

# Exponential Quantum Space Advantage for Approximating Max- $k$ SAT in the Streaming Setting

Haoyu Wang\*      Guangxu Yang†

June 3, 2026

## Abstract

In this paper, we give a one-pass quantum streaming algorithm for Max- $k$ SAT that uses  $\text{polylog}(n)$  space and achieves a 0.7172-approximation on instances with  $n$  variables. In contrast, prior work by Chou, Golovnev, and Velusamy (FOCS 2020) implies that achieving an approximation ratio better than  $\sqrt{2}/2 \approx 0.7071$  for Max- $k$ SAT requires  $\Omega(\sqrt{n})$  space for any classical streaming algorithm. Therefore, it yields an exponential quantum space advantage for Max- $k$ SAT in the streaming setting.

We further give a one-pass quantum streaming algorithm for Max-2OR that uses  $\text{polylog}(n)$  space and achieves a 0.7425-approximation on instances with  $n$  variables. Combining with the known results, it gives a complete classification of quantum space advantages for all Boolean Max-2CSPs.

---

\*Penn State University. Email: [hjw5492@psu.edu](mailto:hjw5492@psu.edu).

†University of Southern California. Email: [guangxuy@usc.edu](mailto:guangxuy@usc.edu). Research supported by NSF CAREER award 2141536.

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Our contributions . . . . .	3
1.2	Related work . . . . .	5
1.3	Technical overview . . . . .	5
<b>2</b>	<b>Preliminaries</b>	<b>7</b>
<b>3</b>	<b>Reducing Max-<math>k</math>SAT to weighted snapshot estimation</b>	<b>8</b>
<b>4</b>	<b>Closeness of the Weighted Snapshot and Pseudosnapshot</b>	<b>13</b>
<b>5</b>	<b>Quantum algorithm for weighted pseudosnapshot estimation</b>	<b>19</b>
5.1	Quantum register . . . . .	21
5.2	One-endpoint estimation . . . . .	22
5.2.1	High-degree case . . . . .	24
5.2.2	Low-degree case . . . . .	31
5.3	Two-endpoint estimation . . . . .	33
5.3.1	High-degree on both endpoints . . . . .	34
5.3.2	Other cases . . . . .	40
5.4	Estimating the weighted pseudosnapshot score . . . . .	44
<b>6</b>	<b>Quantum algorithm for Max-<math>k</math>SAT</b>	<b>50</b>
<b>A</b>	<b>Quantum space lower bound for Max-<math>k</math>SAT</b>	<b>53</b>
<b>B</b>	<b>Exact finite LP certificate for Max-<math>k</math>SAT</b>	<b>54</b>
<b>C</b>	<b>Finite LP certificate for approximation ratio of Max-2OR</b>	<b>54</b>

# 1 Introduction

Streaming algorithms provide a framework for processing massive datasets, especially in settings where the input is too large to be stored in memory [BBD<sup>+</sup>02, Mut05]. In this model, data elements arrive sequentially, and the algorithm must process each element as it appears while using only limited working space. Because of their space efficiency, streaming algorithms have found many applications, including the computation of statistics over data streams [FM85, AMS99] and the estimation of parameters of massive graphs [BYKS02, McG14].

Quantum computing offers the possibility of large resource advantages over classical systems. The main focus has traditionally been time complexity [FGG14], for example, Shor’s algorithm for integer factorization [Sho99] gives an exponential speedup over the best-known classical algorithms. However, proving unconditional exponential time separations from all classical algorithms remains a major challenge [BV97, Wat18]. Space is another crucial resource, especially in the quantum setting, where fault-tolerant qubits are scarce and costly [FMMC12, BMN<sup>+</sup>21]. Quantum streaming algorithms provide a natural framework for studying space-efficient quantum computation and for seeking quantum advantages in space usage [Gal06, NT17, Kal22].

Provable exponential space advantages for quantum streaming algorithms were first obtained in the seminal work of Gavinsky, Kempe, Kerenidis, Raz, and de Wolf [GKK<sup>+</sup>08]. Their problem is designed for this separation, and leaves open whether such advantages could be obtained for natural problems of independent classical interest. Several natural candidates such as Dyck(2) language [JN14, NT17] and triangle counting [Kal22] were subsequently studied. Kallaugher, Parekh, and Voronova achieved the first exponential quantum space advantage for a natural streaming problem [KPV24]. They gave a polylogarithmic-space quantum streaming algorithm that achieves a 0.4844-approximation for Max-DiCut. In contrast, previous work of Chou, Golovnev, and Velusamy [CGV20] implies that any classical streaming algorithm achieving an approximation ratio better than  $4/9 \approx 0.444$  requires  $\Omega(\sqrt{n})$  space.

Since Max-DiCut is only one particular Max-CSP, it naturally raises a broader question about the scope of quantum space advantages for streaming CSPs.

*Which constraint satisfaction problems (CSPs) admit an exponential quantum space advantage in the streaming model?*

In this paper, we prove an exponential quantum space advantage for Max- $k$ SAT in the streaming setting. Max- $k$ SAT is one of the most standard and widely studied Max-CSPs, where each constraint is a disjunction of at most  $k$  literals. Our result also completes the classification of quantum space advantages for Boolean Max-2CSPs when combined with prior work.

## 1.1 Our contributions

Our main contribution is a polylogarithmic-space one-pass quantum streaming algorithm for Max- $k$ SAT with approximation ratio 0.7172.

**Theorem 1.** *For every fixed  $k \geq 2$  and every  $\delta \in (0, 1)$ , there is a one-pass quantum streaming algorithm that, on every Max- $k$ SAT instance with  $n$  variables, outputs a 0.7172-approximation with probability at least  $1 - \delta$ , using  $O(\log^5 n \log \frac{1}{\delta})$  qubits of space.*

In contrast, prior work by [CGV20] implies that achieving an approximation ratio better than  $\sqrt{2}/2 \approx 0.7071$  for Max- $k$ SAT requires  $\Omega(\sqrt{n})$  space for any classical streaming algorithm.

To better understand our result, we classify 2-variable Boolean predicates  $f: \{0, 1\}^2 \rightarrow \{0, 1\}$  into four standard types based on their truth tables: TR (depends on  $\leq 1$  input), OR (exactly

Type $\mathcal{G}$	Special case	Classical	Quantum	Advantage
TR	Trivial	1 [Folklore]	1 [Folklore]	No
OR	Max-2EOR	3/4 [CGV20]	3/4 [KP22]	No
TR + OR	Max-2SAT	$\sqrt{2}/2$ [CGV20]	<b>0.7425, 3/4</b>	<b>Yes</b>
XOR	Max-Cut	1/2 [CGV20]	1/2 [KP22]	No
AND	Max-DiCut	4/9 [CGV20]	[0.4844, 1/2] [KPV24]	Yes

Table 1: Classical and quantum one-pass streaming thresholds for Boolean Max-2CSPs with polylogarithmic space. The “Classical” and “Quantum” columns list the known threshold values from the cited works; an interval means that the quantum threshold is known to lie between the displayed lower and upper bounds. The last column indicates whether the quantum threshold is strictly better than the classical one. The bold entry is the new quantum advantage for Max-2SAT established in this work. Chou, Golovnev, and Velusamy [CGV20] show that the streaming approximability of every Boolean Max-2CSP is governed by the five canonical problems. More precisely, for any predicate family  $\mathcal{F}$ , Max-CSP( $\mathcal{F}$ ) is exactly as hard to approximate as the hardest problem in these predicates.

one zero), XOR (depends on both inputs; exactly two zeros), and AND (exactly three zeros). For any  $\Lambda \in \{\text{OR}, \text{XOR}, \text{AND}\}$ , the Max-2CSP problem is denoted Max-2E $\Lambda$  if its allowed predicate set  $\mathcal{F}$  contains only  $\Lambda$ -type predicates (exact arity two), and Max-2 $\Lambda$  if  $\mathcal{F}$  also includes TR-type predicates. Identifying a predicate family with its contained types, every Boolean Max-2CSP is thus uniquely specified by a subset  $\mathcal{F} \subseteq \{\text{TR}, \text{OR}, \text{XOR}, \text{AND}\}$ .

Theorem 1 also gives a quantum streaming algorithm for Max-2OR, since Max-2OR is equivalent to Max-2SAT. We also optimize the approximation ratio for Max-2OR. The proof is deferred to Appendix C.

**Corollary 1.** *For every  $\delta \in (0, 1)$ , there is a one-pass quantum streaming algorithm that, on every Max-2OR instance, with  $n$  variables, outputs a 0.7425-approximation with probability at least  $1 - \delta$ , using  $O(\log^5 n \log \frac{1}{\delta})$  qubits of space.*

Combining Corollary 1 with the known results [CGV20, KP22] gives a complete classification of quantum space advantages for Boolean Max-2CSPs, summarized in Table 1. Previous work had already settled several cases.

- For pure OR-type constraints (Max-2EOR) and XOR-type constraints (Max-Cut), quantum streaming algorithms satisfy the same space lower bounds as classical algorithms [KP22], and hence give no quantum space advantage.
- For AND-type constraints (Max-DiCut), Kallaugher, Parekh, and Voronova give a polylogarithmic-space quantum streaming algorithm achieving a 0.4844-approximation [KPV24], whereas the classical lower bound of Chou, Golovnev, and Velusamy rules out any ratio better than 4/9 in  $o(\sqrt{n})$  space [CGV20].

Thus, only the TR+OR case, corresponding to Max-2OR / Max-2SAT, remained open. Our result closes this gap by showing that this problem also has a quantum space advantage in the streaming setting.

To complement our results, we also prove a quantum streaming lower bound for Max- $k$ SAT. This is achieved via a simple reduction from Max-Cut, leveraging its known quantum streaming space lower bound [KP22]. The proof is deferred to Appendix A.

**Theorem 2** (Quantum streaming lower bound for Max- $k$ SAT). *Fix any  $k \geq 2$  and  $\gamma > 0$ , any one-pass quantum streaming algorithm that achieves a  $(0.75 + \gamma)$ -approximation for Max- $k$ SAT requires  $\Omega(n)$  qubits of space.*

Theorem 2 also holds for Exact Max- $k$ SAT. As a byproduct, it separates Max- $k$ SAT from Exact Max- $k$ SAT in terms of quantum space advantage.

## 1.2 Related work

**Classical streaming bounds for Max- $k$ SAT** The streaming approximability of CSPs has been studied for more than a decade; see the surveys and expository articles [Sud22, Ass23, Sin25]. Chou, Golovnev, and Velusamy [CGV20] established a tight space complexity threshold for Max- $k$ SAT around the approximation ratio of  $\sqrt{2}/2 \approx 0.707$ . Specifically, they demonstrated that a  $(\sqrt{2}/2 - \varepsilon)$ -approximation streaming algorithm for Max- $k$ SAT can be achieved using only  $O(\varepsilon^{-2} \log n)$  space. On the negative side, they proved a matching lower bound: for any  $k \geq 2$  and  $\varepsilon > 0$ , any streaming algorithm achieving a strictly better approximation ratio of  $(\sqrt{2}/2 + \varepsilon)$  for Max- $k$ SAT requires  $\Omega(\sqrt{n})$  space.

**Quantum streaming algorithms for Max-CSPs** Quantum streaming algorithms for CSP-type problems were previously studied by Kallaugher and Parekh [KP22], who showed any quantum streaming algorithms need  $\Omega(n)$  space to beat the trivial  $1/2$  approximation for Max-Cut problem. In contrast, Kallaugher, Parekh, and Voronova recently proved an exponential quantum space advantage for Max-DiCut [KPV24]: they gave a polylogarithmic-space quantum streaming algorithm with approximation ratio 0.4844, while the classical lower bound of [CGV20] rules out any ratio better than  $4/9 \approx 0.4444$  in  $o(\sqrt{n})$  space.

## 1.3 Technical overview

Our proof extends the snapshot-to-pseudosnapshot framework of Kallaugher, Parekh, and Voronova [KPV24] from Max-DiCut to Max- $k$ SAT. At a high level, we first reduce Max- $k$ SAT to estimating a weighted snapshot, then replace this final-stream object by a one-pass pseudosnapshot, and finally estimate the pseudosnapshot using quantum sketches.

**Step 1: From Max- $k$ SAT to a weighted snapshot.** The first step is to reduce Max- $k$ SAT to estimating a weighted snapshot. The snapshot is designed for a fixed oblivious rounding rule: each variable is placed into a bucket according to its weighted signed bias, and the rounding probability of each literal is then determined only by this bucket and the literal sign.

Snapshot-based estimation was introduced for Max-DiCut by Saxena, Singer, Sudan, and Velusamy [SSSV23], who asked whether the approach can be extended beyond Max-DiCut. We answer this question for Max- $k$ SAT. Interestingly, combining our approach with the streaming algorithm of [SSSV23] also yields a 0.7425-approximation classical streaming algorithm for Max-2SAT in  $\tilde{O}(\sqrt{n})$  space, thereby resolving an open problem from [CGV20]. Since this paper focuses on quantum advantage, we omit the details of this classical variant.

An edge in Max-DiCut has only two ordered endpoints, whereas a SAT clause is characterized by both its length and the signs of its literals. We therefore defined a *weighted snapshot*. For each variable, we compute a weighted signed bias from its occurrences in clauses with length at most three; the weights make endpoints in shorter clauses count more. Each endpoint is then assigned a sign-bucket pair, consisting of its literal sign and the bias bucket of its variable, and the weighted snapshot  $\text{WSnap}(\Phi)$  records the resulting histograms over these pairs.

The goal of this step is to construct a linear score  $L_{\leq k}(\text{WSnap}(\Phi))$  such that

$$\rho \text{OPT}(\Phi) \leq L_{\leq k}(\text{WSnap}(\Phi)) \leq \text{OPT}(\Phi).$$

Compared with [KPV24], the main barrier is that in Max- $k$ SAT, clauses have different lengths. As the number of endpoints processed simultaneously increases, the operations and analysis associated with quantum algorithms become extremely complex and intractable. In order to bypass this technical barrier, we handle clauses of length at least four separately. Under the certified rounding rule, every such clause is satisfied with probability at least the target ratio  $\rho$ . Thus these long clauses get the fixed contribution  $\rho m_{\geq 4}$ , and all nontrivial snapshot work is only for clauses of length at most three. In other words,

$$L_{\leq k}(\text{WSnap}(\Phi)) = L_{\leq 3}(\text{WSnap}(\Phi_{\leq 3})) + \rho m_{\geq 4}.$$

The main difficulty inside  $L_{\leq 3}(\text{WSnap}(\Phi_{\leq 3}))$  is the ternary case. The exact satisfaction probability of a ternary clause depends on three endpoints at once, but directly querying three endpoints would still make the quantum bookkeeping hard to control<sup>1</sup>. We therefore replace the exact ternary score by a pairwise-decomposable surrogate, namely a linear combination of one-endpoint and two-endpoint quantities. The unary and binary scores are kept as their exact independent-rounding satisfaction probabilities.

We certify that this surrogate is still good enough: the resulting short-clause score  $L_{\leq 3}(\text{WSnap}(\Phi))$  achieves ratio  $\rho = 0.717275$ . This is done by solving a finite linear program and verifying the resulting rational certificate exactly over  $\mathbb{Q}$ .

**Step 2: The weighted snapshot and pseudosnapshot are close.** The weighted snapshot uses final variable biases, which are only known after the stream ends. As in [KPV24], we replace it by a weighted *pseudosnapshot*, whose endpoint sign-bucket pairs are assigned when a clause arrives so that the it can be estimated in one pass.

The idea is to use prefix information at the time of arrival together with exact suffix information: the sketch encodes the prefix before the query, and once a query selects a endpoint, the remaining suffix counts can be computed exactly.

For Max- $k$ SAT, the data being tracked are different from Max-DiCut. We must track tagged total degrees and tagged positive counts, instead of only directed degree and out-degree quantities, and the same sign-bucket assignment rule must handle unary, binary, and ternary clauses. Our modification is to duplicate each endpoint from a short clause into multiple copies, called tagged units. If the clause length is  $\ell$ , the endpoint creates  $w_\ell$  copies; this multiplicity records the clause length. Each copy also carries a positive/negative tag, recording the literal sign. Thus every variable has a tagged total degree and a tagged positive count, which play the role of the degree and out-degree quantities in [KPV24]. We then use the same high-degree/low-degree split: On high-degree scales, the prefix total degree is rounded to the lower end of its scale, and the positive prefix count is sampled randomly. On low-degree scales, both prefix quantities are kept exactly.

To compare these arrival-time sign-bucket pairs with a fixed bucket assignment, we add smoothing noise to the final biases and prove that the pseudosnapshot is  $O(\varepsilon m)$ -close to the fixed-label snapshot in  $\ell_1$  distance. The fixed bucket assignment obtained this way may not satisfy the original bucket constraints exactly; they are only  $\varepsilon$ -admissible. We therefore prove a robust version of the finite certificate, which tolerates these approximate labels and loses only  $O(\varepsilon m)$  in the snapshot score.

---

<sup>1</sup>Our focus is on demonstrating a quantum advantage rather than optimizing the ration  $\rho = 0.7172$ ; better score functions and more involved quantum operations may further improve the approximation ratio.

**Step 3: Quantum estimation of the pseudosnapshot.** The third step is to estimate the weighted pseudosnapshot in one pass. Kallaugher, Parekh, and Voronova [KPV24] estimate the Max-DiCut pseudosnapshot by running a hidden-matching-style sketch separately for each pair of head/tail degree ranges. Their sketch maintains four degree-test families of basis states, corresponding to lower and upper degree tests for the head endpoint and lower and upper degree tests for the tail endpoint; sampled out-degree information is encoded in separated counter blocks. Edge updates are implemented by swaps. After each edge measurement, a cleanup measurement handles boundary basis states so that the state keeps the invariant needed for the lower/upper inclusion-exclusion over the queried degree ranges.

We follow the same high-level idea, but use a different quantum implementation. For the current clause, our sketch tests either one endpoint or a pair of endpoints. Each test asks whether certain threshold basis states are present in the current superposition. The updates are elementary shifts along threshold-state chains, and the measurements are only single-threshold or pair-threshold queries. Thus the coordinates of the weighted pseudosnapshot are estimated by the same basic operation: test whether the relevant threshold basis states are present when the clause arrives.

For each tested endpoint, the sketch needs two prefix numbers of the underlying variable: the tagged total prefix degree  $d$  and a positive prefix count  $c$ . On high-degree scales,  $c$  is the sampled positive prefix count; on low-degree scales, it is the exact positive prefix count. To implement the update by shifts, we encode these two numbers into one counter

$$\Gamma(d, c) = d + Bc,$$

where  $B$  is chosen randomly for the current degree range. The price of this simple encoding is impersonation. An endpoint whose true prefix degree is outside the queried scale may still land on a queried level of  $\Gamma$  and be counted by mistake. The random choice of  $B$  makes these accidental hits rare. We charge impersonation and show that its total contribution is  $O(\varepsilon m)$ . This avoids the separate cleanup measurement used in [KPV24].

Combining the first three steps gives a constant-success one-pass quantum algorithm for Max- $k$ SAT. Finally, we amplify over independent public seeds and take a median, obtaining success probability  $1 - \delta$  while keeping polylogarithmic space.

## 2 Preliminaries

**Definition 1.** For an instance  $\Phi$  of a Max-CSP( $\Pi$ ) problem, we denote the number of constraints in  $\Phi$  by  $m = |\Phi| = \text{poly}(n)$ . We denote the set of Boolean variables of  $\Phi$  by  $\mathcal{V} = \{v_1, \dots, v_n\}$ . A constraint in  $\Phi$  consists of a predicate from the constraint language  $\Pi$  applied to a tuple of variables from  $\mathcal{V}$ .

For an assignment  $\sigma : \mathcal{V} \rightarrow \{0, 1\}$  of the variables of  $\Phi$ , we denote the number of constraints of  $\Phi$  satisfied by  $\sigma$  as  $\text{Val}_\Phi(\sigma)$ . We denote the maximum number of simultaneously satisfiable constraints in  $\Phi$  as  $\text{OPT}(\Phi)$ :

$$\text{OPT}(\Phi) = \max_{\sigma} \text{Val}_\Phi(\sigma).$$

For  $\alpha \in [0, 1]$ , an algorithm  $\mathcal{A}$  is an  $\alpha$ -approximation to the Max-CSP( $\Pi$ ) problem if on any input  $\Phi$ ,  $\mathcal{A}$  outputs  $Z$ , such that with probability  $1 - \delta$ , it holds that

$$\text{OPT}(\Phi) \geq Z \geq \alpha \cdot \text{OPT}(\Phi).$$

For example, when  $\alpha = 1$ , the algorithm solves Max-CSP( $\Pi$ ) exactly with probability  $1 - \delta$ .

Suppose each variable  $v \in \mathcal{V}$  has been assigned a bucket index  $i(v) \in \{0, \dots, L-1\}$ . Given a rounding vector  $r = (r_0, \dots, r_{L-1}) \in [0, 1]^L$ , let  $\mathcal{R}_r(\Phi)$  be the product distribution over assignments defined by

$$\Pr[X_v = 1] = r_{i(v)}, \quad \Pr[X_v = 0] = 1 - r_{i(v)},$$

independently for all  $v \in \mathcal{V}$ . The independent rounding value of  $\Phi$  with respect to  $r$  is

$$\mathcal{E}_r(\Phi) = \mathbb{E}_{\sigma \sim \mathcal{R}_r(\Phi)} [\text{Val}_\Phi(\sigma)].$$

As a well-known specific example, the Max- $k$ SAT problem is a special case of Max-CSP where a constraint is a clause, and a clause in  $\Phi$  is a disjunction (OR) of at most  $k$  literals (where a literal is either a variable  $x_i$  or its negation  $\neg x_i$ ).

**Quantum streaming for Max-CSP** In this paper, we consider the Max-CSP( $\Pi$ ) problem processed by a one-pass quantum streaming algorithm. The algorithm has sequential access to an input instance  $\Phi$ , which it processes left-to-right using a limited quantum workspace.

