects $V^\sigma_{\star\text{-con}}$, where $V^\sigma_{\star\text{-con}}$ is as defined in Definition 3.3.*
- *Define $\mathcal{C}^\sigma_{\star\text{-frozen}} \triangleq \mathcal{C}^\sigma_{\text{frozen}} \cap \mathcal{C}^\sigma_{\star\text{-con}}$.*
- *Define $V^\sigma_\star \triangleq \{v \in V \mid \sigma(v) = \star\}$ to be the set of variables set to $\star$ in σ.*

We are now ready to specify our choice of the truncation condition $f(\cdot)$.

DEFINITION 3.5. (CHOICE OF THE TRUNCATION CONDITION $f(\cdot)$) *The truncation condition $f : \mathcal{Q}^* \to \{\texttt{True}, \texttt{False}\}$ is chosen as*

$$f(\sigma) \triangleq \mathbb{1}\left[|V^\sigma_\star| + \Delta \cdot |\mathcal{C}^\sigma_{\star\text{-frozen}}| \geq L\Delta\right]$$

for some integer parameter $L > 1$ to be specified later.

3.4.2 The choice of the upper bound function To specify our choice of the upper bound function, we will introduce the notion of "generalized $\{2,3\}$-tree", which is a combinatorial structure refined from the $\{2,3\}$-trees used in the analysis of [21] and [17].

Given a hypergraph $H=(V,\mathcal{E})$, let $\mathsf{Lin}(H)$ be the line graph of H whose vertex set is the hyperedges in $\mathcal{E}$ and two hyperedges in $\mathcal{E}$ are adjacent in $\mathsf{Lin}(H)$ if and only if they share some vertex in H. Let $\mathrm{dist}_{\mathsf{Lin}(H)}(\cdot,\cdot)$ be the shortest path distance in $\mathsf{Lin}(H)$.

DEFINITION 3.6. *(generalized $\{2,3\}$-tree) Given a hypergraph $H=(V,\mathcal{E})$, A* generalized $\{2,3\}$-tree $T=U\cup E$*, where $U\subseteq V$ and $E\subseteq\mathcal{E}$, is a subset of vertices and edges of H such that the followings hold:*

1. *For all distinct $u,v\in E$, $dist_{\mathsf{Lin}(H)}(u,v)\geq 2$.*
2. *It holds for the directed graph $G(T,\mathcal{A})$ that there is a vertex $r\in T$ (called a root) which can reach all other vertices through directed paths, where the $G(T,\mathcal{A})$ is constructed on the vertex set T as that, for any $u,v\in T$ there is an arc $(u,v)\in\mathcal{A}$ if and only if at least one of the following conditions is satisfied:*
   - *$u,v\in E$ and $dist_{\mathsf{Lin}(H)}(u,v)=2$ or 3;*
   - *$u\in U,v\in E$ and there exists $e\in\mathcal{E}$ such that $u\in e\wedge dist_{\mathsf{Lin}(H)}(v,e)=1$;*
   - *$u\in E,v\in U$ and there exists $e\in\mathcal{E}$ such that $v\in e\wedge dist_{\mathsf{Lin}(H)}(u,e)=1$ or 2;*
   - *$u,v\in U$ and there exists $e\in\mathcal{E}$ such that $u,v\in e$.*

Furthermore, any rooted directed spanning tree of the directed graph $G(T,\mathcal{A})$ constructed as above is called an auxiliary tree *of the generalized $\{2,3\}$-tree T.*

The generalized $\{2,3\}$-tree in Definition 3.6 is inspired by the the notion of $\{2,3\}$-tree defined for the line graph $\mathsf{Lin}(H)$ [1]. We generalize this notion to the original hypergraph H to simultaneously depict the distances between vertices and hyperedges in H. One can verify that every $\{2,3\}$-tree in the line graph $\mathsf{Lin}(H)$ is some generalized $\{2,3\}$-tree in the hypergraph H. Moreover, a generalized $\{2,3\}$-tree $T=U\cup E$ further restricts that each vertex in U is close to its nearest neighbour in T.

Specifically, when the underlying hypergraph in Definition 3.6 is the hypergraph representation $H_\Phi=(V,\mathcal{C})$ of some CSP Φ, a generalized $\{2,3\}$-tree $T\subseteq V\cup\mathcal{C}$ in H_Φ becomes a subset of variables and constraints.

Given a subset $T\subset V\cup\mathcal{C}$ of variables and constraints, we use $T=U\circ E$ to denote $T=U\cup E$ where $U\subseteq V$ and $E\subseteq\mathcal{C}$. We are now ready to specify our choice of the upper bound function $\mathcal{F}(\cdot)$.

DEFINITION 3.7. (CHOICE OF THE UPPER BOUND FUNCTION $\mathcal{F}(\cdot)$) *The upper bound function $\mathcal{F}:\mathcal{Q}^*\to\mathbb{R}$ is fixed as follows.*

For any subset of vertices and constraints $T=U\circ E$ and any partial assignment $\sigma\in\mathcal{Q}^*$, define

$$F(\sigma,T)\triangleq(1-q\theta)^{|U|}\prod_{c\in E}\left(\alpha^{-1}\mathbb{P}[\neg c\mid\sigma](1+\eta)^k\right). \tag{3.9}$$

For any integer $t>0$, define

$$\mathcal{T}^t\triangleq\{T\mid T=U\circ E \text{ is a generalized } \{2,3\}\text{-tree in } H_\Phi \text{ satisfying} |U|+\Delta\cdot|E|=t\} \tag{3.10}$$

Moreover, for any integer $t>0$ and $v\in V$, define

$$\mathcal{T}_v^t\triangleq\left\{T\in\mathcal{T}^t \mid \text{ there exists an auxiliary tree of } T \text{ rooted at } v\right\} \tag{3.11}$$

Finally, for any partial assignment $\sigma\in\mathcal{Q}^*$, we define

$$\mathcal{F}(\sigma)\triangleq\sum_{i=L}^{L\Delta}\sum_{v\in V}\sum_{T\in\mathcal{T}_v^i}F(\sigma,T\setminus\{v\}), \tag{3.12}$$

where L is the same unspecified parameter as in the definition of the truncation condition.

4 Analysis of the counting algorithm

In this section, we present the analysis of our deterministic approximate counting algorithm. We will prove the following theorem.

THEOREM 4.1. *With the $f(\cdot)$ and $F(\cdot)$ as specified respectively in Definition 3.5 and Definition 3.7, for any CSP formula $\Phi = (V, \mathcal{C})$ satisfying (1.2) and $0 < \varepsilon < 1$, the main counting algorithm (given in Section 3.1) returns a $\widehat{Z}$ satisfying $(1-\varepsilon)Z_\Phi \le \widehat{Z} \le (1+\varepsilon)Z_\Phi$, within time $O\left(\left(\frac{n}{\varepsilon}\right)^{\mathrm{poly}(\log q, \Delta, k)}\right)$.*

4.1 Invariants and local uniformity In this subsection, we present some basic facts that guarantee our algorithm is well-defined. The following two invariants are respectively satisfied by the MarginalApproximator and RecursiveApproximator subroutine called within the counting algorithm (formally proved in Lemma 4.1).

CONDITION 4.1. (INVARIANT FOR MarginalApproximator) *The followings hold for the input tuple (Φ, σ, v):*

- *$\Phi = (V, \mathcal{Q}, \mathcal{C})$ is a CSP formula, $\sigma \in \mathcal{Q}^*$ is a feasible partial assignment, and $v \in V$ is a variable;*
- *v is not σ-fixed and $\sigma(v) = \star$, and for all $u \in V$, $\sigma(u) \in Q_u \cup \{\star\}$;*
- *$\mathbb{P}[\neg c \mid \sigma] \le \alpha q$ for all $c \in \mathcal{C}$.*

CONDITION 4.2. (INVARIANT FOR RecursiveApproximator) *The followings hold for the input tuple (Φ, σ, v):*

- *$\Phi = (V, \mathcal{Q}, \mathcal{C})$ is a CSP formula, $\sigma \in \mathcal{Q}^*$ is a feasible partial assignment, and $v \in V$ is a variable;*
- *$\sigma(v) = \star$;*
- *$\mathbb{P}[\neg c \mid \sigma] \le \alpha q$ for all $c \in \mathcal{C}$.*

LEMMA 4.1. *When the input CSP formula $\Phi = (V, \mathcal{Q}, \mathcal{C})$ satisfies (1.2), the invariants are satisfied during the execution of the algorithm:*

1. *whenever MarginalApproximator(Φ, σ, v) is called, Condition 4.1 is satisfied by its input (Φ, σ, v);*
2. *whenever RecursiveApproximator(Φ, σ, v) is called, Condition 4.2 is satisfied by its input (Φ, σ, v).*

We then prove Lemma 4.1. Before that, we formally define the sequence of partial assignments that evolve in the main counting algorithm.

DEFINITION 4.1. (PARTIAL ASSIGNMENTS IN MAIN COUNTING ALGORITHM) *Let $X^0, X^1, \dots, X^n \in \mathcal{Q}^*$ denote the sequence of partial assignments, where $X^0 = \star^V$ and for every $1 \le i \le n$, X^i is the partial assignment X after the i-th iteration of the for-loop in Line 2 of the main counting algorithm.*

The following two lemmas are immediate by [17, Lemma 5.8] and [17, Lemma 5.9], respectively. We then omit the proof.

LEMMA 4.2. *For the $X^0, X^1, \dots, X^n$ in Definition 4.1, it holds for all $0 \le i \le n$ that X^i is feasible and*

$$\forall c \in \mathcal{C}, \qquad \mathbb{P}[\neg c \mid X^i] \le \alpha q. \tag{4.13}$$

LEMMA 4.3. *Assume Condition 4.2 for (Φ, σ, v). For any $u \in V$, if u is not σ-fixed, then $(\Phi, \sigma_{u \leftarrow a}, v)$ satisfies Condition 4.2 for any $a \in Q_u \cup \{\star\}$.*

The invariant of Condition 4.1 for MarginalApproximator stated in Lemma 4.1-(1) follows directly from Lemma 4.2. The invariant of Condition 4.2 for RecursiveApproximator stated in Lemma 4.1-(2) follows from Lemma 4.3, because during the execution, the algorithm will only change an input partial assignment σ to $\sigma_{u \leftarrow a}$ for $u = \mathsf{NextVar}(\sigma)$ and for $a \in Q_u \cup \{\star\}$, and that by the definition of $\mathsf{NextVar}(\cdot)$ we have u is not σ-fixed. Therefore, Lemma 4.1 is proved.

The next proposition, shows that θ_v always lower bounds the marginal probability $\mu_v^\sigma(\cdot)$ for (Φ, σ, v) satisfying Condition 4.2. Combining with Lemma 4.1-(2), we have shown the well-definedness of the distribution D_v^σ defined in (3.7) and Algorithm 2.

PROPOSITION 4.1. *Assuming Condition 4.2 for the input* (Φ, σ, v), *it holds that* $\min_{x \in Q_v} \mu_v^\sigma(x) \geq \theta_v$ *and for* $u = \mathsf{NextVar}(\sigma)$, *if* $u \neq \perp$ *then it also holds that* $\min_{x \in Q_u} \mu_u^\sigma(x) \geq \theta_u$.

Recall α defined in (3.5) and θ_v, η defined in (3.6). The following corollary implied by the "local uniformity" property is immediate by [17, Proposition 3.9] and directly proves Proposition 4.1.

COROLLARY 4.1. *For any CSP formula* $\Phi = (V, \mathcal{Q}, \mathcal{C})$ *and any partial assignment* $\sigma \in \mathcal{Q}^*$, *if*

$$\forall c \in \mathcal{C}, \quad \mathbb{P}[\neg c \mid \sigma] \leq \alpha q,$$

then σ *is feasible, and for any* $v \in V \setminus \Lambda(\sigma)$, *and any* $x \in Q_v$

$$\mu_v^\sigma(x) \geq \theta_v.$$

4.2 The recursive cost tree A key combinatorial structure used in our proof for Theorem 4.1 is the Recursive Cost Tree (RCT). For each $v \in V$, we further define $\mathcal{Q}_v^\star \triangleq Q_v \cup \{\star\}$ as the extended domain for accessment. Note that a difference between the RCT here and that in [17] is that the RCT here stops growing once the truncation condition is satisfied.

DEFINITION 4.2. (RECURSIVE COST TREE) *For any partial assignment* $\sigma \in \mathcal{Q}^*$, *let* $\mathcal{T}_\sigma = (T_\sigma, \rho_\sigma)$, *where* T_σ *is a rooted tree with nodes* $V(T_\sigma) \subseteq \mathcal{Q}^*$ *and* $\rho_\sigma : V(T_\sigma) \to [0,1]$ *is a labeling of nodes in* T_σ, *be constructed as:*

1. *The root of* T_σ *is* σ, *with* $\rho_\sigma(\sigma) = 1$ *and depth of* σ *being 0;*
2. *for* $i = 0, 1, \ldots$: *for all nodes* $X \in V(T_\sigma)$ *of depth* i *in the current* T_σ,
   - (a) *if* $\mathsf{NextVar}(X) = \perp$ *or* $f(X) = \texttt{True}$, *then leave* X *as a leaf node in* T_σ;
   - (b) *otherwise, supposed* $u = \mathsf{NextVar}(X)$, *append* $\{X_{u \leftarrow x} \mid x \in \mathcal{Q}_u^\star\}$ *as the* $q_u + 1$ *children to the node* X *in* T_σ, *and label them as:*

$$\forall x \in \mathcal{Q}_u^\star, \quad \rho_\sigma(X_{u \leftarrow x}) = \begin{cases} (1 - q_u \theta_u)\rho_\sigma(X) & \text{if } x = \star, \\ \mu_u^\sigma(x) \cdot \rho_\sigma(X) & \text{if } x \in Q_u. \end{cases}$$

The resulting $\mathcal{T}_\sigma = (T_\sigma, \rho_\sigma)$ *is called the* recursive cost tree (RCT) rooted at σ.

For any RCT $\mathcal{T}_\sigma$, let $\mathcal{L}(\mathcal{T}_\sigma)$ be the set of leaf nodes in $\mathcal{T}_\sigma$. Let $\mathcal{L}_g(\mathcal{T}_\sigma) \triangleq \{X \in \mathcal{L}(\mathcal{T}_\sigma) : f(X) = \texttt{False}\}$ and $\mathcal{L}_b(\mathcal{T}_\sigma) \triangleq \{X \in \mathcal{L}(\mathcal{T}_\sigma) : f(X) = \texttt{True}\}$ be the sets of leaf nodes in $\mathcal{T}_\sigma$ that don't and do satisfy the truncation condition, respectively. We also define the following function $\lambda(\cdot)$ on $\mathcal{T}_\sigma$:

$$\lambda(\mathcal{T}_\sigma) \triangleq \sum_{X \in \mathcal{L}_b(\mathcal{T}_\sigma)} \rho_\sigma(X). \tag{4.14}$$

Recall the definition of total variation distance. Let μ and ν be two probability distributions over the same sample space Ω_S. The total variation distance between u and v is defined by

$$d_{\mathrm{TV}}(\mu, \nu) \triangleq \frac{1}{2} \sum_{x \in \Omega_S} |\mu(x) - \nu(x)|.$$

The total variation distance between the distribution returned by the subroutine $\mathsf{MarginalApproximator}$ and the true marginal distribution is upper bounded through $\lambda(\cdot)$.

LEMMA 4.4. *For any* (Φ, σ, v) *satisfying Condition 4.1, it holds that*

$$d_{\mathrm{TV}}(\hat{\mu}, \mu_v^\sigma) \leq \lambda(\mathcal{T}_{\sigma_{v \leftarrow \star}}),$$

where $\hat{\mu}$ *is the distribution returned by* $\mathsf{MarginalApproximator}(\Phi, \sigma, v)$.

We then prove Lemma 4.4. The following recursive relation for RCT is immediate by definition.

PROPOSITION 4.2. *Let* $\sigma \in \mathcal{Q}^*$, $u = \mathsf{NextVar}(\sigma)$. *If* $u \neq \perp$ *and* $f(\sigma) = \texttt{False}$, *then*

$$\lambda(\mathcal{T}_\sigma) = (1 - q_u\theta_u)\lambda\left(\mathcal{T}_{\sigma_{u\leftarrow\star}}\right) + \sum_{x\in Q_u}\left(\mu_u^\sigma(x)\cdot\lambda\left(\mathcal{T}_{\sigma_{u\leftarrow x}}\right)\right)$$

We have the following lemma which bounds the total variation distance between the distribution returned by the subroutine RecursiveApproximator and the "overflow" marginal distribution $\mathcal{D}$.

LEMMA 4.5. *For any* (Φ, σ, v) *satisfying Condition 4.2, it holds that*

$$d_{\mathrm{TV}}(\hat{\mathcal{D}}, \mathcal{D}) \le \lambda(\mathcal{T}_\sigma),$$

where $\hat{\mathcal{D}}$ *is the distribution returned by* $\mathsf{RecursiveApproximator}(\Phi, \sigma, v)$ *and* $\mathcal{D} \triangleq \mathcal{D}_v^\sigma = \frac{\mu_v^\sigma - \theta_v}{1 - q_v\theta_v}$.

Proof. We prove the lemma by an induction on the structure of the RCT. The base case is when T_σ is a single root. Thus we have $\sigma \in \mathcal{L}(\mathcal{T}_\sigma)$. By Item 2a of Definition 4.2, we also have $f(\sigma) = \texttt{True}$ or $\mathsf{NextVar}(\sigma) = \perp$. For these two possibilities, we always have $d_{\mathrm{TV}}(\hat{\mathcal{D}}, \mathcal{D}) \le \lambda(\mathcal{T}_\sigma)$.

1. If $f(\sigma) = \texttt{True}$, we have $\sigma \in \mathcal{L}_b(\mathcal{T}_\sigma)$ by $\sigma \in \mathcal{L}(\mathcal{T}_\sigma)$. Thus, $\lambda(\mathcal{T}_\sigma) = \rho_\sigma(\sigma) = 1 \ge d_{\mathrm{TV}}(\hat{\mathcal{D}}, \mathcal{D})$.

2. Otherwise, $f(\sigma) = \texttt{False}$ and $\mathsf{NextVar}(\sigma) = \perp$. Thus, the condition in Line 5 of $\mathsf{RecursiveApproximator}(\Phi, \sigma, v)$ is not satisfied and Lines 4-12 are skipped. According to Line 16, we have $\hat{\mathcal{D}}$ is exactly $\mathcal{D}$. Therefore $d_{\mathrm{TV}}(\hat{\mathcal{D}}, \mathcal{D}) = 0 \le \lambda(\mathcal{T}_\sigma)$, because $\lambda(\mathcal{T}_\sigma)$ is nonnegative.