First, the algorithm receives the number of variables  $n$  and the stream length  $m$  (equivalently, any constant-factor upper bound on  $m$ ). It then initializes a quantum work register  $W$  consisting of  $s(n)$  qubits to a fixed state  $\rho_0$ , typically  $\rho_0 = |0^{s(n)}\rangle\langle 0^{s(n)}|$ .

The algorithm receives the constraints  $C_1, \dots, C_m$  in  $\Phi$  one by one (in a possibly adversarial order). Let  $\Sigma$  be the set of all possible constraints on  $n$  variables. For each constraint  $\sigma \in \Sigma$ , the algorithm possesses a corresponding quantum channel  $\mathcal{A}_\sigma$  acting on  $W$ . As the stream is processed, the state of the work register evolves as:

$$\rho_i = \mathcal{A}_{C_i}(\rho_{i-1}), \quad i = 1, 2, \dots, m.$$

After the last update, the algorithm measures the work register  $W$ . Based on the measurement outcome, it outputs a value  $Z$ .

Throughout this framework, the space complexity of the algorithm is the number of qubits  $s(n)$  stored in the work register between two consecutive constraint stream updates, where  $s$  is typically a small function of  $n$ . We place no restriction on the running time or computational complexity of the quantum update channels. We also allow the algorithm read-only access to public random bits, and these random bits are not stored in the work register and are not charged to the streaming space.

### 3 Reducing Max- $k$ SAT to weighted snapshot estimation

In this section, we reduce Max- $k$ SAT to the problem of estimating a weighted snapshot. The snapshot is designed to support a fixed oblivious rounding rule: each variable is first assigned a type according to its signed bias, and the rounding probability of each literal is then determined only by this type and by the literal sign.

We first simplify clauses. First, we can delete the repeated literals. Clauses that contain both a literal and its negation are always satisfied, so they are removed from the stream and accounted for by an exact offset maintained separately. For the remaining simplified instance  $\Phi$ , let

$$\Phi_{\leq 3} := \{C \in \Phi : |C| \in \{1, 2, 3\}\}, \quad \Phi_{\geq 4} := \{C \in \Phi : |C| \geq 4\}, \quad m_{\geq 4} := |\Phi_{\geq 4}|.$$

Only the short-clause subinstance  $\Phi_{\leq 3}$  is used to define these variable types and the snapshot score. Clauses in  $\Phi_{\geq 4}$  do not enter the snapshot; since they have at least four literals, they are handled separately by a deterministic floor  $\rho \cdot m_{\geq 4}$ , where  $\rho$  is a constant determined later.

Before defining the "snapshot" of instances, we first define the bias of the variables in  $\Phi_{\leq 3}$ .

**Definition 2.** For  $\ell \in \{1, 2, 3\}$ , let  $P_v^{(\ell)}$  and  $N_v^{(\ell)}$  denote the numbers of positive and negative occurrences of  $v$  in  $\ell$ -clauses of  $\Phi_{\leq 3}$ . Set

$$p_v := 4P_v^{(1)} + 2P_v^{(2)} + P_v^{(3)}, \quad q_v := 4N_v^{(1)} + 2N_v^{(2)} + N_v^{(3)}, \quad d_v := p_v + q_v.$$

The bias of  $v$  is defined as

$$b_v := \begin{cases} (p_v - q_v)/d_v, & d_v > 0, \\ 0, & d_v = 0. \end{cases}$$

Note that  $b_v \in [-1, 1]$ . The optimized weights  $(4, 2, 1)$  determine how strongly unary, binary, and ternary endpoints contribute to the signed type of a variable.

The next definition turns these signed types into a rounding rule.

**Definition 3** (Short-clause rounding probability). Let  $\mathcal{I} = \{I_0, \dots, I_{L-1}\}$  be a partition of  $[-1, 1]$  into a constant number of intervals, written with deterministic tie-breaking as

$$I_i = [\ell_i, u_i) \quad (i < L - 1), \quad I_{L-1} = [\ell_{L-1}, u_{L-1}].$$

Let  $r = (r_0, \dots, r_{L-1}) \in [0, 1]^L$  be a vector of rounding probabilities. For a literal sign  $s \in \{+, -\}$  and a bucket  $i \in \{0, \dots, L - 1\}$ , define

$$q_+(i) := r_i, \quad q_-(i) := 1 - r_i.$$

For a label  $\alpha = (s, i)$ , write  $q_\alpha := q_s(i)$ .

Once  $[-1, 1]$  is partitioned into buckets, a variable  $v$  with  $b_v \in I_i$  is rounded to true with probability  $r_i$ . Thus, a positive literal is satisfied with probability  $r_i$ , while a negative literal is satisfied with probability  $1 - r_i$ . The snapshot will only remember the literal sign and the bucket index, not the exact value of  $b_v$ .

After the rounding probabilities are fixed, every endpoint of a short clause has a constant-size label  $\alpha = (s, i)$ : the literal sign  $s$ , together with the bucket  $i$  of its variable. The snapshot below records only the histograms of these labels that will be needed. We only keep one-endpoint and two-endpoint projections, because these are the statistics that the quantum estimator can handle.

**Definition 4** (Weighted snapshot). Let  $i(v)$  be the unique bucket containing  $v$ 's bias  $b_v$ . Order every clause canonically, say by increasing variable index. Denote the variable on the  $j$ -th endpoint of clause  $C$  by  $v_j(C)$ . Write the sign-bucket pair by

$$\alpha_j(C) = (s_j, i(v_j(C))) \in \{+, -\} \times \{0, \dots, L - 1\},$$

where  $s_j$  is the literal sign of the  $j$ -th endpoint. The weighted short snapshot  $\text{WSnap}(\Phi_{\leq 3})$  consists of:

- one-endpoint coordinates

$$U_{\ell, j, \alpha}(\Phi) := \#\{C \in \Phi : |C| = \ell, \alpha_j(C) = \alpha\}, \quad 1 \leq j \leq \ell \leq 3;$$

- two-endpoint coordinates

$$P_{\ell, j, t, \alpha, \beta}(\Phi) := \#\{C \in \Phi : |C| = \ell, \alpha_j(C) = \alpha, \alpha_t(C) = \beta\}, \quad 1 \leq j < t \leq \ell \leq 3;$$

- the ternary clause count  $M_3(\Phi) := \#\{C \in \Phi : |C| = 3\}$ .

The true satisfaction probability of a ternary clause,  $1 - \prod_{j=1}^3 (1 - q_{\alpha_j})$ , is a three-variable function. If we used it directly, the snapshot would have to store a full three-dimensional histogram of ternary endpoint labels. To avoid using ternary endpoint labels, we define a pairwise-decomposable score  $H$  that relies only on a one-endpoint function,

$$c_0 \in \mathbb{Q}, \quad u_j : \{+, -\} \times \{0, \dots, L-1\} \rightarrow \mathbb{Q} \quad (j = 1, 2, 3)$$

and a two-endpoint function,

$$p_{jt} : (\{+, -\} \times \{0, \dots, L-1\})^2 \rightarrow \mathbb{Q} \quad (1 \leq j < t \leq 3).$$

**Definition 5** (Pairwise-decomposable function). The function  $H : (\{+, -\} \times \{0, \dots, L-1\})^3 \rightarrow \mathbb{R}$  is defined by

$$H(\alpha_1, \alpha_2, \alpha_3) := c_0 + \sum_{j=1}^3 u_j(\alpha_j) + \sum_{1 \leq j < t \leq 3} p_{jt}(\alpha_j, \alpha_t).$$

Thus, the total ternary contribution can be written using only  $M_3$ , the one-endpoint marginals  $U_{3,j,\alpha}$ , and two-endpoint marginals  $P_{3,j,t,\alpha,\beta}$ .

We can now assign a numerical value to a weighted snapshot.

**Definition 6** (Weighted snapshot score). Fix the parameters  $\Theta = (\mathcal{I}, r, c_0, \{u_j\}_{j=1,2,3}, \{p_{jt}\}_{1 \leq j < t \leq 3})$ . For a short instance  $\Phi_{\leq 3}$ , its weighted snapshot score is

$$\begin{aligned} L_{\leq 3}^{\Theta}(\text{WSnap}(\Phi_{\leq 3})) &:= \sum_{\alpha} U_{1,1,\alpha} q_{\alpha} + \sum_{\alpha,\beta} P_{2,1,2,\alpha,\beta} (1 - (1 - q_{\alpha})(1 - q_{\beta})) \\ &+ c_0 M_3 + \sum_{j=1}^3 \sum_{\alpha} u_j(\alpha) U_{3,j,\alpha} + \sum_{1 \leq j < t \leq 3} \sum_{\alpha,\beta} p_{jt}(\alpha, \beta) P_{3,j,t,\alpha,\beta}. \end{aligned}$$

For a full simplified Max- $k$ -SAT instance  $\Phi$ , define

$$L_{\leq k}^{\Theta}(\Phi) := L_{\leq 3}^{\Theta}(\text{WSnap}(\Phi_{\leq 3})) + \rho m_{\geq 4}.$$

The first line scores unary and binary clauses by their exact independent-rounding satisfaction probabilities. The second line scores ternary clauses using the pairwise-decomposable function  $H$ , expanded in terms of the coordinates stored in  $\text{WSnap}(\Phi_{\leq 3})$ .

**Lemma 1** (Finite short-clause certificate). *There exists a tuple  $\Theta^* = (\mathcal{I}^*, r^*, c^*, u^*, p^*)$  such that every short instance  $\Phi_{\leq 3}$  satisfies*

$$\rho \text{OPT}(\Phi_{\leq 3}) \leq L_{\leq 3}^{\Theta^*}(\text{WSnap}(\Phi_{\leq 3})) \leq \mathcal{E}_{r^*}(\Phi_{\leq 3}).$$

where  $\rho = 0.717275$  and  $\mathcal{E}_{r^*}(\Phi_{\leq 3})$  denotes the expected number of clauses satisfied by the independent randomized assignment induced by  $r^*$ .

*Proof. Step 1: Finding conditions for upper bound:* Fix any parameter tuple  $\Theta = (\mathcal{I}, r, c_0, u, p)$ . For unary and binary clauses, the score appearing in  $L_{\leq 3}^{\Theta}(\text{WSnap}(\Phi_{\leq 3}))$  is already the exact independent-rounding satisfaction probability. Therefore, the only extra condition needed for the upper bound concerns ternary clauses.

For a ternary clause, if  $\Theta$  satisfies

$$H_{\Theta}(\alpha_1, \alpha_2, \alpha_3) \leq 1 - \prod_{j=1}^3 (1 - q_{\alpha_j}) \quad \text{for every ternary label triple } (\alpha_1, \alpha_2, \alpha_3),$$

then  $L_{\leq 3}^{\Theta}(\text{WSnap}(\Phi_{\leq 3}))$  is at most the expected number of satisfied clauses  $\mathcal{E}_{\tau}(\Phi_{\leq 3})$  under the independent randomized assignment induced by  $\Theta$ .

*Step 2: Finding conditions for lower bound:* Fix an optimal assignment  $x^*$  for the short instance  $\Phi$ . We refine each clause into *typed atoms*. A typed atom  $c$  consists of a clause length  $\ell \in \{1, 2, 3\}$ , endpoint signs  $(s_1, \dots, s_{\ell}) \in \{+, -\}^{\ell}$ , endpoint buckets  $(i_1, \dots, i_{\ell}) \in \{0, \dots, L-1\}^{\ell}$ , and reference-assignment bits  $(\tau_1, \dots, \tau_{\ell}) \in \{0, 1\}^{\ell}$ . We write  $c = (\ell; s_1, \dots, s_{\ell}; i_1, \dots, i_{\ell}; \tau_1, \dots, \tau_{\ell})$ . For  $\alpha_j = (s_j, i_j)$ , define the local snapshot score of  $c$  by

$$h_{\Theta}(c) := \begin{cases} q_{\alpha_1}, & \ell = 1, \\ 1 - (1 - q_{\alpha_1})(1 - q_{\alpha_2}), & \ell = 2, \\ H_{\Theta}(\alpha_1, \alpha_2, \alpha_3), & \ell = 3. \end{cases}$$

Its optimal-assignment contribution is

$$o(c) := \mathbf{1} \left[ \bigvee_{j=1}^{\ell} \text{Lit}(s_j, \tau_j) = 1 \right],$$

where  $\text{Lit}(+, \tau) = \tau$ ,  $\text{Lit}(-, \tau) = 1 - \tau$ .

For each typed atom  $c$ , let  $N_c$  be the number of clauses of  $\Phi$  whose length, endpoint signs, endpoint buckets, and endpoint bits under  $x^*$  are described by  $c$ . Then the typed atoms give the exact decomposition

$$L_{\leq 3}^{\Theta}(\text{WSnap}(\Phi_{\leq 3})) = \sum_c N_c h_{\Theta}(c), \quad \text{OPT}(\Phi_{\leq 3}) = \text{Val}_{\Phi}(x^*) = \sum_c N_c o(c).$$

Now fix a bucket  $I_i = [\ell_i, u_i]$  and a bit  $\tau \in \{0, 1\}$ . Let  $D_{i,\tau} = \sum_{v: i(v)=i, x_v^*=\tau} (p_v + q_v)$  be the total endpoint mass in bucket  $i$  among variables whose optimal bit is  $\tau$ , and let  $A_{i,\tau} = \sum_{v: i(v)=i, x_v^*=\tau} (p_v - q_v)$  be the corresponding signed endpoint mass. Endpoint masses use the same weights as the bias:  $w_1 = 4, w_2 = 2, w_3 = 1$ .

Writing  $\chi(+)=1$  and  $\chi(-)=-1$ , an endpoint of an  $\ell$ -clause with sign  $s_j$ , bucket  $i_j$ , and optimal bit  $\tau_j$  contributes  $w_{\ell}$  to  $D_{i_j, \tau_j}$  and  $w_{\ell} \chi(s_j)$  to  $A_{i_j, \tau_j}$ . The bucket-feasibility constraints are

$$g_{i,\tau}^{\text{lo}} := \ell_i D_{i,\tau} - A_{i,\tau} \leq 0, \quad g_{i,\tau}^{\text{up}} := A_{i,\tau} - u_i D_{i,\tau} \leq 0.$$

These are the global constraints, saying that variables assigned to bucket  $I_i$  must have average bias inside that interval.

The atom-level contributions to these constraints are defined by

$$g_{i,\tau}^{\text{lo}}(c) := \sum_{\substack{1 \leq j \leq \ell \\ i_j = i, \tau_j = \tau}} w_{\ell} (\ell_i - \chi(s_j)), \quad g_{i,\tau}^{\text{up}}(c) := \sum_{\substack{1 \leq j \leq \ell \\ i_j = i, \tau_j = \tau}} w_{\ell} (\chi(s_j) - u_i).$$

Thus, the global constraints also decompose exactly as

$$g_{i,\tau}^{\text{lo}}(\Phi_{\leq 3}) = \sum_c N_c g_{i,\tau}^{\text{lo}}(c), \quad g_{i,\tau}^{\text{up}}(\Phi_{\leq 3}) = \sum_c N_c g_{i,\tau}^{\text{up}}(c).$$

Suppose there exist nonnegative multipliers  $\lambda_{i,\tau}^{\text{lo}}, \lambda_{i,\tau}^{\text{up}} \geq 0$  such that every typed atom  $c$  satisfies

$$h_{\Theta}(c) - \rho o(c) + \sum_{i,\tau} \left( \lambda_{i,\tau}^{\text{lo}} g_{i,\tau}^{\text{lo}}(c) + \lambda_{i,\tau}^{\text{up}} g_{i,\tau}^{\text{up}}(c) \right) \geq 0.$$

Multiplying this inequality by  $N_c$  and summing over all typed atoms gives

$$L_{\leq 3}^{\Theta}(\text{WSnap}(\Phi_{\leq 3})) - \rho \text{OPT}(\Phi_{\leq 3}) + \sum_{i,\tau} \left( \lambda_{i,\tau}^{\text{lo}} g_{i,\tau}^{\text{lo}}(\Phi_{\leq 3}) + \lambda_{i,\tau}^{\text{up}} g_{i,\tau}^{\text{up}}(\Phi_{\leq 3}) \right) \geq 0.$$

Since  $\Phi_{\leq 3}$  is a valid instance, all bucket constraints are nonpositive, and since all multipliers are nonnegative, the correction term is nonpositive. Therefore

$$L_{\leq 3}^{\Theta}(\text{WSnap}(\Phi_{\leq 3})) \geq \rho \text{OPT}(\Phi_{\leq 3}).$$

So the lower bound reduces to finding a tuple  $\Theta$ , a ratio  $\rho$ , and nonnegative multipliers  $\lambda$  satisfying the typed-atom inequalities.

*Step 3: Solving and verifying a finite LP:* After fixing the certificate form, all unknowns are finite-dimensional:

$$r, c_0, \quad u_j(\alpha), \quad p_{jt}(\alpha, \beta), \quad \lambda_{i,\tau}^{\text{lo}}, \quad \lambda_{i,\tau}^{\text{up}}, \quad \rho.$$

This makes the search space finite and manageable. We first fix a constant-size uniform bucket partition  $\mathcal{I}$  of the bias domain  $[-1, 1]$  and the rounding vector  $r$  by a separate finite numerical search. The choices of  $I$  and  $r$  are recorded in Appendix B.

After this choice is fixed, all quantities  $q_\alpha$ , the unary/binary satisfaction probabilities, and the ternary upper bounds  $1 - \prod_j (1 - q_{\alpha_j})$  are rational constants. The resulting linear program maximizes  $\rho$  subject to two classes of constraints:

- for every ternary endpoint-label triple,

$$H_{\Theta}(\alpha_1, \alpha_2, \alpha_3) \leq 1 - \prod_{j=1}^3 (1 - q_{\alpha_j});$$

- for every typed atom  $c$ ,

$$h_{\Theta}(c) - \rho o(c) + \sum_{i,\tau} \left( \lambda_{i,\tau}^{\text{lo}} g_{i,\tau}^{\text{lo}}(c) + \lambda_{i,\tau}^{\text{up}} g_{i,\tau}^{\text{up}}(c) \right) \geq 0.$$

Because the bucket set and the atom types are both finite, this is a finite LP. We use codes to solve the LP and verify the  $\Theta^*$  that satisfies both conditions. The corresponding code is available in the repository: <https://github.com/Guangxu-Yang/maxksat-lp-certificate>.

We first solve the LP numerically, obtaining a floating-point candidate  $(c, u, p, \lambda, \rho)$ . It is used only to find a good candidate certificate, not to prove the theorem. We then round all floating-point parameters to nearby rational values and run the exact verifier. This script checks every ternary upper constraint and every typed-atom inequality exactly over  $\mathbb{Q}$ . The exact rational certificate used here has  $\rho = 0.717275$ . The exact rational tuple that passes all checks is denoted by  $\Theta^*$ . Therefore,  $\Theta^*$  simultaneously satisfies the upper bound condition and the lower bound condition.

Applying Step 1 and Step 2 to this verified tuple  $\Theta^*$  gives

$$\rho \text{OPT}(\Phi_{\leq 3}) \leq L_{\leq 3}^{\Theta^*}(\text{WSnap}(\Phi_{\leq 3})) \leq \mathcal{E}_{r^*}(\Phi_{\leq 3}).$$

Here  $\mathcal{E}_{r^*}(\Phi_{\leq 3})$  denotes the expected number of satisfied clauses under the independent randomized assignment induced by  $r^*$ .  $\square$

**Lemma 2** (Reduction to weighted snapshot estimation). *Fix the certificate  $\Theta^*$  from Lemma 1, then every Max- $k$ SAT instance  $\Phi$  with  $k \geq 2$  satisfies*

$$\rho \text{OPT}(\Phi) \leq L_{\leq k}^{\Theta^*}(\Phi) \leq \text{OPT}(\Phi).$$

*Proof.* Recall that

$$L_{\leq k}^{\Theta^*}(\Phi) = L_{\leq 3}^{\Theta^*}(\text{WSnap}(\Phi_{\leq 3})) + \rho m_{\geq 4}.$$

*Upper bound.* We first handle the long clauses. Under the rounding vector  $r^*$ , every literal fails with probability at most 0.72917<sup>2</sup>. Therefore, every clause of length  $d \geq 4$  is satisfied with probability at least

$$1 - \prod_{j=1}^d (1 - q_{\alpha_j}) \geq 1 - (0.72917)^d \geq 1 - (0.72917)^4 > 0.717275 = \rho.$$

Thus, the term  $\rho \cdot m_{\geq 4}$  is a valid lower floor for the long clauses, and is also at most their expected contribution under the same independent rounding.

Apply the independent randomized assignment determined by  $\Theta^*$ . Lemma 1 gives

$$L_{\leq 3}^{\Theta^*}(\text{WSnap}(\Phi_{\leq 3})) \leq \mathcal{E}_{r^*}(\Phi_{\leq 3}).$$

The long-clause estimate above shows that  $\rho \cdot m_{\geq 4}$  is at most the expected number of satisfied long clauses. Therefore,

$$L_{\leq k}^{\Theta^*}(\Phi) \leq \mathcal{E}_{r^*}(\Phi) \leq \text{OPT}(\Phi).$$

*Lower bound.* Let  $x^*$  be an optimal assignment for  $\Phi$ . Let  $S^*$  be the number of short clauses satisfied by  $x^*$ , and let  $G^*$  be the number of long clauses satisfied by  $x^*$ . Then

$$\text{OPT}(\Phi) = S^* + G^*.$$

Since the restriction of  $x^*$  to  $\Phi_{\leq 3}$  is feasible for the short subinstance,  $\text{OPT}(\Phi_{\leq 3}) \geq S^*$ . Hence, Lemma 1 gives

$$L_{\leq 3}^{\Theta^*}(\text{WSnap}(\Phi_{\leq 3})) \geq \rho \text{OPT}(\Phi_{\leq 3}) \geq \rho S^*.$$

Also  $G^* \leq m_{\geq 4}$ , so  $\rho \cdot m_{\geq 4} \geq \rho G^*$ .

Adding the last two inequalities yields

$$L_{\leq k}^{\Theta^*}(\Phi) = L_{\leq 3}^{\Theta^*}(\text{WSnap}(\Phi_{\leq 3})) + \rho \cdot m_{\geq 4} \geq \rho \cdot (S^* + G^*) = \rho \cdot \text{OPT}(\Phi).$$

□

## 4 Closeness of the Weighted Snapshot and Pseudosnapshot

In section 3, the Max- $k$ -SAT value was reduced to the weighted snapshot score  $L_{\leq 3}$  together with the deterministic long-clause term  $\rho \cdot m_{\geq 4}$ . The obstacle is that the true short snapshot  $\text{WSnap}$  depends on final variable biases, which are not available during the stream.

To obtain a one-pass streaming algorithm, we replace the weighted snapshot by a weighted pseudosnapshot that can be estimated in the streaming setting. Throughout this subsection, we fix the optimized tuple  $\Theta^*$  and the ratio  $\rho$  from Lemma 1 and Lemma 2. We suppress the superscript  $\Theta^*$  from  $L_{\leq 3}$ .

<sup>2</sup>Under the certified rounding vector  $r^*$ , recorded in Appendix B, every literal is satisfied with probability at least 0.27083

**Definition 7** (Tagged unit). For a short clause  $C$  with  $\ell = |C| \leq 3$ , its  $j$ -th endpoint  $(C, j)$  creates the tagged units

$$(C, j, h), \quad h \in [w_\ell],$$

where  $w_1 = 4$ ,  $w_2 = 2$ , and  $w_3 = 1$ . The tagged unit  $(C, j, h)$  is associated to the variable  $v_j(C)$ , and is positive iff the  $j$ -th literal of  $C$  is positive. A tagged unit is simply a copy of the  $j$ -th endpoint with index  $h \in [w_\ell]$ . Unless necessary, we don't distinguish between the endpoint and its tagged units.

We define the degree of a variable  $v_j(C)$  as the total number of occurrences of its associated tagged units. We then partition the degree range into the following scales. Fix  $\varepsilon \in (0, 1/100)$ , and define geometric scales for

$$D_0 := 0, \quad D_a := \lfloor (1 + \varepsilon^3)^a \rfloor \quad (a = 1, \dots, A),$$

where  $A$  is the smallest interger that satisfies  $D_A \geq 4m$ . In particular  $A = O_\varepsilon(\log m)$ . When a positive tagged unit arrives in the streaming, our algorithm employs a hash function to probabilistically decide whether it should be counted. This selective counting strategy is designed to save memory space.

**Definition 8** (Hash function). For each scale  $a$  with  $2D_a \geq \kappa$ , let  $f_a$  be a fully independent hash function on positive tagged units with selection probability  $\kappa/(2D_a)$ , where  $\kappa = \text{poly}(1/\varepsilon)$  is a sufficiently large constant. Namely, for any positive tagged unit  $(C, j, r)$ ,

$$f_a(C, j, r) = \begin{cases} 1 & \text{with probability } \frac{\kappa}{2D_a}. \\ 0 & \text{with probability } 1 - \frac{\kappa}{2D_a}. \end{cases}$$

When  $f_a = 1$ , we say  $(C, j, r)$  is selected by  $f_a$ . Otherwise we say it's not selected.

When  $2D_a < \kappa$ , the cost of counting the exact degree is cheap, so no hash function is needed for these scales. Let  $g : V \rightarrow [-\varepsilon, \varepsilon]$  be independent uniform smoothing noise. We define the weighted pseudobias as follows.

**Definition 9** (Weighted pseudobias). For a short clause  $C$  incident to a variable  $v$ , let  $d_v^{\leq C}, p_v^{\leq C}$  and  $d_v^{> C}, p_v^{> C}$  denote the tagged total and positive counts up to and including  $C$ , and strictly after  $C$ , respectively. Empty scales with  $D_a = D_{a+1}$  are ignored. Let  $a$  be the smallest nonempty scale index satisfying

$$D_a < d_v^{\leq C} \leq D_{a+1}.$$

If  $2D_a < \kappa$ , set

$$\tilde{d}_v^{\leq C} := d_v^{\leq C}, \quad \tilde{p}_v^{\leq C} := p_v^{\leq C}.$$

If  $2D_a \geq \kappa$ , set  $\tilde{d}_v^{\leq C} := D_a$ , let  $s_v^{\leq C}$  be the number of positive tagged units in the prefix selected by  $f_a$ , and set

$$\tilde{p}_v^{\leq C} := \frac{2D_a}{\kappa} \min\{s_v^{\leq C}, \kappa\}.$$

The weighted pseudobias at the arrival of  $C$  is

$$\tilde{b}_v^C := \text{clip}_{[-1,1]} \left( \frac{2(\tilde{p}_v^{\leq C} + p_v^{> C})}{\tilde{d}_v^{\leq C} + d_v^{> C}} - 1 + g(v) \right).$$

We now use these arrival-time pseudobias to assign the same sign-bucket labels that were used in the weighted snapshot, but separately for each endpoint at the moment its clause arrives.

**Definition 10** (Weighted pseudosnapshot). For a clause  $C$  of length  $\ell$ , write  $v_j(C)$  and  $s_j(C)$  for the variable and literal sign of its  $j$ -th endpoint. Let  $\tilde{i}_j(C)$  be the unique bucket index satisfying  $\tilde{b}_{v_j(C)}^C \in I_{\tilde{i}_j(C)}$ , using the same half-open bucket convention as Definition 3, and set

$$\lambda_j(C) := (s_j(C), \tilde{i}_j(C)) \in \{+, -\} \times \{0, \dots, L-1\}.$$

The weighted pseudosnapshot  $\text{WPsSnap}(\Phi_{\leq 3})$  is a vector with the following coordinates:

$$U_{\ell,j,\alpha} := \#\{C \in \Phi_{\leq 3} : |C| = \ell, \lambda_j(C) = \alpha\}, \quad 1 \leq j \leq \ell \leq 3,$$

$$P_{\ell,j,t,\alpha,\beta} := \#\{C \in \Phi_{\leq 3} : |C| = \ell, (\lambda_j(C), \lambda_t(C)) = (\alpha, \beta)\}, \quad 1 \leq j < t \leq \ell \leq 3,$$

and

$$M_3 := \#\{C \in \Phi_{\leq 3} : |C| = 3\}.$$

Thus  $\text{WPsSnap}(\Phi_{\leq 3})$  has the same coordinate set as  $\text{WSnap}(\Phi_{\leq 3})$ , but endpoint labels are assigned using the arrival-time pseudobias  $\tilde{b}_v^C$  rather than the final bias  $b_v$ .

The pseudosnapshot still uses labels that may vary with the arrival time of a clause. For the analysis, it is useful to compare it with a simpler object in which every variable has one fixed label throughout the whole stream. We obtain this fixed label by applying the same smoothing noise to the final bias  $b_v$ .

**Definition 11** (Smoothed labels and fixed-label snapshot). For each variable  $v$ , define  $\bar{b}_v := \text{clip}_{[-1,1]}(b_v + g(v))$ . Let  $\beta_v$  be the unique bucket index such that  $\bar{b}_v \in I_{\beta_v}$ , using the half-open bucket convention from Definition 3. The fixed-label short snapshot  $\text{WSnap}_\beta(\Phi_{\leq 3})$  is defined using the same coordinates as  $\text{WSnap}(\Phi_{\leq 3})$ , except that every endpoint of variable  $v$  is assigned the bucket label  $\beta_v$ .

Next, we show that with high probability over the public randomness, the pseudosnapshot is close to a fixed-label snapshot. The fixed labels need not be the true final buckets, but they are close enough: if a variable is labeled by bucket  $I_i$ , then its true bias lies within an  $\varepsilon$ -neighborhood of  $I_i$ .

**Lemma 3** (Short-clause pseudosnapshot stability). *With probability at least  $1 - O(\varepsilon) - e^{-\Omega(\varepsilon^6 \kappa)}$ , the label map  $\beta$  from the smoothed labels construction satisfies*

$$\|\text{WPsSnap}(\Phi_{\leq 3}) - \text{WSnap}_\beta(\Phi_{\leq 3})\|_1 \leq C_{\text{st}} \varepsilon m$$

for an absolute constant  $C_{\text{st}}$ . Moreover, if  $I_i = [\ell_i, u_i]$ , then  $\beta$  is  $\varepsilon$ -admissible in the sense that whenever  $\beta_v = i$ ,  $b_v \in [\ell_i - \varepsilon, u_i + \varepsilon]$ .

*Proof.* Fix a variable  $v$ . An endpoint of a short clause  $C$  incident to  $v$  uses scale  $a = a(C, v)$  if  $D_a < d_v^{\leq C} \leq D_{a+1}$ . Call this endpoint *overflowing* if  $2D_a \geq \kappa$  and  $s_v^{\leq C} \geq \kappa$ .

We first show that, except for a small initial segment and possible overflow endpoints, the pseudobias is close to the final smoothed bias. If  $d_v < 1/\varepsilon$ , then all prefix counts are exact once  $\kappa > 2/\varepsilon$ . If  $d_v \geq 1/\varepsilon$ , discard the first  $\lfloor \varepsilon d_v \rfloor$  tagged units of  $v$ ; their total mass is at most  $\varepsilon d_v$ .

After this discard, the prefix degree of  $v$  visits only  $O(\varepsilon^{-3} \log \frac{1}{\varepsilon})$  nonempty geometric intervals. Fix one sampled interval  $(D_a, D_{a+1}]$ , and set  $\theta_a = \kappa/(2D_a)$ . At the first tagged unit in the interval, let  $p$  and  $s$  be the true and sampled positive prefix counts. Then  $\mathbb{E}[s] \leq \theta_a D_{a+1} \leq (1 + \varepsilon^3)\kappa/2$ , so Bernstein's inequality gives

$$|s - \theta_a p| \leq c_1 \varepsilon^3 \kappa$$

except with probability  $e^{-\Omega(\varepsilon^6\kappa)}$ . During the rest of the interval, the true degree changes by only

$$D_{a+1} - D_a = O(\varepsilon^3 D_a),$$

So a Chernoff bound shows that the number of additionally sampled positive tagged units is  $O(\varepsilon^3\kappa)$ , except with probability  $e^{-\Omega(\varepsilon^3\kappa)}$ . Hence, throughout this interval,

$$|s_v^{\leq C} - \theta_a p_v^{\leq C}| \leq c_2 \varepsilon^3 \kappa$$

for every non-overflow endpoint  $C$ .

Scaling by  $2D_a/\kappa$  and using  $D_a < d_v^{\leq C} \leq D_{a+1}$  gives

$$|\tilde{p}_v^{\leq C} - p_v^{\leq C}| \leq c_3 \varepsilon^3 d_v^{\leq C}. \quad (1)$$

Also, in sampled intervals,

$$|\tilde{d}_v^{\leq C} - d_v^{\leq C}| = d_v^{\leq C} - D_a \leq D_{a+1} - D_a \leq c_4 \varepsilon^3 d_v^{\leq C}. \quad (2)$$

In low-degree intervals, both prefix quantities are exact, so the same bounds hold with left-hand side 0. A union bound over the intervals visited by  $v$ , with the polynomial factor in  $1/\varepsilon$  absorbed into the exponent by choosing the constant in  $\kappa = \text{poly}(1/\varepsilon)$  large enough, shows that (1) and (2) hold for every non-overflow endpoint of  $v$  after the discarded prefix, except with probability  $e^{-\Omega(\varepsilon^6\kappa)}$ .

On this concentration event, write

$$p_v = p_v^{\leq C} + p_v^{> C}, \quad d_v = d_v^{\leq C} + d_v^{> C}.$$

Since  $0 \leq p_v \leq d_v$ , the two prefix estimates (1) and (2) imply

$$\left| \frac{2(\tilde{p}_v^{\leq C} + p_v^{> C})}{\tilde{d}_v^{\leq C} + d_v^{> C}} - \frac{2p_v}{d_v} \right| \leq O\left(\frac{\varepsilon^3 d_v^{\leq C}}{d_v}\right) = O(\varepsilon^3).$$

The clipping map is 1-Lipschitz. Therefore

$$\left| \tilde{b}_v^C - \bar{b}_v \right| = O(\varepsilon^3), \quad \bar{b}_v := \text{clip}_{[-1,1]}(b_v + g(v)),$$

for every such endpoint.

This bias estimate implies label agreement unless the smoothed bias lies very close to a bucket boundary. If  $b_v + g(v)$  is farther than  $c\varepsilon^3$  from every bucket boundary, then every endpoint satisfying the concentration event receives the same bucket in  $\text{WPsSnap}(\Phi_{\leq 3})$  as in  $\text{WSnap}_\beta(\Phi_{\leq 3})$ . Since  $g(v)$  is uniform on  $[-\varepsilon, \varepsilon]$  and there are only constantly many bucket boundaries,

$$\Pr[\text{dist}(b_v + g(v), \partial\mathcal{I}) \leq c\varepsilon^3] = O(\varepsilon^2).$$

It remains to show that the exceptional endpoints have small total mass. First consider overflow. If  $2D_a < \kappa$ , overflow is impossible. If  $2D_a \geq \kappa$ , then

$$s_v^{\leq C} \sim \text{Bin}\left(p_v^{\leq C}, \frac{\kappa}{2D_a}\right), \quad \mathbb{E}[s_v^{\leq C}] \leq \frac{\kappa D_{a+1}}{2D_a} \leq 0.6\kappa$$

for  $\varepsilon < 1/100$ . Thus

$$\Pr[s_v^{\leq C} \geq \kappa] \leq e^{-\Omega(\kappa)}.$$

Let  $W_{\text{of}}$  be the total tagged mass of overflowing endpoints. Since  $\sum_v d_v = 4m_1 + 4m_2 + 3m_3 \leq 4m$ ,

$$\mathbb{E}[W_{\text{of}}] \leq e^{-\Omega(\kappa)} m.$$

Increasing the constant in  $\kappa = \text{poly}(1/\varepsilon)$  if necessary, we may assume  $\mathbb{E}[W_{\text{of}}] \leq \varepsilon^2 m$ . Markov's inequality gives

$$\Pr[W_{\text{of}} > \varepsilon m] \leq \varepsilon. \quad (3)$$

Call a tagged unit bad if it is in a discarded initial segment, its variable fails the concentration event, its endpoint overflows, or  $b_v + g(v)$  is within  $c\varepsilon^3$  of a bucket boundary. These four classes have small total mass. The discarded initial segments have mass at most

$$\varepsilon \sum_v d_v \leq 4\varepsilon m.$$

The concentration-failure class has an expected mass at most  $e^{-\Omega(\varepsilon^6 \kappa)} \sum_v d_v$  and hence has mass  $O(\varepsilon m)$  with failure probability  $e^{-\Omega(\varepsilon^6 \kappa)}$ , after the same adjustment of the constant in  $\kappa$ . The boundary class has expected mass  $O(\varepsilon^2) \sum_v d_v$ , and is  $O(\varepsilon m)$  with probability  $1 - O(\varepsilon)$ . The overflow class is controlled by (3). Thus, the total bad tagged mass is  $O(\varepsilon m)$  with probability  $1 - O(\varepsilon) - e^{-\Omega(\varepsilon^6 \kappa)}$ .

Now delete, only for the analysis, every short clause containing a bad tagged unit. Each short clause contains at most four tagged units, so the number of deleted clauses is  $O(\varepsilon m)$ . On every remaining endpoint, the label used by  $\text{WPsSnap}(\Phi_{\leq 3})$  is exactly the bucket of  $\bar{b}_v$ , hence it is the same label used by  $\text{WSnap}_\beta(\Phi_{\leq 3})$ . Since each short clause contributes to only constantly many snapshot coordinates, the deleted clauses contribute only  $O(\varepsilon m)$  to the  $\ell_1$  distance:

$$\|\text{WPsSnap}(\Phi_{\leq 3}) - \text{WSnap}_\beta(\Phi_{\leq 3})\|_1 \leq C_{\text{st}} \varepsilon m$$

for a constant  $C_{\text{st}}$ .

Finally, if  $\beta_v = i$ , then  $\bar{b}_v \in I_i = [\ell_i, u_i]$ . Since  $|g(v)| \leq \varepsilon$ , we obtain  $b_v \in [\ell_i - \varepsilon, u_i + \varepsilon]$ . Hence  $\beta$  is  $\varepsilon$ -admissible.  $\square$

Lemma 3 reduces the analysis of the pseudosnapshot to the analysis of a fixed-label snapshot whose labels are only approximately feasible. We therefore need a robust version of the finite certificate from Lemma 1: the certificate should still give a good lower bound when the bucket constraints hold with a small additive slack.

**Lemma 4** (Approximate-feasibility short-clause certificate). *Let  $\Psi = \Phi_{\leq 3}$ . Let  $\beta$  be a  $\xi$ -admissible label map for its variables and  $\mathcal{D}_\beta$  be the independent assignment that sets  $x_v = 1$  with probability  $r_{\beta_v}^*$ . Then there exists a constant  $C_{\text{cert}}$ , depending only on the verified finite certificate, such that for every assignment  $x$ ,*

$$\rho \text{Val}_\Psi(x) - C_{\text{cert}} \xi |\Psi| \leq L_{\leq 3}(\text{WSnap}_\beta(\Psi)) \leq \mathbb{E}_{X \sim \mathcal{D}_\beta}[\text{Val}_\Psi(X)].$$

*Proof.* For the upper bound, unary and binary clauses are counted with their exact satisfaction probabilities under  $\mathcal{D}_\beta$ . For ternary clauses, the verified constraint

$$H_{\Theta^*}(\alpha_1, \alpha_2, \alpha_3) \leq 1 - \prod_{j=1}^3 (1 - q_{\alpha_j})$$

shows that the surrogate contribution is at most the true satisfaction probability. Summing over clauses gives

$$L_{\leq 3}(\text{WSnap}_\beta(\Psi)) \leq \mathbb{E}_{X \sim \mathcal{D}_\beta}[\text{Val}_\Psi(X)] \leq \text{OPT}(\Psi).$$

For the lower bound, fix an assignment  $x$ . Refine every clause into typed atoms using the labels from  $\beta$  and the endpoint bits of  $x$ . More explicitly, a typed atom records the clause length  $\ell \in \{1, 2, 3\}$ , the endpoint signs  $s_1, \dots, s_\ell$ , the endpoint labels  $\beta_{v_1}, \dots, \beta_{v_\ell}$ , and the bits  $x_{v_1}, \dots, x_{v_\ell}$ . Let  $N_c$  be the number of clauses of  $\Psi$  of typed atom  $c$ . As in the exact certificate, write  $h(c)$  for the local contribution of  $c$  to  $L_{\leq 3}$ , and  $o(c)$  for whether the clause represented by  $c$  is satisfied by  $x$ . Then

$$L_{\leq 3}(\text{WSnap}_\beta(\Psi)) = \sum_c N_c h(c), \quad \text{Val}_\Psi(x) = \sum_c N_c o(c).$$

The bucket constraints decompose over the same typed atoms. Let  $w_1 = 4, w_2 = 2, w_3 = 1$ , and write  $\chi(+)=1, \chi(-)=-1$ . For  $I_i = [\ell_i, u_i]$ , define  $D_{i,\tau}$  and  $A_{i,\tau}$  using the label map  $\beta$  and the reference assignment  $x$ : an endpoint of an  $\ell$ -clause with sign  $s_j$ , label  $i_j$ , and bit  $\tau_j$  contributes  $w_\ell$  to  $D_{i_j, \tau_j}$  and  $w_\ell \chi(s_j)$  to  $A_{i_j, \tau_j}$ . The atom contributions

$$g_{i,\tau}^{\text{lo}}(c) = \sum_{\substack{1 \leq j \leq \ell \\ i_j = i, \tau_j = \tau}} w_\ell (\ell_i - \chi(s_j)), \quad g_{i,\tau}^{\text{up}}(c) = \sum_{\substack{1 \leq j \leq \ell \\ i_j = i, \tau_j = \tau}} w_\ell (\chi(s_j) - u_i)$$

satisfy

$$g_{i,\tau}^{\text{lo}}(\Psi) = \sum_c N_c g_{i,\tau}^{\text{lo}}(c), \quad g_{i,\tau}^{\text{up}}(\Psi) = \sum_c N_c g_{i,\tau}^{\text{up}}(c).$$

Multiplying the verified atom inequality by  $N_c$  and summing over all typed atoms gives

$$L_{\leq 3}(\text{WSnap}_\beta(\Psi)) - \rho \text{Val}_\Psi(x) + \sum_{i,\tau} \left( \lambda_{i,\tau}^{\text{lo}} g_{i,\tau}^{\text{lo}}(\Psi) + \lambda_{i,\tau}^{\text{up}} g_{i,\tau}^{\text{up}}(\Psi) \right) \geq 0.$$

Since  $\beta$  is  $\xi$ -admissible, every variable assigned to bucket  $I_i = [\ell_i, u_i]$  satisfies

$$\ell_i - \xi \leq b_v \leq u_i + \xi.$$

After summing over variables in bucket  $i$  and bit class  $\tau$ , this gives

$$g_{i,\tau}^{\text{lo}}(\Psi) \leq \xi D_{i,\tau}, \quad g_{i,\tau}^{\text{up}}(\Psi) \leq \xi D_{i,\tau}.$$

Because all multipliers are fixed nonnegative constants and  $\sum_{i,\tau} D_{i,\tau} \leq 4|\Psi|$ , the correction term is at most  $C_{\text{cert}} \xi |\Psi|$ . Therefore

$$L_{\leq 3}(\text{WSnap}_\beta(\Psi)) \geq \rho \text{Val}_\Psi(x) - C_{\text{cert}} \xi |\Psi|.$$

□

We now combine the two ingredients. Lemma 3 shows that  $\text{WPsSnap}(\Phi_{\leq 3})$  is close to  $\text{WSnap}_\beta(\Phi_{\leq 3})$  for an  $\varepsilon$ -admissible label map  $\beta$ . Lemma 4 then applies to this fixed-label snapshot and loses only  $O(\varepsilon m)$ . Since  $L_{\leq 3}$  is a fixed linear functional, the  $L_1$ -closeness of the two snapshots changes the score by only  $O(\varepsilon m)$ . Adding the deterministic long-clause term  $\rho m_{\geq 4}$  gives the approximate value guarantee below.