For the induction step, we assume that T_σ is a tree of depth > 0, which implies $f(\sigma) = \texttt{False}$ and $\mathsf{NextVar}(\sigma) = u \neq \perp$ for some $u \in V$ by Item 2a of Definition 4.2. Let $\hat{\mathcal{D}}_u^{\sigma_{u\leftarrow\star}}$ be the probability vector returned by $\mathsf{RecursiveApproximator}(\Phi, \sigma_{u\leftarrow\star}, u)$ and $\hat{\mathcal{D}}_v^{\sigma_{u\leftarrow x}}$ be the probability vector returned by $\mathsf{RecursiveApproximator}(\Phi, \sigma_{u\leftarrow x}, v)$ for each $x \in Q_u$. Given a probability vector $\mathbf{p}$, we use $\mathbf{p}(j)$ to denote the j-th item of $\mathbf{p}$. By Line 7-Line 12 of Algorithm 2, we have for each $j \in Q_v$,

$$\begin{aligned}
\hat{\mathcal{D}}(j) &= \sum_{i\in Q_u}\left(\theta_u + (1-q_u\theta_u)\hat{\mathcal{D}}_u^{\sigma_{u\leftarrow\star}}(i)\right)\cdot\hat{\mathcal{D}}_v^{\sigma_{u\leftarrow i}}(j)\\
&= \sum_{i\in Q_u}\left(\theta_u + (1-q_u\theta_u)\mathcal{D}_u^{\sigma_{u\leftarrow\star}}(i) + (1-q_u\theta_u)\left(\hat{\mathcal{D}}_u^{\sigma_{u\leftarrow\star}}(i) - \mathcal{D}_u^{\sigma_{u\leftarrow\star}}(i)\right)\right)\cdot\hat{\mathcal{D}}_v^{\sigma_{u\leftarrow i}}(j)\\
&= \sum_{i\in Q_u}\left(\mu_u^\sigma(i) + (1-q_u\theta_u)\left(\hat{\mathcal{D}}_u^{\sigma_{u\leftarrow\star}}(i) - \mathcal{D}_u^{\sigma_{u\leftarrow\star}}(i)\right)\right)\cdot\hat{\mathcal{D}}_v^{\sigma_{u\leftarrow i}}(j),
\end{aligned} \tag{4.15}$$

where the last equality is by the definition of $\mathcal{D}(\cdot)$. Moreover, by the chain rule, we also have for each $j \in Q_v$,

$$\mathcal{D}(j) = \sum_{i\in Q_u}\mu_u^\sigma(i)\cdot\mathcal{D}_v^{\sigma_{u\leftarrow i}}(j). \tag{4.16}$$

Combining (4.15) with (4.16), we have

$$\hat{\mathcal{D}}(j) - \mathcal{D}(j) = \sum_{i\in Q_u}\left(\mu_u^\sigma(i)\cdot(\hat{\mathcal{D}}_v^{\sigma_{u\leftarrow i}}(j) - \mathcal{D}_v^{\sigma_{u\leftarrow i}}(j)) + (1-q_u\theta_u)\left(\hat{\mathcal{D}}_u^{\sigma_{u\leftarrow\star}}(i) - \mathcal{D}_u^{\sigma_{u\leftarrow\star}}(i)\right)\cdot\hat{\mathcal{D}}_v^{\sigma_{u\leftarrow i}}(j)\right).$$

Combining with the triangle inequality for absolute values, we have

$$\left|\hat{\mathcal{D}}(j) - \mathcal{D}(j)\right| \le \sum_{i\in Q_u}\left(\mu_u^\sigma(i)\cdot\left|\hat{\mathcal{D}}_v^{\sigma_{u\leftarrow i}}(j) - \mathcal{D}_v^{\sigma_{u\leftarrow i}}(j)\right| + (1-q_u\theta_u)\cdot\left|\hat{\mathcal{D}}_u^{\sigma_{u\leftarrow\star}}(i) - \mathcal{D}_u^{\sigma_{u\leftarrow\star}}(i)\right|\cdot\hat{\mathcal{D}}_v^{\sigma_{u\leftarrow i}}(j)\right).$$

Therefore, we have

$$
\begin{aligned}
d_{\mathrm{TV}}(\hat{\mathcal{D}},\mathcal{D}) =& \frac{1}{2}\sum_{j\in Q_v}\left|\hat{\mathcal{D}}(j)-\mathcal{D}(j)\right| \\
\le& \frac{1}{2}\sum_{i\in Q_u}\Bigg(\mu_u^\sigma(i)\cdot\sum_{j\in Q_v}\left|\hat{\mathcal{D}}_v^{\sigma_{u\leftarrow i}}(j)-\mathcal{D}_v^{\sigma_{u\leftarrow i}}(j)\right| \\
&\qquad +(1-q_u\theta_u)\cdot\left|\hat{\mathcal{D}}_u^{\sigma_{u\leftarrow\star}}(i)-\mathcal{D}_u^{\sigma_{u\leftarrow\star}}(i)\right|\cdot\sum_{j\in Q_v}\hat{\mathcal{D}}_v^{\sigma_{u\leftarrow i}}(j)\Bigg) \qquad (4.17)\\
\le& \sum_{i\in Q_u}\left(u_u^\sigma(i)\cdot d_{\mathrm{TV}}(\hat{\mathcal{D}}_v^{\sigma_{u\leftarrow i}},\mathcal{D}_v^{\sigma_{u\leftarrow i}})+\frac{1}{2}\cdot(1-q_u\theta_u)\cdot\left|\hat{\mathcal{D}}_u^{\sigma_{u\leftarrow\star}}(i)-\mathcal{D}_u^{\sigma_{u\leftarrow\star}}(i)\right|\right) \\
\le& (1-q_u\theta_u)\cdot d_{\mathrm{TV}}(\hat{\mathcal{D}}_u^{\sigma_{u\leftarrow\star}},\mathcal{D}_u^{\sigma_{u\leftarrow\star}})+\sum_{i\in Q_u}\left(\mu_u^\sigma(i)\cdot d_{\mathrm{TV}}(\hat{\mathcal{D}}_v^{\sigma_{u\leftarrow i}},\mathcal{D}_v^{\sigma_{u\leftarrow i}})\right).
\end{aligned}
$$

Note that by Item 2b in Definition 4.2, for each $x\in\mathcal{Q}_u^\star$, the subtree in T_σ rooted by $\sigma_{u\leftarrow x}$ is precisely the T_X in the RCT $\mathcal{T}_X=(T_X,\rho_X)$ rooted at $X=\sigma_{u\leftarrow x}$. Moreover, by Lemma 4.3, Condition 4.2 is still satisfied by $(\Phi,\sigma_{u\leftarrow x},v)$. Thus, by induction hypothesis, we have for each $x\in\mathcal{Q}_u^\star$,

$$d_{\mathrm{TV}}(\hat{\mathcal{D}}_v^{\sigma_{u\leftarrow x}},\mathcal{D}_v^{\sigma_{u\leftarrow x}})\le\lambda(\mathcal{T}_{\sigma_{u\leftarrow x}}).$$

Combining with (4.17), it follows that

$$
\begin{aligned}
d_{\mathrm{TV}}(\hat{\mathcal{D}},\mathcal{D}) \le& (1-q_u\theta_u)\cdot d_{\mathrm{TV}}(\hat{\mathcal{D}}_u^{\sigma_{u\leftarrow\star}},\mathcal{D}_u^{\sigma_{u\leftarrow\star}})+\sum_{i\in Q_u}\left(\mu_u^\sigma(i)\cdot d_{\mathrm{TV}}(\hat{\mathcal{D}}_v^{\sigma_{u\leftarrow i}},\mathcal{D}_v^{\sigma_{u\leftarrow i}})\right) \\
\le& (1-q_u\theta_u)\lambda(\mathcal{T}_{\sigma_{u\leftarrow\star}})+\sum_{x\in Q_u}\mu_u^\sigma(x)\lambda(\mathcal{T}_{\sigma_{u\leftarrow x}}) \\
=& \lambda(\mathcal{T}_\sigma).
\end{aligned}
$$

where the equality follows by Proposition 4.2. □

For any (Φ,σ,v) satisfying Condition 4.1, one can verify that $(\Phi,\sigma_{v\leftarrow\star},v)$ satisfies Condition 4.2. Let $\hat{\mathcal{D}}$ be the distribution returned by $\mathsf{RecursiveApproximator}(\Phi,\sigma,v)$. By Lemma 4.5 we have $d_{\mathrm{TV}}(\hat{\mathcal{D}},\mathcal{D}_v^\sigma)\le\lambda(\mathcal{T}_{\sigma_{v\leftarrow\star}})$. Thus, by Lines 1-3 of $\mathsf{MarginalApproximator}(\Phi,\sigma,v)$, we have

$$d_{\mathrm{TV}}(\hat{\mu},\mu_v^\sigma)=(1-q_v\theta_v)d_{\mathrm{TV}}(\hat{\mathcal{D}},\mathcal{D}_v^\sigma)\le(1-q_v\theta_v)\lambda(\mathcal{T}_{\sigma_{v\leftarrow\star}})\le(1-q\theta)\lambda(\mathcal{T}_{\sigma_{v\leftarrow\star}})\le\lambda(\mathcal{T}_{\sigma_{v\leftarrow\star}}),$$

where the equality is by the definition of $\mathcal{D}_v^\sigma$. This proves Lemma 4.4.

4.3 A random path simulating RCT The recursive cost tree in Definition 4.2 inspires the following random process of partial assignments. Given a partial assignment σ and a variable $v\in V\setminus\Lambda(\sigma)$, define

$$
\begin{aligned}
&\gamma_v^\sigma(\star)=\frac{1-q_v\cdot\theta_v}{2-q_v\cdot\theta_v},\\
\forall x\in Q_v,\qquad &\gamma_v^\sigma(x)=\frac{\mu_v^\sigma(x)}{2-q_v\cdot\theta_v}.
\end{aligned}
$$

It is obvious to see that $\gamma_v^\sigma(\cdot)$ is a well-defined probability distribution over $\mathcal{Q}_v^\star$.

DEFINITION 4.3. (THE $\mathsf{Path}(\sigma)$ PROCESS) *For any $\sigma\in\mathcal{Q}^*$, $\mathsf{Path}(\sigma)=(\sigma_0,\sigma_1,\ldots,\sigma_\ell)$ is a random sequence of partial assignments generated from the initial $\sigma_0=\sigma$ as that for $i=0,1,\ldots$:*

1. *if* $\mathsf{NextVar}(\sigma_i)=\perp$ *or* $f(\sigma_i)=\texttt{True}$, *the sequence stops at* σ_i*;*

2. *otherwise* $u = \mathsf{NextVar}(\sigma_i) \in V$, *the partial assignment* $\sigma_{i+1} \in \mathcal{Q}^*$ *is generated from* σ_i *by randomly giving* $\sigma(u)$ *a value* $x \in \mathcal{Q}_u^\star$, *such that*

$$\forall x \in \mathcal{Q}_u^\star, \qquad \mathbf{Pr}\left[\sigma_{i+1} = (\sigma_i)_{u \leftarrow x}\right] = \gamma_u^\sigma(x).$$

The length $\ell(\sigma)$ of $\mathsf{Path}(\sigma) = \left(\sigma_0, \sigma_1, \cdots, \sigma_{\ell(\sigma)}\right)$ is a random variable whose distribution is determined by σ. We simply write $\ell = \ell(\sigma)$ and $\mathsf{Path}(\sigma) = (\sigma_0, \sigma_1, \cdots, \sigma_\ell)$ if σ is clear from the context. It is straightforward to verify that $\mathsf{Path}(\sigma)$ satisfies the Markov property.

The significance of the random process $\mathsf{Path}(\sigma)$ is that it is related to the total variation distance between the distribution returned by $\mathsf{MarginalApproximator}(\Phi, \sigma, v)$ and μ_v^σ through the following function $H(\cdot)$. For any two partial assignments τ_1, τ_2, define

$$\chi(\tau_1, \tau_2) \triangleq \prod_{v \in \Lambda^+(\tau_1) \setminus \Lambda^+(\tau_2)} (2 - q_v \theta_v).$$

Given any sequence $P = (\sigma_0, \sigma_1, \ldots, \sigma_\ell) \in (\mathcal{Q}^*)^{\ell+1}$ with $\ell \geq 0$, define

$$H(P) \triangleq \mathbb{1}\left[f(\sigma_\ell) = \mathtt{True}\right] \cdot \chi(\sigma_\ell, \sigma_0). \tag{4.18}$$

Recall the $\lambda(\mathcal{T}_\sigma)$ defined in (4.14). We have the following lemma.

LEMMA 4.6. *For any partial assignment* $\sigma \in \mathcal{Q}^*$, *the following holds for* $P = \mathsf{Path}(\sigma)$:

$$\mathbb{E}\left[H(P)\right] = \lambda(\mathcal{T}_\sigma)$$

Proof. We show the lemma by an induction on the structure of the RCT. The base case is when T_σ is a single root. Thus we have $\sigma \in \mathcal{L}(\mathcal{T}_\sigma)$. By Item 2a of Definition 4.2, we have $\mathsf{NextVar}(\sigma) = \perp$ or $f(\sigma) = \mathtt{True}$. Also, by Definition 4.3 we have $\mathsf{Path}(\sigma) = (\sigma)$ and $\ell(\sigma) = 0$. Then we always have $\lambda(\mathcal{T}_\sigma) = H(\mathsf{Path}(\sigma))$ no matter whether $f(\sigma) = \mathtt{True}$:

1. If $f(\sigma) = \mathtt{True}$, then we have $\sigma \in \mathcal{L}_b(\mathcal{T}_\sigma)$ by $\sigma \in \mathcal{L}(\mathcal{T}_\sigma)$. Thus, $\lambda(\mathcal{T}_\sigma) = \rho_\sigma(\sigma) = 1$. Meanwhile, by $\mathsf{Path}(\sigma) = (\sigma)$, we have $\sigma_\ell = \sigma_0 = \sigma$. Thus, $f(\sigma_\ell) = f(\sigma) = \mathtt{True}$ and $\Lambda^+(\sigma_\ell) \setminus \Lambda^+(\sigma_0) = \emptyset$. We have $H(\mathsf{Path}(\sigma)) = 1$. In summary, we have $\lambda(\mathcal{T}_\sigma) = H(\mathsf{Path}(\sigma))$.
2. Otherwise, $f(\sigma) = \mathtt{False}$. We have $\sigma \notin \mathcal{L}_b(\mathcal{T}_\sigma)$ and $\mathcal{L}_b(\mathcal{T}_\sigma) = \emptyset$. Thus, we have $\lambda(\mathcal{T}_\sigma) = 0$. Also, by $\mathsf{Path}(\sigma) = (\sigma)$, we have $\sigma_\ell = \sigma$. Combining with $f(\sigma) = \mathtt{False}$, we have $f(\sigma_\ell) = f(\sigma) = \mathtt{False}$. Thus $H(\mathsf{Path}(\sigma)) = 0$. In summary, $\lambda(\mathcal{T}_\sigma) = H(\mathsf{Path}(\sigma))$.

For the induction step, we assume that T_σ is a tree of depth > 0. Thus by Item 2a of Definition 4.2, we have $f(\sigma) = \mathtt{False}$ and $\mathsf{NextVar}(\sigma) = u \neq \perp$ for some $u \in V$. According to Item 2 of Definition 4.3, we have $\ell(\sigma) \geq 1$ and

$$\forall x \in \mathcal{Q}_u^\star, \quad \mathbf{Pr}\left[\sigma_1 = \sigma_{u \leftarrow x}\right] = \gamma_u^\sigma(x). \tag{4.19}$$

Moreover, by the Markov property of $\mathsf{Path}(\sigma)$, given $\sigma_1 = \sigma_{u \leftarrow x}$ for each $x \in \mathcal{Q}_u^\star$, the subsequence $(\sigma_1, \sigma_2, \cdots, \sigma_\ell)$ is identically distributed as $\mathsf{Path}(\sigma_{u \leftarrow x})$. In addition, it can be verified that for any sequence of partial assignments $P = (\tau_0, \tau_1, \ldots, \tau_\ell)$ with $\ell \geq 1$ satisfying $\mathbf{Pr}\left[\mathsf{Path}(\sigma) = P\right] > 0$,

$$H(P) = (2 - q_u \theta_u) H((\tau_1, \ldots, \tau_\ell)). \tag{4.20}$$

There are two possibilities:

1. If $f(\tau_\ell) = \mathtt{False}$, then we have $H(P) = H((\tau_1, \ldots, \tau_\ell)) = 0 = (2 - q_u\theta_u) H((\tau_1, \ldots, \tau_\ell))$.
2. Otherwise, $f(\tau_\ell) = \mathtt{True}$. By $\mathbf{Pr}\left[\mathsf{Path}(\sigma) = P\right] > 0$, we have $\tau_0 = \sigma$, $\Lambda^+(\tau_1) = \Lambda^+(\tau_0) \cup \{\mathsf{NextVar}(\tau_0)\}$, and $\Lambda^+(\tau_0) \subsetneq \Lambda^+(\tau_1) \subseteq \Lambda^+(\tau_\ell)$. Combining with $\mathsf{NextVar}(\sigma) = u$, we have $\mathsf{NextVar}(\tau_0) = \mathsf{NextVar}(\sigma) = u$, $\Lambda^+(\tau_1) = \Lambda^+(\tau_0) \cup \{u\}$, and $u \notin \Lambda^+(\tau_0)$. Combining with $\Lambda^+(\tau_1) \subseteq \Lambda^+(\tau_\ell)$, we have $\Lambda^+(\tau_\ell) \setminus \Lambda^+(\tau_0) = \{u\} \uplus (\Lambda^+(\tau_\ell) \setminus \Lambda^+(\tau_1))$. Therefore, we have

$$\begin{aligned} H(P) &= \mathbb{1}\left[f(\tau_\ell) = \mathtt{True}\right] \cdot \chi(\tau_\ell, \tau_0) \\ &= \chi(\tau_\ell, \tau_0) \\ &= (2 - q_u\theta_u)\chi(\tau_\ell, \tau_1) \\ &= (2 - q_u\theta_u) H((\tau_1, \ldots, \tau_\ell)). \end{aligned}$$

By the law of total expectation, we have

$$\begin{aligned}\mathbb{E}\left[H(\mathsf{Path}(\sigma))\right] &= \sum_{x\in\mathcal{Q}_u^\star}\left(\mathbf{Pr}\left[\sigma_1=\sigma_{u\leftarrow x}\right]\cdot\mathbb{E}\left[H(\mathsf{Path}(\sigma))\mid\sigma_1=\sigma_{u\leftarrow x}\right]\right)\\ &= \sum_{x\in\mathcal{Q}_u^\star}\left(\gamma_u^\sigma(x)\cdot(2-q_u\theta_u)\mathbb{E}\left[H(\mathsf{Path}(\sigma_{u\leftarrow x}))\right]\right)\\ &=(1-q_u\theta_u)\mathbb{E}\left[H(\mathsf{Path}(\sigma_{u\leftarrow\star}))\right]+\sum_{x\in Q_u}\left(\mu_u^\sigma(x)\mathbb{E}\left[H(\mathsf{Path}(\sigma_{u\leftarrow x}))\right]\right),\end{aligned}\tag{4.21}$$

where the second equality is by (4.19) and (4.20). Note that by Item 2b in Definition 4.2, for each $x\in\mathcal{Q}_u^\star$, the subtree in T_σ rooted by $\sigma_{u\leftarrow x}$ is precisely the T_X in the RCT $\mathcal{T}_X=(T_X,\rho_X)$ rooted at $X=\sigma_{u\leftarrow x}$. Thus, by induction hypothesis, we have

$$\forall x\in\mathcal{Q}_u^\star, \mathbb{E}\left[H(\mathsf{Path}(\sigma_{u\leftarrow x}))\right]=\lambda(\mathcal{T}_{\sigma_{u\leftarrow x}})$$

Combining with (4.21), we have

$$\mathbb{E}\left[H(\mathsf{Path}(\sigma))\right]=(1-q_u\theta_u)\lambda(\mathcal{T}_{\sigma_{u\leftarrow\star}})+\sum_{x\in Q_u}\mu_u^\sigma(x)\lambda(\mathcal{T}_{\sigma_{u\leftarrow x}})=\lambda(\mathcal{T}_\sigma),$$

where the last equality is by Proposition 4.2. □

4.4 Correctness of the counting algorithm

In this subsection we bound the total variation distance between the distribution returned by Algorithm 1 and the true marginal distribution by the upper bound function $F(\cdot)$. The whole subsection will devote to proving the following proposition.

PROPOSITION 4.3. *For any input (Φ,σ,v) satisfying Condition 4.1, it holds that*

$$d_{\mathrm{TV}}(\xi,\mu_v^\sigma)\le F(\sigma),$$

where ξ is the distribution returned by $\mathsf{MarginalApproximator}(\Phi,\sigma,v)$.

4.4.1 Generalized $\{2,3\}$-tree witness for truncation

Given $\sigma\in\mathcal{Q}^*$ and generalized $\{2,3\}$-tree $T=U\circ E$ in H_Φ, let $\mathsf{Path}(\sigma)=(\sigma_0,\sigma_1,\dots,\sigma_\ell)$. Define the event $\mathcal{E}_T^\sigma$ as

$$\mathcal{E}_T^\sigma: U=V_\star^{\sigma_\ell}\wedge E\subseteq\mathcal{C}_{\star\text{-frozen}}^{\sigma_\ell}.\tag{4.22}$$

Let $\sigma\in\mathcal{Q}^*$ be a partial assignment such that only one variable $v\in V$ has $\sigma(v)=\star$. The following lemma shows that if the path $\mathsf{Path}(\sigma)=(\sigma_0,\sigma_1,\dots,\sigma_\ell)$ generated from such σ gets truncated at Item 1 of Definition 4.3 for satisfying $f(\sigma_\ell)=\mathtt{True}$, then there must be a large generalized $\{2,3\}$-tree in H_Φ. Its proof is deferred to Appendix A.1.

LEMMA 4.7. *Let $\sigma\in\mathcal{Q}^*$ be a partial assignment with exactly one variable $v\in V$ having $\sigma(v)=\star$ and $\mathsf{Path}(\sigma)=(\sigma_0,\sigma_1,\dots,\sigma_\ell)$. Suppose $f(\sigma_\ell)=\mathtt{True}$, then there exists a generalized $\{2,3\}$-tree $T=U\circ E$ in H_Φ with some auxiliary tree rooted at v satisfying*

$$L\le|U|+\Delta\cdot|E|\le L\Delta$$

such that $\mathcal{E}_T^\sigma$ happens.

4.4.2 Probability bounds for generalized $\{2,3\}$-tree witness

Recall the function $H(\cdot)$ defined in (4.18) and the event $\mathcal{E}_T^\sigma$ defined in (4.22). A crucial lemma we will show is given as follows, which gives a probability bound for certain generalized $\{2,3\}$-tree witness in H_Φ.

LEMMA 4.8. *Let $\sigma\in\mathcal{Q}^*$ be a partial assignment with exactly one variable $v\in V$ having $\sigma(v)=\star$ satisfying $\mathbb{P}[\neg c\mid\sigma]\le\alpha q$ for all $c\in\mathcal{C}$. Let T be any generalized $\{2,3\}$-tree in H_Φ, then we have*

$$\mathbf{Pr}\left[\mathcal{E}_T^\sigma\right]\cdot\mathbb{E}\left[H(\mathsf{Path}(\sigma))\mid\mathcal{E}_T^\sigma\right]\le F(\sigma,T\setminus\{v\}).$$

For any constraint $c \in \mathcal{C}$ and partial assignment $\sigma \in \mathcal{Q}^*$, we define

$$Z(\sigma, c) \triangleq |\mathsf{vbl}(c) \setminus \Lambda(\sigma)|.$$

as the number of unassigned variables in $\mathsf{vbl}(c)$.

For any generalized $\{2,3\}$-tree T in H_Φ and partial assignment $\sigma \in \mathcal{Q}^*$, we further define

$$g(\sigma, T) \triangleq \prod_{v \in U \setminus V_\star^\sigma} (1 - q_v\theta_v) \prod_{c \in E} \left(\alpha^{-1}\mathbb{P}[\neg c \mid \sigma](1+\eta)^{Z(\sigma,c)}\right). \tag{4.23}$$

To prove Lemma 4.8, it is sufficient to show the following.