**Lemma 5** (From weighted pseudosnapshot to value). *There exist constants  $C_0, C_1 > 0$ , depending only on the certificate, such that with probability at least  $1 - O(\varepsilon) - e^{-\Omega(\varepsilon^6 \kappa)}$ ,*

$$\rho \text{OPT}(\Phi) - C_1 \varepsilon m \leq L_{\leq 3}(\text{WPsSnap}(\Phi_{\leq 3})) + \rho m_{\geq 4} - C_0 \varepsilon m \leq \text{OPT}(\Phi).$$

*Proof.* Work on the good event of Lemma 3. Since  $L_{\leq 3}$  is a fixed linear functional with constant coefficients,

$$|L_{\leq 3}(\text{WPsSnap}(\Phi_{\leq 3})) - L_{\leq 3}(\text{WSnap}_\beta(\Phi_{\leq 3}))| \leq C_{\text{lin}} \varepsilon m$$

for some absolute constant  $C_{\text{lin}}$ .

For the upper bound, apply Lemma 4 with  $\xi = \varepsilon$  and  $\Psi = \Phi_{\leq 3}$ :

$$L_{\leq 3}(\text{WSnap}_\beta(\Phi_{\leq 3})) \leq \mathbb{E}_{X \sim \mathcal{D}_\beta}[\text{Val}_{\Phi_{\leq 3}}(X)].$$

Moreover, the same label map  $\beta$  defines an independent randomized assignment on all variables of  $\Phi$ . Under this assignment, every long clause has a satisfaction probability at least  $\rho$ . Hence

$$L_{\leq 3}(\text{WSnap}_\beta(\Phi_{\leq 3})) + \rho m_{\geq 4} \leq \mathbb{E}_{X \sim \mathcal{D}_\beta}[\text{Val}_\Phi(X)] \leq \text{OPT}(\Phi).$$

Together with the previous display, this gives

$$L_{\leq 3}(\text{WPsSnap}(\Phi_{\leq 3})) + \rho m_{\geq 4} \leq \text{OPT}(\Phi) + C_{\text{lin}} \varepsilon m.$$

Subtracting a sufficiently large multiple of  $\varepsilon m$  yields the upper bound.

For the lower bound, let  $x^*$  be an optimal assignment for  $\Phi$ . Let  $S^*$  and  $G^*$  be the numbers of satisfied short and long clauses, respectively, so that  $\text{OPT}(\Phi) = S^* + G^*$ . Applying Lemma 4 with  $\Psi = \Phi_{\leq 3}$ ,  $\xi = \varepsilon$ , and  $x = x^*|_{\Phi_{\leq 3}}$ , we obtain

$$L_{\leq 3}(\text{WSnap}_\beta(\Phi_{\leq 3})) \geq \rho S^* - C_{\text{cert}} \varepsilon m.$$

Also  $G^* \leq m_{\geq 4}$ , so  $\rho m_{\geq 4} \geq \rho G^*$ . Therefore

$$L_{\leq 3}(\text{WSnap}_\beta(\Phi_{\leq 3})) + \rho m_{\geq 4} \geq \rho \text{OPT}(\Phi) - C_{\text{cert}} \varepsilon m.$$

Using the linear closeness bound again,

$$L_{\leq 3}(\text{WPsSnap}(\Phi_{\leq 3})) + \rho m_{\geq 4} \geq \rho \text{OPT}(\Phi) - C'_1 \varepsilon m$$

for some constant  $C'_1 > 0$ . After subtracting  $C_0 \varepsilon m$ , we obtain

$$L_{\leq 3}(\text{WPsSnap}(\Phi_{\leq 3})) + \rho m_{\geq 4} - C_0 \varepsilon m \geq \rho \text{OPT}(\Phi) - C_1 \varepsilon m.$$

Finally, every nonconstant clause is satisfied by a uniformly random assignment with probability at least  $1/2$ , so  $\text{OPT}(\Phi) \geq m/2$ . Hence, the additive loss  $C_1 \varepsilon m$  is an  $O(\varepsilon)$   $\text{OPT}(\Phi)$  multiplicative loss.  $\square$

## 5 Quantum algorithm for weighted pseudosnapshot estimation

In this section, we give a quantum streaming algorithm for the pseudosnapshot.

**Lemma 6** (Quantum estimation). *For any fixed constant  $\varepsilon \in (0, 1/100]$ , there is a one-pass quantum streaming algorithm for estimating  $L_{\leq 3}(\text{WPsSnap}(\Phi_{\leq 3}))$  using  $O_\varepsilon(\log^5(n+m))$  qubits and  $O_\varepsilon(\log^5(n+m))$  classical bits. The output  $\widehat{L}_{\leq 3}$  satisfies*

$$\left| \widehat{L}_{\leq 3} - L_{\leq 3}(\text{WPsSnap}(\Phi_{\leq 3})) \right| \leq \varepsilon m$$

with probability at least  $15/16$ .

Throughout this section, we use the scale sequence, public hash functions  $f_a$ , smoothing noise  $g$ , tagged-unit counts, and buckets from the pseudosnapshot construction in Definition 10. We take  $A$  to be the smallest index with  $D_A \geq 4m$ . Hence every  $D_a = O_\varepsilon(m)$  and  $A = O_\varepsilon(\log m)$ . Empty scales with  $D_a = D_{a+1}$  are ignored. We also choose  $\kappa$  to be a sufficiently large positive integer.

For a clause  $C$  and  $j \in \{1, \dots, |C|\}$ , let  $v_j(C)$  be the variable of the  $j$ -th literal. The endpoint  $v_j(C)$  has a sign  $s \in \{+, -\}$ , and the pseudobias  $\widetilde{b}_{v_j(C)}$ .  $\widetilde{b}_{v_j(C)}$  can be computed at the arrival of  $C$ . It lies in a bucket among  $[-1, 1]$ , which is indexed by  $i \in \{0, \dots, L-1\}$ . Let

$$\lambda_j(C) := (s, i) \in \{+, -\} \times \{0, \dots, L-1\}$$

denote the sign-bucket pair. All clauses considered in this section are short clauses, with  $\ell := |C| \leq 3$ . Each endpoint of an  $\ell$ -clause generates  $w_\ell$  tagged units on the same variable, where  $w_1 = 4$ ,  $w_2 = 2$ , and  $w_3 = 1$ . A tagged unit is positive if the corresponding literal is positive.

For later reference, define the true scale of the endpoint  $(C, r)$  by

$$a(C, r) := \min \left\{ s : s \text{ is nonempty and } D_s < d_{v_r(C)}^{\leq C} \leq D_{s+1} \right\}.$$

Thus  $a(C, r)$  is the scale that the  $r$ -th endpoint of  $C$  actually belongs to, according to its prefix total degree at the arrival of  $C$ .

We will use a scale refinement of the coordinates of  $\text{WPsSnap}(\Phi_{\leq 3})$  as follows. Each original one-endpoint or two-endpoint coordinate is split according to the scale of the relevant prefix total degree. For each original coordinate, the sum of its scale-refined coordinates over all relevant scales is exactly the corresponding coordinate of  $\text{WPsSnap}(\Phi_{\leq 3})$ . Moreover, we partition clauses by scales  $(D_a, D_{a+1}]$ . For the indices  $a, b \in \{0, \dots, A-1\}$  where  $A = O_\varepsilon(\log m)$ , and the sign-bucket pairs  $\alpha, \beta \in \{+, -\} \times \{0, \dots, L-1\}$ , define

$$\begin{aligned} U_{1,1,\alpha}^{(a)} &:= \# \left\{ C \in \Phi_{\leq 3} : |C|=1, D_a < d_{v_1(C)}^{\leq C} \leq D_{a+1}, \lambda_1(C)=\alpha \right\}, \\ P_{2,1,2,\alpha,\beta}^{(a,b)} &:= \# \left\{ C \in \Phi_{\leq 3} : |C|=2, D_a < d_{v_1(C)}^{\leq C} \leq D_{a+1}, D_b < d_{v_2(C)}^{\leq C} \leq D_{b+1}, (\lambda_1(C), \lambda_2(C))=(\alpha, \beta) \right\}, \\ U_{3,j,\alpha}^{(a)} &:= \# \left\{ C \in \Phi_{\leq 3} : |C|=3, D_a < d_{v_j(C)}^{\leq C} \leq D_{a+1}, \lambda_j(C)=\alpha \right\}, \\ P_{3,j,t,\alpha,\beta}^{(a,b)} &:= \# \left\{ C \in \Phi_{\leq 3} : |C|=3, D_a < d_{v_j(C)}^{\leq C} \leq D_{a+1}, D_b < d_{v_t(C)}^{\leq C} \leq D_{b+1}, (\lambda_j(C), \lambda_t(C))=(\alpha, \beta) \right\}. \end{aligned}$$

With this scale refinement, the weighted pseudosnapshot score admits the decomposition

$$\begin{aligned} L_{\leq 3}(\text{WPsSnap}(\Phi_{\leq 3})) &= \sum_a \sum_\alpha q_\alpha U_{1,1,\alpha}^{(a)} + \sum_{a,b} \sum_{\alpha,\beta} (1 - (1 - q_\alpha)(1 - q_\beta)) \cdot P_{2,1,2,\alpha,\beta}^{(a,b)} \\ &\quad + c_0 M_3 + \sum_{j=1}^3 \sum_a \sum_\alpha u_j(\alpha) U_{3,j,\alpha}^{(a)} + \sum_{1 \leq j < t \leq 3} \sum_{a,b} \sum_{\alpha,\beta} p_{j,t}(\alpha, \beta) \cdot P_{3,j,t,\alpha,\beta}^{(a,b)}. \end{aligned} \tag{4}$$

The coefficients  $q_\alpha, q_\beta, c_0, u_j(\alpha)$ , and  $p_{jt}(\alpha, \beta)$  are fixed constants from Lemma 1. The term  $M_3$  is the number of length-3 clauses, and it can be counted exactly with  $O(\log m)$  classical bits.

It remains to estimate the four kinds of scale-refined coordinates above. We call  $U_{1,1,\alpha}^{(a)}$  and  $U_{3,j,\alpha}^{(a)}$  *one-endpoint* coordinates, because they test one endpoint of a clause. We call  $P_{2,1,2,\alpha,\beta}^{(a,b)}$  and  $P_{3,j,t,\alpha,\beta}^{(a,b)}$  *two-endpoint* coordinates. For each coordinate, we run a quantum sketch algorithm, described in Sections 5.2 and 5.3. The sketches are combined to form the target quantum algorithm, and we prove Lemma 6 in Section 5.4. We first record the quantum-register notation used by all sketches.

## 5.1 Quantum register

We use a *label*  $S$  to denote the corresponding basis state  $|S\rangle$ . For convenience, we do not distinguish between them unless necessary. Let  $U$  be the set of all labels. If a basis state has a nonzero amplitude, we call it *live*. Let  $T \subseteq U$  be the set of live non-anchor labels. We write

$$|Q(T)\rangle := \frac{|\perp\rangle + \sum_{x \in T} |x\rangle}{\sqrt{1 + |T|}}.$$

Here  $|\perp\rangle$  is a fixed anchor basis state. It is never measured; it is included only so that after every failed measurement, the post-measurement state is still valid, including the case  $T = \emptyset$ . In all sketches below, we will use the quantum register with size

$$M := \lceil C_\varepsilon m \rceil,$$

where  $C_\varepsilon$  is a large enough constant.

We mainly use the following three quantum operations in our algorithm.

- *Permutation*. For a permutation  $\pi$  of  $U$ , apply the unitary that fixes  $|\perp\rangle$  and sends  $|x\rangle$  to  $|\pi(x)\rangle$ .
- *Single-query*. For  $x \in U$ , measure  $|Q(T)\rangle$  with  $\{|x\rangle\langle x|, I - |x\rangle\langle x|\}$ .

For  $x \in T$ : if we get  $|x\rangle$ , output 1, and the quantum state collapses. Otherwise, output 0 and the new state becomes  $|Q(T \setminus \{x\})\rangle$ ;

For  $x \notin T$ : it outputs 0, and the state is unchanged.

- *Pair-query*. For distinct  $x, y \in U$ , measure  $|Q(T)\rangle$  with

$$|\psi_{x,y}^+\rangle := \frac{|x\rangle + |y\rangle}{\sqrt{2}}, \quad |\psi_{x,y}^-\rangle := \frac{|x\rangle - |y\rangle}{\sqrt{2}},$$

and their orthogonal complement. The query outputs 1 on outcome  $|\psi_{x,y}^+\rangle$ , outputs  $-1$  on outcome  $|\psi_{x,y}^-\rangle$ , and outputs 0 on the orthogonal-complement outcome. The state update and probabilities are:

1. If both  $x$  and  $y$  belong to  $T$ , then

$$\Pr[+1] = \frac{2}{1 + |T|}, \quad \Pr[-1] = 0.$$

If we get the orthogonal complement, the new state becomes  $|Q(T \setminus \{x, y\})\rangle$ .

2. If exactly one of them, say  $z$ , belongs to  $T$ , then

$$\Pr[+1] = \Pr[-1] = \frac{1}{2(1 + |T|)}.$$

If the outcome is the orthogonal complement, the new state is  $|Q(T \setminus \{z\})\rangle$ .

3. If neither belongs to  $T$ , the state is unchanged.

**Formulas of single/pair-query.** The single-query and pair-query operations satisfy the following expectation formulas. If the state before the query is  $|Q(T)\rangle$ , then:

- if  $x \in U$  and  $\zeta_x \in \{0, 1\}$  is the output of the single-query, then

$$\mathbb{E}[\zeta_x \mid |Q(T)\rangle] = \frac{1}{|T| + 1} \mathbf{1}[x \in T];$$

- if  $x, y \in U$  and  $\chi_{x,y} \in \{-1, 0, +1\}$  is the output of the pair-query, then

$$\mathbb{E}[\chi_{x,y} \mid |Q(T)\rangle] = \frac{2}{|T| + 1} \mathbf{1}[x \in T] \mathbf{1}[y \in T].$$

## 5.2 One-endpoint estimation

Fix a one-endpoint coordinate  $U_{\ell,j,\alpha}^{(a)}$ , where  $\alpha = (s_\alpha, i_\alpha)$  denotes a sign-bucket pair. Recall that when  $D_a \geq \kappa/2$ , we call  $U_{\ell,j,\alpha}^{(a)}$  *high-degree*. Otherwise, so  $D_a < \kappa/2$ , call it *low-degree*. We handle these two cases with slightly different strategies in Section 5.2.1 and Section 5.2.2, respectively.

The rough idea is to sample a clause during the stream with low probability. If a sampled clause belongs to  $U_{\ell,j,\alpha}^{(a)}$ , the output is scaled by  $M$  or  $M/2$  to cancel the low probability; otherwise it outputs 0.

Thus, to test if the clause belongs to  $U_{\ell,j,\alpha}^{(a)}$ , we should check all the relative constraints on the scale  $a$ , clause length  $\ell$ , endpoint position  $j$ , literal sign  $s_\alpha$  and bucket index  $i_\alpha$ . Indeed, the clause length, endpoint position, and literal sign are known when the clause arrives. The scale test is handled by quantum labels that track prefix degree. The bucket test is evaluated only after a successful query, using the queried candidate prefix value and the exact suffix counts.

Moreover, for the scale and bucket tests, the sketch needs the prefix total degree  $d$  and the prefix positive count  $c$ . To implement the updates by simple shifts, we store these two counts through the single threshold value  $\Gamma_a(d, c) = d + B_a c$ . A different pair  $(d', c')$  can sometimes reach the same queried threshold, especially when the endpoint is outside the queried scale. These accidental threshold crossings are the only extra counts considered below. We define the degree condition under which they can occur and record the properties needed later.

**Random-base threshold encoding.** Fix a sufficiently large constant  $K_{\text{base}} := K_{\text{base}}(\varepsilon)$ . For every nonempty scale  $a$ , choose an independent public-random integer  $B_a$  uniformly from

$$\left\{ \lceil K_{\text{base}} D_{a+1} \rceil, \lceil K_{\text{base}} D_{a+1} \rceil + 1, \dots, \lceil 2K_{\text{base}} D_{a+1} \rceil \right\}.$$

In particular, after enlarging  $K_{\text{base}}$  if needed, we have  $B_a > D_{a+1} + 1$  for every scale  $a$ . For a prefix positive count  $c$ , define

$$\Gamma_a(d, c) := d + B_a c.$$

In a high-degree sketch,  $c$  is the sampled positive-prefix count  $s_v^{\leq C}$ , while in a low-degree sketch,  $c$  is the exact positive-prefix count  $p_v^{\leq C}$ .

For every  $q \in \{0, \dots, \kappa - 1\}$  and every in-range degree  $D_a < d \leq D_{a+1}$ ,

$$\mathbf{1}[\Gamma_a(d, c) \geq B_a q + D_a + 1] - \mathbf{1}[\Gamma_a(d, c) \geq B_a q + D_{a+1} + 1] = \mathbf{1}[c = q]. \quad (5)$$

Likewise, for every  $D_a < d, d_0 \leq D_{a+1}$  and every  $c_0 \geq 0$ ,

$$\mathbf{1}[\Gamma_a(d, c) \geq B_a c_0 + d_0] - \mathbf{1}[\Gamma_a(d, c) \geq B_a c_0 + d_0 + 1] = \mathbf{1}[d = d_0, c = c_0]. \quad (6)$$

These are the only identities needed on genuine scale- $a$  endpoints. For the high-degree one-endpoint coordinate output, we will also use the cumulative form: for every  $q \geq 1$ , every integer  $c \geq 0$ , and every  $D_a < d \leq D_{a+1}$ ,

$$\mathbf{1}[\Gamma_a(d, c) \geq qB_a] = \mathbf{1}[c \geq q]. \quad (7)$$

This follows from  $B_a > D_{a+1}$ .

For later use, set the largest multiplier of  $B_a$  that can appear in the false-scale condition

$$H_a := \begin{cases} \kappa, & D_a \geq \kappa/2, \\ D_{a+1}, & D_a < \kappa/2. \end{cases}$$

**Impersonation.** We say that an endpoint with prefix degree  $d$  *impersonates scale  $a$*  if  $d > D_{a+1}$  and

$$d \in \bigcup_{h=1}^{H_a} [hB_a + D_a + 1, hB_a + D_{a+1}]. \quad (8)$$

To make the subsequent formulas more concise, let  $\text{Imp}_{C,r}^a$  be the event that endpoint  $(C, r)$  impersonates scale  $a$ .

This condition is deliberately broader than the event that the endpoint actually gives a nonzero contribution to the wrong queried scale. Every such nonzero contribution from outside the queried scale satisfies this condition, but some endpoints satisfying it may still contribute zero because their positive-prefix count does not put the encoded value  $d + B_a c$  at the queried level.

**Lemma 7** (Impersonation). *Assume  $D_a < \kappa/2$ . Fix a queried low-degree level  $B_a c_0 + d_0$ , where  $D_a < d_0 \leq D_{a+1}$  and  $0 \leq c_0 \leq d_0$ . Let  $d^*, c^*$  be nonnegative integers. If*

$$\Gamma_a(d^*, c^*) = B_a c_0 + d_0$$

*and  $d^* \notin (D_a, D_{a+1}]$ , then  $d^* > D_{a+1}$  and the endpoint with prefix degree  $d^*$  impersonates scale  $a$ .*

Thus, in a low-degree single-level query, every nonzero contribution from outside the queried scale is counted by the impersonation condition.

*Proof.* Because  $B_a > D_{a+1} + 1$ , both  $d_0$  and every positive  $d^* \leq D_a$  are below  $B_a$ . If  $d^* = 0$ , equality modulo  $B_a$  is also impossible because  $d_0 > 0$ . Thus  $d^* \leq D_a$  cannot satisfy  $d^* + B_a c^* = B_a c_0 + d_0$ : equality modulo  $B_a$  would force  $d^* = d_0 > D_a$ .

It remains to consider  $d^* > D_{a+1}$ . From the equality,

$$d^* = (c_0 - c^*)B_a + d_0.$$

The integer  $h := c_0 - c^*$  must be positive; otherwise  $d^* \leq d_0 \leq D_{a+1}$ . Also  $h \leq c_0 \leq d_0 \leq D_{a+1} = H_a$  in the low-degree case. Hence

$$d^* \in [hB_a + D_a + 1, hB_a + D_{a+1}]$$

for some  $1 \leq h \leq H_a$ , which is exactly the event that the endpoint impersonates scale  $a$ .  $\square$

### 5.2.1 High-degree case

For each  $q \in \{0, 1, \dots, \kappa\}$ , we maintain an independent  $q$ -Sketch. The case  $q = \kappa$  accounts for all endpoints with  $s_v^{\leq C} \geq \kappa$  through the capped value  $\min\{s_v^{\leq C}, \kappa\}$ , so high-degree one-endpoint coordinates need no separate overflow term.

Each  $q$ -Sketch consists of two independent sub-sketches, called the *Upper- $q$ -Sketch* and the *Lower- $q$ -Sketch*. Their outputs are combined to give an estimate for the high-degree coordinate.

We describe the explicit form of the quantum register as follows. There are six kinds of labels/basic states in  $U$ . We use the global register size  $M = \lceil C_\varepsilon m \rceil$ , with  $C_\varepsilon$  chosen large enough for the fixed  $\varepsilon > 0$ .

1. The fixed anchor basis state  $|\perp\rangle$ ;
2. Spare labels:  $S_1, \dots, S_M$ ;
3.  $H$ -labels:  $H(v, t)$ , where  $v \in [n]$ ,  $t \in [M]$ ;
4.  $E$ -labels:  $E(v, t)$ , where  $v \in [n]$ ,  $t \in [M]$ ;
5.  $H$ -garbage-labels:  $G_H(v, t)$ , where  $v \in [n]$ ,  $t \in [M]$ ;
6.  $E$ -garbage-labels:  $G_E(v, t)$ , where  $v \in [n]$ ,  $t \in [M]$ ;

Besides, we maintain a pointer  $\mathcal{P}$  for the spare labels via classical register, which costs  $O(\log M) = O(\log m)$  bits. Because the basis state is specified by a variable index and a level in  $[M]$ , this quantum register costs  $O(\log(nm)) = O(\log n)$  qubits, for  $m = \text{poly}(n)$ .

**What the labels are for.** The spare labels are only temporary labels used to make every update a permutation. They are not queried directly.

The  $H$ -labels stores the prefix total degree information. Every time a tagged unit on  $u$  arrives, we increase the  $H(u, \cdot)$  labels, so the queried label  $H(u, t)$  should be thought of as the level  $t$  of the prefix total degree. The queries on  $H$ -labels will tell us whether the prefix degree has already entered the scale  $(D_a, D_{a+1}]$ .

The  $E$ -labels stores the encoded quantity. Besides the one-step update for every tagged unit, a selected positive tagged unit applies  $B_a$  additional one-step shifts to the  $E$ -labels. So the queried label  $E(u, t)$  should be thought of as recording

$$\Gamma_a(d_u^{\leq C}, s_u^{\leq C}) = B_a s_u^{\leq C} + d_u^{\leq C}.$$

The queries on  $E$ -labels will tell us which sampled positive-prefix count is consistent with the current prefix.

The garbage labels do not store any statistic we need. They are only there so that every update is a permutation on a finite set of basis labels.

**Definition 12** (Increment operation). Given a variable  $u \in [n]$  and the current pointer  $\mathcal{P}$ , define  $\text{inc}(H, u, 1)$  as follows.

1. Apply the cyclic permutation of basis labels whose cycle is

$$S_{\mathcal{P}} \rightarrow H(u, 1) \rightarrow H(u, 2) \rightarrow \dots \rightarrow H(u, M) \rightarrow G_H(u, M) \rightarrow \dots \rightarrow G_H(u, 1) \rightarrow S_{\mathcal{P}}.$$

We call this one elementary shift. Its quantum part is unitary because it is a permutation of basis labels.

2. Move the pointer to the next spare label. If  $\mathcal{P} < M$ , set  $\mathcal{P} \leftarrow \mathcal{P} + 1$ . If  $\mathcal{P} = M$ , set  $\mathcal{P} \leftarrow 1$ . Thus the pointer runs through  $S_1, S_2, \dots, S_M$  and then starts again from  $S_1$ .

For any integer  $r \geq 1$ ,  $\text{inc}(H, u, r)$  means repeating this elementary shift  $r$  times, updating  $\mathcal{P}$  after each repetition, saying

$$\text{inc}(H, u, r) := \underbrace{\text{inc}(H, u, 1) \circ \dots \circ \text{inc}(H, u, 1)}_{r \text{ times}}.$$

For  $E$ -labels, we use the same shift operation on the chain  $E(u, \cdot)$ ; in particular,  $\text{inc}(E, u, B_a)$  means applying this one-step shift  $B_a$  times, so it advances  $\mathcal{P}$  by  $B_a$ .

The concrete implementations differ slightly for the four sub-sketches: Lower- $q$ -Sketch ( $q \geq 1$ ), Upper- $q$ -Sketch ( $q \geq 1$ ), Lower-0-Sketch and Upper-0-Sketch. But the overall structure of the algorithm is the same. Therefore, we present the **Lower- $q$ -Sketch** ( $q \geq 1$ ) in detail, while for other sub-sketches we focus primarily on the aspects that differ. In this subsection, call a clause  $C$  *target-shape* for the one-endpoint coordinate  $U_{\ell, j, \alpha}^{(a)}$  if  $|C| = \ell$  and the  $j$ -th literal has sign  $s_\alpha$ . To make the subsequent formulas more concise, let  $\text{Tgt}_C^{\ell, j, \alpha}$  be the event that  $C$  is target-type for  $U_{\ell, j, \alpha}^{(a)}$ .

**Initialization:** The quantum register is initialized as:

$$|Q(T_0)\rangle = \frac{|\perp\rangle + |S_1\rangle + \dots + |S_{M-1}\rangle}{\sqrt{M}},$$

where the live set

$$T_0 := \{S_1, S_2, \dots, S_{M-1}\}.$$

Initialize  $\mathcal{P} = 1$ .

When a clause  $C$  arrives, the quantum algorithm performs two types of operations in turn: update and query.

**Update.** Decompose the clause  $C$  into tagged units and handle them one by one. Suppose the tagged unit is on variable  $u$  and the live set is  $T$ . Perform the following increment operations in turn. Each operation applies a unitary permutation to the quantum register and updates the pointer as specified above.

**Step-1.**  $\text{inc}(H, u, 1)$ .

**Step-2.**  $\text{inc}(E, u, 1)$ .

**Step-3.** If this tagged unit is positive and selected by  $f_a$  (See Definition 8 for details), perform  $\text{inc}(E, u, B_a)$ .

**Query.** If the clause  $C$  is not target-shape, do nothing and end. Otherwise denote the  $j$ -th literal by  $v := v_j(C)$ . Perform the pair-query

$$(H(v, D_a + 1), E(v, qB_a))$$

on the current quantum register. Denote the output of pair-query by  $\chi_C^\ell \in \{-1, 0, +1\}$ .

**Branch 1:** if  $\chi_C^\ell = 0$ , we say this query *fails*. Then do nothing and end.

**Branch 2:** if  $\chi_C^\ell \in \{-1, +1\}$ , we say this query is *successful*. Then this quantum register performs no further updates or queries. During the rest of the stream, the sub-sketch only keeps the following classical data:

1. store the variable  $v$  and clause  $C$ ;
2. store the nonzero measurement outcome  $\chi_C^\ell$ ;
3. store  $q$ ;
4. in the following streaming of clauses, count the suffix total degree  $d_v^{>C}$  and suffix positive count  $p_v^{>C}$  accurately.

The classical part costs  $O(\log(n+m))$  bits. We say a sub-sketch is *active* if the quantum part has not ended, equivalently, if it has not entered Branch 2 yet.

**Output.** If a high-degree sub-sketch succeeds on  $C$ , then after the rest of the stream has been processed it has the exact suffix counts  $p_v^{>C}$  and  $d_v^{>C}$ . For every  $r \in \{0, \dots, \kappa\}$ , define

$$F_r(C) := \mathbf{1} \left[ \text{clip}_{[-1,1]} \left( \frac{2 \left( \frac{2D_a}{\kappa} r + p_v^{>C} \right)}{D_a + d_v^{>C}} - 1 + g(v) \right) \in I_{i_\alpha} \right].$$

Finally, the Lower- $q$ -Sketch ( $q \geq 1$ ) outputs

$$L_q(C) := \frac{M}{2} \cdot \chi_C^\ell \cdot (F_q(C) - F_{q-1}(C)).$$

In all other cases, the Lower- $q$ -Sketch ( $q \geq 1$ ) outputs 0.

Next we describe the other three sub-sketches. Most of the algorithm is the same, so we only describe the parts that differ from the Lower- $q$ -Sketch ( $q \geq 1$ ).

**Upper- $q$ -Sketch** ( $q \geq 1$ ) In the query step, we perform the pair-query on

$$(H(v, D_{a+1} + 1), E(v, qB_a))$$

rather than  $(H(v, D_a + 1), E(v, qB_a))$  for Lower- $q$ -Sketch.

Similarly, define  $\chi_C^u \in \{-1, 0, 1\}$  for the output of the pair-query. In its successful branch, the Upper- $q$ -Sketch stores the nonzero value of  $\chi_C^u$  together with the same classical data. For Branch 2, define the final output as

$$U_q(C) := \frac{M}{2} \cdot \chi_C^u \cdot (F_q(C) - F_{q-1}(C)).$$

In all other cases, it outputs 0.

**Lower-0-Sketch and Upper-0-Sketch.** For the 0 case, the main difference is that the assumed prefix positive count is 0. Hence we do not need to maintain this information, and we no longer measure the  $E$ -labels or  $E$ -garbage-labels. The register only needs the  $H$ -labels, and the update step per tagged unit is the single increment  $\text{inc}(H, u, 1)$ .

In the query step, the Lower-0-Sketch performs a single-query on  $H(v, D_a + 1)$  and the Upper-0-Sketch performs a single-query on  $H(v, D_{a+1} + 1)$ .

Define  $\xi_C^\ell$  and  $\xi_C^u$  to denote the outputs of the single-query from Lower-0-Sketch and Upper-0-Sketch respectively. In a successful branch the sketch stores this nonzero output, which equals 1, and the same classical data and suffix counters as above. Define the final outputs as

$$L_0(C) := M \cdot \xi_C^\ell \cdot F_0(C), \quad U_0(C) := M \cdot \xi_C^u \cdot F_0(C),$$

In all other cases, output 0.

**Correctness.** Fix a one-endpoint sub-sketch and a label  $F(u, t)$  that it may query, where  $F$  could denote  $H$  or  $E$ -labels. We first show that, as long as the pointer has not wrapped around,  $F(u, t)$  is live exactly when  $d_u + Bc_u \geq t$ . Lemma 9 will then show that the assumption fails with probability at most  $1/160$ .

For this fixed  $F$  and  $u$ , the sub-sketch queries only this level  $t$ . Just before such a query, let  $d_u$  count the calls  $\text{inc}(F, u, 1)$ , and let  $c_u$  count the calls  $\text{inc}(F, u, B_a)$ . If this family has no such  $\text{inc}(F, u, B_a)$  calls, set  $c_u = 0$ .

**Lemma 8.** *Suppose  $t < M$ , and suppose that before every query of this same label, the whole sub-sketch has used fewer than  $M$  elementary shifts in total. Then, with  $d_u$  and  $c_u$  as defined above, immediately before every such query,*

$$F(u, t) \in T \iff d_u + B_a c_u \geq t.$$

**Remark 1.** The same proof also applies to the  $Z_1$ - and  $Z_2$ -labels used later, because those sketches also update first and query afterward. See Section 5.3 for details. Therefore, we will not repeat the proof for the two-endpoint case, but instead directly cite this lemma.

*Proof.* First ignore failed queries. Because the whole sub-sketch has used fewer than  $M$  elementary shifts, the pointer has not wrapped around. On the chain  $F(u, \cdot)$ , the calls counted by  $d_u$  and  $c_u$  move live labels forward by  $d_u + B_a c_u$  levels. Hence  $F(u, t)$  is live exactly when  $d_u + B_a c_u \geq t$ .

It remains to check that failed queries do not break this statement. A failed query can only remove the label being tested. Suppose it removes  $F(u, t)$ , and suppose the same label is queried again later. That later query can occur only on a later clause whose queried endpoint is again  $u$ . In the algorithm, the update step of that later clause is performed before the query, and this update applies at least one one-step shift to the same chain  $F(u, \cdot)$ . Since the total number of shifts is still below  $M$ , the pointer has not returned to a previously used spare position.

If  $t = 1$ , that shift moves a live spare label into  $F(u, 1)$ . If  $t > 1$ , then  $F(u, t - 1)$  was live when  $F(u, t)$  was removed: the live labels had already reached level  $t$ , and this sub-sketch does not query level  $t - 1$  of the same chain. The next shift moves that live label into  $F(u, t)$ . Thus a failed deletion of  $F(u, t)$  is repaired before the same label is queried again. The same condition for being live therefore holds before every query of  $F(u, t)$ .  $\square$

The previous lemma needs two assumptions: the queried level  $t$  is below  $M$ , and the total number of elementary shifts in the sub-sketch is below  $M$ . The next lemma checks both. The level check is deterministic after  $C_\varepsilon$  is chosen large enough; the shift bound is the part that uses the public randomness. Although the low-degree one-endpoint sketches are described later, we include them in the same statement because they use the same shift operation.

**Lemma 9** (No-wrap event for one-endpoint sketches). *Choose the constant  $C_\varepsilon$  in  $M = \lceil C_\varepsilon m \rceil$  large enough. Then every queried level in these sub-sketches lies in  $[M]$ . Moreover, there is an event  $\mathcal{G}_1$ , depending only on the public hash functions, such that  $\Pr[\mathcal{G}_1] \geq 1 - 1/160$ , and on  $\mathcal{G}_1$ , every one-endpoint sub-sketch performs fewer than  $M$  elementary shifts during the whole stream.*

*Proof.* The stream contains at most  $4m$  tagged units in total. If  $m = 0$ , no update is applied and the claim is immediate. Assume  $m \geq 1$ .

First consider a low-degree sub-sketch, so  $D_a < \kappa/2$ . Then  $D_{a+1} = O_\varepsilon(1)$ . Every tagged unit applies  $O_\varepsilon(1)$  elementary shifts: it always applies the one-step updates, and a positive tagged unit applies at most  $B_a = O_\varepsilon(1)$  additional shifts. Therefore the total number of elementary shifts in this sub-sketch is deterministically  $O_\varepsilon(m)$ .

Now consider a high-degree scale  $a$ , so  $D_a \geq \kappa/2$ . Let  $X_a$  be the number of positive tagged units selected by the public hash  $f_a$  in the whole stream. There is some  $N_a \leq 4m$  such that

$$X_a \sim \text{Bin}\left(N_a, \frac{\kappa}{2D_a}\right), \quad \mathbb{E}[X_a] \leq \frac{2\kappa m}{D_a}.$$

Choose a large constant  $K_\varepsilon$ . A Chernoff bound gives

$$\Pr\left[X_a > K_\varepsilon \frac{m}{D_a}\right] \leq \exp\left(-c_\varepsilon \frac{m}{D_a}\right)$$

for some constant  $c_\varepsilon > 0$ .

On the event

$$X_a \leq K_\varepsilon \frac{m}{D_a},$$

the extra number of elementary shifts caused by sampled positive tagged units is at most

$$B_a X_a \leq (2K_{\text{base}} D_{a+1} + 1) \cdot K_\varepsilon \frac{m}{D_a} = O_\varepsilon(m),$$

because  $D_{a+1} \leq (1+\varepsilon^3)D_a + 1$ . The deterministic one-step updates contribute another  $O(m)$  shifts, so the total number of elementary shifts is  $O_\varepsilon(m)$ .

It remains to check the queried levels. In the low-degree one-endpoint sketches, the queried levels are

$$B_a c + d, \quad B_a c + d + 1,$$

where  $D_a < d \leq D_{a+1}$  and  $0 \leq c \leq d$ . Since  $D_{a+1} = O_\varepsilon(1)$ , these levels are  $O_\varepsilon(1)$ .

In the high-degree one-endpoint sketches, the queried levels are

$$D_a + 1, \quad D_{a+1} + 1, \quad qB_a \quad (0 \leq q \leq \kappa).$$

Here  $D_{a+1} = O_\varepsilon(m)$ ,  $B_a = O_\varepsilon(D_{a+1})$ , and  $\kappa = O_\varepsilon(1)$ , so these levels are  $O_\varepsilon(m)$ . After increasing  $C_\varepsilon$  if necessary, all queried levels lie in  $[M]$ .

The event above depends only on the scale  $a$ , not on the endpoint position, sign-bucket label, or the value of  $q$ . Thus, once it holds for scale  $a$ , every one-endpoint sub-sketch on this scale uses  $O_\varepsilon(m)$  elementary shifts. After increasing  $C_\varepsilon$  again if necessary, this number is less than  $M$ .

Now let  $\mathcal{G}_1$  be the event that  $X_a \leq K_\varepsilon \frac{m}{D_a}$  holds for every high-degree scale  $a$ . Low-degree scales need no probability bound, as shown above. Because the scales are geometric, there are only  $O_\varepsilon(1)$  scales in each dyadic range  $D_a \in [m/2^{r+1}, m/2^r]$ . There are also  $O_\varepsilon(1)$  high-degree scales with  $m < D_a \leq 4m$ . Their total failure probability can be made no larger than the first term of the dyadic sum by increasing  $K_\varepsilon$ . Thus,

$$\sum_{a: D_a \geq \kappa/2} \Pr\left[X_a > K_\varepsilon \frac{m}{D_a}\right] \leq C'_\varepsilon \sum_{r \geq 0} \exp(-c'_\varepsilon 2^r) < \frac{1}{160}$$

after increasing  $K_\varepsilon$  if necessary. Therefore,  $\Pr[\mathcal{G}_1] \geq 1 - \frac{1}{160}$  on  $\mathcal{G}_1$ , every one-endpoint sub-sketch uses fewer than  $M$  elementary shifts over the whole stream, and every queried level lies in  $[M]$ . This proves the lemma.  $\square$

If  $\mathcal{G}_1$  fails, the algorithm still runs legally: the pointer is cyclic, each update is still a permutation of the finite set of labels, and each query is still one of the measurements defined earlier. What may fail is only the simple interpretation of queried labels by prefix counts.

On  $\mathcal{G}_1$ , Lemma 8 applies to every label queried by a one-endpoint sketch. Thus, immediately before each query, an  $H$ -label is live exactly when the corresponding prefix degree has reached its queried level, and an  $E$ -label is live exactly when the prefix value encoded by  $\Gamma_a$  has reached its queried level. These are the only facts about the live set used in the estimates below.

**Estimation and Analysis.** All the following arguments are conditioned on the event  $\mathcal{G}_1$  from Lemma 9.

The lower and upper sketches first isolate the degree range  $(D_a, D_{a+1}]$ . The  $q$ -sketches then recover the sampled positive-prefix count by a telescoping sum. Thus the proof below has two steps: first isolate the scale, and then use the telescoping sum over  $q$  to evaluate the bucket condition from the sampled positive-prefix count.

Extend the outputs by 0 to every clause that is not target-shape. For each clause  $C$ , define

$$\widehat{U}_{\ell,j,\alpha}^{(a)}(C) := L_0(C) - U_0(C) + \sum_{q=1}^{\kappa} (L_q(C) - U_q(C)). \quad (9)$$

Set

$$\widehat{U}_{\ell,j,\alpha}^{(a)} := \sum_{C \in \Phi_{\leq 3}} \widehat{U}_{\ell,j,\alpha}^{(a)}(C).$$

Fix a target-shape clause  $C$  and write  $v := v_j(C)$ . For each  $q \in \{0, 1, \dots, \kappa\}$ , let

$$F_q(C) := \mathbf{1} \left[ \text{clip}_{[-1,1]} \left( \frac{2 \left( \frac{2D_a}{\kappa} q + p_v^{>C} \right)}{D_a + d_v^{>C}} - 1 + g(v) \right) \in I_{i_\alpha} \right].$$

Recall that  $s_v^{\leq C}$  denotes the number of positive tagged units on variable  $v$  up to clause  $C$ , which are selected by  $f_a$  in Step-3 of the update step. By Definition 10, on the scale  $(D_a, D_{a+1}]$  the weighted pseudobias uses

$$\widetilde{d}_v^{\leq C} = D_a, \quad \widetilde{p}_v^{\leq C} = \frac{2D_a}{\kappa} \min\{s_v^{\leq C}, \kappa\}.$$

Hence

$$\mathbf{1}[\lambda_j(C) = \alpha] = F_{\min\{s_v^{\leq C}, \kappa\}}(C).$$

Therefore,

$$\mathbf{1}[\lambda_j(C) = \alpha] = F_0(C) + \sum_{q=1}^{\kappa} (F_q(C) - F_{q-1}(C)) \mathbf{1}[s_v^{\leq C} \geq q]. \quad (10)$$

Recall that a sub-sketch is *active* if its quantum part has not ended, equivalently, if it has not entered Branch 2 in the query step.

**Lemma 10** (Active probability). *Fix one sub-sketch and fix the public randomness. Suppose that, conditioned on the event that this sub-sketch is active just before the query on a clause  $C$ , its live set  $T$  has size  $M' - 1$ , where  $M' = 1 + |T|$ . Then*

$$\Pr[\text{this sub-sketch is active just before the query on } C] = \frac{M'}{M}.$$

*Proof.* List the earlier queries of this sub-sketch as  $\text{Query}_1, \dots, \text{Query}_r$ . For  $i = 1, \dots, r + 1$ , let  $M_i$  be the value of  $1 + |T|$  just before  $\text{Query}_i$ , conditioned on the event that the sub-sketch is still active at that time. Let  $M_{r+1} = M'$ . Then  $M_1 = M$ .

A failed query can delete zero, one, or two queried live labels, so  $M_i - M_{i+1} \in \{0, 1, 2\}$ . A failed query leaves the sub-sketch active. Therefore

$$\Pr[\text{the sub-sketch is active after } \text{Query}_i \mid \text{it is active before } \text{Query}_i] = 1 - \frac{M_i - M_{i+1}}{M_i} = \frac{M_{i+1}}{M_i}.$$

Multiplying these conditional probabilities over  $i = 1, \dots, r$  gives

$$\Pr[\text{the sub-sketch is still active just before the query on } C] = \prod_{i=1}^r \frac{M_{i+1}}{M_i} = \frac{M'}{M}.$$

□

Next, we analyze the levels  $q \in \{1, \dots, \kappa\}$  and  $q = 0$  separately. Recall that  $\text{Tgt}_C^{\ell, j, \alpha}$  denotes the event that  $|C| = \ell$  and the  $j$ -th literal of  $C$  has sign  $s_\alpha$ , i.e.  $C$  is the target-type for  $U_{\ell, j, \alpha}^{(a)}$ .