LEMMA 4.9. *Let $\sigma \in \mathcal{Q}^*$ be a partial assignment satisfying $\mathbb{P}[\neg c \mid \sigma] \le \alpha q$ for all $c \in \mathcal{C}$, and let $\mathsf{Path}(\sigma) = (\sigma_0, \sigma_1, \dots, \sigma_\ell)$. Then for any generalized $\{2,3\}$-tree $T = U \circ E$ in H_Φ,*

$$\mathbf{Pr}[\mathcal{E}_T^\sigma] \cdot \mathbb{E}[H(\mathsf{Path}(\sigma)) \mid \mathcal{E}_T^\sigma] \le g(\sigma, T). \tag{4.24}$$

Proof. We show the lemma by a structural induction on $\mathsf{Path}(\sigma)$. The base case is when $\mathsf{Path}(\sigma) = (\sigma)$. Then we have $\ell(\sigma) = 0$ and $\sigma = \sigma_\ell$. In this case, $\mathcal{E}_T^\sigma$ is the deterministic event $U = V_\star^\sigma \wedge E \subseteq \mathcal{C}_{\star\text{-frozen}}^\sigma$. If $U \neq V_\star^\sigma$ or $E \not\subseteq \mathcal{C}_{\star\text{-frozen}}^\sigma$, we have $\mathbf{Pr}[\mathcal{E}_T^\sigma] = 0$ and the lemma is immediate. Otherwise, we have $U = V_\star^\sigma \wedge E \subseteq \mathcal{C}_{\star\text{-frozen}}^\sigma$. By $\sigma = \sigma_\ell$ and (4.18), we have $H(\mathsf{Path}(\sigma)) \le 1$. Thus, we have

$$\mathbf{Pr}[\mathcal{E}_T^\sigma] \cdot \mathbb{E}[H(\mathsf{Path}(\sigma)) \mid \mathcal{E}_T^\sigma] \le H(\mathsf{Path}(\sigma)) \le 1.$$

Meanwhile, by $E \subseteq \mathcal{C}_{\star\text{-frozen}}^\sigma$ and Definition 3.4, we have $E \subseteq \mathcal{C}_{\star\text{-frozen}}^\sigma \subseteq \mathcal{C}_{\text{frozen}}^\sigma$. Thus, for each $c \in E$, we have c is σ-frozen. Combining with Definition 3.2, we have $\mathbb{P}[\neg c \mid \sigma] > \alpha$. Combining with $U = V_\star^\sigma$, we have

$$g(\sigma, T) = \prod_{c \in E} \left(\alpha^{-1}\mathbb{P}[\neg c \mid \sigma](1+\eta)^{Z(\sigma,c)}\right) \ge \prod_{c \in E} \left(\alpha^{-1}\alpha(1+\eta)^{Z(\sigma,c)}\right) \ge 1 \ge \mathbf{Pr}[\mathcal{E}_T^\sigma].$$

The base case is proved.

For the induction steps, we assume that $\ell(\sigma) \ge 1$, which by Item 1 of Definition 4.3, says that $\mathsf{NextVar}(\sigma) = u \neq \perp$ for some $u \in V$ and $f(\sigma) = \texttt{False}$. According to Item 2 of Definition 4.3, we have

$$\forall x \in \mathcal{Q}_u^\star, \quad \mathbf{Pr}[\sigma_1 = \sigma_{u \leftarrow x}] = \gamma_u^\sigma(x).$$

Thus, by the law of total probability, we have

$$\begin{aligned}
&\mathbf{Pr}[\mathcal{E}_T^\sigma] \cdot \mathbb{E}[H(\mathsf{Path}(\sigma)) \mid \mathcal{E}_T^\sigma] \\
=& \sum_{x \in \mathcal{Q}_u^\star} \left(\mathbf{Pr}[\sigma_1 = \sigma_{u \leftarrow x}]\, \mathbf{Pr}[\mathcal{E}_T^\sigma \mid \sigma_1 = \sigma_{u \leftarrow x}]\, \mathbb{E}[H(\mathsf{Path}(\sigma)) \mid \sigma_1 = \sigma_{u \leftarrow x} \wedge \mathcal{E}_T^\sigma]\right) \\
=& \sum_{x \in \mathcal{Q}_u^\star} \left(\gamma_u^\sigma(x) \cdot \mathbf{Pr}[\mathcal{E}_T^\sigma \mid \sigma_1 = \sigma_{u \leftarrow x}]\, \mathbb{E}[H(\mathsf{Path}(\sigma)) \mid \sigma_1 = \sigma_{u \leftarrow x} \wedge \mathcal{E}_T^\sigma]\right)
\end{aligned} \tag{4.25}$$

Moreover, by (4.18) we have

$$\begin{aligned}
&\mathbb{E}[H(\mathsf{Path}(\sigma)) \mid \sigma_1 = \sigma_{u \leftarrow x} \wedge \mathcal{E}_T^\sigma] \\
=&\mathbb{E}[\mathbb{1}[f(\sigma_\ell) = \texttt{True}] \cdot \chi(\sigma_\ell, \sigma_0) \mid \sigma_1 = \sigma_{u \leftarrow x} \wedge \mathcal{E}_T^\sigma] \\
=& (2 - q_u\theta_u)\, \mathbb{E}[\mathbb{1}[f(\sigma_\ell) = \texttt{True}] \cdot \chi(\sigma_\ell, \sigma_1) \mid \sigma_1 = \sigma_{u \leftarrow x} \wedge \mathcal{E}_T^\sigma]
\end{aligned} \tag{4.26}$$

In addition, by the Markov property, given $\sigma_1 = \tau \triangleq \sigma_{u \leftarrow x}$ for each $x \in \mathcal{Q}_u^\star$, the subsequence $(\sigma_1, \sigma_2, \cdots, \sigma_\ell)$ is identically distributed as $\mathsf{Path}(\tau)$. Thus, we have σ_ℓ is identically distributed as $\tau_{\ell(\tau)}$. combining with (4.18) and (4.22) , we have

$$\begin{aligned}
&\mathbb{E}[\mathbb{1}[f(\sigma_\ell) = \texttt{True}] \cdot \chi(\sigma_\ell, \sigma_1) \mid \sigma_1 = \tau \wedge \mathcal{E}_T^\sigma] \\
=&\mathbb{E}\left[\mathbb{1}\left[f(\tau_{\ell(\tau)}) = \texttt{True}\right] \cdot \chi(\tau_{\ell(\tau)}, \sigma_1) \mid \sigma_1 = \tau \wedge \mathcal{E}_T^\tau\right] \\
=&\mathbb{E}\left[\mathbb{1}\left[f(\tau_{\ell(\tau)}) = \texttt{True}\right] \cdot \chi(\tau_{\ell(\tau)}, \tau) \mid \mathcal{E}_T^\tau\right] \\
=&\mathbb{E}[H(\mathsf{Path}(\tau)) \mid \mathcal{E}_T^\tau].
\end{aligned}$$

Combining with (4.26), we have

$$\text{(4.27)}\qquad \begin{aligned}\mathbb{E}\left[H(\mathsf{Path}(\sigma)) \mid \sigma_1 = \sigma_{u\leftarrow x} \wedge \mathcal{E}_T^\sigma\right] &= (2 - q_u\theta_u)\,\mathbb{E}\left[H(\mathsf{Path}(\sigma_1)) \mid \sigma_1 = \sigma_{u\leftarrow x} \wedge \mathcal{E}_T^\sigma\right] \\ &= (2 - q_u\theta_u)\,\mathbb{E}\left[H(\mathsf{Path}(\sigma_{u\leftarrow x})) \mid \mathcal{E}_T^{\sigma_{u\leftarrow x}}\right].\end{aligned}$$

Recall that given $\sigma_1 = \sigma_{u\leftarrow x}$ for each $x \in \mathcal{Q}_u^\star$, σ_ℓ is identically distributed as $\tau_{\ell(\tau)}$ where $\tau = \sigma_{u\leftarrow x}$. Combining with (4.22), we have $\mathbf{Pr}\left[\mathcal{E}_T^\sigma \mid \sigma_1 = \sigma_{u\leftarrow x}\right] = \mathbf{Pr}\left[\mathcal{E}_T^{\sigma_{u\leftarrow x}}\right]$. Combining with (4.25) and (4.27), we have

$$\text{(4.28)}\qquad \mathbf{Pr}\left[\mathcal{E}_T^\sigma\right] \cdot \mathbb{E}\left[H(\mathsf{Path}(\sigma)) \mid \mathcal{E}_T^\sigma\right] = \sum_{x\in\mathcal{Q}_u^\star} \left((2 - q_u\theta_u)\,\gamma_u^\sigma(x) \cdot \mathbf{Pr}\left[\mathcal{E}_T^{\sigma_{u\leftarrow x}}\right] \cdot \mathbb{E}\left[H(\mathsf{Path}(\sigma_{u\leftarrow x}) \mid \mathcal{E}_T^{\sigma_{u\leftarrow x}}\right]\right).$$

We then show the induction step for two cases respectively, namely the case when $u \in U$ and the case when $u \notin U$. At first we assume $u \in U$. Given $x \in Q_u$ and $\tau = \sigma_{u\leftarrow x}$, by $\tau(u) = x$, we also have $\tau_{\ell(\tau)}(u) = x \neq \star$. Thus $u \notin V_\star^{\tau_{\ell(\tau)}}$. Combining with $u \in U$, we have $U \neq V_\star^{\tau_{\ell(\tau)}}$. Combining with (4.22), we have $\mathcal{E}_T^\tau$ does not happen. In summary, for each $x \in Q_u$, $\mathcal{E}_T^{\sigma_{u\leftarrow x}}$ does not happen and $\mathbf{Pr}\left[\mathcal{E}_T^{\sigma_{u\leftarrow x}}\right] = 0$. Combining with (4.28), we have

$$\text{(4.29)}\qquad \mathbf{Pr}\left[\mathcal{E}_T^\sigma\right] \cdot \mathbb{E}\left[H(\mathsf{Path}(\sigma)) \mid \mathcal{E}_T^\sigma\right] = (1 - q_u\theta_u) \cdot \mathbf{Pr}\left[\mathcal{E}_T^{\sigma_{u\leftarrow\star}}\right] \cdot \mathbb{E}\left[H(\mathsf{Path}(\sigma_{u\leftarrow\star})) \mid \mathcal{E}_T^{\sigma_{u\leftarrow\star}}\right].$$

In addition, by $\sigma \in \mathcal{Q}^*$ is a partial assignment satisfying $\mathbb{P}[\neg c \mid \sigma] \leq \alpha q$ for all $c \in \mathcal{C}$, one can also verify $\mathbb{P}[\neg c \mid \sigma_{u\leftarrow x}] \leq \alpha q$ for all $c \in \mathcal{C}$ and $x \in \mathcal{Q}_u^\star$ by a similar argument as Lemma 4.3. Thus by the induction hypothesis, for each $x \in \mathcal{Q}_u^\star$ we have

$$\text{(4.30)}\qquad \mathbf{Pr}\left[\mathcal{E}_T^{\sigma_{u\leftarrow x}}\right] \cdot \mathbb{E}\left[H(\mathsf{Path}(\sigma_{u\leftarrow x})) \mid \mathcal{E}_T^{\sigma_{u\leftarrow x}}\right] \leq g(\sigma_{u\leftarrow x}, T).$$

Combining with (4.29), we have

$$\text{(4.31)}\qquad \begin{aligned}&\mathbf{Pr}\left[\mathcal{E}_T^\sigma\right] \cdot \mathbb{E}\left[H(\mathsf{Path}(\sigma)) \mid \mathcal{E}_T^\sigma\right] \\ \leq &(1 - q_u\theta_u) \cdot g(\sigma_{u\leftarrow\star}, T) \\ = &(1 - q_u\theta_u) \prod_{v\in U\setminus V_\star^{\sigma_{u\leftarrow\star}}} (1 - q_v\theta_v) \prod_{c\in E} \left(\alpha^{-1}\mathbb{P}[\neg c \mid \sigma_{u\leftarrow\star}](1+\eta)^{Z(\sigma_{u\leftarrow\star},c)}\right) \\ = &(1 - q_u\theta_u) \prod_{v\in U\setminus V_\star^{\sigma_{u\leftarrow\star}}} (1 - q_v\theta_v) \prod_{c\in E} \left(\alpha^{-1}\mathbb{P}[\neg c \mid \sigma](1+\eta)^{Z(\sigma,c)}\right),\end{aligned}$$

where the last equality is by $\mathbb{P}[\neg c \mid \sigma_{u\leftarrow\star}] = \mathbb{P}[\neg c \mid \sigma]$ and $Z(\sigma_{u\leftarrow\star}, c) = Z(\sigma, c)$ for each σ and c. In addition, by $u = \mathsf{NextVar}\,(\sigma)$, we have $\sigma(u) = \text{☆} \neq \star$. Thus, $u \notin V_\star^\sigma$. Meanwhile, by $\sigma_{u\leftarrow\star}(u) = \star$, we have $u \in V_\star^{u\leftarrow\star}$. Thus, $V_\star^{\sigma_{u\leftarrow\star}} = V_\star^\sigma \biguplus \{u\}$. Combining with $u \in U$, we have $U \setminus V_\star^\sigma = (U \setminus V_\star^{\sigma_{u\leftarrow\star}}) \biguplus \{u\}$. Therefore,

$$(1 - q_u\theta_u) \prod_{v\in U\setminus V_\star^{\sigma_{u\leftarrow\star}}} (1 - q_v\theta_v) = \prod_{v\in U\setminus V_\star^\sigma} (1 - q_v\theta_v).$$

Combining with (4.31), (4.24) is immediate. This finishes the induction step for the case when $u \in U$.

In the following, we assume $u \notin U$. Given $\tau = \sigma_{u\leftarrow\star}$, by $\tau(u) = \star$, we also have $\tau_{\ell(\tau)}(u) = \star$. Thus $u \in V_\star^{\tau_{\ell(\tau)}}$. Combining with $u \notin U$, we have $U \neq V_\star^{\tau_{\ell(\tau)}}$. Combining with (4.22), we have $\mathcal{E}_T^\tau = \mathcal{E}_T^{\sigma_{u\leftarrow\star}}$ does not happen and $\mathbf{Pr}\left[\mathcal{E}_T^{\sigma_{u\leftarrow\star}}\right] = 0$. Combining with (4.28), we have

$$\mathbf{Pr}\left[\mathcal{E}_T^\sigma\right] \cdot \mathbb{E}\left[H(\mathsf{Path}(\sigma)) \mid \mathcal{E}_T^\sigma\right] = \sum_{x\in Q_u} \left(\mu_u^\sigma(x) \cdot \mathbf{Pr}\left[\mathcal{E}_T^{\sigma_{u\leftarrow x}}\right] \cdot \mathbb{E}\left[H(\mathsf{Path}(\sigma_{u\leftarrow x}) \mid \mathcal{E}_T^{\sigma_{u\leftarrow x}}\right]\right).$$

Combining with (4.30), we have

$$\text{(4.32)}\qquad \begin{aligned}&\mathbf{Pr}\left[\mathcal{E}_T^\sigma\right] \cdot \mathbb{E}\left[H(\mathsf{Path}(\sigma)) \mid \mathcal{E}_T^\sigma\right] \\ \leq &\sum_{x\in Q_u} \left(\mu_u^\sigma(x) \cdot g(\sigma_{u\leftarrow x}, T)\right) \\ = &\sum_{x\in Q_u} \left(\mu_u^\sigma(x) \prod_{v\in U\setminus V_\star^{\sigma_{u\leftarrow x}}} (1 - q_v\theta_v) \prod_{c\in E} \left(\alpha^{-1}\mathbb{P}[\neg c \mid \sigma_{u\leftarrow x}](1+\eta)^{Z(\sigma_{u\leftarrow x},c)}\right)\right).\end{aligned}$$

In addition, by T is a generalized $\{2,3\}$-tree and Item 1 of Definition 3.6, we have $\mathsf{vbl}(c) \cap \mathsf{vbl}(c') \neq \emptyset$ for any different $c, c' \in E$. Thus, there exists at most one unique constraint $c_0 \in E$ such that $u \in \mathsf{vbl}(c_0)$. Let $S = E \setminus \{c_0\}$ if $u \in \mathsf{vbl}(E)$ and $S = E$ otherwise. Thus for each $c \in S$, we have $u \notin \mathsf{vbl}(c)$. Then $\mathbb{P}[\neg c \mid \sigma_{u\leftarrow x}] = \mathbb{P}[\neg c \mid \sigma]$ and $Z(\sigma_{u\leftarrow x}, c) = Z(\sigma, c)$ for each $x \in Q_u$. Therefore,

$$\begin{aligned}
&\sum_{x\in Q_u}\left(\mu_u^\sigma(x)\prod_{c\in E}\left(\mathbb{P}[\neg c \mid \sigma_{u\leftarrow x}](1+\eta)^{Z(\sigma_{u\leftarrow x},c)}\right)\right)\\
&=\prod_{c\in S}\left(\mathbb{P}[\neg c \mid \sigma_{u\leftarrow x}](1+\eta)^{Z(\sigma_{u\leftarrow x},c)}\right)\sum_{x\in Q_u}\left(\mu_u^\sigma(x)\prod_{c\in E\setminus S}\left(\mathbb{P}[\neg c \mid \sigma_{u\leftarrow x}](1+\eta)^{Z(\sigma_{u\leftarrow x},c)}\right)\right)\\
&=\prod_{c\in S}\left(\mathbb{P}[\neg c \mid \sigma](1+\eta)^{Z(\sigma,c)}\right)\sum_{x\in Q_u}\left(\mu_u^\sigma(x)\prod_{c\in E\setminus S}\left(\mathbb{P}[\neg c \mid \sigma_{u\leftarrow x}](1+\eta)^{Z(\sigma_{u\leftarrow x},c)}\right)\right).
\end{aligned} \tag{4.33}$$

In addition, by Corollary 2.1 and the assumption that $\mathbb{P}[\neg c \mid \sigma] \le \alpha q$ for all $c \in \mathcal{C}$, we have for each $x \in Q_u$, $\mu_u^\sigma(x) \le q_u^{-1}(1+\eta)$. Therefore,

$$\sum_{x\in Q_u}\left(\mu_u^\sigma(x)\cdot\mathbb{P}[\neg c_0 \mid \sigma_{u\leftarrow x}]\right) \le (1+\eta)\cdot q_u^{-1}\sum_{x\in Q_u}\mathbb{P}[\neg c_0 \mid \sigma_{u\leftarrow x}] = (1+\eta)\cdot\mathbb{P}[\neg c_0 \mid \sigma].$$

Thus, we have

$$\begin{aligned}
&\sum_{x\in Q_u}\left(\mu_u^\sigma(x)\mathbb{P}[\neg c_0 \mid \sigma_{u\leftarrow x}](1+\eta)^{Z(\sigma_{u\leftarrow x},c_0)}\right)\\
&=\sum_{x\in Q_u}\left(\mu_u^\sigma(x)\mathbb{P}[\neg c_0 \mid \sigma_{u\leftarrow x}](1+\eta)^{Z(\sigma,c_0)-1}\right)\\
&\le\mathbb{P}[\neg c_0 \mid \sigma](1+\eta)^{Z(\sigma,c_0)}.
\end{aligned} \tag{4.34}$$

Moreover, we always have

$$\sum_{x\in Q_u}\left(\mu_u^\sigma(x)\prod_{c\in E\setminus S}\left(\mathbb{P}[\neg c \mid \sigma_{u\leftarrow x}](1+\eta)^{Z(\sigma_{u\leftarrow x},c)}\right)\right) \le \prod_{c\in E\setminus S}\left(\mathbb{P}[\neg c \mid \sigma](1+\eta)^{Z(\sigma,c)}\right), \tag{4.35}$$

where we assume that a product over an empty set is 1. Because $E \setminus S$ is either $\{c_0\}$ or an empty set. If $E \setminus S = \{c_0\}$, (4.35) is immediate by (4.34). Otherwise, $E \setminus S = \emptyset$. Then both sides of (4.35) are equal to 1. Combining (4.33) with (4.35), we have

$$\sum_{x\in Q_u}\left(\mu_u^\sigma(x)\prod_{c\in E}\left(\mathbb{P}[\neg c \mid \sigma_{u\leftarrow x}](1+\eta)^{Z(\sigma_{u\leftarrow x},c)}\right)\right) \le \prod_{c\in E}\left(\mathbb{P}[\neg c \mid \sigma](1+\eta)^{Z(\sigma,c)}\right). \tag{4.36}$$

Moreover, by $u = \mathsf{NextVar}(\sigma)$, we have $\sigma(u) = \star \neq \bigstar$. Thus, $u \notin V_\bigstar^\sigma$. Meanwhile, by $\sigma_{u\leftarrow x}(u) = x \neq \bigstar$, we also have $u \notin V_\bigstar^{u\leftarrow x}$ for each $x \in Q_u$. Thus, $U \setminus V_\bigstar^\sigma = U \setminus V_\bigstar^{\sigma_{u\leftarrow x}}$. Combining with (4.32) and (4.36), (4.24) is immediate. This finishes the induction step for the case when $u \notin U$. The lemma is proved. ☐

Combining Lemma 4.9 with the condition that $\sigma \in \mathcal{Q}^*$ is a partial assignment with exactly one variable $v \in V$ having $\sigma(v) = \bigstar$ and comparing (3.9) with (4.23), Lemma 4.8 is proved.

We are now ready to prove Proposition 4.3.