**For**  $q \in \{1, \dots, \kappa\}$ . Suppose the live set just before the query on  $C$  has size  $M' - 1$ .

By definition,

$$L_q(C) = \frac{M}{2} \chi_C^\ell (F_q(C) - F_{q-1}(C)).$$

The Lower- $q$ -Sketch queries the pair

$$(H(v, D_a + 1), E(v, qB_a)).$$

So the pair-query formula gives

$$\mathbb{E}[\chi_C^\ell \mid |Q(T)|] = \frac{2}{M'} \mathbf{1}[H(v, D_a + 1) \in T] \mathbf{1}[E(v, qB_a) \in T].$$

Therefore

$$\mathbb{E}[L_q(C) \mid |Q(T)|] = \frac{M}{M'} (F_q(C) - F_{q-1}(C)) \cdot \mathbf{1}[\text{Tgt}_C^{\ell, j, \alpha}] \cdot \mathbf{1}[H(v, D_a + 1) \in T] \cdot \mathbf{1}[E(v, qB_a) \in T].$$

On the event  $\mathcal{G}_1$  from Lemma 9, the queried label  $H(v, D_a + 1)$  is live if and only if

$$d_v^{\leq C} \geq D_a + 1,$$

and the queried label  $E(v, qB_a)$  is live if and only if

$$\Gamma_a(d_v^{\leq C}, s_v^{\leq C}) \geq qB_a.$$

So

$$\mathbb{E}[L_q(C) \mid |Q(T)|] = \frac{M}{M'} (F_q(C) - F_{q-1}(C)) \cdot \mathbf{1}[\text{Tgt}_C^{\ell, j, \alpha}] \cdot \mathbf{1}[d_v^{\leq C} \geq D_a + 1] \cdot \mathbf{1}[\Gamma_a(d_v^{\leq C}, s_v^{\leq C}) \geq qB_a].$$

Similarly,

$$\mathbb{E}[U_q(C) \mid |Q(T)|] = \frac{M}{M'} (F_q(C) - F_{q-1}(C)) \cdot \mathbf{1}[\text{Tgt}_C^{\ell, j, \alpha}] \cdot \mathbf{1}[d_v^{\leq C} \geq D_{a+1} + 1] \cdot \mathbf{1}[\Gamma_a(d_v^{\leq C}, s_v^{\leq C}) \geq qB_a].$$

By Lemma 10,

$$\Pr[\text{this sub-sketch is active just before the query on } C] = \frac{M'}{M}.$$

Multiplying by this factor gives

$$\mathbb{E}[L_q(C) - U_q(C)] = (F_q(C) - F_{q-1}(C)) \cdot \mathbf{1}[\text{Tgt}_C^{\ell, j, \alpha}] \cdot \mathbf{1}[D_a < d_v^{\leq C} \leq D_{a+1}] \cdot \mathbf{1}[\Gamma_a(d_v^{\leq C}, s_v^{\leq C}) \geq qB_a].$$

Now assume

$$D_a < d_v^{\leq C} \leq D_{a+1}.$$

By (7), the last condition is equivalent to  $s_v^{\leq C} \geq q$ . Hence

$$\mathbb{E}[L_q(C) - U_q(C)] = (F_q(C) - F_{q-1}(C)) \mathbf{1}[\text{Tgt}_C^{\ell, j, \alpha}, D_a < d_v^{\leq C} \leq D_{a+1}, s_v^{\leq C} \geq q]. \quad (11)$$

**For  $q = 0$ .** Now the query is a single-query. Since

$$L_0(C) = M \cdot \xi_C^\ell \cdot F_0(C),$$

and the Lower-0-Sketch queries  $H(v, D_a + 1)$ , the single-query formula gives

$$\mathbb{E}[\xi_C^\ell \mid |Q(T)\rangle] = \frac{1}{M'} \mathbf{1}[H(v, D_a + 1) \in T].$$

Therefore

$$\mathbb{E}[L_0(C) \mid |Q(T)\rangle] = \frac{M}{M'} F_0(C) \mathbf{1}[\text{Tgt}_C^{\ell,j,\alpha}] \mathbf{1}[H(v, D_a + 1) \in T].$$

On the event  $\mathcal{G}_1$ , this becomes

$$\mathbb{E}[L_0(C) \mid |Q(T)\rangle] = \frac{M}{M'} F_0(C) \mathbf{1}[\text{Tgt}_C^{\ell,j,\alpha}, d_v^{\leq C} \geq D_a + 1].$$

Similarly, the Upper-0-Sketch queries  $H(v, D_{a+1} + 1)$ , so

$$\mathbb{E}[U_0(C) \mid |Q(T)\rangle] = \frac{M}{M'} F_0(C) \mathbf{1}[\text{Tgt}_C^{\ell,j,\alpha}, d_v^{\leq C} \geq D_{a+1} + 1].$$

Using Lemma 10 again, we get

$$\mathbb{E}[L_0(C) - U_0(C)] = F_0(C) \mathbf{1}[\text{Tgt}_C^{\ell,j,\alpha}, D_a < d_v^{\leq C} \leq D_{a+1}]. \quad (12)$$

Finally, combining Equality (9) (10) (11) and (12), define

$$A_a(C) := \mathbf{1}[\text{Tgt}_C^{\ell,j,\alpha}, D_a < d_v^{\leq C} \leq D_{a+1}].$$

Then we have

$$\begin{aligned} \mathbb{E}[\widehat{U}_{\ell,j,\alpha}^{(a)}(C)] &= A_a(C) \left( F_0(C) + \sum_{q=1}^{\kappa} (F_q(C) - F_{q-1}(C)) \mathbf{1}[s_v^{\leq C} \geq q] \right) \\ &= \mathbf{1}[\text{Tgt}_C^{\ell,j,\alpha}, D_a < d_v^{\leq C} \leq D_{a+1}, \lambda_j(C) = \alpha]. \end{aligned} \quad (13)$$

Therefore

$$\mathbb{E}[\widehat{U}_{\ell,j,\alpha}^{(a)}] = U_{\ell,j,\alpha}^{(a)}.$$

### 5.2.2 Low-degree case

For the low-degree  $U_{\ell,j,\alpha}^{(a)}$ , the relevant prefix total degree and prefix positive count of  $v_j(C)$  range over a constant-size set. We perform sketches for all the possible  $(d, c)$  pairs, where  $d \in (D_a, D_{a+1}]$  and  $c \in [0, d]$ . Each pair represents the candidate values  $d_v^{\leq C} = d$  and  $p_v^{\leq C} = c$ . Moreover, for each  $(d, c)$  we maintain a  $(d, c)$ -Sketch.

Each  $(d, c)$ -Sketch consists of analogous Lower- $(d, c)$ -Sketch and Upper- $(d, c)$ -Sketch. They each have an identical quantum register. The quantum register and update operations are the same as in the high-degree case, except for the positive-count update described next.

For the update step, Step-3 is replaced as follows: if this tagged unit is positive, perform  $\text{inc}(E, u, B_a)$ .

For the query step, the Lower- $(d, c)$ -Sketch performs a single-query on

$$E(v, B_a c + d)$$

and the Upper- $(d, c)$ -Sketch performs single-query on

$$E(v, B_a c + d + 1).$$

In the low-degree sketch, the query tests only the value  $\Gamma_a(d, c)$ ; it does not separately test whether the prefix degree lies in  $(D_a, D_{a+1}]$ . Therefore an endpoint whose true prefix degree is outside this interval may still pass the queried threshold. The condition (8) was chosen to count every such possible extra contribution. If this condition does not occur, the queried difference isolates the intended low-degree state  $(d, c)$ .

**Output.** After a successful query on a target-shape clause  $C$ , the corresponding Lower- $(d, c)$ -Sketch or Upper- $(d, c)$ -Sketch stores the nonzero single-query output, which equals 1, and computes

$$F_{d,c}(C) := \mathbf{1} \left[ \text{clip}_{[-1,1]} \left( \frac{2(c + p_v^{>C})}{d + d_v^{>C}} - 1 + g(v) \right) \in I_{i_\alpha} \right]$$

by substituting the exact prefix values

$$d_v^{\leq C} = d, \quad p_v^{\leq C} = c$$

into the weighted pseudobias formula of Definition 10. Define

$$L_{d,c}(C) := M \cdot \xi_{d,c}^\ell(C) \cdot F_{d,c}(C), \quad U_{d,c}(C) := M \cdot \xi_{d,c}^u(C) \cdot F_{d,c}(C),$$

where  $\xi_{d,c}^\ell(C), \xi_{d,c}^u(C) \in \{0, 1\}$  are the one-query outputs of the Lower- $(d, c)$ -Sketch and Upper- $(d, c)$ -Sketch, respectively. Extend both outputs by 0 to every clause that is not target-shape.

**Estimation and Analysis.** For each clause  $C$ , define

$$\widehat{U}_{\ell,j,\alpha}^{(a)}(C) := \sum_{d=D_a+1}^{D_{a+1}} \sum_{c=0}^d (L_{d,c}(C) - U_{d,c}(C)),$$

and

$$\widehat{U}_{\ell,j,\alpha}^{(a)} := \sum_{C \in \Phi_{\leq 3}} \widehat{U}_{\ell,j,\alpha}^{(a)}(C).$$

Fix a clause  $C$  and write  $v := v_j(C)$ . If the live set just before the query in the Lower- $(d, c)$ -Sketch has size  $M' - 1$ , then the one-query property gives

$$\mathbb{E}[L_{d,c}(C) \mid |Q(T)\rangle] = \frac{M}{M'} F_{d,c}(C) \mathbf{1} \left[ \text{Tgt}_C^{\ell,j,\alpha}, \Gamma_a(d_v^{\leq C}, p_v^{\leq C}) \geq B_a c + d \right].$$

Similarly,

$$\mathbb{E}[U_{d,c}(C) \mid |Q(T)\rangle] = \frac{M}{M'} F_{d,c}(C) \mathbf{1} \left[ \text{Tgt}_C^{\ell,j,\alpha}, \Gamma_a(d_v^{\leq C}, p_v^{\leq C}) \geq B_a c + d + 1 \right].$$

Since the probability that the quantum part of this sub-sketch is active just before the query is  $M'/M$ ,

$$\mathbb{E}[L_{d,c}(C) - U_{d,c}(C)] = F_{d,c}(C) \mathbf{1} \left[ \text{Tgt}_C^{\ell,j,\alpha}, \Gamma_a(d_v^{\leq C}, p_v^{\leq C}) = B_a c + d \right].$$

If the endpoint  $v_j(C)$  does not impersonate scale  $a$ , then Lemma 7 rules out hits from degrees outside  $(D_a, D_{a+1}]$ , and the threshold identity (6) applies inside the interval. Thus

$$\Gamma_a(d_v^{\leq C}, p_v^{\leq C}) = B_a c + d \iff d_v^{\leq C} = d, \quad p_v^{\leq C} = c.$$

Therefore, on the event that  $v_j(C)$  does not impersonate scale  $a$ ,

$$\mathbb{E}[L_{d,c}(C) - U_{d,c}(C)] = F_{d,c}(C) \mathbf{1} \left[ \text{Tgt}_C^{\ell,j,\alpha}, d_v^{\leq C} = d, p_v^{\leq C} = c \right].$$

Whenever

$$d_v^{\leq C} = d, \quad p_v^{\leq C} = c,$$

the values substituted in  $F_{d,c}(C)$  are exactly those in Definition 10, so

$$F_{d,c}(C) = \mathbf{1}[\lambda_j(C) = \alpha].$$

Summing over all pairs  $(d, c)$  therefore gives

$$\mathbb{E} \left[ \widehat{U}_{\ell,j,\alpha}^{(a)}(C) \right] = \mathbf{1} \left[ \text{Tgt}_C^{\ell,j,\alpha}, D_a < d_v^{\leq C} \leq D_{a+1}, \lambda_j(C) = \alpha \right]$$

whenever the endpoint  $v_j(C)$  does not impersonate scale  $a$ .

For a fixed clause  $C$  and scale  $a$ , at most one pair  $(d, c)$  can satisfy  $\Gamma_a(d_v^{\leq C}, p_v^{\leq C}) = B_a c + d$  with  $D_a < d \leq D_{a+1}$ , because equality for two pairs  $(d, c)$  and  $(d', c')$  would imply  $d - d' = B_a(c' - c)$ , and  $|d - d'| < B_a$  forces  $c = c'$  and then  $d = d'$ . Hence, for this fixed clause and scale, the total contribution of all low-degree  $(d, c)$ -sketches has absolute value at most 1. Therefore, in all cases

$$\left| \mathbb{E} \left[ \widehat{U}_{\ell,j,\alpha}^{(a)}(C) \right] - \mathbf{1} \left[ \text{Tgt}_C^{\ell,j,\alpha}, D_a < d_v^{\leq C} \leq D_{a+1}, \lambda_j(C) = \alpha \right] \right| \leq \mathbf{1} \left[ \text{Tgt}_C^{\ell,j,\alpha} \right] \cdot \mathbf{1} \left[ \text{Imp}_{C,j}^a \right]. \quad (14)$$

Recall that  $\text{Imp}_{C,j}^a$  denotes the event that endpoint  $(C, j)$  impersonates scale  $a$ .

Hence

$$\left| \mathbb{E} \left[ \widehat{U}_{\ell,j,\alpha}^{(a)} \right] - U_{\ell,j,\alpha}^{(a)} \right| \leq \sum_{C \in \Phi_{\leq 3}} \mathbf{1} \left[ \text{Tgt}_C^{\ell,j,\alpha}, \text{Imp}_{C,j}^a \right]. \quad (15)$$

### 5.3 Two-endpoint estimation

Fix a two-endpoint coordinate  $P_{\ell,j,t,\alpha,\beta}^{(a,b)}$ . Here  $1 \leq j < t \leq \ell$ ,  $\alpha = (s_\alpha, i_\alpha)$  and  $\beta = (s_\beta, i_\beta)$  are sign-bucket pairs.  $a$  and  $b$  correspond to the scales  $(D_a, D_{a+1}]$  and  $(D_b, D_{b+1}]$ .

We call a clause  $C$  *target-shape* for the two-endpoint coordinate  $P_{\ell,j,t,\alpha,\beta}^{(a,b)}$  if  $|C| = \ell$ , the  $j$ -th literal of  $C$  has sign  $s_\alpha$ , and the  $t$ -th literal of  $C$  has sign  $s_\beta$ . To make the subsequent formulas more concise, let  $\text{Tgt}_C^{\ell,j,t,\alpha,\beta}$  be the event that  $C$  is the target-type for  $P_{\ell,j,t,\alpha,\beta}^{(a,b)}$ .

For the first endpoint  $v_j(C)$ , when  $2D_a \geq \kappa$ , recall the definition

$$s_{v_j(C)}^{\leq C} := \# \left\{ \text{positive tagged units of } v_j(C) \text{ that are selected by } f_a \text{ up to } C \right\}.$$

For the second endpoint  $v_t(C)$ , when  $2D_b \geq \kappa$ ,

$$s_{v_t(C)}^{\leq C} := \# \left\{ \text{positive tagged units of } v_t(C) \text{ that are selected by } f_b \text{ up to } C \right\}.$$

Impersonation also plays an important role in the error counting here. Recall that  $a(C, r)$  denotes the true scale of the endpoint  $(C, r)$ , namely the scale satisfying

$$D_{a(C,r)} < d_{v_r(C)}^{\leq C} \leq D_{a(C,r)+1}.$$

For any queried nonempty scale  $s$ , we say that the endpoint  $(C, r)$  impersonates scale  $s$  if it satisfies the scale- $s$  impersonation condition from (8). In the error count below, we only count such events when  $s \neq a(C, r)$ . This condition only checks the prefix degree, and it is chosen to include every case in which  $(C, r)$  actually gives a nonzero contribution to a queried wrong scale  $s$ .

To see this, let  $c_{\text{enc}}$  be the positive-prefix count encoded by the scale- $s$  sketch on this endpoint: it is the sampled count on a high-degree scale and the exact positive-prefix count on a low-degree scale. If such a nonzero contribution occurs, then for some queried count index  $c_0$  and some  $d_0$  with  $D_s < d_0 \leq D_{s+1}$ ,

$$d_{v_r(C)}^{\leq C} + B_s c_{\text{enc}} = B_s c_0 + d_0.$$

Hence

$$d_{v_r(C)}^{\leq C} = (c_0 - c_{\text{enc}})B_s + d_0.$$

For a wrong-scale nonzero contribution, the integer  $h := c_0 - c_{\text{enc}}$  is one of the values allowed in (8). Therefore this endpoint is included in the larger set defined by the impersonation condition. The converse is not claimed: an endpoint may satisfy the interval condition in (8) but still contribute zero, because its encoded positive-prefix count may not place  $d + B_s c_{\text{enc}}$  at the queried level.

The coordinate output defined in this section is the actual algorithmic output for  $P_{\ell,j,t,\alpha,\beta}^{(a,b)}$ . We compare it directly with the target coordinate, which counts clauses whose two queried endpoints have true scales  $a$  and  $b$ . Once the no-wrap event below makes the queried labels represent the intended threshold inequalities, a fixed queried scale pair  $(a, b)$  can differ from the target coordinate only in the following two ways:

1. a clause whose true scale pair is not  $(a, b)$  can still give a nonzero term to the coordinate output for  $(a, b)$ , because the first endpoint satisfies the scale- $a$  impersonation condition or the second endpoint satisfies the scale- $b$  impersonation condition;
2. a true high-degree endpoint overflows, meaning that its sampled positive-prefix count is at least  $\kappa$  while the sketch only queries the values  $0, 1, \dots, \kappa - 1$ .

The first case can make the queried coordinate include a clause that the target coordinate does not include; the second case can make the queried coordinate miss a clause that the target coordinate includes.

### 5.3.1 High-degree on both endpoints

We first discuss the case  $D_a \geq \kappa/2$  and  $D_b \geq \kappa/2$ . The remaining cases, where at least one scale is low-degree, are handled in Section 5.3.2; only the queried levels and positive-count updates change.

In this case both endpoints use the sampled positive-prefix counts  $s_v^{\leq C}$  from Definition 10, and we keep only the levels  $\{0, 1, \dots, \kappa - 1\}$ . We use the random-base threshold encoding from the one-endpoint section on both sides:

$$\Gamma_a(d, c) = d + B_a c, \quad \Gamma_b(d, c) = d + B_b c,$$

where  $B_a$  and  $B_b$  are the public-random bases chosen in the one-endpoint section. Set

$$\Sigma_a := \{0, 1, \dots, \kappa - 1\}, \quad \Sigma_b := \{0, 1, \dots, \kappa - 1\}.$$

For each pair  $(\sigma, \tau) \in \Sigma_a \times \Sigma_b$ , we maintain four independent sub-sketches:

1. Lower-Lower- $(\sigma, \tau)$ -Sketch,
2. Lower-Upper- $(\sigma, \tau)$ -Sketch,
3. Upper-Lower- $(\sigma, \tau)$ -Sketch,
4. Upper-Upper- $(\sigma, \tau)$ -Sketch.

**Quantum register.** Each of the four sub-sketches has an identical quantum register. We use the global register size  $M = \lceil C_\varepsilon m \rceil$ , with  $C_\varepsilon$  chosen large enough for the fixed  $\varepsilon > 0$ . The basis labels are:

1. a fixed anchor basis state  $|\perp\rangle$ ;
2. spare labels  $S_1, \dots, S_M$ ;
3. first-family labels  $Z_1(v, r)$ , where  $v \in [n]$  and  $r \in [M]$ ;
4. second-family labels  $Z_2(v, r)$ , where  $v \in [n]$  and  $r \in [M]$ ;
5. first-family garbage labels  $G_1(v, r)$ , where  $v \in [n]$  and  $r \in [M]$ ;
6. second-family garbage labels  $G_2(v, r)$ , where  $v \in [n]$  and  $r \in [M]$ .

As in the one-endpoint case, we maintain a pointer  $\mathcal{P}$  in a classical register. The pointer runs through the spare labels  $S_1, S_2, \dots, S_M$ : after it reaches  $S_M$ , the next shift uses  $S_1$  again.

In the one-endpoint sketch, the  $H$ - and  $E$ -labels are both used for the same endpoint. Here  $Z_1$  is used for  $v_j(C)$  and  $Z_2$  is used for  $v_i(C)$ , so each side uses one label family to encode the threshold value for that endpoint. Thus each side uses one label family for the combined value  $\Gamma_a(d, c)$  or  $\Gamma_b(d, c)$ , rather than keeping separate labels for the total degree and the positive-prefix count.

For  $i = 1, 2$ ,  $\text{inc}(Z_i, u, r)$  means applying the same one-step shift to the chain  $Z_i(u, \cdot)$  exactly  $r$  times; after each one-step shift, the pointer moves to the next spare label, with  $S_M$  followed by  $S_1$ . As long as fewer than  $M$  elementary shifts have been applied, the queried labels behave exactly as in the one-way shifts

$$S_{\mathcal{P}} \rightarrow Z_1(u, 1) \rightarrow Z_1(u, 2) \rightarrow \dots \rightarrow Z_1(u, M) \rightarrow G_1(u, M) \rightarrow \dots \rightarrow G_1(u, 1) \rightarrow S_{\mathcal{P}},$$

and

$$S_{\mathcal{P}} \rightarrow Z_2(u, 1) \rightarrow Z_2(u, 2) \rightarrow \dots \rightarrow Z_2(u, M) \rightarrow G_2(u, M) \rightarrow \dots \rightarrow G_2(u, 1) \rightarrow S_{\mathcal{P}}.$$

The initial state is

$$|Q(T_0)\rangle = \frac{|\perp\rangle + |S_1\rangle + \dots + |S_{M-1}\rangle}{\sqrt{M}}, \quad T_0 := \{S_1, S_2, \dots, S_{M-1}\}.$$

And initialize  $\mathcal{P} = 1$ . This register uses  $O(\log(n + m))$  qubits, and the pointer uses  $O(\log(m))$  classical bits.

**Update.** Process the tagged units of the current clause one by one. Suppose the tagged unit is on variable  $u$ . In each of the four sub-sketches, perform

1.  $\text{inc}(Z_1, u, 1)$ ;
2.  $\text{inc}(Z_2, u, 1)$ ;
3. if this tagged unit is positive and selected by  $f_a$ , perform  $\text{inc}(Z_1, u, B_a)$ ;
4. if this tagged unit is positive and selected by  $f_b$ , perform  $\text{inc}(Z_2, u, B_b)$ .