Proof of Proposition 4.3. Let $\tau = \sigma_{v\leftarrow\bigstar}$. If $H(\mathsf{Path}(\tau)) > 0$, we have $\mathbb{1}[f(\tau_\ell) = \mathtt{True}]\cdot\chi(\tau_\ell, \tau_0) > 0$ by (4.18). Combining with $\chi(\tau_\ell, \tau_0) \ge 0$, we have $f(\tau_\ell)$ is true if $H(\mathsf{Path}(\tau)) > 0$. Therefore, by Definition 3.5 we have $|V_\bigstar^{\tau_\ell}| + \Delta\cdot|\mathcal{C}_{\bigstar\text{-frozen}}^{\tau_\ell}| \ge L\Delta$. Combining with Lemma 4.7 we have there always exists a generalized $\{2,3\}$-tree $T = U \circ E$ in H_Φ with some auxiliary tree rooted at v such that $L \le \Delta\cdot|E| + |U| \le L\Delta$ and $\mathcal{E}_T^\tau$ happens. Let $\mathcal{U}$

denote the set $\{Y \in \mathcal{T}_v^t : L \le t \le L\Delta \wedge v \in V\}$, then we have $T \in \mathcal{U}$. In summary, if $H(\mathsf{Path}(\tau)) > 0$, there exists some $T \in \mathcal{U}$ such that $\mathcal{E}_T^\tau$ happens. Therefore by the law of total expectation and the nonnegativity of $H(\mathsf{Path}(\tau))$, we have

$$\mathbb{E}\left[H(\mathsf{Path}(\tau))\right] \le \sum_{T \in \mathcal{U}} \mathbf{Pr}\left[\mathcal{E}_T^\tau\right] \cdot \mathbb{E}\left[H(\mathsf{Path}(\tau)) \mid \mathcal{E}_T^\tau\right]. \tag{4.37}$$

Thus, we have

$$\begin{aligned}
& d_{\mathrm{TV}}(\xi, \mu_v^\sigma) \\
\text{(by Lemmas 4.4 and 4.6)} \quad & \le \mathbb{E}\left[H(\mathsf{Path}(\tau))\right] \\
\text{(by (4.37))} \quad & \le \sum_{T \in \mathcal{U}} \mathbf{Pr}\left[\mathcal{E}_T^\tau\right] \cdot \mathbb{E}\left[H(\mathsf{Path}(\tau)) \mid \mathcal{E}_T^\tau\right] \\
\text{(by Lemma 4.8)} \quad & \le \sum_{T \in \mathcal{U}} F(\tau, T \setminus \{v\}) \\
& \le \sum_{i=L}^{L\Delta} \sum_{v \in V} \sum_{T \in \mathcal{T}_v^i} F(\tau, T \setminus \{v\}) \\
\text{(by (3.12))} \quad & = \digamma(\tau) \\
& = \digamma(\sigma).
\end{aligned}$$

□

4.5 Efficiency of the counting algorithm We then show the efficiency of the algorithm, given that the upper bound function $\digamma(\cdot)$ is small. Recall the definition of X^n in Definition 4.1. We will show two crucial propositions, namely Proposition 4.4 and Proposition 4.5. Proposition 4.4 bounds the running time of the marginal approximator subroutine called within the main counting algorithm, and Proposition 4.5 the running time on the exhaustive enumeration part in the main counting algorithm if $\digamma(X^n)$ is small.

PROPOSITION 4.4. *For any (Φ, σ, v) satisfying Condition 4.1, let $T_{\mathsf{MA}}(\Phi, \sigma, v)$ denote the running time of* MarginalApproximator(Φ, σ, v). *Then $T_{\mathsf{MA}}(\Phi, \sigma, v) \le \mathrm{poly}(n, q^{kL\Delta^2})$.*

PROPOSITION 4.5. *If $\digamma(X^n) < 1$, then $T_{\mathsf{Enu}}(\Phi, X^n) = \mathrm{poly}(n, q^{kL\Delta^2})$, where $T_{\mathsf{Enu}}(\Phi, X^n)$ denotes the running time of the exhaustive enumeration in the main counting algorithm.*

For each $\sigma \in \mathcal{Q}^*$ and $v \in V^\sigma$, $H_v^\sigma = (V_v^\sigma, \mathcal{C}_v^\sigma)$ denotes the connected component in H^σ that contains the vertex/variable v, where H^σ is the hypergraph representation for the CSP formula Φ^σ obtained from the simplification of Φ under σ.

We further stipulate that $H_v^\sigma = (V_v^\sigma, \mathcal{C}_v^\sigma) = (\emptyset, \emptyset)$ is the empty hypergraph when $v \in \Lambda(\sigma)$.

Recall that $\mathcal{T}_\sigma = (T_\sigma, \rho_\sigma)$ is the RCT rooted at σ. To show Proposition 4.4, we need the following lemma, which bounds the efficiency of the subroutine RecursiveApproximator.

LEMMA 4.10. *Let (Φ, σ, v) be the input to* RecursiveApproximator *(Algorithm 2) satisfying Condition 4.2, and let $T_{\mathsf{RA}}(\Phi, \sigma, v)$ denote the running time of* RecursiveApproximator(Φ, σ, v). *It holds that*

$$T_{\mathsf{RA}}(\Phi, \sigma, v) \le |\mathcal{T}_\sigma| \cdot \mathrm{poly}(n, \Delta, q^k) + O\left(\sum_{\tau \in \mathcal{L}_g(\mathcal{T}_\sigma)} \left(\left(k\,|\mathcal{C}_v^\tau| + q\,|V_v^\tau|\right) \cdot q^{|V_v^\tau|}\right)\right).$$

The following lemma will be used in the proof of Lemma 4.10. Its proof is similar to that of Lemma 4.1-(2) and omitted here.

LEMMA 4.11. *Let (Φ, σ, v) satisfy Condition 4.2. Then for each node τ in T_σ, (Φ, τ, v) also satisfies Condition 4.2.*

By Proposition 4.1 and Lemma 4.11 , we have the following lemma.

LEMMA 4.12. *Let (Φ, σ, v) satisfy Condition 4.2. Recall that $\mathcal{T}_\sigma = (T_\sigma, \rho_\sigma)$ is the RCT rooted at σ. For any leaf node τ in T_σ, let P be the path from σ to τ in T_σ. Then $\mathbf{Pr}[\mathsf{Path}(\sigma) = P] > 0$.*

Proof. Let $(\tau_0 = \sigma, \tau_1, \tau_2, \tau_3, \cdots, \tau_r = \tau)$ be the path from σ to τ in T_σ. Given $0 \le i < r$, let $u_i = \mathsf{NextVar}(\tau_i)$. Then we have $f(\tau_i) \neq \texttt{True}$ and $u_i \neq \perp$. Otherwise, by Item 2a of Definition 4.2, we have τ_i is a leaf of T_σ, which is contradictory with $i < r$. By Item 2b of Definition 4.2, we also have $\tau_{i+1} \in \{(\tau_i)_{u_i \leftarrow x} \mid x \in \mathcal{Q}^\star_{u_i}\}$. In addition, by Lemma 4.11, we have (Φ, τ_i, v) satisfies Condition 4.2. Combining with Proposition 4.1, we have $\mu^{\tau_i}_{u_i}(x) \ge \theta_{u_i}$ for each $x \in \mathcal{Q}^\star_{u_i}$. Thus, combining $f(\tau_i) \neq \texttt{True}$, $u_i = \mathsf{NextVar}(\tau_i) \neq \perp$, $\tau_{i+1} \in \{(\tau_i)_{u_i \leftarrow x} \mid x \in \mathcal{Q}^\star_{u_i}$ with Definition 4.3, we have

$$\begin{aligned}\mathbf{Pr}[\sigma_{i+1} = \tau_{i+1} \mid \sigma_0 = \tau_0, \cdots, \sigma_i = \tau_i] &\ge (2 - q_{u_i}\theta_{u_i})^{-1} \min\left\{1 - q_{u_i}\theta_{u_i}, \min_{x \in Q_{u_i}} \left\{\mu^{\tau_i}_{u_i}(x)\right\}\right\} \\ &\ge (2 - q_{u_i}\theta_{u_i})^{-1} \min\{1 - q_{u_i}\theta_{u_i}, \theta_{u_i}\} > 0,\end{aligned}$$

where the second inequality is by that $\mu^{\tau_i}_{u_i}(x) \ge \theta_{u_i}$ for each $x \in \mathcal{Q}^\star_{u_i}$. Thus, the lemma is immediate by the chain rule. □

By [17, Proposition 6.28], the following lemma is immediate.

LEMMA 4.13. *For any $\sigma \in \mathcal{Q}^*$, $\mathsf{NextVar}(\sigma)$ and $f(\sigma)$ can be computed in $\mathrm{poly}(n, \Delta, q^k)$ cost.*

Now we can prove Lemma 4.10.

Proof of Lemma 4.10. We prove this lemma by an induction on the structure of RCT. The base case is when T_σ is just a single root, in which case $\mathsf{NextVar}(\sigma) = \perp$ or $f(\sigma) = \texttt{True}$. If $f(\sigma) = \texttt{True}$, the condition in Line 1 of $\mathsf{RecursiveApproximator}(\Phi, \sigma, v)$ is not satisfied and Lines 3-13 are omitted. Thus, we have $T_{\mathsf{RA}}(\Phi, \sigma, v) \le \mathrm{poly}(n, \Delta, q^k)$. Otherwise, $f(\sigma) = \texttt{False}$ and $\mathsf{NextVar}(\sigma) = \perp$. By T_σ is just a single root and $f(\sigma) = \texttt{False}$, we have $\sigma \in \mathcal{L}_g(\mathcal{T}_\sigma)$. By $\mathsf{NextVar}(\sigma) = \perp$, we have the condition in Line 5 of $\mathsf{RecursiveApproximator}(\Phi, \sigma, v)$ is not satisfied and Lines 3-12 are omitted. Combining with Lemma 4.13, we have

$$\begin{aligned}T_{\mathsf{RA}}(\Phi, \sigma, v) &\le \mathrm{poly}(n, \Delta, q^k) + O\left((k\,|\mathcal{C}^\tau_v| + q\,|V^\tau_v|) \cdot q^{|V^\tau_v|}\right) \\ &\le \mathrm{poly}(n, \Delta, q^k) + O\left(\sum_{\tau \in \mathcal{L}_g(\mathcal{T}_\sigma)} \left((k\,|\mathcal{C}^\tau_v| + q\,|V^\tau_v|) \cdot q^{|V^\tau_v|}\right)\right),\end{aligned}$$

where the first inequality is by (Φ, σ, v) satisfies Condition 4.2 and the standard guarantee on the running time of exhaustive enumeration, and the last inequality is by $\sigma \in \mathcal{L}_g(\mathcal{T}_\sigma)$. The base case is proved.

For the induction step, we assume that T_σ is a tree of depth > 0. Thus by Definition 4.2, $f(\sigma) = \texttt{False}$ and $\mathsf{NextVar}(\sigma) = u \neq \perp$ for some $u \in V$. Thus the condition in Line 1 of $\mathsf{RecursiveApproximator}(\Phi, \sigma, v)$ is not satisfied and the condition in Line 5 is satisfied. According to Lines 6-16, one can verify that

$$T_{\mathsf{RA}}(\Phi, \sigma, v) \le \mathrm{poly}(n, \Delta, q^k) + T_{\mathsf{RA}}(\Phi, \sigma_{u \leftarrow \star}, u) + \sum_{x \in Q_u} T_{\mathsf{RA}}(\Phi, \sigma_{u \leftarrow x}, v). \tag{4.38}$$

Let $S \triangleq \{\sigma_{u \leftarrow x} : x \in Q_u \cup \{\star\}\}$. By the induction hypothesis, we have

$$\begin{aligned}&T_{\mathsf{RA}}(\Phi, \sigma_{u \leftarrow \star}, u) + \sum_{x \in Q_u} T_{\mathsf{RA}}(\Phi, \sigma_{u \leftarrow x}, v) \\ &\le \sum_{\tau \in S}\left(|\mathcal{T}_\tau| \cdot \mathrm{poly}(n, \Delta, q^k) + O\left(\sum_{X \in \mathcal{L}_g(\mathcal{T}_\tau)} \left((k\,|\mathcal{C}^X_v| + q\,|V^X_v|) \cdot q^{|V^X_v|}\right)\right)\right).\end{aligned} \tag{4.39}$$

Meanwhile, by Definition 4.2, one can verify that T_σ is a tree consisting of a root σ and $q_v + 1$ subtrees T_τ where $\tau \in S$. Thus, we have $1 + \sum_{\tau \in S} |\mathcal{T}_\tau| = |\mathcal{T}_\sigma|$ and

$$\bigcup_{\tau \in S} \mathcal{L}_g(\mathcal{T}_\tau) = \mathcal{L}_g(\mathcal{T}_\sigma).$$

Moreover, it is easy to verify that $\mathcal{L}_g(\mathcal{T}_\tau) \cap \mathcal{L}_g(\mathcal{T}_{\tau'}) = \emptyset$ for different $\tau, \tau' \in S$. Combing with (4.39), we have

$$\begin{aligned}
&T_{\mathsf{RA}}(\Phi, \sigma_{u\leftarrow\star}, u) + \sum_{x\in Q_u} T_{\mathsf{RA}}(\Phi, \sigma_{u\leftarrow x}, v) \\
&\leq (|\mathcal{T}_\sigma| - 1)\cdot \mathrm{poly}(n, \Delta, q^k) + O\left(\sum_{\tau\in\mathcal{L}_g(\mathcal{T}_\sigma)} \left((k\,|\mathcal{C}_v^\tau| + q\,|V_v^\tau|)\cdot q^{|V_v^\tau|}\right)\right).
\end{aligned}$$

Combining with (4.38), we have

$$T_{\mathsf{RA}}(\Phi, \sigma, v) \leq |\mathcal{T}_\sigma|\cdot \mathrm{poly}(n, \Delta, q^k) + O\left(\sum_{\tau\in\mathcal{L}_g(\mathcal{T}_\sigma)} \left((k\,|\mathcal{C}_v^\tau| + q\,|V_v^\tau|)\cdot q^{|V_v^\tau|}\right)\right),$$

which finishes the proof of the induction step. Then the lemma is immediate. □

Let $\sigma \in \mathcal{Q}^*$ be a partial assignment such that only one variable $v \in V$ has $\sigma(v) = \star$. The following lemma shows that the path $\mathsf{Path}(\sigma) = (\sigma_0, \sigma_1, \ldots, \sigma_\ell)$ generated from such σ cannot be too long. Its proof is deferred to Appendix A.2.

LEMMA 4.14. *Let $\sigma \in \mathcal{Q}^*$ be a partial assignment with exactly one variable $v \in V$ having $\sigma(v) = \star$, and let $\mathsf{Path}(\sigma) = (\sigma_0, \sigma_1, \ldots, \sigma_\ell)$. Then it always holds that*

$$\ell \leq kL\Delta^2.$$

The following lemma further relates the size of $\mathcal{C}_v^{\sigma_\ell}$ with the sizes of $V_\star^{\sigma_\ell}$ and $\mathcal{C}_{\star\text{-frozen}}^{\sigma_\ell}$. It is formally proved in Appendix A.3.

LEMMA 4.15. *Let $\sigma \in \mathcal{Q}^*$ be a partial assignment with exactly one variable $v \in V$ having $\sigma(v) = \star$. Let $\mathsf{Path}(\sigma) = (\sigma_0, \sigma_1, \cdots, \sigma_\ell)$. If $\mathsf{NextVar}(\sigma_\ell) = \perp$, we have $|\mathcal{C}_v^{\sigma_\ell}| \leq \Delta \cdot |\mathcal{C}_{\star\text{-frozen}}^{\sigma_\ell}| + \Delta \cdot |V_\star^{\sigma_\ell}| \leq L\Delta^2$.*

Combining Lemmas 4.12, 4.14 and 4.15 , we have the following corollary.

COROLLARY 4.2. *Let (Φ, σ, v) satisfy Condition 4.2 and v is the only vertex in V with $\sigma(v) = \star$. Recall that $\mathcal{T}_\sigma = (T_\sigma, \rho_\sigma)$ is the RCT rooted at σ. For each leaf τ in T_σ, we have the depth of τ in T_σ is no more than $kL\Delta^2$. Moreover, if $\tau \in \mathcal{L}_g(\mathcal{T}_\sigma)$, then $|\mathcal{C}_v^\tau| \leq \Delta \cdot |\mathcal{C}_{\star\text{-frozen}}^\tau| + \Delta \cdot |V_\star^\tau| \leq L\Delta^2$.*

Now we can prove Proposition 4.4.

Proof of Proposition 4.4. Given (Φ, σ, v) satisfying Condition 4.1, by Algorithm 1 the nontrivial costs in $\mathsf{MarginalApproximator}(\Phi, \sigma, v)$ is to calculate $\mathsf{RecursiveApproximator}(\Phi, \sigma_{v\leftarrow\star}, v)$. Thus, to prove the proposition, it is sufficient to prove that $T_{\mathsf{RA}}(\Phi, \sigma_{v\leftarrow\star}, v) \leq \mathrm{poly}(n, q^{kL\Delta^2})$. Let $\tau = \sigma_{v\leftarrow\star}$. By (Φ, σ, v) satisfies Condition 4.1, we have (Φ, τ, v) satisfies Condition 4.2 and v is the only vertex in V with $\tau(v) = \star$. Thus we have

$$\begin{aligned}
&\sum_{X\in\mathcal{L}_g(\mathcal{T}_\tau)} \left((k\,|\mathcal{C}_v^X| + q\,|V_v^X|)\cdot q^{|V_v^X|}\right) \\
&\leq \sum_{X\in\mathcal{L}_g(\mathcal{T}_\tau)} \left((k\,|\mathcal{C}_v^X| + qk\,|\mathcal{C}_v^X| + q)\cdot q^{k|\mathcal{C}_v^X|+1}\right) \\
&\leq \sum_{X\in\mathcal{L}_g(\mathcal{T}_\tau)} \left((kL\Delta^2 + qkL\Delta^2)\cdot q^{2kL\Delta^2}\right) \\
&\leq |\mathcal{T}_\tau| \cdot (kL\Delta^2 + 2qkL\Delta^2)\cdot q^{2kL\Delta^2},
\end{aligned}$$

where the first inequality is by $|V_v^\tau| \leq k\,|\mathcal{C}_v^\tau| + 1$, the second one is by Corollary 4.2, and the last one is by $\mathcal{L}_g(\mathcal{T}_\tau) \leq |\mathcal{T}_\tau|$. Combining with Lemma 4.10, we have

$$T_{\mathsf{RA}}(\Phi, \tau, v) \leq |\mathcal{T}_\tau|\cdot \mathrm{poly}(n, \Delta, q^k) + |\mathcal{T}_\tau|\cdot O\left((kL\Delta^2 + 2qkL\Delta^2)\cdot q^{2kL\Delta^2}\right) \leq |\mathcal{T}_\tau|\cdot \mathrm{poly}(n, q^{kL\Delta^2}).$$

By Corollary 4.2, we have the depth of T_τ is at most $kL\Delta^2$. In addition, we have

$$|\mathcal{T}_\tau| \le \sum_{i=0}^{kL\Delta^2} (1+q)^i \le 2(1+q)^{kL\Delta^2}$$

by T_τ is a tree where each node has at most $q+1$ children, Therefore, we have

$$T_{\mathsf{RA}}(\Phi,\tau,v) \le |\mathcal{T}_\tau| \cdot \mathrm{poly}(n, q^{kL\Delta^2}) \le \mathrm{poly}(n, q^{kL\Delta^2}),$$

which finishes the proof. □

Recall the definition of X^n in Definition 4.1. We need the following lemma, which is an analogy of Lemma 4.7, whose proof is deferred to Appendix A.4.

LEMMA 4.16. *For every $v \in V$, there exists a generalized $\{2,3\}$-tree $T = \{v\} \circ E$ in H_Φ where $E \subseteq \mathcal{C}^{X^n}_{\mathsf{frozen}}$ and $\Delta^2 |E| \ge \left|\mathcal{C}^{X^n}_v\right|$. In addition, if $\left|\mathcal{C}^{X^n}_v\right| \ge L\Delta^2$, then there exists a generalized $\{2,3\}$-tree $T = \{v\} \circ E$ in H_Φ with some auxiliary tree rooted at v where $E \subseteq \mathcal{C}^{X^n}_{\mathsf{frozen}}$ and $L \le 1 + \Delta \cdot |E| \le L\Delta$.*

We are now ready to prove Proposition 4.5.

Proof of Proposition 4.5. For simplification, let $\tau = X^n$. Let $\{(V_i^\tau, \mathcal{C}_i^\tau) \mid 1 \le i \le K\}$ denote all connected components in H_{Φ^τ}. By the standard guarantee on the running time of exhaustive enumeration, it is sufficient to show that $|V_i^\tau| \le kL\Delta^2 + 1$ for each $i \le K$. Assume for contradiction that there exists some $i \in [K]$ such that $|V_i^\tau| > kL\Delta^2 + 1$. Then we have $|\mathcal{C}_v^\tau| > L\Delta^2$ by $|V_v^\tau| \le k\,|\mathcal{C}_v^\tau| + 1$. Combining with Lemma 4.16, we have there exists a generalized $\{2,3\}$-tree $T = \{v\} \circ E$ in H_Φ with some auxiliary tree rooted at v where $E \subseteq \mathcal{C}^\tau_{\mathsf{frozen}}$ and $L \le 1 + \Delta \cdot |E| \le L\Delta$. Let $\mathcal{U}$ denote the set $\{Y \in \mathcal{T}_v^t : L \le t \le L\Delta\}$, then we have $T \in \mathcal{U}$. Therefore, we have

$$\begin{aligned}
& F(\tau) \\
\text{(by (3.12))} \quad & = \sum_{i=L}^{L\Delta} \sum_{v \in V} \sum_{S \in \mathcal{T}_v^i} F(\tau, S \setminus \{v\}) \\
\text{(by } F(\cdot,\cdot) \ge 0, \text{ and } T \in \mathcal{U}) \quad & \ge F(\tau, T \setminus \{v\}) \\
\text{(by (3.9))} \quad & = \prod_{c \in E} \alpha^{-1} \mathbb{P}[\neg c \mid \tau](1+\eta)^k \\
\text{(by } (1+\eta)^k > 1) \quad & \ge \prod_{c \in E} \left(\alpha^{-1} \mathbb{P}[\neg c \mid \tau]\right) \\
\text{(by } E \subseteq \mathcal{C}^\tau_{\mathsf{frozen}}) \quad & \ge 1,
\end{aligned}$$

which is contradictory with $F(\tau) < 1$. Thus, we have $|V_i^\tau| \le kL\Delta^2 + 1$ for each $i \le K$ and the lemma is immediate. □

4.6 Analysis of the main counting algorithm We are now ready to present the analysis for the main counting algorithm. Recall the sequence of partial assignments $X^0, X^1, \ldots, X^n$ that evolve in the main counting algorithm defined in Definition 4.1.

A crucial lemma we will show is the following, which states $F(X)$ is small throughout the main counting algorithm.

LEMMA 4.17. *If $16\mathrm{e}p\Delta^3 \le \alpha$, $\eta \le (2k)^{-1}$, $1 - q\theta \le (8\mathrm{e}k\Delta)^{-1}$ and $L \ge 9$, then it holds for all $0 \le i \le n$ that*

$$F(X^i) < 8n\Delta \cdot 2^{-\lfloor \frac{L}{2\Delta} \rfloor}.$$

We first show the following lemma, which states that $F(\star^V)$ is small.

LEMMA 4.18. *Under the condition of Lemma 4.17,*

$$F(\star^V) < 8n\Delta \cdot 2^{-\lfloor \frac{L}{2\Delta} \rfloor}.$$

Proof. Recall by (3.11) and (3.12) that

$$F(\star^V) = \sum_{i=L}^{L\Delta} \sum_{v \in V} \sum_{T \in \mathcal{T}_v^i} \left((1-q\theta)^{|U|-1} \cdot \left(p\alpha^{-1}(1+\eta)^k\right)^{|E|} \right) \tag{4.40}$$

By (3.11), in order to bound the term in $F(\star^V)$, instead of bounding (4.40) over all possible generalized $\{2,3\}$-tree T, it is sufficient to bound the term in (4.40) over all possible auxiliary trees T^*. This is because distinct generalized $\{2,3\}$-trees must have distinct auxiliary trees by Definition 3.6.

Fix any $v \in V$. We define the following multi-type Galton-Watson process that generates a rooted tree such that each possible auxiliary trees of some generalized $\{2,3\}$-tree rooted at v appears as a subgraph with the corresponding probability.

Definition 4.4. (A multi-type Galton-Watson process) *For each $v \in V$, we define the following multi-type Galton-Watson process that generates a rooted directed tree T^* with vertex set $V(T^*) \subseteq V \cup \mathcal{C}$ that generates all possible auxiliary trees of generalized $\{2,3\}$-trees rooted at v as a subgraph with the corresponding probability:*

1. *The root of T^* is v and the depth of v is 0.*
2. *For $i = 0, 1, \ldots,$: for all nodes $v \in V(T^*)$ of depth i in the current T^**
   - (a) *If $v \in V$:*
     - i. *For each $u \in V$ such that there exists $c \in \mathcal{C}$ where $v, u \in \mathsf{vbl}(c)$, add u as a child of v independently with probability $1 - q\theta$. Note that there are at most $k\Delta$ such u.*
     - ii. *For each $c \in \mathcal{C}$ such that there exists $c' \in \mathcal{C}$ where $v \in \mathsf{vbl}(c') \wedge dist_{G(\mathcal{C})}(c, c') = 1$, add c as a child of v independently with probability $p\alpha^{-1}(1+\eta)^k$. Note that there are at most Δ^2 such c.*
   - (b) *If $v \in \mathcal{C}$:*
     - i. *For each $u \in V$ such that there exists $c \in \mathcal{C}$ where $u \in \mathsf{vbl}(c) \wedge dist_{G(\mathcal{C})}(v, c) = 1$ or 2, add u as a child of v independently with probability $1 - q\theta$. Note that there are at most $k\Delta^2$ such u.*
     - ii. *For each $c \in \mathcal{C}$ such that $dist_{G(\mathcal{C})}(c, c') = 2$ or 3, add c as a child of v independently with probability $p\alpha^{-1}(1+\eta)^k$. Note that there are at most Δ^3 such c.*

Note that the process in Definition 4.4 may generate trees that violate the rule of auxiliary trees, as vertices may be repeated and constraints in the tree may not be pairwise disjoint. Nevertheless, by comparing with Definition 3.6, it can be verified that this process generates all possible auxiliary trees T rooted at v of some generalized $\{2,3\}$-tree as a subgraph with probability exactly $F(\star^V, V(T))$. By (3.12), it then suffices to bound the sum of the probability such process generating a tree T as a subgraph over all possible auxiliary trees T rooted at v satisfying $L \le |V(T^*) \cap V| + \Delta \cdot |V(T^*) \cap \mathcal{C}| \le L\Delta$. However, this is still not convenient enough for calculation, so we further define the following two-type Galton-Watson process.

Definition 4.5. (A two-type Galton-Watson process) *We define the following multi-type Galton-Watson process that generates a tree T with vertex set $V(T)$ consisting nodes of two types.*

1. *The root of T, r_T, is of type 1 or type 2, and of depth 0.*
2. *For $i = 0, 1, \ldots,$: for all nodes $v \in V(T)$ of depth i in the current T*
   - (a) *If v is of type 1:*
     - i. *Independently repeat $k\Delta$ times: generate a node of type 1 as the child of v with probability $1 - q\theta$.*
     - ii. *Independently repeat Δ^2 times: generate a node of type 2 as the child of v with probability $p\alpha^{-1}(1+\eta)^k$.*
   - (b) *If v is of type 2:*
     - i. *Independently repeat $k\Delta^2$ times: generate a node of type 1 as the child of v with probability $1 - q\theta$.*

ii. *Independently repeat Δ^3 times: generate a node of type 2 as the child of v with probability $p\alpha^{-1}(1+\eta)^k$.*

It is easy to construct an injection between each tree generated by the process in Definition 4.4 and each tree generated by the process in Definition 4.5 with the same probability. Then it is sufficient to bound the sum of probability of the process in Definition 4.5 generating a tree T as a subgraph over all T satisfying $L \le a+\Delta\cdot b \le L\Delta$ with root r_T, where a and b represent the number of type 1 nodes and type 2 nodes in $V(T)$, respectively.

If r_T is of type 1, let $f_1(x)$ be the generating function for the random tree generated in the process in Definition 4.5 where for each $m \ge 0$, the m-th coefficient $[x^m]f_1(x)$ represents the sum of probability that the process in Definition 4.5 generates a tree T as a subgraph over all directed tree T rooted at r_T satisfying $a + \Delta \cdot b = m$, where a, b are the numbers of type 1 and type 2 nodes in $V(T)$, respectively. Otherwise, r_T is of type 2, and we define $f_2(x)$ similarly. Let $p_1 = 1 - q\theta$ and $p_2 = p\alpha^{-1} \cdot (1+\eta)^k$. By Definition 4.5 we have

$$f_1 = x(1+p_1f_1)^{k\Delta} \cdot (1+p_2f_2)^{\Delta^2},$$
$$f_2 = x^{\Delta}(1+p_1f_1)^{k\Delta^2} \cdot (1+p_2f_2)^{\Delta^3}.$$

Thus, we have $f_2 = f_1^{\Delta}$ and then

$$f_1 = x(1+p_1f)^{k\Delta} \cdot (1+p_2f_1^{\Delta})^{\Delta^2}.$$

Let g be the functional inverse of f_1. Formally, $g(f_1(x)) = f_1(g(x)) = x$. By $f_1(0) = 0$, we also have $g(0) = 0$. In addition, by $f_1(x)$ is a probability generating function, we have $[x^n]f_1(x) \le 1$. Therefore,

$$\lim_{y\to 0} \frac{f_1(y) - f_1(0)}{y - 0} = [x]f_1(x) \le 1.$$

Thus, we have

$$g'(0) = \lim_{u\to 0} \frac{g(u) - 0}{u - 0} = \lim_{f_1(y)\to 0} \frac{y - 0}{f_1(y) - 0} = \lim_{y\to 0} \frac{y - 0}{f_1(y) - 0} \ne 0,$$

where the last equality is by $f_1(0) = 0$.

By $g(0) = 0$ and $g'(0) \ne 0$, the condition of Lagrange inversion theorem is satisfied. By applying the theorem we have for $m \ge 1$,

$$\begin{aligned}
[x^m]f_1(x) &= \frac{1}{m}[u^{m-1}]\left(\frac{u}{g(u)}\right)^m \\
&= \frac{1}{n}[u^{m-1}]\left((1+p_1u)^{k\Delta} \cdot (1+p_2u^{\Delta})^{\Delta^2}\right)^m \\
&= \frac{1}{m}\sum_{i=0}^{\lfloor \frac{m}{\Delta} \rfloor}\left([u^{(m-1-\Delta i)}]\left(1+p_1u\right)^{k\Delta m} \cdot [u^{(\Delta i)}]\left(1+p_2u^{\Delta}\right)^{\Delta^2 m}\right) \\
&\le \frac{1}{m}\left(\sum_{i=0}^{\lfloor \frac{m}{2\Delta} \rfloor}[u^{(m-1-\Delta i)}]\left(1+p_1u\right)^{k\Delta m} + \sum_{i=\lfloor \frac{m}{2\Delta} \rfloor+1}^{\lfloor \frac{m}{\Delta} \rfloor}[u^{(\Delta i)}]\left(1+p_2u^{\Delta}\right)^{\Delta^2 m}\right)
\end{aligned} \tag{4.41}$$

In addition, let $t = m - 1 - \Delta i$. For each $i \in \lfloor \frac{m}{2\Delta} \rfloor$, we have

$$[u^t]\left(1+p_1u\right)^{mk\Delta} = p_1^t \cdot \binom{mk\Delta}{t}$$

Moreover, for each $m \ge 8$ and $i \in \lfloor \frac{m}{2\Delta} \rfloor$, we have $mk\Delta \le 1.2(m-1)k\Delta \le 4tk\Delta$. Thus

$$\binom{mk\Delta}{t} \le \binom{4tk\Delta}{t} \le \left(\frac{4\mathrm{e}tk\Delta}{t}\right)^t \le \left(4\mathrm{e}k\Delta\right)^t,$$

where the second inequality is by that for each $0 < \gamma \leq \beta$ where γ, β are integers,

$$\binom{\beta}{\gamma} \leq \left(\frac{\mathrm{e}\beta}{\gamma}\right)^{\gamma}.$$

Thus, we have

$$[u^t]\left(1+p_1 u\right)^{mk\Delta} = p_1^t \cdot \binom{nk\Delta}{t} \leq p_1^t \left(4\mathrm{e}k\Delta\right)^t = \left(4\mathrm{e}p_1 k\Delta\right)^t. \tag{4.42}$$

Similarly, for each $\lfloor \frac{m}{2\Delta} \rfloor < i \leq \lfloor \frac{m}{\Delta} \rfloor$, we have

$$[u^{\Delta i}]\left(1+p_2 u^{\Delta}\right)^{m\Delta^2} = p_2^i \cdot \binom{m\Delta^2}{i}.$$

Moreover, for each $m \geq 8$ and $\lfloor \frac{m}{2\Delta} \rfloor < i \leq \lfloor \frac{m}{\Delta} \rfloor$, we have $m\Delta^2 \leq 2i\Delta^3$. Thus

$$\binom{m\Delta^2}{i} \leq \binom{2i\Delta^3}{i} \leq \left(\frac{2\mathrm{e}i\Delta^3}{i}\right)^i = \left(2\mathrm{e}\Delta^3\right)^i,$$

where the second inequality is also by that for each $0 < \gamma \leq \beta$ where γ, β are integers,

$$\binom{\beta}{\gamma} \leq \left(\frac{\mathrm{e}\beta}{\gamma}\right)^{\gamma}.$$

Thus, we have

$$[u^{\Delta i}]\left(1+p_2 u^{\Delta}\right)^{m\Delta^2} = p_2^i \cdot \binom{m\Delta^2}{i} \leq p_2^i \left(2\mathrm{e}\Delta^3\right)^i = \left(2\mathrm{e}p_2\Delta^3\right)^i.$$

Combining with (4.41) and (4.42), we have

$$[x^m]f_1(x) \leq m^{-1}\left(\sum_{i=0}^{\lfloor \frac{m}{2\Delta} \rfloor} \left(4\mathrm{e}k\Delta\right)^{m-1-\Delta i} + \sum_{i=\lfloor \frac{m}{2\Delta} \rfloor+1}^{\lfloor \frac{m}{\Delta} \rfloor} \left(2\mathrm{e}p_2\Delta^3\right)^i\right)$$

Note that by $16\mathrm{e}p\Delta^3 \leq \alpha$, $\eta \leq (2k)^{-1}$ and $1 - q\theta \leq (8\mathrm{e}k\Delta)^{-1}$ we have $p_1 \leq (8\mathrm{e}k\Delta)^{-1}$ and $p_2 \leq (8\mathrm{e}\Delta^3)^{-1}$, hence we have for each $m \geq 8$,

$$[x^m]f_1(x) \leq m^{-1}\left(\sum_{i=0}^{\lfloor \frac{m}{2\Delta} \rfloor} 2^{(\Delta i+1-m)} + \sum_{i=\lfloor \frac{m}{2\Delta} \rfloor+1}^{\lfloor \frac{m}{\Delta} \rfloor} 2^{-i}\right) \leq m^{-1}\left(2^{\left(2-\frac{m}{2}\right)} + 2^{-\lfloor \frac{m}{2\Delta} \rfloor}\right) \leq 2^{-\lfloor \frac{m}{2\Delta} \rfloor}.$$

Therefore by the analysis above we have

$$\digamma(\star^V) \leq n \cdot \sum_{i=L}^{L\Delta} 2^{-\lfloor \frac{i-1}{2\Delta} \rfloor} \leq 8n\Delta \cdot 2^{-\lfloor \frac{L}{2\Delta} \rfloor}.$$

□

The following lemma is a property for the upper bound function $\digamma(\cdot)$ defined in Definition 3.7 .

LEMMA 4.19. *For any partial assignment $\sigma \in \mathcal{Q}^*$, any variable $u \in V$ with $\sigma(u) = \star$, we have*

$$\sum_{x \in Q_u} \digamma(\sigma_{u \leftarrow x}) = q_u \cdot \digamma(\sigma).$$

Proof. Given $T = U \circ E \in \mathcal{T}_v^i$ for any $L \le i \le L\Delta$ and $v \in V$, by the definition of $\mathcal{T}_v^i$, we have $\mathsf{vbl}(c) \cap \mathsf{vbl}(c') \neq \emptyset$ for any different $c, c' \in E$. Then there exists at most one unique constraint $c_0 \in E$ such that $u \in \mathsf{vbl}(c_0)$. Let $S = T \setminus \{c_0\}$ if $u \in \mathsf{vbl}(E)$ and $S = T$ otherwise. Thus for each $c \in S \cap \mathcal{C}$, we have $u \notin \mathsf{vbl}(c)$. Then $\mathbb{P}[\neg c \mid \sigma_{u\leftarrow x}] = \mathbb{P}[\neg c \mid \sigma]$ and $F(\sigma_{u\leftarrow x}, c) = F(\sigma, c)$ for each $x \in Q_u$. Therefore,

$$F(\sigma_{u\leftarrow x}, S) = (1-q\theta)^{|S\cap V|} \prod_{c\in S\cap\mathcal{C}} F(\sigma_{u\leftarrow x}, c) = (1-q\theta)^{|S\cap V|} \prod_{c\in S\cap\mathcal{C}} F(\sigma_u, c) = F(\sigma, S).$$

Note that by the definition of $F(\cdot,\cdot)$ it is easy to verify that for any partial assignment $\tau \in \mathcal{Q}^*$ and any two non-intersecting subsets $S_1, S_2 \subseteq V \cup \mathcal{C}$ we have

$$F(\tau, S_1 \cup S_2) = F(\tau, S_1) \cdot F(\tau, S_2).$$

Thus we have

$$\sum_{x\in Q_u} F(\sigma_{u\leftarrow x}, T) = \sum_{x\in Q_u} \left(F(\sigma_{u\leftarrow x}, T\setminus S) \cdot F(\sigma_{u\leftarrow x}, S)\right) = \left(\sum_{x\in Q_u} F(\sigma_{u\leftarrow x}, T\setminus S)\right) F(\sigma, S).$$

In addition, we claim that

$$\sum_{x\in Q_u} F(\sigma_{u\leftarrow x}, T\setminus S) = q_u F(\sigma, T\setminus S). \tag{4.43}$$

Thus, we have

$$\sum_{x\in Q_u} F(\sigma_{u\leftarrow x}, T) = q_u F(\sigma, T\setminus S) F(\sigma, S) = q_u F(\sigma, T).$$

Therefore, combining with (3.12) we have

$$\begin{aligned}
\sum_{x\in Q_u} F(\sigma_{u\leftarrow x}) &= \sum_{x\in Q_u} \sum_{i=L}^{L\Delta} \sum_{v\in V} \sum_{T\in\mathcal{T}_v^i} F(\sigma_{u\leftarrow x}, T\setminus\{v\}) \\
&= \sum_{i=L}^{L\Delta} \sum_{v\in V} \sum_{T\in\mathcal{T}_v^i} \sum_{x\in Q_u} F(\sigma_{u\leftarrow x}, T\setminus\{v\}) \\
&= q_u \sum_{i=L}^{L\Delta} \sum_{v\in V} \sum_{T\in\mathcal{T}_v^i} F(\sigma, T\setminus\{v\}) \\
&= q_u \cdot F(\sigma).
\end{aligned}$$

In the following, we prove (4.43). If $S = T \setminus \{c_0\}$, by $\sigma(u) = \star$ we have

$$\sum_{x\in Q_u} \mathbb{P}[\neg c \mid \sigma_{u\leftarrow x}] = q_u \mathbb{P}[\neg c \mid \sigma].$$

Thus

$$\begin{aligned}
\sum_{x\in Q_u} F(\sigma_{u\leftarrow x}, T\setminus S) &= \sum_{x\in Q_u} F(\sigma_{u\leftarrow x}, c_0) \\
&= \sum_{x\in Q_u} \left(\alpha^{-1}\mathbb{P}[\neg c_0 \mid \sigma_{u\leftarrow x}](1+\eta)^k\right) \\
&= \alpha^{-1}(1+\eta)^k \sum_{x\in Q_u} \mathbb{P}[\neg c_0 \mid \sigma_{u\leftarrow x}] = q_u \alpha^{-1}(1+\eta)^k \mathbb{P}[\neg c_0 \mid \sigma] \\
&= q_u F(\sigma, c_0) \\
&= q_u F(\sigma, T\setminus S).
\end{aligned}$$

Otherwise, $T \setminus S$ is empty. We also have

$$\sum_{x \in Q_u} F(\sigma_{u \leftarrow x}, T \setminus S) = \sum_{x \in Q_u} \prod_{c \in T \setminus S} F(\sigma_{u \leftarrow x}, c) = q_u \prod_{c \in T \setminus S} F(\sigma, c) = q_u F(\sigma, T \setminus S),$$

where we assume that a product over an empty set is 1. Thus, we have (4.43) always holds and the lemma is proved. $\square$

We are now ready to prove Lemma 4.17.

Proof of Lemma 4.17. We prove the lemma by induction on i where $i \in \{0, 1, \cdots, n\}$. Recall the sequence of partial assignments $X^0, X^1, \ldots, X^n$ that evolve in the main counting algorithm defined in Definition 4.1. For the base case where $i = 0$, we have $X^0 = \star^V$. By Lemma 4.18, we have

$$\mathcal{F}(\star^V) < 8n\Delta \cdot 2^{-\lfloor \frac{L}{2\Delta} \rfloor}.$$

For the induction step, it then suffices to show for each $i \in [n]$,

$$\mathcal{F}(X^i) \leq \mathcal{F}(X^{i-1}). \tag{4.44}$$

Recall v_i in Line 2 of the main counting algorithm. Given $i \in [n]$, we have $X^{i-1}(v_i) = \star$. If v_i is involved in some X^{i-1}-frozen constraint, we have the condition in Line 2 of the main counting algorithm is not satisfied, then we have $X^i = X^{i-1}$ and $\mathcal{F}(X^i) = \mathcal{F}(X^{i-1})$. Otherwise the condition is satisfied, and by Line 2(b) of the main counting algorithm we have

$$\mathcal{F}(X^i) = \min_{x \in Q_v} \mathcal{F}(X^{i-1}_{v_i \leftarrow x}) \leq q_{v_i}^{-1} \sum_{x \in Q_v} \mathcal{F}(X^{i-1}_{v_i \leftarrow x}) = \mathcal{F}(X^{i-1}),$$

where the last equality is by Lemma 4.19 and $X^{i-1}(v_i) = \star$. $\square$

We are now ready to prove Theorem 4.1.

Proof of Theorem 4.1. Let

$$\alpha = \left(16\mathrm{e}^2 q^2 k \Delta^2\right)^{-1}$$

and

$$L = 100\Delta \left\lceil \log\left(\frac{qn\Delta}{\varepsilon}\right) \right\rceil.$$

We choose the truncation condition $f(\cdot)$ in Definition 3.5, the upper bound function $\mathcal{F}(\cdot)$ in Definition 3.7.

We first prove the bound on the running time of the main counting algorithm. The followings are the nontrivial costs in the main counting algorithm:

- the cost of estimate the marginal distribution $\mu_{v_i}^X$ with MarginalApproximator in Line 2(a);
- the cost of calculating $\mathcal{F}(\cdot)$ in Line 2(b);
- the cost of the exhaustive enumeration in Line 3.