**Query.** If the current clause  $C$  is not target-shape, do nothing. Otherwise let

$$u := v_j(C), \quad v := v_t(C).$$

The Lower-Lower- $(\sigma, \tau)$ -Sketch applies the pair-query to

$$\left( Z_1(u, B_a\sigma + D_a + 1), Z_2(v, B_b\tau + D_b + 1) \right).$$

The other three sub-sketches use the same pair-query with the corresponding threshold:

$$\begin{aligned} & \left( Z_1(u, B_a\sigma + D_a + 1), Z_2(v, B_b\tau + D_{b+1} + 1) \right), \\ & \left( Z_1(u, B_a\sigma + D_{a+1} + 1), Z_2(v, B_b\tau + D_b + 1) \right), \\ & \left( Z_1(u, B_a\sigma + D_{a+1} + 1), Z_2(v, B_b\tau + D_{b+1} + 1) \right). \end{aligned}$$

Denote the four outputs of pair-query by

$$\chi_{\sigma,\tau}^{\ell\ell}(C), \quad \chi_{\sigma,\tau}^{\ell u}(C), \quad \chi_{\sigma,\tau}^{u\ell}(C), \quad \chi_{\sigma,\tau}^{uu}(C) \in \{-1, 0, 1\}.$$

If the output is 0, the sketch continues. If the output is  $\pm 1$ , the quantum part of that sub-sketch stops, and we store the nonzero output sign together with  $u$ ,  $v$ ,  $\sigma$ , and  $\tau$ . During the rest of the stream, we count

$$d_u^{>C}, p_u^{>C}, d_v^{>C}, p_v^{>C}$$

exactly using classical counters.

The no-wrap lemma below justifies these threshold tests before each query.

**Output.** If the pair-query output is nonzero, the sub-sketch computes

$$\tilde{b}_u := \text{clip}_{[-1,1]} \left( \frac{2 \left( \frac{2D_a}{\kappa} \sigma + p_u^{>C} \right)}{D_a + d_u^{>C}} - 1 + g(u) \right), \quad \tilde{b}_v := \text{clip}_{[-1,1]} \left( \frac{2 \left( \frac{2D_b}{\kappa} \tau + p_v^{>C} \right)}{D_b + d_v^{>C}} - 1 + g(v) \right),$$

and

$$F_{\sigma,\tau}(C) := \mathbf{1} \left[ \tilde{b}_u \in I_{i_\alpha}, \tilde{b}_v \in I_{i_\beta} \right].$$

This is exactly the indicator that the two endpoints have sign-bucket pairs  $(\alpha, \beta)$ , when  $\sigma$  and  $\tau$  are used as the candidate sampled positive-prefix counts for the two endpoints. The four final outputs are

$$\begin{aligned} Y_{\sigma,\tau}^{\ell\ell}(C) &:= \frac{M}{2} \chi_{\sigma,\tau}^{\ell\ell}(C) F_{\sigma,\tau}(C), & Y_{\sigma,\tau}^{\ell u}(C) &:= \frac{M}{2} \chi_{\sigma,\tau}^{\ell u}(C) F_{\sigma,\tau}(C), \\ Y_{\sigma,\tau}^{u\ell}(C) &:= \frac{M}{2} \chi_{\sigma,\tau}^{u\ell}(C) F_{\sigma,\tau}(C), & Y_{\sigma,\tau}^{uu}(C) &:= \frac{M}{2} \chi_{\sigma,\tau}^{uu}(C) F_{\sigma,\tau}(C). \end{aligned}$$

If a sub-sketch never has a successful pair-query, its output is 0.

**Correctness.** The same no-wrap issue appears here. Each two-endpoint sub-sketch has two label families,  $Z_1$  and  $Z_2$ , but they share one pointer. The analysis only needs the following fact: each queried label represents its intended threshold inequality as long as the sub-sketch performs fewer than  $M$  elementary shifts in total and every queried level lies in  $[M]$ . Under these two conditions, the pointer never returns to an earlier spare position, so the queried labels still represent the thresholds of  $\Gamma_a$  and  $\Gamma_b$ . The following lemma verifies these two conditions for all two-endpoint sub-sketches.

**Lemma 11** (No-wrap event for two-endpoint sketches). *Choose the constant  $C_\varepsilon$  in  $M = \lceil C_\varepsilon m \rceil$  large enough. Then every level queried in these sub-sketches lies in  $[M]$ . Moreover, there is an event  $\mathcal{G}_2$ , depending only on the public hash functions, such that*

$$\Pr[\mathcal{G}_2] \geq 1 - \frac{1}{160},$$

and on  $\mathcal{G}_2$ , in every sub-sketch appearing in the two-endpoint estimation, the total number of elementary shifts applied during all update steps, counting both the  $Z_1$ - and  $Z_2$ -families, is less than  $M$ .

*Proof.* The stream contains at most  $4m$  tagged units in total. If  $m = 0$ , no update is applied and the claim is immediate. Assume  $m \geq 1$ .

First, consider a low-degree side, say  $D_a < \kappa/2$ . Then  $D_{a+1} = O_\varepsilon(1)$ . The deterministic one-step updates for this family apply  $O(m)$  elementary shifts over the whole stream, and every positive tagged unit applies at most  $B_a = O_\varepsilon(1)$  additional elementary shifts. Therefore, this side applies  $O_\varepsilon(m)$  elementary shifts deterministically. The same argument applies to the second side when  $D_b < \kappa/2$ .

Now consider a high-degree scale  $a$ , so  $D_a \geq \kappa/2$ . Let  $X_a$  be the number of positive tagged units selected by the public hash  $f_a$  in the whole stream. There is some  $N_a \leq 4m$  such that

$$X_a \sim \text{Bin}\left(N_a, \frac{\kappa}{2D_a}\right), \quad \mathbb{E}[X_a] \leq \frac{2\kappa m}{D_a}.$$

Choose a large constant  $K_\varepsilon$ . A Chernoff bound gives

$$\Pr\left[X_a > K_\varepsilon \frac{m}{D_a}\right] \leq \exp\left(-c_\varepsilon \frac{m}{D_a}\right)$$

for some constant  $c_\varepsilon > 0$ .

On the event

$$X_a \leq K_\varepsilon \frac{m}{D_a},$$

the extra number of elementary shifts coming from the sampled positive tagged units on this scale is at most

$$B_a X_a \leq (2K_{\text{base}} D_{a+1} + 1) \cdot K_\varepsilon \frac{m}{D_a} = O_\varepsilon(m),$$

because  $D_{a+1} \leq (1 + \varepsilon^3)D_a + 1$ . The deterministic one-step updates contribute another  $O(m)$  elementary shifts. If a two-endpoint sub-sketch has two high-degree sides, the total number of elementary shifts is the sum of the two corresponding bounds, and is still  $O_\varepsilon(m)$ .

It remains only to check that the queried levels themselves are below  $M$ . On a low-degree side with scale  $a$ , the queried levels are

$$B_a c + d, \quad B_a c + d + 1,$$

with  $D_a < d \leq D_{a+1}$  and  $0 \leq c \leq d$ . Since  $D_{a+1} = O_\varepsilon(1)$ , these levels are  $O_\varepsilon(1)$ .

On a high-degree side with scale  $a$ , the queried levels are

$$B_a\sigma + D_a + 1, \quad B_a\sigma + D_{a+1} + 1 \quad (0 \leq \sigma \leq \kappa - 1),$$

and the setup at the start of this section gives  $D_{a+1} = O_\varepsilon(m)$ . Since  $B_a = O_\varepsilon(D_{a+1})$  and  $\kappa = O_\varepsilon(1)$ , these levels are  $O_\varepsilon(m)$ . The same bounds hold for the second side. After enlarging  $C_\varepsilon$  if needed, all queried levels in both families are below  $M$  deterministically.

The event above depends only on the scale  $a$ . It controls the number of elementary shifts caused by selected positive tagged units for every two-endpoint sub-sketch that uses scale  $a$  on either side, because these sub-sketches use the same selected positive tagged units.

Now let  $\mathcal{G}_2$  be the event that  $X_a \leq K_\varepsilon \frac{m}{D_a}$  holds for every high-degree scale  $a$ . Low-degree sides need no probability bound, as shown above. Because the scales are geometric, there are only  $O_\varepsilon(1)$  scales in each dyadic range  $D_a \in [m/2^{r+1}, m/2^r]$ . There are also  $O_\varepsilon(1)$  high-degree scales with  $m < D_a \leq 4m$ . Their total failure probability can be made no larger than the first term of the dyadic sum by increasing  $K_\varepsilon$ . Thus, the following union bound covers all high-degree scales:

$$\sum_{a: D_a \geq \kappa/2} \Pr \left[ X_a > K_\varepsilon \frac{m}{D_a} \right] \leq C'_\varepsilon \sum_{r \geq 0} \exp(-c'_\varepsilon 2^r) < \frac{1}{160}$$

after enlarging  $K_\varepsilon$  if needed.  $C'_\varepsilon$  and  $c'_\varepsilon$  are constants from the Chernoff bound. Therefore

$$\Pr[\mathcal{G}_2] \geq 1 - \frac{1}{160}.$$

On  $\mathcal{G}_2$ , every two-endpoint sub-sketch applies fewer than  $M$  elementary shifts, and all queried levels in both families are below  $M$ . This proves the lemma.  $\square$

As in the one-endpoint case, the correctness argument only needs the fact from Lemma 8 that a fixed queried label has the intended threshold meaning before each query. Work on  $\mathcal{G}_2$ , fix one elementary sub-sketch, and let  $T_a$  and  $T_b$  be the two thresholds queried by that sub-sketch. Immediately before a query on a target-shape clause  $C$ ,

$$Z_1(u, T_a) \in T \iff \Gamma_a(d_u^{\leq C}, s_u^{\leq C}) \geq T_a,$$

$$Z_2(v, T_b) \in T \iff \Gamma_b(d_v^{\leq C}, s_v^{\leq C}) \geq T_b.$$

A failed pair-query can delete one queried label from the  $Z_1$ -family, one queried label from the  $Z_2$ -family, or both. Before any later query of the same label, the next clause containing the relevant variable applies the unconditional one-step update on that label family, even if the tagged unit is not positive or is not selected. Therefore the fixed queried level is restored before it is tested again. Each elementary sub-sketch fixes one queried threshold in each label family throughout the run.

**Estimation of  $P_{\ell,j,t,\alpha,\beta}^{(a,b)}$ .** All the following argument is on the event  $\mathcal{G}_2$  from Lemma 11.

Extend the outputs of the four sub-sketches by 0 to every clause that is not target-shape. For each target-shape clause  $C$ , define

$$\widehat{P}_{\ell,j,t,\alpha,\beta}^{(a,b)}(C) := \sum_{\sigma=0}^{\kappa-1} \sum_{\tau=0}^{\kappa-1} \left( Y_{\sigma,\tau}^{\ell\ell}(C) - Y_{\sigma,\tau}^{\ell u}(C) - Y_{\sigma,\tau}^{u\ell}(C) + Y_{\sigma,\tau}^{uu}(C) \right),$$

and then set

$$\widehat{P}_{\ell,j,t,\alpha,\beta}^{(a,b)} := \sum_C \widehat{P}_{\ell,j,t,\alpha,\beta}^{(a,b)}(C).$$

Recall that  $\mathbf{Tgt}_C^{\ell,j,t,\alpha,\beta}$  denotes the event that  $|C| = \ell$ , the  $j$ -th literal has sign  $s_\alpha$ , and the  $t$ -th literal has sign  $s_\beta$ , i.e.  $C$  is the target-type for  $P_{\ell,j,t,\alpha,\beta}^{(a,b)}$ .

Fix  $\sigma \in \Sigma_a$ ,  $\tau \in \Sigma_b$ , and one of the four sub-sketches. If the live set just before its query on  $C$  has size  $M' - 1$ , then by the pair-query property we have

$$\mathbb{E}\left[Y_{\sigma,\tau}^{\ell\ell}(C) \mid |Q(T)\rangle\right] = \frac{M}{M'} F_{\sigma,\tau}(C) \cdot \mathbf{1}\left[\mathbf{Tgt}_C^{\ell,j,t,\alpha,\beta}, \Gamma_a(d_u^{\leq C}, s_u^{\leq C}) \geq B_a\sigma + D_a + 1, \Gamma_b(d_v^{\leq C}, s_v^{\leq C}) \geq B_b\tau + D_b + 1\right].$$

Exactly as in the one-endpoint case, the probability that the sub-sketch is still active just before that query is  $M'/M$ , so the unconditional expectation is the same indicator without the prefactor  $M/M'$ . The corresponding formulas for  $Y_{\sigma,\tau}^{\ell u}(C)$ ,  $Y_{\sigma,\tau}^{\ell\ell}(C)$ , and  $Y_{\sigma,\tau}^{uu}(C)$  are obtained by replacing the lower threshold on the first endpoint, the second endpoint, or both, by the corresponding upper threshold.

For the first endpoint, assume that  $v_j(C)$  does not impersonate scale  $a$ . Recall that  $a(C, j)$  means the true scale of the first queried endpoint  $v_j(C)$ . Then the difference of the two threshold indicators for the first endpoint has the direct interpretation

$$\begin{aligned} & \mathbf{1}[\Gamma_a(d_u^{\leq C}, s_u^{\leq C}) \geq B_a\sigma + D_a + 1] - \mathbf{1}[\Gamma_a(d_u^{\leq C}, s_u^{\leq C}) \geq B_a\sigma + D_{a+1} + 1] \\ &= \begin{cases} \mathbf{1}[D_a < d_u^{\leq C} \leq D_{a+1}, s_u^{\leq C} = \sigma], & a = a(C, j) \text{ and } s_u^{\leq C} < \kappa, \\ 0, & a \neq a(C, j) \text{ or } s_u^{\leq C} \geq \kappa. \end{cases} \end{aligned}$$

Indeed, if  $a = a(C, j)$  and  $s_u^{\leq C} < \kappa$ , this is exactly (5). If  $a \neq a(C, j)$  and the endpoint does not impersonate scale  $a$ , then this difference is 0. To see this, if  $d_u^{\leq C} \leq D_a$ , the two threshold indicators are equal; if  $d_u^{\leq C} > D_{a+1}$ , a nonzero difference would force scale- $a$  impersonation, contrary to the assumption. If  $a = a(C, j)$  but  $s_u^{\leq C} \geq \kappa$ , then for every  $\sigma \in \{0, 1, \dots, \kappa - 1\}$  both indicators are 1, so the difference is again 0. The same statement holds for the second endpoint with  $(b, \tau)$  in place of  $(a, \sigma)$ .

Therefore, whenever neither endpoint impersonates its queried scale, we have

$$\begin{aligned} & \mathbb{E}\left[Y_{\sigma,\tau}^{\ell\ell}(C) - Y_{\sigma,\tau}^{\ell u}(C) - Y_{\sigma,\tau}^{\ell\ell}(C) + Y_{\sigma,\tau}^{uu}(C)\right] \\ &= F_{\sigma,\tau}(C) \cdot \mathbf{1}[\mathbf{Tgt}_C^{\ell,j,t,\alpha,\beta}] \\ & \quad \cdot \mathbf{1}[a = a(C, j), D_a < d_u^{\leq C} \leq D_{a+1}, s_u^{\leq C} = \sigma < \kappa] \\ & \quad \cdot \mathbf{1}[b = a(C, t), D_b < d_v^{\leq C} \leq D_{b+1}, s_v^{\leq C} = \tau < \kappa]. \end{aligned}$$

Whenever the last two indicators are 1, the prefix values substituted in  $F_{\sigma,\tau}(C)$  are exactly those of Definition 10, so

$$F_{\sigma,\tau}(C) = \mathbf{1}[(\lambda_j(C), \lambda_t(C)) = (\alpha, \beta)].$$

Summing over  $\sigma$  and  $\tau$  gives

$$\mathbb{E}\left[\widehat{P}_{\ell,j,t,\alpha,\beta}^{(a,b)}(C)\right] = \mathbf{1}[C \text{ contributes to } P_{\ell,j,t,\alpha,\beta}^{(a,b)}]$$

whenever neither endpoint impersonates the queried scale, and neither relevant sampled positive-prefix count overflows.

In all cases, we use a crude bound of absolute value at most 1. For a fixed endpoint and fixed queried scale, at most one value of the index can make the corresponding difference of the two threshold indicators nonzero, because the intervals

$$[B_a\sigma + D_a + 1, B_a\sigma + D_{a+1}]$$

are disjoint. The same holds for the second endpoint.

Recall that  $\text{Imp}_{C,r}^a$  denotes the event that endpoint  $(C,r)$  impersonates scale  $a$ . Since  $0 \leq F_{\sigma,\tau}(C) \leq 1$ , every clause contributes at most 1 in absolute value to one fixed queried coordinate. Therefore, for every clause  $C$ ,

$$\begin{aligned} & \left| \mathbb{E}[\widehat{P}_{\ell,j,t,\alpha,\beta}^{(a,b)}(C)] - \mathbf{1}[C \text{ contributes to } P_{\ell,j,t,\alpha,\beta}^{(a,b)}] \right| \\ & \leq \mathbf{1}[\text{Tgt}_C^{\ell,j,t,\alpha,\beta}] \left( \mathbf{1}[\text{Imp}_{C,j}^a] + \mathbf{1}[\text{Imp}_{C,t}^b] + \mathbf{1}[a = a(C,j), s_{v_j(C)}^{\leq C} \geq \kappa] + \mathbf{1}[b = a(C,t), s_{v_t(C)}^{\leq C} \geq \kappa] \right). \end{aligned}$$

Consequently,

$$\begin{aligned} & \left| \mathbb{E}[\widehat{P}_{\ell,j,t,\alpha,\beta}^{(a,b)}] - P_{\ell,j,t,\alpha,\beta}^{(a,b)} \right| \\ & \leq \sum_{C \in \Phi_{\leq 3}} \mathbf{1}[\text{Tgt}_C^{\ell,j,t,\alpha,\beta}] \left( \mathbf{1}[\text{Imp}_{C,j}^a] + \mathbf{1}[\text{Imp}_{C,t}^b] + \mathbf{1}[a = a(C,j), s_{v_j(C)}^{\leq C} \geq \kappa] + \mathbf{1}[b = a(C,t), s_{v_t(C)}^{\leq C} \geq \kappa] \right) \end{aligned} \tag{16}$$

**Why overflow remains.** The coordinate output only queries sampled counts  $0, 1, \dots, \kappa - 1$ . If a true high-degree endpoint has sampled positive-prefix count at least  $\kappa$ , then that clause may still belong to the true coordinate  $P_{\ell,j,t,\alpha,\beta}^{(a,b)}$ , but each queried difference of threshold indicators on that side is zero. The final proof bounds the total loss from these endpoints by the overflow weight  $W_{\text{of}}$ .

### 5.3.2 Other cases

Assume now that at least one of  $D_a$  and  $D_b$  is low-degree, namely  $D_a < \kappa/2$  or  $D_b < \kappa/2$ . We keep the same four sub-sketches, the same register layout, and the same final inclusion-exclusion formula as above. Only the levels, the positive-count updates, and the queried positions change.

For each endpoint we use the exact positive-prefix count on a low-degree scale and the sampled positive-prefix count on a high-degree scale. Accordingly, set

$$\begin{aligned} \Sigma_a & := \begin{cases} \{0, 1, \dots, \kappa - 1\}, & D_a \geq \kappa/2, \\ \{(d_1, c_1) : D_a < d_1 \leq D_{a+1}, 0 \leq c_1 \leq d_1\}, & D_a < \kappa/2, \end{cases} \\ \Sigma_b & := \begin{cases} \{0, 1, \dots, \kappa - 1\}, & D_b \geq \kappa/2, \\ \{(d_2, c_2) : D_b < d_2 \leq D_{b+1}, 0 \leq c_2 \leq d_2\}, & D_b < \kappa/2. \end{cases} \end{aligned}$$

**Update.** For every tagged unit on a variable  $u$ , first apply the degree update  $\text{inc}(Z_1, u, 1)$  and  $\text{inc}(Z_2, u, 1)$ . Then update the positive count on each side separately. On the first side, apply  $\text{inc}(Z_1, u, B_a)$  if the tagged unit is positive and either  $D_a < \kappa/2$ , or  $D_a \geq \kappa/2$  and the unit is selected by  $f_a$ . On the second side, apply  $\text{inc}(Z_2, u, B_b)$  under the same rule with  $b$  and  $f_b$ . Thus, when both sides are low-degree, a positive tagged unit triggers the  $B_a$ -step update on  $Z_1$  and the  $B_b$ -step update on  $Z_2$ .

**Query.** If the current clause  $C$  is not target-shape, do nothing. Otherwise write

$$u := v_j(C), \quad v := v_t(C).$$

If the first endpoint is low-degree and  $\sigma = (d_1, c_1) \in \Sigma_a$ , then the lower and upper queried positions on the first family labels are

$$Z_1(u, B_a c_1 + d_1), \quad Z_1(u, B_a c_1 + d_1 + 1).$$

If the first endpoint is high-degree  $D_a \geq \kappa/2$  and  $\sigma \in \{0, 1, \dots, \kappa - 1\}$ , we query like the previous section:

$$Z_1(u, B_a \sigma + D_a + 1), \quad Z_1(u, B_a \sigma + D_{a+1} + 1).$$

The second endpoint is treated analogously: if  $D_b < \kappa/2$  and  $\tau = (d_2, c_2) \in \Sigma_b$ , use

$$Z_2(v, B_b c_2 + d_2), \quad Z_2(v, B_b c_2 + d_2 + 1),$$

and if  $D_b \geq \kappa/2$  and  $\tau \in \{0, 1, \dots, \kappa - 1\}$ , use

$$Z_2(v, B_b \tau + D_b + 1), \quad Z_2(v, B_b \tau + D_{b+1} + 1).$$

The four sub-sketches again apply pair-query to the lower/lower, lower/upper, upper/lower, and upper/upper choices. If such a pair-query has output  $\pm 1$ , that sub-sketch stops and stores the nonzero sign, the two variables, and the indices  $\sigma, \tau$ ; the suffix degrees and suffix positive counts are then counted exactly as in the high-high case.