Let T_{Count} be the running time of the main counting algorithm. Recall that T_{Enu} denotes the cost of the exhaustive enumeration in the main counting algorithm, and T_{MA} denotes the cost of MarginalApproximator. Moreover, one can prove that the total cost of calculating $\mathcal{F}(\cdot)$ in the main counting algorithm is $n^{\mathrm{poly}(\log q, \Delta, k)}$. Thus we have

$$T_{\mathsf{Count}} \leq T_{\mathsf{Enu}}(\Phi, X^n) + \sum_{i=1}^{n} T_{\mathsf{MA}}(\Phi, X^{i-1}, v_i) + n^{\mathrm{poly}(\log q, \Delta, k)}, \tag{4.45}$$

By Lemma 4.1 and Proposition 4.4, we also have for each $i \in [n]$,

$$T_{\mathsf{MA}}(\Phi, X^{i-1}, v_i) = \mathrm{poly}(n, q^{kL\Delta^2}).$$

In addition, by Proposition 4.5 we have

$$T_{\mathsf{Enu}}(\Phi, X^n) = \mathrm{poly}(n, q^{kL\Delta^2}) \tag{4.46}$$

Combining with (4.45) and the definition of L, we have

$$T_{\mathsf{Count}} \le \mathrm{poly}(n, q^{kL\Delta^2}) + n^{\mathrm{poly}(\log q, \Delta, k)} = O\left(\left(\frac{n}{\varepsilon}\right)^{\mathrm{poly}(\log q, \Delta, k)}\right).$$

In the following, we prove the bound on the cost of calculating $\mathcal{F}(\cdot)$. For each $v \in V$ and $L \le i \le L\Delta$, by the definition of $\mathcal{T}_v^i$ in (3.10) and (3.11), it is immediate that $\mathcal{T}_v^i$ can be embedded as a subtree in a graph G with vertex set $V \cup \mathcal{C}$ and degree bounded by $2\Delta^3$, where the subtree is rooted at v of size at most i. Note that for any degree-bounded graph G with maximum degree D, the number of subtrees of G with t vertices and a specific vertex as root, is at most $(\mathrm{e}D)^{t-1}/2$ [5, Lemma 2.1]. Then we have

$$\left|\mathcal{T}_v^i\right| \le \frac{(2\mathrm{e}\Delta^3)^{i-1}}{2} \le \frac{(2\mathrm{e}\Delta^3)^{L\Delta}}{2} \le n^{\mathrm{poly}(\log q, \Delta, k)}.$$

To construct the set $\mathcal{T}_v^i$, one can first construct each possible subtree T of G where T is rooted at v and of size no more than i, and then check whether $T \in \mathcal{T}_v^i$. Therefore, the set $\mathcal{T}_v^i$ can also be constructed in $n^{\mathrm{poly}(k,\Delta)}$ cost. Combining with (3.9) and (3.12), we have for any $\sigma \in \mathcal{Q}^*$, $\mathcal{F}(\sigma)$ can be calculated within cost

$$L\Delta \cdot n \cdot n^{\mathrm{poly}(k,\Delta)} \le n^{\mathrm{poly}(\log q, \Delta, k)}.$$

In addition, by Line 2 of the main counting algorithm, we have $\mathcal{F}(\cdot)$ is calculated at most qn times. Thus, the total cost of calculating $\mathcal{F}(\cdot)$ in the main counting algorithm is $n^{\mathrm{poly}(\log q, \Delta, k)}$.

In the next, we prove (4.46). Recall the definition of η and θ in (3.6). By (1.2) and the above definitions of α and L, it then can be verified that

$$p < \alpha < (\mathrm{e}q\Delta)^{-1}, \quad 16\mathrm{e}p\Delta^3 \le \alpha, \quad \eta \le (2k)^{-1}, \quad 1 - q\theta \le (8\mathrm{e}k\Delta)^{-1}, \quad L \ge 9.$$

Therefore, the conditions of Lemma 4.17 are satisfied. Thus, for large enough n we have

$$\mathcal{F}(X^n) < 8n\Delta \cdot 2^{-\lfloor \frac{L}{2\Delta} \rfloor} \le 8n\Delta \left(\frac{qn\Delta}{\varepsilon}\right)^{-50} < 1.$$

Combining with Proposition 4.5, (4.46) is immediate. The upper bound of T_{Count} is proved.

At last, we prove the bound on the relative error. Let $S = \{i \in [n] \mid v_i \notin V_{\mathsf{fix}}^{X^{i-1}}\}$. For each $i \in S$, let ξ_i be the distribution returned by $\mathsf{MarginalApproximator}(\Phi, X^{i-1}, v_i)$. Thus, we have

$$d_{\mathrm{TV}}(\xi_i, \mu_{v_i}^{X^{i-1}}) < 8n\Delta \cdot 2^{-\lfloor \frac{L}{2\Delta} \rfloor} < \frac{\varepsilon}{8nq},$$

where the first inequality is by $L \ge 9$ and Proposition 4.3, and the second inequality is by Lemma 4.17.

Combining with Lemma 4.1 and Proposition 4.1, we have

$$\xi_i(X^i(v_i)) \ge \mu_{v_i}^{X^{i-1}}(X^i(v_i)) - \frac{\varepsilon}{8nq} \ge \mu_{v_i}^{X^{i-1}}(X^i(v_i))\left(1 - \frac{\varepsilon}{4n}\right)$$

Thus, by Line 2(b) of the main counting algorithm, we have

$$\begin{aligned}
\widehat{Z} &= |\mathcal{S}_{X^n}| \cdot \prod_{i \in S} (\xi_i(X^i(v_i)))^{-1} \\
&\le |\mathcal{S}_{X^n}| \cdot \prod_{i \in S} \left((1 + \varepsilon(2n)^{-1})(\mu_{v_i}^{X^{i-1}}(X^i(v_i)))^{-1}\right) \\
&\le (1 + \varepsilon(2n)^{-1})^n \cdot |\mathcal{S}_{X^n}| \cdot \prod_{i \in S} (\mu_{v_i}^{X^{i-1}}(X^i(v_i)))^{-1} \\
&\le (1 + \varepsilon) \cdot |\mathcal{S}_{X^n}| \cdot \prod_{i \in S} (\mu_{v_i}^{X^{i-1}}(X^i(v_i)))^{-1} \\
&= (1 + \varepsilon) Z_\Phi,
\end{aligned}$$

where the last equality is by (3.4). Similarly, one can also prove $(1 - \varepsilon) Z_\Phi \le \widehat{Z}$. $\square$

5 On improving the JPV algorithm

In this section, we explain how to use the generalized $\{2,3\}$-to improve the analysis of the algorithm in [21] with an improved LLL condition $p\Delta^5 \lesssim 1$.

As stated in the technique overview and Section 3, the algorithm presented in [21] uses the same framework for the main counting algorithm, only with the subroutine for estimating the marginal probabilities replaced with the procedure of setting up a linear program to mimicry the transition probabilities of an idealized coupling procedure. We then include the definition of the idealized coupling procedure and the subroutine for estimating the marginal probability in [21] with conformed notation for a complete illustration:

Definition 5.1. (Idealized coupling procedure in [21]) *Fix any tuple (Φ, σ, v) satisfying Condition 4.1 and any $a, b \in Q_v$, the idealized coupling procedure presented in [21] is equivalent to the following:*

1. *Initialize the partial assignments $X \leftarrow X_0 = \sigma_{v\leftarrow a}$, $Y \leftarrow Y_0 = \sigma_{v\leftarrow b}$, $Z \leftarrow \sigma_{v\leftarrow \star}$.*
2. *Choose $u \leftarrow \mathsf{NextVar}(Z)$, if $u = \perp$, terminates.*
3. *Sample a pair of values (x, y) according to the maximal coupling of the marginal distribution of μ_v^X and μ_v^Y.*
4. *Update X by assigning $X \leftarrow X_{v\leftarrow x}$, and update Y by assigning $Y \leftarrow Y_{v\leftarrow y}$. If $x = y$, update $Z \leftarrow Z_{v\leftarrow x}$, otherwise update $Z \leftarrow Z_{v\leftarrow \star}$.*
5. *Return to Item 1.*

For any pair of partial assignments $(X, Y) \in \mathcal{Q}^* \times \mathcal{Q}^*$, we let $\mu_{\mathsf{cp}}(X, Y)$ to denote the probability that the idealized coupling procedure reaches (X, Y).

The idealized coupling procedure in Definition 5.1 inspires the following definition of a (truncated) idealized deterministic rooted decision tree $\mathscr{T}$.

Definition 5.2. ((Truncated) idealized deterministic rooted decision tree $\mathscr{T}$ in [21]) *Fix any tuple (Φ, σ, v) satisfying Condition 4.1 and any two distinct values $a, b \in Q_v$ as in Definition 5.1, the (truncated) idealized deterministic rooted decision tree $\mathscr{T}$ presented in [21] is equivalent to the following:*

1. *The root of $\mathscr{T}$ consists of the partial assignments $(X_0 = \sigma_{v\leftarrow a}, Y_0 = \sigma_{v\leftarrow b})$. Also for any $\sigma, \tau \in \mathcal{Q}^*$, we define a partial assignment $h(\sigma, \tau) : (\mathcal{Q}^*)^2 \rightarrow \mathcal{Q}^*$ that captures the "discrepancy set" between σ and τ as:*

$$\forall v \in V, h(\sigma, \tau)(v) \triangleq \begin{cases} \sigma(v) & \sigma(v) = \tau(v) \\ \star & \text{otherwise} \end{cases}$$

2. *For each node $(X, Y) \in \mathscr{T}$, if $\mathsf{NextVar}(h(X, Y)) = \perp$ or $f(h(X, Y)) = \mathtt{True}$, then (X, Y) is a leaf node of $\mathscr{T}$, where $f(\cdot)$ is some truncation condition.*
3. *Otherwise, let $u = \mathsf{NextVar}(h(X, Y))$. The children of (X, Y) in $\mathscr{T}$ consist of all possible extensions of (X, Y) obtained by assigning a pair of values in $Q_u \times Q_u$ to the variable u. Similar as in Definition 4.2, we also let $\mathcal{L}(\mathscr{T})$ be the set of leaf nodes in $\mathscr{T}$. Let $\mathcal{L}_g(\mathscr{T}) \triangleq \{(X, Y) \in \mathcal{L}(\mathscr{T}) : f(h(X, Y)) = \mathtt{False}\}$ and $\mathcal{L}_b(\mathscr{T}) \triangleq \{X \in \mathcal{L}(\mathscr{T}) : f(h(X, Y)) = \mathtt{True}\}$ be the sets of leaf nodes $(X, Y) \in \mathcal{L}(\mathscr{T})$ with $h(X, Y)$ don't and do satisfy the truncation condition, respectively.*

One may find a striking resemblance between the RCT defined in Definition 4.2 and the idealized deterministic rooted decision tree defined in Definition 5.2. A difference is that the RCT appeared implicitly in the analysis of the counting algorithm presented in Section 3, and the decision tree here is defined explicitly and used directly in the algorithm in [21] to appear later. We interpret such similarity as an intrinsic property of the problem instance, which makes possible the improvement of both algorithms using the same refined combinatorial structure of generalized $\{2,3\}$-tree.

We then show the crucial subroutine for estimating the marginal probability in [21].

DEFINITION 5.3. (SUBROUTINE FOR ESTIMATING THE MARGINAL PROBABILITY IN [21]) *Fix any (Φ,σ,v) satisfying Condition 4.1, the subroutine for estimating the marginal probability presented in [21] is equivalent to the following procedure:*

For any $a,b\in Q_v$, let $\mathscr{T}$ be the idealized deterministic rooted decision tree defined in Definition 5.2. Set up the following linear program with variables r_-,r_+ and $\hat{p}^X_{X,Y},\hat{p}^Y_{X,Y}$ for each $(X,Y)\in\mathscr{T}$:

1. *For all $(X,Y)\in\mathcal{L}(\mathscr{T})$, $0\le\hat{p}^X_{X,Y},\hat{p}^Y_{X,Y}\le 1$.*

2. *For every $(X,Y)\in\mathcal{L}_g(\mathscr{T})$,*
$$r_-\le\frac{\hat{p}^X_{X,Y}\,|\mathcal{S}_X|}{\hat{p}^Y_{X,Y}\,|\mathcal{S}_Y|}\le r_+,$$
where $\frac{|\mathcal{S}_X|}{|\mathcal{S}_Y|}$ is computed through exhaustive enumeration over the connected component in $H_{\Phi^{h(X,Y)}}$ containing v.

3. *$\hat{p}^{X_0}_{X_0,Y_0}=\hat{p}^{Y_0}_{X_0,Y_0}=1$. Moreover, for every node $(X,Y)\in\mathscr{T}\setminus\mathcal{L}(\mathscr{T})$ and $u=\mathsf{NextVar}\,(h(X,Y))$,*
$$\hat{p}^X_{X,Y}=\sum_{b\in Q_u}\hat{p}^{X_{u\leftarrow a}}_{X_{u\leftarrow a},Y_{u\leftarrow b}}\text{ for all }a\in Q_u$$
$$\hat{p}^Y_{X,Y}=\sum_{b\in Q_u}\hat{p}^{Y_{u\leftarrow a}}_{X_{u\leftarrow b},Y_{u\leftarrow a}}\text{ for all }a\in Q_u$$

4. *For every node $(X,Y)\in\mathscr{T}\setminus\mathcal{L}(\mathscr{T})$, letting $u=\mathsf{NextVar}\,(h(X,Y))$, for all $a\in Q_u$,*
$$\sum_{\substack{b\in Q_u\\ b\ne a}}\hat{p}^{X_{u\leftarrow a}}_{X_{u\leftarrow a},Y_{u\leftarrow b}}\le 1-q\theta$$
$$\sum_{\substack{b\in Q_u\\ b\ne a}}\hat{p}^{Y_{u\leftarrow a}}_{X_{u\leftarrow b},Y_{u\leftarrow a}}\le 1-q\theta,$$
where θ,η is defined as in (3.6).

It can be verified the following lemma holds by by taking
$$\hat{p}^X_{X,Y}=\frac{\mu_{\mathsf{cp}}(X,Y)}{\mu[X\mid X_0]},\hat{p}^Y_{X,Y}=\frac{\mu_{\mathsf{cp}}(X,Y)}{\mu[Y\mid Y_0]}$$
for each $(X,Y)\in\mathscr{T}$ and verifying all items in Definition 5.3. Particularly, Item 4 of Definition 5.3 can be shown using a similar argument as in Corollary 4.1.

LEMMA 5.1. *The LP defined in Definition 5.3 is feasible for $r_-=r_+=\frac{|\mathcal{S}_{X_0}|}{|\mathcal{S}_{Y_0}|}$.*

The following lemma holds by Lemma 4.14 and standard guarantees on the running time of linear programming.

LEMMA 5.2. *For every r_-,r_+,η which can be represented in $\mathrm{poly}(n,q)$ bits, the feasibility of the LP defined in Definition 5.3 can be checked in time $\mathrm{poly}(n,q^{k\Delta L})$.*

One crucial thing is that the feasibility of the above LP (for appropriately chosen α and truncation condition $f(\cdot)$) implies that r_- (respectively r_+) is an approximate lower (respectively upper) bound for $|\mathcal{S}_{X_0}|\,/\,|\mathcal{S}_{Y_0}|$. Given this, one will be able to use binary search to approximate $|\mathcal{S}_{X_0}|\,/\,|\mathcal{S}_{Y_0}|$.

For improving the analysis in [21], we choose the truncation condition $f(\cdot)$ in Definition 3.5 for Item 2 of Definition 5.2. It suffices to show the following improved lemma.

LEMMA 5.3. (IMPROVED VERSION OF **Lemma 5.1** IN [21]) *Recall that* $X_0 = \sigma_{v\leftarrow a}, Y_0 = \sigma_{v\leftarrow b}$. *If* $16\mathrm{e}p\Delta^3 \le \alpha$, $\eta \le (2k)^{-1}$, $1 - q\theta \le (8\mathrm{e}k\Delta)^{-1}$ *and* $L > 1$,

$$\frac{1}{|\mathcal{S}_{X_0}|} \sum_{\tau \in \mathcal{S}_{X_0}} \sum_{(X,Y) \in \mathcal{L}_b(\mathscr{T}): X \to \tau} \hat{p}^X_{X,Y} \le F(\sigma)$$

$$\frac{1}{|\mathcal{S}_{Y_0}|} \sum_{\tau \in \mathcal{S}_{Y_0}} \sum_{(X,Y) \in \mathcal{L}_b(\mathscr{T}): Y \to \tau} \hat{p}^Y_{X,Y} \le F(\sigma),$$

where $x \to y$ *means* x *extends* y, *and* $F(\cdot)$ *is the upper bound function in Definition 3.7.*

Given Lemma 5.3, note that by Item 3 of Definition 5.3, we have

$$\sum_{(X,Y) \in \mathcal{L}(\mathscr{T}): X \to \tau} \hat{p}^X_{X,Y} = 1 \text{ for all } \tau \in \mathcal{S}_{X_0} \tag{5.47}$$

Therefore, we have

$$\begin{aligned} |\mathcal{S}_{X_0}| &= \sum_{\tau \in \mathcal{S}_{X_0}} \sum_{(X,Y) \in \mathcal{L}(\mathscr{T}): X \to \tau} \hat{p}^X_{X,Y} \\ &= \sum_{\tau \in \mathcal{S}_{X_0}} \sum_{(X,Y) \in \mathcal{L}_g(\mathscr{T}): X \to \tau} \hat{p}^X_{X,Y} + \sum_{\tau \in \mathcal{S}_{X_0}} \sum_{(X,Y) \in \mathcal{L}_b(\mathscr{T}): X \to \tau} \hat{p}^X_{X,Y} \\ &= \sum_{(X,Y) \in \mathcal{L}_g(\mathscr{T})} \left(\hat{p}^X_{X,Y} \cdot |\mathcal{S}_X|\right) \pm F(\sigma) \cdot |\mathcal{S}_{X_0}|, \end{aligned}$$

where the first equality is by (5.47) and the last equality is by interchanging sums and Lemma 5.3. A similar estimate also holds for $|\mathcal{S}_{Y_0}|$. Thus, we have

$$\begin{aligned} \frac{|\mathcal{S}_{X_0}| \cdot (1 \pm F(\sigma))}{|\mathcal{S}_{Y_0}| \cdot (1 \pm F(\sigma))} &= \frac{\sum_{(X,Y) \in \mathcal{L}_g(\mathscr{T})} \left(\hat{p}^X_{X,Y} \cdot |\mathcal{S}_X|\right)}{\sum_{(X,Y) \in \mathcal{L}_g(\mathscr{T})} \left(\hat{p}^Y_{X,Y} \cdot |\mathcal{S}_Y|\right)} \\ \text{(By Item 2 of Definition 5.3)} \quad &\in \left[\frac{r_- \cdot \sum_{(X,Y) \in \mathcal{L}_g(\mathscr{T})} \left(\hat{p}^Y_{X,Y} \cdot |\mathcal{S}_Y|\right)}{\sum_{(X,Y) \in \mathcal{L}_g(\mathscr{T})} \left(\hat{p}^Y_{X,Y} \cdot |\mathcal{S}_Y|\right)}, \frac{r_+ \cdot \sum_{(X,Y) \in \mathcal{L}_g(\mathscr{T})} \left(\hat{p}^Y_{X,Y} \cdot |\mathcal{S}_Y|\right)}{\sum_{(X,Y) \in \mathcal{L}_g(\mathscr{T})} \left(\hat{p}^Y_{X,Y} \cdot |\mathcal{S}_Y|\right)}\right] \\ &\in [r_-, r_+]. \end{aligned}$$

With this guarantee, one may approximate $|\mathcal{S}_{X_0}| / |\mathcal{S}_{Y_0}|$ and therefore the estimate marginal distribution μ^σ_v with total variation distance between the true marginal distribution bounded above by some linear function with $F(\sigma)$. Note that Proposition 4.4 also holds by the same reasoning for the replaced marginal approximator. Moreover, as we only replaced the subroutine for approximating marginals, both Proposition 4.5 and Lemma 4.17 still hold. Then combining Lemma 5.2 and going through the same proof as Theorem 4.1, one can improve the analysis for the algorithm presented in [21] to work in the regime of $p\Delta^5 \lesssim 1$.

Proof of Lemma 5.3. Consider the following process of generating a random root-to-leaf paths of $\mathscr{T}$. At a non-leaf node $(X,Y) \in \mathscr{T} \setminus \mathcal{L}(\mathscr{T})$, sample a value a for $u = \mathsf{NextVar}\,(h(X,Y))$ according to μ^X_v and set $X' \leftarrow X_{u\leftarrow a}$. Then, choose a random element $b \in Q_u$ and go to the node $(X', Y_{u\leftarrow b}) \in \mathscr{T}$, where the probability of choosing each $b \in Q_u$ is

$$p(X,Y,X',Y_{u(b)}) = \frac{\hat{p}^{X'}_{X',Y_{u\leftarrow b}}}{\hat{p}^X_{X,Y}}, \tag{5.48}$$

Note that by Item 3 of Definition 5.3, one can verify that $p(X, Y, X', Y_{u\leftarrow(\cdot)})$ is a probability distribution. Let (X^*, Y^*) denote the random leaf of $\mathscr{T}$ returned by this process and let $\hat{\mu}$ denote the probability distribution on $\mathcal{L}(\mathscr{T})$ induced by this process.

Let $(X_\ell, Y_\ell) \in \mathcal{L}(\mathscr{T})$ and denote the corresponding root-to-leaf path by $(X_0, Y_0), \dots, (X_\ell, Y_\ell)$. Then,

$$\hat{\mu}[(X^*, Y^*) = (X_\ell, Y_\ell)] = \prod_{t=1}^{\ell} \mu(X_t \mid X_{t-1}) \times \prod_{t=1}^{\ell} p(X_{t-1}, Y_{t-1}, X_t, Y_t) = \frac{|\mathcal{S}_{X_\ell}|}{|\mathcal{S}_{X_0}|} \cdot \frac{\hat{p}^{X_\ell}_{X_\ell, Y_\ell}}{\hat{p}^{X_0}_{X_0, Y_0}} = \frac{|\mathcal{S}_{X_\ell}|}{|\mathcal{S}_{X_0}|} \cdot \hat{p}^{X_\ell}_{X_\ell, Y_\ell},$$

where the first equality is by chain rule, the second one by (5.48) and the last one by Item 3 of Definition 5.3. Therefore,

$$\begin{aligned}\frac{1}{|\mathcal{S}_{X_0}|} \sum_{\tau \in \mathcal{S}_{X_0}} \sum_{(X,Y)\in\mathcal{L}_b(\mathscr{T}): X\to\tau} \hat{p}^X_{X,Y} &= \sum_{\tau \in \mathcal{S}_{X_0}} \sum_{(X,Y)\in\mathcal{L}_b(\mathscr{T}): X\to\tau} \frac{\hat{\mu}[(X^*, Y^*) = (X, Y)]}{|\mathcal{S}_X|} \\ &= \sum_{(X,Y)\in\mathcal{L}_b(\mathscr{T})} \hat{\mu}[(X^*, Y^*) = (X, Y)] = \hat{\mu}[(X^*, Y^*) \in \mathcal{L}_b(\mathscr{T})].\end{aligned}$$

For each $0 \le i \le \ell$ we let $Z_i = h(X_i, Y_i)$ and let $Z^* = h(X^*, Y^*)$. Note that by Item 2 of Definition 5.2, the stopping rule of the above process only depends on $h(X, Y)$, and one can then verify that $\hat{\mu}[(X^*, Y^*) \in \mathcal{L}_b(\mathscr{T})] = \hat{\mu}[f(Z^*) = \texttt{True}]$.

It remains to bound $\hat{\mu}[f(Z^*) = \texttt{True}]$. Note that this process of generating a root-to-leaf path also leads to a process that generates a sequence of constraints $Z_0, \dots, Z_\ell$ that satisfy the following two properties:

1. if $\mathsf{NextVar}(Z_i) = \perp$ or $f(Z_i) = \texttt{True}$, the sequence stops at Z_i;
2. otherwise $u = \mathsf{NextVar}(Z_i) \in V$, the partial assignment $Z_{i+1} \in \mathcal{Q}^*$ is generated from Z_i by randomly giving u a value $x \in \mathcal{Q}_u^\star$, such that
   (a) $\mathbf{Pr}[Z_{i+1} = (Z_i)_{u\leftarrow\star}] \le 1 - q\theta$.
   (b) $\forall x \in Q_u, \mathbf{Pr}[Z_{i+1} = (Z_i)_{u\leftarrow x}] \le \frac{1}{q_u}(1+\eta)$

We then show these properties. Item 1 is from the stopping rule of the process of generating a root-to-leaf path. Item 2a is from combining (5.48) and Item 4 of Definition 5.3. Item 2b is from combining the rule of the process of generating a root-to-leaf path and that

$$\forall x \in Q_u, \mathbf{Pr}[Z_{i+1} = (Z_i)_{u\leftarrow x}] \le \mathbf{Pr}[X_{i+1} = (X_i)_{u\leftarrow x}] \le \mu_v^{X_i}(x) \le \frac{1}{q_u}(1+\eta).$$

Note that the process above that generates $Z_0, \dots, Z_\ell$ is pretty much similar to the process Path defined in Definition 4.3. Moreover, it can be verified that the proofs in Lemma 4.9, Lemma 4.8 and eventually in Proposition 4.3 still can apply for this process, and going through these proofs for this process leads to the desired result. ☐

References

[1] N. Alon, *A parallel algorithmic version of the local lemma*, Random Struct. Algorithms, 2 (1991), pp. 367–378. (Conference version in *FOCS*'91).

[2] K. Anand and M. Jerrum, *Perfect sampling in infinite spin systems via strong spatial mixing*, SIAM J. Comput., 51 (2022), pp. 1280–1295.

[3] A. Barvinok, *Combinatorics and complexity of partition functions*, vol. 30 of Algorithms and Combinatorics, Springer, Cham, 2016.

[4] J. Beck, *An algorithmic approach to the Lovász local lemma.*, Random Struct. Algorithms, 2 (1991), pp. 343–365.

[5] C. Borgs, J. Chayes, J. Kahn, and L. Lovász, *Left and right convergence of graphs with bounded degree*, Random Struct. Algorithms, 42 (2013), pp. 1–28.

[6] M. Dyer, A. Frieze, and R. Kannan, *A random polynomial-time algorithm for approximating the volume of convex bodies*, J. ACM, 38 (1991), pp. 1–17.

[7] P. Erdős and L. Lovász, *Problems and results on 3-chromatic hypergraphs and some related questions*, Infinite and finite sets, volume 10 of Colloquia Mathematica Societatis János Bolyai, (1975), pp. 609–628.

[8] W. Feng, H. Guo, and J. Wang, *Improved bounds for randomly colouring simple hypergraphs*, in RANDOM, vol. 245 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022, pp. 25:1–25:17. (full version in arXiv:2202.05554).

[9] W. Feng, H. Guo, Y. Yin, and C. Zhang, *Fast sampling and counting k-SAT solutions in the local lemma regime*, J. ACM, 68 (2021), pp. Art. 40, 42.

[10] W. Feng, K. He, and Y. Yin, *Sampling constraint satisfaction solutions in the local lemma regime*, in STOC, ACM, 2021, pp. 1565–1578.

[11] A. Galanis, L. A. Goldberg, H. Guo, and K. Yang, *Counting solutions to random CNF formulas*, in ICALP, vol. 168 of LIPIcs, 2020, pp. 53:1–53:14.

[12] A. Galanis, H. Guo, and J. Wang, *Inapproximability of counting hypergraph colourings*, ACM Trans. Comput. Theory, (2022). To appear.

[13] H. Guo, M. Jerrum, and J. Liu, *Uniform sampling through the Lovász local lemma*, J. ACM, 66 (2019), pp. Art. 18, 31.

[14] H. Guo, C. Liao, P. Lu, and C. Zhang, *Counting hypergraph colorings in the local lemma regime*, SIAM J. Comput., 48 (2019), pp. 1397–1424.

[15] B. Haeupler, B. Saha, and A. Srinivasan, *New constructive aspects of the lovász local lemma*, J. ACM, 58 (2011), pp. 28:1–28:28. (Conference version in *FOCS*'10).

[16] K. He, X. Sun, and K. Wu, *Perfect sampling for (atomic) Lovász local lemma*, arXiv, abs/2107.03932 (2021).

[17] K. He, C. Wang, and Y. Yin, *Sampling Lovász local lemma for general constraint satisfaction solutions in near-linear time*, arXiv, abs/2204.01520 (2022). (To appear in *FOCS*'22).

[18] T. Helmuth, W. Perkins, and G. Regts, *Algorithmic pirogov–sinai theory*, Probability Theory and Related Fields, 176 (2020), pp. 851–895.

[19] V. Jain, W. Perkins, A. Sah, and M. Sawhney, *Approximate counting and sampling via local central limit theorems*, in STOC, ACM, 2022, pp. 1473–1486.

[20] V. Jain, H. T. Pham, and T. D. Vuong, *On the sampling Lovász local lemma for atomic constraint satisfaction problems*, arXiv, abs/2102.08342 (2021).

[21] ——, *Towards the sampling lovász local lemma*, in 62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022, IEEE, 2021, pp. 173–183.

[22] M. Jenssen, P. Keevash, and W. Perkins, *Algorithms for #BIS-hard problems on expander graphs*, SIAM J. Comput., 49 (2020), pp. 681–710.

[23] M. R. Jerrum, L. G. Valiant, and V. V. Vazirani, *Random generation of combinatorial structures from a uniform distribution*, Theoret. Comput. Sci., 43 (1986), pp. 169–188.

[24] A. Moitra, *Approximate counting, the Lovász local lemma, and inference in graphical models*, J. ACM, 66 (2019), pp. 10:1–10:25. (Conference version in *STOC*'17).

[25] V. Patel and G. Regts, *Deterministic polynomial-time approximation algorithms for partition functions and graph polynomials*, SIAM J. Comput., 46 (2017), pp. 1893–1919.

[26] G. Qiu, Y. Wang, and C. Zhang, *A perfect sampler for hypergraph independent sets*, in ICALP, vol. 229 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022, pp. 103:1–103:16.

[27] D. Štefankovič, S. Vempala, and E. Vigoda, *Adaptive simulated annealing: a near-optimal connection between sampling and counting*, J. ACM, 56 (2009), pp. Art. 18, 36.

[28] D. Weitz, *Counting independent sets up to the tree threshold*, in STOC, ACM, 2006, pp. 140–149.

A Generalized $\{2,3\}$-tree as witnesses for useful properties

In this section, we prove several technical lemmas (Lemma 4.7, Lemma 4.14, Lemma 4.15 and Lemma 4.16). Lemma 4.14 states that for some partial assignment $\sigma \in \mathcal{Q}^*$, the length of $\mathsf{Path}(\sigma) = (\sigma_0, \ldots, \sigma_\ell)$ is bounded. Lemma 4.15 relates the size of $\mathcal{C}_v^{\sigma_\ell}$ with the sizes of $V_\star^{\sigma_\ell}$ and $\mathcal{C}_{\star\text{-frozen}}^{\sigma_\ell}$. Lemma 4.7 and Lemma 4.16 state that for certain properties such as $V_\star^\sigma$ or $\mathcal{C}_{\star\text{-frozen}}^\sigma$, when the class of variables/constraints with that property becomes too large, a generalized $\{2,3\}$-tree with certain properties inevitably appears within the class.

To aid our proof, we introduce the definition of G_{VC}, a graph with a vertex set over all variables and constraints of the CSP formula.

Definition A.1. (Graph of variables and constraints) *Let $\Phi = (V, \mathcal{Q}, \mathcal{C})$ be the CSP formula. Define $G_{\mathsf{VC}} = (V \cup \mathcal{C}, E)$ as the graph where vertices are $V \cup \mathcal{C}$ and there is an edge between two vertices u, v if and only if one of the following holds:*

1. *$u, v \in V$ and there exists some $c \in \mathcal{C}$ such that $u, v \in \mathsf{vbl}(c)$.*
2. *$u, v \in \mathcal{C}$ and $dist_{\mathsf{Lin}(H_\Phi)}(u, v) = 1$ or 2.*
3. *$u \in V, v \in \mathcal{C}$ and there exists some $c \in \mathcal{C}$ such that $u \in \mathsf{vbl}(c) \wedge dist_{\mathsf{Lin}(H_\Phi)}(c, v) = 1$.*

Furthermore, for any $S \subseteq V \cup \mathcal{C}$, we let $G_{\mathsf{VC}}(S)$ denote the subgraph of G_{VC} induced by S.

We state the following lemma, which is immediate by [17, Lemma 6.32].

Lemma A.1. *Let $\sigma \in \mathcal{Q}^*$ and $\mathsf{Path}(\sigma) = (\sigma_0, \sigma_1, \cdots, \sigma_\ell)$. For every $0 \le i \le j \le \ell$, it holds that*

$$V_\star^{\sigma_i} \subseteq V_\star^{\sigma_j} \quad \text{and} \quad \mathcal{C}_{\mathcal{P}}^{\sigma_i} \subseteq \mathcal{C}_{\mathcal{P}}^{\sigma_j},$$

where $\mathcal{P}$ can be any property $\mathcal{P} \in \{\,\mathsf{frozen},\ \star\text{-}\mathsf{con},\ \star\text{-}\mathsf{frozen}\,\}$.

A.1 Proof of Lemma 4.7 We first need the two following lemmas.

Lemma A.2. *Assume the condition of Lemma A.3. Then $G_{\mathsf{VC}}\left(\mathcal{C}_{\star\text{-frozen}}^{\sigma_i} \cup V_\star^{\sigma_i}\right)$ is connected for each $0 \le i \le \ell$.*

Proof. We prove this lemma by induction on i. For simplicity, we say a variable or constraint c is connected to a subset $S \subseteq V \cup \mathcal{C}$ in G_{VC} if c is connected to some $c' \in S$. The base case is when $i = 0$. By the condition of the lemma, v is the only variable satisfying $\sigma(v) = \star$. Combining with $\sigma_0 = \sigma$, we have v is the only variable satisfying $\sigma_0(v) = \star$. Therefore, $V_\star^{\sigma_0} = \{v\}$. In addition, we have the following claim: each $c \in \mathcal{C}_{\star\text{-frozen}}^{\sigma_0}$ is connected to v in $G_{\mathsf{VC}}(\mathcal{C}_{\star\text{-frozen}}^{\sigma_0} \cup V_\star^{\sigma_0})$. Combining with the claim, we have $G_{\mathsf{VC}}(\mathcal{C}_{\star\text{-frozen}}^{\sigma_0} \cup V_\star^{\sigma_0})$ is connected.

Now we prove the claim, which completes the proof of the base case. By $c \in \mathcal{C}_{\star\text{-frozen}}^{\sigma_0}$, we have $c \in \mathcal{C}_{\star\text{-con}}^{\sigma_0} \cap \mathcal{C}_{\mathsf{frozen}}^{\sigma_0}$. By $c \in \mathcal{C}_{\star\text{-con}}^{\sigma_0}$ and Definition 3.3, we have $V_{\star\text{-con}}^{\sigma_0} \cap \mathsf{vbl}(c) \neq \emptyset$. Combining with v is the only variable satisfying $\sigma_0(v) = \star$ and the definition of $V_{\star\text{-con}}^{\sigma_0}$, we have there exists a connected path $c_1^{\sigma_0}, c_2^{\sigma_0}, \cdots, c_t^{\sigma_0} = c^{\sigma_0} \in \mathcal{C}^{\sigma_0}$ such that $\sigma_0(v) = \star$, $v \in \mathsf{vbl}(c_1^{\sigma_0})$ and $\mathsf{vbl}\left(c_j^{\sigma_0}\right) \subseteq V^{\sigma_0} \cap V_{\mathsf{fix}}^{\sigma_0}$ for each $j < t$. If $c = c_1$, then $v \in \mathsf{vbl}(c)$ and the claim is immediate by the definition of G_{VC}. In the following, we assume $c \neq c_1$. Let $w_j \in \left(\mathsf{vbl}\left(c_j^{\sigma_0}\right) \cap \mathsf{vbl}\left(c_{j+1}^{\sigma_0}\right)\right)$ for each $j < t$. Then $w_j \notin \Lambda(\sigma_0)$. By $w_j \in \mathsf{vbl}\left(c_j^{\sigma_0}\right)$ and $\mathsf{vbl}\left(c_j^{\sigma_0}\right) \subseteq V_{\mathsf{fix}}^{\sigma_0}$, we have $w_j \in V_{\mathsf{fix}}^{\sigma_0}$. Combining with $w_j \notin \Lambda(\sigma_0)$, we have either $\sigma_0(w_j) = \star$, where we set $\widehat{c}_j = w_j$; or $w_j \in \mathsf{vbl}(\widehat{c}_j)$ for some $\widehat{c}_j \in \mathcal{C}_{\mathsf{frozen}}^{\sigma_0}$. Note that $\widehat{c}_j$ can be either a variable or a constraint. In the former case, we have $\widehat{c}_j \in V_\star^{\sigma_0}$. In the latter case, By w_j is connected to v in $H_{\mathsf{fix}}^{\sigma_0}$ through the path $c_1^{\sigma_0}, c_2^{\sigma_0}, \cdots, c_j^{\sigma_0}$, we have $w_j \in V_{\star\text{-con}}^{\sigma_0}$. Thus, we have $\widehat{c}_j \in \mathcal{C}_{\star\text{-con}}^{\sigma_0}$ by Definition 3.3. Combining with $\widehat{c}_j \in \mathcal{C}_{\mathsf{frozen}}^{\sigma_0}$, we have $\widehat{c}_j \in \mathcal{C}_{\star\text{-frozen}}^{\sigma_0}$. In summary, we always have $\widehat{c}_j \in V_\star^{\sigma_0} \cup \mathcal{C}_{\star\text{-frozen}}^{\sigma_0}$. Moreover, for each $j < t-1$, if $\widehat{c}_j \in \mathcal{C}$, we have $w_j \in \mathsf{vbl}\left(c_{j+1}^{\sigma_0}\right) \cap \mathsf{vbl}\left(\widehat{c}_j^{\sigma_0}\right)$, otherwise we have $\widehat{c}_j = w_j$. Thus by Definition A.1, it can be verified that $\widehat{c}_j$ and $\widehat{c}_{j+1}$ are adjacent in G_{VC}. In addition, if $\widehat{c}_1 \in \mathcal{C}$, we have $w_1 \in \mathsf{vbl}(c_1) \cap \mathsf{vbl}(\widehat{c}_1)$, otherwise we have $\mathsf{vbl}(\widehat{c}_1) = w_1 \in \mathsf{vbl}(c_1)$, hence c_1 and $\widehat{c}_1$ are adjacent in G_{VC}. Similarly, we have $\widehat{c}_{t-1}$ and c_t are adjacent in G_{VC} Thus, we have $v, c_1, \widehat{c}_1, \widehat{c}_2, \cdots, \widehat{c}_{t-1}, c_t = c$ is a connected path in G_{VC}. Combining with $v \in V_\star^{\sigma_0}$ and $\widehat{c}_j \in \mathcal{C}_{\star\text{-frozen}}^{\sigma_0}$ for each $j < t$, the claim is immediate.

For the induction step, we prove this lemma for each $i > 0$. We claim that each $v \in V_\star^{\sigma_i}$ is connected to $V_\star^{\sigma_{i-1}}$ in $G_{\mathsf{VC}}(\mathcal{C}_{\star\text{-frozen}}^{\sigma_i} \cup V_\star^{\sigma_i})$. In addition, we can prove each $c \in \mathcal{C}_{\star\text{-frozen}}^{\sigma_i}$ is connected to $V_\star^{\sigma_{i-1}}$ in $G_{\mathsf{VC}}(\mathcal{C}_{\star\text{-frozen}}^{\sigma_i} \cup V_\star^{\sigma_i})$ by a similar argument to the base case. Moreover, by the induction hypothesis we have $G_{\mathsf{VC}}(\mathcal{C}_{\star\text{-frozen}}^{\sigma i-1} \cup V_\star^{\sigma_{i-1}})$ is connected. Combining with $\mathcal{C}_{\star\text{-frozen}}^{\sigma_{i-1}} \subseteq \mathcal{C}_{\star\text{-frozen}}^{\sigma_i}$ and $V_\star^{\sigma_{i-1}} \subseteq V_\star^{\sigma_i}$ by Lemma A.1, we have $G_{\mathsf{VC}}(\mathcal{C}_{\star\text{-frozen}}^{\sigma_i} \cup V_\star^{\sigma_i})$ is connected.

Now we prove the claim that each $v \in V_\star^{\sigma_i}$ is connected to $V_\star^{\sigma_{i-1}}$ in G_{VC}, which completes the proof of the lemma. If $v \in V_\star^{\sigma_{i-1}}$, the claim is immediate by $V_\star^{\sigma_{i-1}} \subseteq V_\star^{\sigma_i}$. In the following, we assume $v \in V_\star^{\sigma_i} \setminus V_\star^{\sigma_{i-1}}$, where by Definition 4.3 we have $v = \mathsf{NextVar}(\sigma_{i-1})$. By the definition of $\mathsf{NextVar}(\cdot)$, we have $v \in V_{\star\text{-inf}}^{\sigma_{i-1}}$ and then $v \in \mathsf{vbl}(\widehat{c})$ for some constraint $\widehat{c} \in \mathcal{C}_{\star\text{-con}}^{\sigma_{i-1}}$. In addition, by $\widehat{c} \in \mathcal{C}_{\star\text{-con}}^{\sigma_{i-1}}$ one can verify that there exists a variable $w \neq v$ and a connected path $c_1^{\sigma_{i-1}}, c_2^{\sigma_{i-1}}, \cdots, c_t^{\sigma_{i-1}} = \widehat{c}^{\sigma_{i-1}} \in \mathcal{C}^{\sigma_{i-1}}$ such that $\sigma_{i-1}(w) = \star$, $w \in \mathsf{vbl}\left(c_1^{\sigma_{i-1}}\right)$ and $\mathsf{vbl}\left(c_j^{\sigma_{i-1}}\right) \subseteq V^{\sigma_{i-1}} \cap V_{\mathsf{fix}}^{\sigma_{i-1}}$ for each $j < t$. Then there are two possibilities for $\widehat{c}$.

- If $\widehat{c} = c_1$, we have $v, w \in \mathsf{vbl}(c_1)$. Therefore, v is connected to w in $G_{\mathsf{VC}}(\mathcal{C}_{\star\text{-frozen}}^{\sigma_{i-1}} \cup V_\star^{\sigma_{i-1}} \cup \{v\})$. Also by $\sigma_{i-1}(w) = \star$ we have $w \in V_\star^{\sigma_{i-1}}$. In addition, by Lemma A.1 we have

  $$\mathcal{C}_{\star\text{-frozen}}^{\sigma_{i-1}} \cup V_\star^{\sigma_{i-1}} \cup \{v\} \subseteq \mathcal{C}_{\star\text{-frozen}}^{\sigma_{i-1}} \cup V_\star^{\sigma_i} \subseteq \mathcal{C}_{\star\text{-frozen}}^{\sigma_i} \cup V_\star^{\sigma_i}.$$

  Thus the claim is immediate.

- Otherwise, $\widehat{c} \neq c_1$. Similarly to the base case, one can find a connected path $c_1, \widehat{c}_1, \widehat{c}_2, \cdots, \widehat{c}_{t-1}$, $c_t = \widehat{c}$ in G_{VC}, where $w \in \mathsf{vbl}(c_1)$, $\widehat{c}_j \in V_\star^{\sigma_{i-1}} \cup \mathcal{C}_{\star\text{-frozen}}^{\sigma_{i-1}}$ for each $j < t$, and there exists $w_{t-1} \in \mathsf{vbl}(c_t) \cap \mathsf{vbl}(\widehat{c}_{t-1})$. Recall that $v \in \mathsf{vbl}(\widehat{c})$ and $w \in \mathsf{vbl}(c_1)$ Thus, $w, c_1, \widehat{c}_1, \widehat{c}_2, \cdots, \widehat{c}_{t-1}, v$ is also a connected path in G_{VC}. Combining with $w \in V_\star^{\sigma_{i-1}}$, $\widehat{c}_j \in V_\star^{\sigma_{i-1}} \cup \mathcal{C}_{\star\text{-frozen}}^{\sigma_{i-1}}$ for each $j < t$, we have v is connected to $V_\star^{\sigma_{i-1}}$ in $G_{\mathsf{VC}}(\mathcal{C}_{\star\text{-frozen}}^{\sigma_i} \cup V_\star^{\sigma_i})$. Thus the claim is immediate.

□

LEMMA A.3. *Let $\sigma \in \mathcal{Q}^*$ be a partial assignment satisfying that exactly one variable $v \in V$ has $\sigma(v) = \star$. Let $\mathsf{Path}(\sigma) = (\sigma_0, \sigma_1, \ldots, \sigma_\ell)$. Then there always exists a generalized $\{2,3\}$-tree $T = U \circ E$ in H_Φ with some auxiliary tree rooted at v such that*

$$U = V_\star^{\sigma_\ell}, \quad E \subseteq \mathcal{C}_{\star\text{-frozen}}^{\sigma_\ell}, \quad \text{and} \quad \Delta \cdot |E| \geq |\mathcal{C}_{\star\text{-frozen}}^{\sigma_\ell}|$$

Proof. We construct T along with one of its auxiliary tree T^* by greedily starting from a single root v. We maintain a set B of "valid vertices", initially set as $B \leftarrow \mathcal{C}_{\star\text{-frozen}}^{\sigma_\ell} \cup V_\star^{\sigma_\ell} \setminus \{v\}$. Each time we choose a vertex u in B that is nearest to T in G_{VC}, i.e., $u = \arg\min_{w \in B} \min_{x \in T} \mathrm{dist}_{G_{\mathsf{VC}}}(w, x)$, then let w be the vertex nearest to u in T. We add w along with the arc (w, u) in T^*, then update B as follows:

(a) If $u \in V$, then we update $B \leftarrow B \setminus \{u\}$

(b) If $u \in \mathcal{C}$, we update $B \leftarrow B \setminus \Gamma^+(u)$, where $\Gamma^+(u) = \{c \in \mathcal{C} \mid \mathsf{vbl}(u) \cap \mathsf{vbl}(c) \neq \emptyset\}$.

If $B = \emptyset$, the process stops. We claim that when the process stops, we have $T = U \circ E$ is a generalized $\{2,3\}$-tree in H_Φ satisfying $U = V_\star^{\sigma_\ell}, E \subseteq \mathcal{C}_{\mathsf{frozen}}^{\sigma_\ell}$ and $\Delta \cdot |E| \geq \mathcal{C}_{\mathsf{frozen}}^{\sigma_\ell}$.

We first show that T is a generalized $\{2,3\}$-tree in H_Φ. Item 1 of Definition 3.6 is immediate by Item (b) of the process. It then suffices to show T^* is a valid auxiliary tree. For Item 2 of Definition 3.6, note that by Lemma A.2 we have $G_{\mathsf{VC}}(\mathcal{C}_{\star\text{-frozen}}^{\sigma_\ell} \cup V_\star^{\sigma_\ell})$ is connected. Also from the process, we know each $u \in \mathcal{C}_{\star\text{-frozen}}^{\sigma_\ell} \cup V_\star^{\sigma_\ell}$ is either added into T or removed in Item (b). If u is removed in Item (b), then $u \in \mathcal{C}$ and there exists $c \in \mathcal{C}$ such that $\mathsf{vbl}(c) \cap \mathsf{vbl}(u) \neq \emptyset$ and c is added into T. For each $u \neq v \in T$, let w be the only father of u in T^*, then we have the following cases:

- $\mathrm{dist}_{G_{\mathsf{VC}}}(u, w) = 1$: Then the arc (u, w) must satisfy Item 2 of Definition 3.6 by comparing Definition A.1 with Item 2 of Definition 3.6.

- Otherwise it must follow that $w \in \mathcal{C}$ and $\mathrm{dist}_{G_{\mathsf{VC}}}(u, w') = 1$ for some constraint $w' \in \mathcal{C}$ removed during Item (b) such that $\mathsf{vbl}(w) \cap \mathsf{vbl}(w') \neq \emptyset$. This is by our choice of u and w from the process and the fact that $G_{\mathsf{VC}}(\mathcal{C}_{\star\text{-frozen}}^{\sigma_\ell} \cup V_\star^{\sigma_\ell})$ is connected. Then the arc (u, w) must also satisfy Item 2 of Definition 3.6 by comparing Definition A.1 with Item 2 of Definition 3.6.

This shows that T^* is a valid auxiliary tree rooted at v; therefore, T is a generalized $\{2,3\}$-tree in H_Φ satisfying the condition.

The claims $U = V_\star^{\sigma_\ell}$ and $E \subseteq \mathcal{C}_{\mathsf{frozen}}^{\sigma_\ell}$ are trivial from the process. Note that in each step of the process at most Δ vertices in $\mathcal{C}_{\star\text{-frozen}}^{\sigma_\ell}$ or one vertex in $V_\star^{\sigma_\ell}$ are removed. Therefore we have $\Delta \cdot |E| \geq |\mathcal{C}_{\star\text{-frozen}}^{\sigma_\ell}|$. This completes the proof. ☐

Now we are ready to prove Lemma 4.7.

Proof of Lemma 4.7. Note that by Definition 3.5, $f(\sigma_\ell) = \texttt{True}$ says $|V_\star^{\sigma_\ell}| + \Delta \cdot |\mathcal{C}_{\star\text{-frozen}}^{\sigma_\ell}| \geq L\Delta$. We then analyze two cases separately: $\ell = 0$ and $\ell \geq 1$.

For the case when $\ell = 0$, by $\sigma \in \mathcal{Q}^*$ is a partial assignment with exactly one variable $v \in V$ having $\sigma(v) = \star$, it suffices to take $U = \{v\}$ and repeatedly add available constraints and corresponding edges in $\mathcal{C}_{\star\text{-frozen}}^{\sigma_\ell}$ into the auxiliary tree T^* to construct $T = U \circ E$, as in the proof of Lemma A.3, until $\Delta \cdot |E| + |U| \geq L$. Then we have $L \leq \Delta \cdot |E| + |U| \leq L\Delta$ by combining Lemma A.3, the assumption $|V_\star^{\sigma_\ell}| + \Delta \cdot |\mathcal{C}_{\star\text{-frozen}}^{\sigma_\ell}| \geq L\Delta$ and $L + \Delta \leq L\Delta$ from the assumption that $L > 1$ and $\Delta \geq 2$.

Otherwise we have $\ell \geq 1$. By Item 1 of Definition 4.3 we have $\left|V_\star^{\sigma_{\ell-1}}\right| + \Delta \cdot \left|\mathcal{C}_{\star\text{-frozen}}^{\sigma_{\ell-1}}\right| < L\Delta$ and hence by Lemma A.3 there exists a generalized $\{2,3\}$-tree $T' = U' \circ E'$ in H_Φ satisfying $|U'| + \Delta \cdot |E'| < L\Delta$ with some auxiliary tree rooted at v such that $\mathcal{E}_{T'}^{\sigma_{\ell-1}}$ happens. Let $u = \mathsf{NextVar}(\sigma_{\ell-1})$. We then construct $T = U \circ E$ from T' by adding u and the corresponding edge into T^* if $\sigma_\ell(u) = \star$, and then repeatedly adding available constraints in $\mathcal{C}_{\star\text{-frozen}}^{\sigma_\ell}$ and corresponding edges into T^* until $\Delta \cdot |E| + |U| \geq L$. Then we have $L \leq \Delta \cdot |E| + |U| \leq L\Delta$ by combining Lemma A.3, the assumption $|V_\star^{\sigma_\ell}| + \Delta \cdot |\mathcal{C}_{\star\text{-frozen}}^{\sigma_\ell}| \geq L\Delta$ and $L + \Delta \leq L\Delta$ from the assumption that $L > 1$ and $\Delta \geq 2$. ☐

A.2 Proof of Lemma 4.14

Proof of Lemma 4.14. Fix any $0 \leq i \leq \ell$. We claim that for each $0 \leq j < i$,

1. either there exist some c_j, c'_j such that $\mathsf{NextVar}(\sigma_j) \in \mathsf{vbl}(c_j)$, $c'_j \subseteq \mathcal{C}_{\star\text{-frozen}}^{\sigma_i}$, and $\mathsf{vbl}(c_j) \cap \mathsf{vbl}(c'_j) \neq \emptyset$;
2. or there exist some c_j, u_j such that $\mathsf{NextVar}(\sigma_j), u_j \in \mathsf{vbl}(c_j)$ and $u_j \in V_\star^{\sigma_i}$.

Therefore, for each $0 \leq j < i$, $\mathsf{NextVar}(\sigma_j)$ is in a constraint c where either $\mathsf{vbl}(c) \cap \mathsf{vbl}(c') \neq \emptyset$ for some $c' \in \mathcal{C}_{\star\text{-frozen}}^{\sigma_i}$, or $u \in \mathsf{vbl}(c)$ for some $u \in V_\star^{\sigma_i}$. Combining with $|\mathsf{vbl}(c)| \leq k$, we have for each $0 \leq i \leq \ell$,

$$
\begin{aligned}
i &\leq k \cdot |\{c \in \mathcal{C} : \mathsf{vbl}(c) \cap \mathsf{vbl}(c') \neq \emptyset \text{ for some } c' \in \mathcal{C}_{\star\text{-frozen}}^{\sigma_i} \text{ or } u \in \mathsf{vbl}(c) \text{ for some } u \in V_\star^{\sigma_i}\}| \\
&\leq k\Delta \cdot (|\mathcal{C}_{\star\text{-frozen}}^{\sigma_i}| + |V_\star^{\sigma_i}|) \leq k\Delta \cdot (\Delta \cdot |\mathcal{C}_{\star\text{-frozen}}^{\sigma_i}| + |V_\star^{\sigma_i}|).
\end{aligned}
$$

The case when $\ell = 0$ is trivial. We then assume $\ell \geq 1$. By Item 1 of Definition 4.3, we then have two cases.

- If $\Delta \cdot |\mathcal{C}_{\star\text{-frozen}}^{\sigma_\ell}| + |V_\star^{\sigma_\ell}| < L\Delta$, we directly obtain $\ell \leq k\Delta \cdot (\Delta \cdot |\mathcal{C}_{\star\text{-frozen}}^{\sigma_\ell}| + |V_\star^{\sigma_\ell}|) < kL\Delta^2$.
- If $\Delta \cdot |\mathcal{C}_{\star\text{-frozen}}^{\sigma_\ell}| + |V_\star^{\sigma_\ell}| \geq L\Delta$, we have $\Delta \cdot \left|\mathcal{C}_{\star\text{-frozen}}^{\sigma_{\ell-1}}\right| + \left|V_\star^{\sigma_{\ell-1}}\right| < L\Delta$ by Item 1 of Definition 4.3, hence $\ell - 1 < kL\Delta^2$ and therefore $\ell \leq kL\Delta^2$.

Now we prove the claim. Note that by $\mathsf{Path}(\sigma) = (\sigma_0, \ldots, \sigma_\ell)$, $0 \leq i \leq \ell$ and Definition 4.3, we have $\mathsf{NextVar}(\sigma_j) \neq \perp$ for each $0 \leq j < i$. Assume that $\mathsf{NextVar}(\sigma_j) = u_j$. By Definition 3.3, we have $u_j \in V_{\star\text{-inf}}^{\sigma_j} \neq \emptyset$. Combining with the definition of $V_{\star\text{-inf}}^{\sigma_j}$, we have there exists some $c_j \in \mathcal{C}^{\sigma_j}$, $w_j \in V_{\star\text{-con}}^{\sigma_j}$ such that $u_j, w_j \in \mathsf{vbl}(c_j)$. By $w_j \in V_{\star\text{-con}}^{\sigma_j}$, we have $w_j \in V^{\sigma_j} \cap V_{\mathsf{fix}}^{\sigma_j}$. By $w_j \in V^{\sigma_j}$, we have $w_j \notin \Lambda(\sigma_j)$. Combining with $w_j \in V_{\mathsf{fix}}^{\sigma_j}$, we have either $\sigma_j(w_j) = \star$ or $w_j \in \widehat{c}_j$ for some $\widehat{c}_j \in \mathcal{C}_{\mathsf{frozen}}^{\sigma_j}$. If $\sigma_j(w_j) = \star$, we have $w_j \in V_\star^{\sigma_j} \subseteq V_\star^{\sigma_i}$ and c_j, w_j satisfies Item 2. Otherwise, $w_j \in \mathsf{vbl}(\widehat{c}_j)$ for some $\widehat{c}_j \in \mathcal{C}_{\mathsf{frozen}}^{\sigma_j}$. In addition, by $w_j \in V_{\star\text{-con}}^{\sigma_j}$ and $w_j \in \mathsf{vbl}(\widehat{c}_j)$, we have $\widehat{c}_j \in \mathcal{C}_{\star\text{-con}}^{\sigma_j}$. Combining with $\widehat{c}_j \in \mathcal{C}_{\mathsf{frozen}}^{\sigma_j}$, we have $\widehat{c}_j \in \mathcal{C}_{\star\text{-frozen}}^{\sigma_j}$. By $w_j \in \mathsf{vbl}(c_j)$ and $w_j \in \mathsf{vbl}(\widehat{c}_j)$, we have $\mathsf{vbl}(c_j) \cap \mathsf{vbl}(\widehat{c}_j) \neq \emptyset$ and $c_j, \widehat{c}_j$ satisfies Item 1. This justifies the claim. ☐

A.3 Proof of Lemma 4.15

Proof of Lemma 4.15. It is sufficient to show that $|\mathcal{C}^{\sigma_\ell}_{\star\text{-con}}| \le \Delta \cdot |\mathcal{C}^{\sigma_\ell}_{\star\text{-frozen}}| + \Delta \cdot |V^{\sigma_\ell}_{\star}|$. We show this by proving that for each $c \in \mathcal{C}^{\sigma_\ell}_{\star\text{-con}}$, either there exists some $u \in \mathsf{vbl}(c)$ such that $u \in V^{\sigma_\ell}_{\star}$, or there exists some $c' \in \mathcal{C}^{\sigma_\ell}_{\star\text{-frozen}}$ such that $\mathsf{vbl}(c) \cap \mathsf{vbl}(c') \neq \emptyset$.

For each $c \in \mathcal{C}^{\sigma_\ell}_{\star\text{-con}}$, by Definition 3.3, we have there exists some $u \in V^{\sigma_\ell}_{\star\text{-con}} \cap \mathsf{vbl}(c^{\sigma_\ell})$. By $u \in V^{\sigma_\ell}_{\star\text{-con}}$, we have $u \in V^{\sigma_\ell} \cap V^{\sigma_\ell}_{\mathsf{fix}}$. By $u \in V^{\sigma_\ell}$, we have $u \notin \Lambda(\sigma_\ell)$. Combining with $u \in V^{\sigma_\ell}_{\mathsf{fix}}$, we have either $\sigma_\ell(u) = \star$ or $u \in c'$ for some $c' \in \mathcal{C}^{\sigma_\ell}_{\mathsf{frozen}}$. If $\sigma_\ell(u) = \star$, we have $u \in V^{\sigma_\ell}_{\star}$. Otherwise, $u \in c'$ for some $c' \in \mathcal{C}^{\sigma_\ell}_{\mathsf{frozen}}$. In addition, we also have $c' \in \mathcal{C}^{\sigma_\ell}_{\star\text{-con}}$ by $u \in V^{\sigma_\ell}_{\star\text{-con}}$ and $u \in \mathsf{vbl}(c')$. Combining with $c' \in \mathcal{C}^{\sigma_\ell}_{\mathsf{frozen}}$, we have $c' \in \mathcal{C}^{\sigma_\ell}_{\star\text{-frozen}}$. This completes the proof. ☐

A.4 Proof of Lemma 4.16 Let $X = X^n$ where $X^0, X^1, \cdots, X^n$ is the partial assignment sequence in Definition 4.1. The following lemma is immediate by [17, Lemma C.2].

LEMMA A.4. $V \setminus \Lambda(X) \subseteq \mathsf{vbl}\left(\mathcal{C}^X_{\mathsf{frozen}}\right)$.

Now we can prove Lemma 4.16.

Proof of Lemma 4.16. Let $\{\Phi^X_i = (V^X_i, \mathcal{C}^X_i)\} \mid 1 \le i \le K\}$ be the decomposition of Φ^X. If $v \notin V_i$ for each $i \in [k]$, we have $\mathcal{C}^X_v = \emptyset$ and the lemma is trivial. In the following, we assume *w.l.o.g.* that $v \in V^X_i$ for some $i \in K$. Then we have $\Phi^X_v = (V^X_i, \mathcal{C}^X_i)$. Let

$$S \triangleq \left\{c \in \mathcal{C}^X_{\mathsf{frozen}} \mid c^X \in \mathcal{C}^X_i\right\}.$$

At first, we prove that there exists some $c_v \in S$ such that $v \in \mathsf{vbl}(c_v)$. By $v \in V^X_i$, we have $v \notin \Lambda(X)$. Combining with Lemma A.4, we have there exists some $c_v \in \mathcal{C}^X_{\mathsf{frozen}}$ such that $v \in \mathsf{vbl}(c_v)$. In addition, by $v \in \mathsf{vbl}(c_v)$ and $c_v \in \mathcal{C}^X_{\mathsf{frozen}}$, we also have $c^X_v \in \mathcal{C}^X_i$. Combining with $c_v \in \mathcal{C}^X_{\mathsf{frozen}}$, we have $c_v \in S$.

Now we prove $\left|\mathcal{C}^X_i\right| \le \Delta |S|$. For each $c^X \in \mathcal{C}^X_i$, we have there exists a connected path $c^X_1, c^X_2, \cdots, c^X_t = c^X \in \mathcal{C}^X_i$ such that $v \in \mathsf{vbl}\left(c^X_1\right)$. Let $v' \in \mathsf{vbl}\left(c^X\right)$. We have $v' \notin \Lambda(X)$. Combining with Lemma A.4, we have $v' \in \mathsf{vbl}(\widehat{c})$ for some $\widehat{c} \in \mathcal{C}^X_{\mathsf{frozen}}$. Then we have $\widehat{c}^X \in \mathcal{C}^X_i$ because there exists a connected path $c^X_1, c^X_2, \cdots, c^X_t, \widehat{c}^X \in \mathcal{C}^X_i$ where $v \in \mathsf{vbl}\left(c^X_1\right)$. Combining $\widehat{c} \in \mathcal{C}^X_{\mathsf{frozen}}$ with $\widehat{c}^X \in \mathcal{C}^X_i$, we have $c \in S$. In summary, for each each $c^X \in \mathcal{C}^X_i$, there exists some $\widehat{c} \in S$ such that $\mathsf{vbl}\left(c^X\right) \cap \mathsf{vbl}\left(\widehat{c}^X\right) \neq \emptyset$. Thus, we have $\left|\mathcal{C}^X_i\right| \le \Delta |S|$.

In the next, we prove that $G_{\mathsf{VC}}(S)$ is connected. It is enough to prove that any two different constraints $c, \widehat{c} \in S$ are connected in $G_{\mathsf{VC}}(S)$. Given $c, \widehat{c} \in S$, we have $c^X, \widehat{c}^X$ are in $\mathcal{C}^X_i$. Therefore, we have there exists a connected path $c^X = c^X_1, c^X_2, \cdots, c^X_t = \widehat{c}^X \in \mathcal{C}^X_i$. If $t \le 3$, obviously c and $\widehat{c}$ are connected in $G^2(S)$. In the following, we assume that $t > 3$. Let $w_j \in \left(\mathsf{vbl}\left(c^X_j\right) \cap \mathsf{vbl}\left(c^X_{j+1}\right)\right)$ for each $j < t$. Then we have $w_j \notin \Lambda(X)$. Combining with Lemma A.4, we have $w_j \in \mathsf{vbl}\left(\widehat{c}^X_j\right)$ for some $\widehat{c}_j \in \mathcal{C}^X_{\mathsf{frozen}}$. Moreover, we also have $\widehat{c}^X \in \mathcal{C}^X_i$, because $\widehat{c}^X_j$ is connected to c^X through $c^X_2, \cdots, c^X_j \in \mathcal{C}^X_i$. Thus, we have $\widehat{c}_j \in S$. In addition, for each $\widehat{c}_j, \widehat{c}_{j+1}$ where $j < t-1$, we have $\widehat{c}_j$ and $\widehat{c}_{j+1}$ are connected in $G^2(\mathcal{C})$, because $w_j \in \mathsf{vbl}(\widehat{c}_j) \cap \mathsf{vbl}(c_{j+1})$ and $w_{j+1} \in \mathsf{vbl}(\widehat{c}_{j+1}) \cap \mathsf{vbl}(c_{j+1})$. Thus, the constraints $c = c_1, \widehat{c}_1, \widehat{c}_2, \cdots, \widehat{c}_{t-1}, c_t = \widehat{c}$ forms a connected path in G_{VC}. Combining with $\widehat{c}_j \in S$ for each $j \le t-1$ and $c, \widehat{c} \in S$, we have the constraints $c, \widehat{c}$ are connected in $G_{\mathsf{VC}}(S)$.

In summary, we have $c_v \in S \subseteq \mathcal{C}^X_{\mathsf{frozen}}$, $\Delta |S| \ge \left|\mathcal{C}^X_i\right|$ and $G_{\mathsf{VC}}(S)$ is connected. Combing with $v \in \mathsf{vbl}(c_v)$ we have $G_{\mathsf{VC}}(S \cup \{v\})$ is also connected. By going through the process in the proof of Lemma A.3, we have there exists a subset of constraints and vertices $T \subseteq S \cup \{v\}$ such that $T = \{v\} \circ E$ is a generalized $\{2,3\}$-tree in H_Φ with some auxiliary tree rooted at v and

$$|E| \ge |S| / \Delta \ge \left|\mathcal{C}^X_i\right| / \Delta^2 = \left|\mathcal{C}^X_v\right| / \Delta^2.$$

In addition, if $\left|\mathcal{C}^X_v\right| \ge L\Delta^2$, then similar to the proof of Lemma 4.7,we take $U = \{v\}$ and repeatedly add available constraints and corresponding edges in S into the auxiliary tree T^* to construct $T = U \circ E$ until $\Delta \cdot |E| + |U| \ge L$. Then we have $L \le \Delta \cdot |E| + |U| \le L\Delta$ by combining $S \ge \left|\mathcal{C}^X_v\right| / \Delta \ge L\Delta$ and $L + \Delta \le L\Delta$ from the assumption that $L > 1$ and $\Delta \ge 2$, and the lemma is immediate. ☐