**Output.** After a successful pair-query on a clause  $C$ , compute  $F_{\sigma, \tau}(C)$  by using the following candidate prefix values. If  $D_a < \kappa/2$  and  $\sigma = (d_1, c_1) \in \Sigma_a$ , use

$$\hat{d}_u = d_1, \quad \hat{p}_u = c_1.$$

If  $D_a \geq \kappa/2$  and  $\sigma \in \{0, 1, \dots, \kappa - 1\}$ , use

$$\hat{d}_u = D_a, \quad \hat{p}_u = \frac{2D_a}{\kappa} \sigma.$$

If  $D_b < \kappa/2$  and  $\tau = (d_2, c_2) \in \Sigma_b$ , use

$$\hat{d}_v = d_2, \quad \hat{p}_v = c_2.$$

If  $D_b \geq \kappa/2$  and  $\tau \in \{0, 1, \dots, \kappa - 1\}$ , use

$$\hat{d}_v = D_b, \quad \hat{p}_v = \frac{2D_b}{\kappa} \tau.$$

Then set

$$F_{\sigma, \tau}(C) := \mathbf{1} \left[ \begin{array}{l} \text{clip}_{[-1,1]} \left( \frac{2(\hat{p}_u + p_u^{>C})}{\hat{d}_u + d_u^{>C}} - 1 + g(u) \right) \in I_{i_\alpha}, \\ \text{clip}_{[-1,1]} \left( \frac{2(\hat{p}_v + p_v^{>C})}{\hat{d}_v + d_v^{>C}} - 1 + g(v) \right) \in I_{i_\beta} \end{array} \right].$$

The four final outputs are

$$\begin{aligned} Y_{\sigma, \tau}^{\ell\ell}(C) &:= \frac{M}{2} \chi_{\sigma, \tau}^{\ell\ell}(C) F_{\sigma, \tau}(C), & Y_{\sigma, \tau}^{\ell u}(C) &:= \frac{M}{2} \chi_{\sigma, \tau}^{\ell u}(C) F_{\sigma, \tau}(C), \\ Y_{\sigma, \tau}^{u\ell}(C) &:= \frac{M}{2} \chi_{\sigma, \tau}^{u\ell}(C) F_{\sigma, \tau}(C), & Y_{\sigma, \tau}^{uu}(C) &:= \frac{M}{2} \chi_{\sigma, \tau}^{uu}(C) F_{\sigma, \tau}(C). \end{aligned}$$

If the sub-sketch never has a successful pair-query, its output is 0.

We now give some notations that cover all remaining cases.

**Endpoint formulas.** For the analysis of a target-shape clause  $C$ , write

$$u := v_j(C), \text{ and } v := v_t(C).$$

For the first endpoint, define

$$N_a(C) := \begin{cases} \Gamma_a(d_u^{\leq C}, s_u^{\leq C}), & D_a \geq \kappa/2, \\ \Gamma_a(d_u^{\leq C}, p_u^{\leq C}), & D_a < \kappa/2. \end{cases}$$

Define  $N_b(C)$  analogously for the second endpoint, using  $\Gamma_b$ ,  $d_v^{\leq C}$ , and either  $s_v^{\leq C}$  or  $p_v^{\leq C}$  according to whether  $D_b$  is high- or low-degree.

If  $D_a \geq \kappa/2$  and  $\sigma \in \{0, 1, \dots, \kappa - 1\}$ , set

$$L_a(\sigma) = B_a\sigma + D_a + 1, \quad U_a(\sigma) = B_a\sigma + D_{a+1} + 1.$$

Define

$$\Delta_a(C, \sigma) := \mathbf{1}[N_a(C) \geq L_a(\sigma)] - \mathbf{1}[N_a(C) \geq U_a(\sigma)].$$

Whenever the first endpoint does not impersonate scale  $a$ , we have

$$\Delta_a(C, \sigma) = \begin{cases} \mathbf{1}[D_a < d_u^{\leq C} \leq D_{a+1}, s_u^{\leq C} = \sigma], & a = a(C, j) \text{ and } s_u^{\leq C} < \kappa, \\ 0, & a \neq a(C, j) \text{ or } s_u^{\leq C} \geq \kappa. \end{cases}$$

For the zero case with  $a \neq a(C, j)$ , the same check applies: degrees below the queried scale make the two threshold indicators equal, while degrees above the queried scale would force impersonation if the difference of threshold indicators were nonzero.

If  $D_a < \kappa/2$  and  $\sigma = (d_1, c_1)$ , set

$$L_a(\sigma) = B_a c_1 + d_1, \quad U_a(\sigma) = B_a c_1 + d_1 + 1.$$

Whenever the first endpoint does not impersonate scale  $a$ , we have

$$\Delta_a(C, \sigma) = \begin{cases} \mathbf{1}[d_u^{\leq C} = d_1, p_u^{\leq C} = c_1], & a = a(C, j), \\ 0, & a \neq a(C, j). \end{cases}$$

The second endpoint is identical; the functions  $L_b(\tau)$  and  $U_b(\tau)$  are defined by the same formulas with  $b, v, \tau$  in place of  $a, u, \sigma$ .

On  $\mathcal{G}_2$ , Lemma 8 says that each queried label is live exactly when its threshold inequality is true. Thus, for a target-shape clause  $C$ ,

$$Z_1(u, L_a(\sigma)) \in T \iff N_a(C) \geq L_a(\sigma), \quad Z_1(u, U_a(\sigma)) \in T \iff N_a(C) \geq U_a(\sigma),$$

and the same statement holds for  $Z_2$  with  $N_b, L_b, U_b$ . This is the only live-label fact used in the expectation calculation below.

**Estimation.** All the following argument is on the event  $\mathcal{G}_2$  from Lemma 11.

For each clause  $C$ , define

$$\widehat{P}_{\ell, j, t, \alpha, \beta}^{(a, b)}(C) := \sum_{(\sigma, \tau) \in \Sigma_a \times \Sigma_b} \left( Y_{\sigma, \tau}^{\ell \ell}(C) - Y_{\sigma, \tau}^{\ell u}(C) - Y_{\sigma, \tau}^{u \ell}(C) + Y_{\sigma, \tau}^{u u}(C) \right),$$

and then set

$$\widehat{P}_{\ell,j,t,\alpha,\beta}^{(a,b)} := \sum_C \widehat{P}_{\ell,j,t,\alpha,\beta}^{(a,b)}(C).$$

For a clause that is not target-shape, all four outputs are 0, so the bound below is immediate. Now fix a target-shape clause  $C$  and one pair  $(\sigma, \tau) \in \Sigma_a \times \Sigma_b$ , and write

$$u := v_j(C), \quad v := v_t(C).$$

Suppose the live set just before the query in the current sub-sketch has size  $M' - 1$ .

The Lower-Lower- $(\sigma, \tau)$ -Sketch queries the pair

$$(Z_1(u, L_a(\sigma)), Z_2(v, L_b(\tau))).$$

So the pair-query formula gives

$$\mathbb{E}[Y_{\sigma,\tau}^{\ell\ell}(C) \mid |Q(T)\rangle] = \frac{M}{M'} F_{\sigma,\tau}(C) \cdot \mathbf{1}[\text{Tgt}_C^{\ell,j,t,\alpha,\beta}] \cdot \mathbf{1}[N_a(C) \geq L_a(\sigma)] \cdot \mathbf{1}[N_b(C) \geq L_b(\tau)].$$

Similarly,

$$\mathbb{E}[Y_{\sigma,\tau}^{\ell u}(C) \mid |Q(T)\rangle] = \frac{M}{M'} F_{\sigma,\tau}(C) \cdot \mathbf{1}[\text{Tgt}_C^{\ell,j,t,\alpha,\beta}] \cdot \mathbf{1}[N_a(C) \geq L_a(\sigma)] \cdot \mathbf{1}[N_b(C) \geq U_b(\tau)],$$

$$\mathbb{E}[Y_{\sigma,\tau}^{u\ell}(C) \mid |Q(T)\rangle] = \frac{M}{M'} F_{\sigma,\tau}(C) \cdot \mathbf{1}[\text{Tgt}_C^{\ell,j,t,\alpha,\beta}] \cdot \mathbf{1}[N_a(C) \geq U_a(\sigma)] \cdot \mathbf{1}[N_b(C) \geq L_b(\tau)],$$

$$\mathbb{E}[Y_{\sigma,\tau}^{uu}(C) \mid |Q(T)\rangle] = \frac{M}{M'} F_{\sigma,\tau}(C) \cdot \mathbf{1}[\text{Tgt}_C^{\ell,j,t,\alpha,\beta}] \cdot \mathbf{1}[N_a(C) \geq U_a(\sigma)] \cdot \mathbf{1}[N_b(C) \geq U_b(\tau)].$$

By Lemma 10, the probability that this sub-sketch is still active just before the query is  $M'/M$ . Therefore

$$\mathbb{E}\left[Y_{\sigma,\tau}^{\ell\ell}(C) - Y_{\sigma,\tau}^{\ell u}(C) - Y_{\sigma,\tau}^{u\ell}(C) + Y_{\sigma,\tau}^{uu}(C)\right] = F_{\sigma,\tau}(C) \cdot \mathbf{1}[\text{Tgt}_C^{\ell,j,t,\alpha,\beta}] \cdot \Delta_a(C, \sigma) \cdot \Delta_b(C, \tau),$$

Say that the first endpoint is consistent with  $\sigma$  on scale  $a$  if its prefix data agree with the index  $\sigma$  on scale  $a$ : on a high-degree side this means

$$D_a < d_u^{\leq C} \leq D_{a+1}, \quad s_u^{\leq C} = \sigma,$$

and on a low-degree side, if  $\sigma = (d_1, c_1)$ , it means

$$d_u^{\leq C} = d_1, \quad p_u^{\leq C} = c_1.$$

Define consistency with  $\tau$  on the second endpoint in the same way. Assume that neither endpoint impersonates its queried scale. Then the endpoint formulas above identify the two differences of the two threshold indicators exactly. If, in addition, every relevant high-degree sampled positive-prefix count is below  $\kappa$  (that is, if  $a = a(C, j)$  and  $D_a \geq \kappa/2$  then  $s_u^{\leq C} < \kappa$ , and likewise on the second endpoint), we obtain

$$\begin{aligned} & \mathbb{E}\left[Y_{\sigma,\tau}^{\ell\ell}(C) - Y_{\sigma,\tau}^{\ell u}(C) - Y_{\sigma,\tau}^{u\ell}(C) + Y_{\sigma,\tau}^{uu}(C)\right] \\ &= F_{\sigma,\tau}(C) \cdot \mathbf{1}[\text{Tgt}_C^{\ell,j,t,\alpha,\beta}] \cdot \mathbf{1}[\text{the first endpoint is consistent with } \sigma] \\ & \quad \cdot \mathbf{1}[\text{the second endpoint is consistent with } \tau]. \end{aligned}$$

Whenever both endpoints are consistent with their indices, the prefix values substituted in  $F_{\sigma,\tau}(C)$  are the true prefix values from Definition 10. Hence

$$F_{\sigma,\tau}(C) = \mathbf{1}[(\lambda_j(C), \lambda_t(C)) = (\alpha, \beta)]$$

on that event. Summing over all  $(\sigma, \tau) \in \Sigma_a \times \Sigma_b$  gives

$$\mathbb{E}[\widehat{P}_{\ell,j,t,\alpha,\beta}^{(a,b)}(C)] = \mathbf{1}[C \text{ contributes to } P_{\ell,j,t,\alpha,\beta}^{(a,b)}]$$

whenever neither endpoint impersonates the queried scale and no relevant high-degree endpoint overflows.

In all cases, the absolute value of the contribution of this fixed clause to this fixed coordinate is at most 1. For a fixed endpoint and fixed scale, at most one index in  $\Sigma_a$  can make the difference of the two threshold indicators nonzero: in the high-degree case the intervals

$$[B_a\sigma + D_a + 1, B_a\sigma + D_{a+1}]$$

are disjoint, and in the low-degree case the singleton levels  $B_a c_1 + d_1$  are distinct because  $B_a > D_{a+1}$ . The same holds for the second endpoint, with levels  $B_b c_2 + d_2$ . Hence, for every clause  $C$ ,

$$\begin{aligned} & \left| \mathbb{E}[\widehat{P}_{\ell,j,t,\alpha,\beta}^{(a,b)}(C)] - \mathbf{1}[C \text{ contributes to } P_{\ell,j,t,\alpha,\beta}^{(a,b)}] \right| \\ & \leq \mathbf{1}[\text{Tgt}_C^{\ell,j,t,\alpha,\beta}] \cdot \left( \mathbf{1}[\text{Imp}_{C,j}^a] + \mathbf{1}[\text{Imp}_{C,t}^b] + \mathbf{1}[a = a(C, j), D_a \geq \kappa/2, s_{v_j(C)}^{\leq C} \geq \kappa] \right. \\ & \quad \left. + \mathbf{1}[b = a(C, t), D_b \geq \kappa/2, s_{v_t(C)}^{\leq C} \geq \kappa] \right). \end{aligned}$$

Consequently,

$$\begin{aligned} & \left| \mathbb{E}[\widehat{P}_{\ell,j,t,\alpha,\beta}^{(a,b)}] - P_{\ell,j,t,\alpha,\beta}^{(a,b)} \right| \\ & \leq \sum_{C \in \Phi_{\leq 3}} \mathbf{1}[\text{Tgt}_C^{\ell,j,t,\alpha,\beta}] \cdot \left( \mathbf{1}[\text{Imp}_{C,j}^a] + \mathbf{1}[\text{Imp}_{C,t}^b] + \mathbf{1}[a = a(C, j), D_a \geq \kappa/2, s_{v_j(C)}^{\leq C} \geq \kappa] \right. \\ & \quad \left. + \mathbf{1}[b = a(C, t), D_b \geq \kappa/2, s_{v_t(C)}^{\leq C} \geq \kappa] \right). \end{aligned} \quad (17)$$

The first two terms upper-bound possible errors from endpoints counted by the impersonation condition for one of the queried scales. The last two terms record overflow on true high-degree endpoints.

#### 5.4 Estimating the weighted pseudosnapshot score

Run in parallel, during the same pass over the stream, all sketches defined for the one-endpoint and two-endpoint coordinates, together with the exact classical counter for  $M_3$ . Let  $Z$  denote the output of one such full run after applying exactly the same linear combination as  $L_{\leq 3}(\text{WPsSnap}(\Phi_{\leq 3}))$ :

$$\begin{aligned} Z := & \sum_a \sum_{\alpha} q_{\alpha} \widehat{U}_{1,1,\alpha}^{(a)} + \sum_{a,b} \sum_{\alpha,\beta} (1 - (1 - q_{\alpha})(1 - q_{\beta})) \widehat{P}_{2,1,2,\alpha,\beta}^{(a,b)} \\ & + c_0 M_3 + \sum_{j=1}^3 \sum_a \sum_{\alpha} u_j(\alpha) \widehat{U}_{3,j,\alpha}^{(a)} + \sum_{1 \leq j < t \leq 3} \sum_{a,b} \sum_{\alpha,\beta} p_{jt}(\alpha, \beta) \widehat{P}_{3,j,t,\alpha,\beta}^{(a,b)}. \end{aligned}$$

Fix the public randomness used for the hash functions, smoothing noise, and the random bases  $B_a$ , and call it  $\omega$ . All quantities defined from the pseudosnapshot are then deterministic; we

suppress their dependence on  $\omega$ . In the rest of the proof, whenever  $\omega$  is fixed, every expectation and variance is taken only over the outcomes of the quantum measurements.

The constants are fixed in the following order. First, choose the positive integer  $\kappa$  large enough for the sampled-prefix estimates and the overflow bound used below. With this  $\kappa$  fixed, choose  $K_{\text{base}}$  for the impersonation bound, choose  $K_\varepsilon$  for the no-wrap tail bounds, and finally choose  $C_\varepsilon$  in  $M = \lceil C_\varepsilon m \rceil$  so that every update count and queried level fits in the finite register.

Let  $\mathcal{G}_1$  be the good event from Lemma 9, and let  $\mathcal{G}_2$  be the good event from Lemma 11. Define

$$\mathcal{G} := \mathcal{G}_1 \cap \mathcal{G}_2.$$

By the two lemmas and a union bound,

$$\Pr[\mathcal{G}^c] \leq \Pr[\mathcal{G}_1^c] + \Pr[\mathcal{G}_2^c] \leq \frac{1}{160} + \frac{1}{160} = \frac{1}{80}.$$

For every short-clause endpoint  $(C, r)$ , let  $a(C, r)$  be its true scale, and let

$$\mathcal{I}(C, r) := \{s : s \text{ is nonempty, } s \neq a(C, r), (C, r) \text{ impersonates scale } s\}$$

be the set of wrong queried scales for which  $(C, r)$  satisfies the impersonation condition. Also write

$$\mathcal{R}(C, r) := \{a(C, r)\} \cup \mathcal{I}(C, r)$$

for the true scale of  $(C, r)$  together with these wrong queried scales. Recall that  $\text{Imp}_{C,r}^a$  denotes the event that endpoint  $(C, r)$  impersonates scale  $s$ . Define

$$W_{\text{imp}} := \sum_{C \in \Phi_{\leq 3}} \sum_{r=1}^{|C|} |\mathcal{I}(C, r)| = \sum_{C \in \Phi_{\leq 3}} \sum_{r=1}^{|C|} \sum_{\substack{s \text{ nonempty} \\ s \neq a(C, r)}} \mathbf{1}[\text{Imp}_{C,r}^s].$$

Thus  $W_{\text{imp}}$  counts all endpoint-scale pairs that may contribute to a wrong queried scale. It may count too many endpoint-scale pairs, but it does not miss any endpoint-scale pair that gives a nonzero contribution to a wrong queried scale.

Let  $W_{\text{of}}$  be the total tagged weight of short-clause endpoints whose true scale  $a(C, r)$  satisfies  $2D_{a(C,r)} \geq \kappa$  and whose sampled positive-prefix count at that scale is at least  $\kappa$ :

$$W_{\text{of}} := \sum_{C \in \Phi_{\leq 3}} \sum_{r=1}^{|C|} w_{|C|} \mathbf{1}\left[2D_{a(C,r)} \geq \kappa \text{ and } s_{v_r(C)}^{\leq C} \geq \kappa\right].$$

The expectations in the next two auxiliary bounds are over the public randomness before  $\omega$  is fixed. After these bounds, expectations again follow the convention stated at the start of the proof.

**Lemma 12** (Overflow weight bound).

$$\mathbb{E}[W_{\text{of}}] \leq e^{-\Omega(\kappa)} m.$$

*Proof.* Fix one short-clause endpoint  $(C, r)$ , and let  $a = a(C, r)$ . If  $2D_a < \kappa$ , then this endpoint is not counted by  $W_{\text{of}}$ . If  $2D_a \geq \kappa$ , then

$$s_{v_r(C)}^{\leq C} \sim \text{Bin}\left(p_{v_r(C)}^{\leq C}, \frac{\kappa}{2D_a}\right), \quad \mathbb{E}[s_{v_r(C)}^{\leq C}] \leq \frac{\kappa D_{a+1}}{2D_a} \leq 0.6\kappa$$

for sufficiently small fixed  $\varepsilon$ . A Chernoff bound gives  $\Pr[s_{v_r(C)}^{\leq C} \geq \kappa] \leq e^{-\Omega(\kappa)}$ . The sum of the tagged weights of all short-clause endpoints is at most  $4m$ , so linearity of expectation gives the claim.  $\square$

**Lemma 13** (Impersonation-count bound). *There is a constant  $C_\varepsilon^* > 0$  depending only on  $\varepsilon$  such that*

$$\mathbb{E}[W_{\text{imp}}] \leq C_\varepsilon^* \frac{m \log(K_{\text{base}} + 2)}{K_{\text{base}}}.$$

*Proof.* Fix one endpoint  $(C, r)$  with prefix degree  $d$  and true scale  $a(C, r)$ . Fix a wrong nonempty scale  $s \neq a(C, r)$ . If  $(C, r)$  impersonates scale  $s$ , then for some  $1 \leq h \leq H_s$ ,

$$d \in [hB_s + D_s + 1, hB_s + D_{s+1}].$$

Equivalently,

$$B_s \in \left[ \frac{d - D_{s+1}}{h}, \frac{d - D_s - 1}{h} \right].$$

For a fixed  $h$ , this interval contains  $O((D_{s+1} - D_s + 1)/h + 1) = O_\varepsilon(D_{s+1}/h)$  integers, whereas  $B_s$  is uniform on an interval of length  $\Theta(K_{\text{base}}D_{s+1})$ . Hence

$$\Pr[(C, r) \text{ impersonates scale } s \text{ through this } h] = O_\varepsilon\left(\frac{1}{K_{\text{base}}h}\right).$$

Summing over  $1 \leq h \leq H_s$  gives

$$\Pr[\text{Imp}_{C,r}^s] = O_\varepsilon\left(\frac{1}{K_{\text{base}}}\right).$$

The harmonic factor is absorbed into  $O_\varepsilon(\cdot)$ , because  $H_s \leq \kappa = \text{poly}(\varepsilon)$  on high-degree scales and  $H_s = D_{s+1} = O_\varepsilon(1)$  on low-degree scales.

Moreover, impersonating scale  $s$  implies

$$d \leq H_s(2K_{\text{base}}D_{s+1} + 1) + D_{s+1}.$$

Notice that  $D_s$  grows exponentially by  $s$ . If  $s$  is high-degree then  $H_s = \kappa$ , so only the  $O_\varepsilon(\log(K_{\text{base}} + 2))$  scales with

$$D_{s+1} \in \left[ \frac{d}{2K_{\text{base}}\kappa + O_\varepsilon(1)}, d \right)$$

are relevant. If  $s$  is low-degree then  $D_{s+1} < \kappa = O_\varepsilon(1)$ , so there are only  $O_\varepsilon(1)$  such scales. Therefore

$$\mathbb{E}[|\mathcal{I}(C, r)|] = \sum_{\substack{s \text{ nonempty} \\ s \neq a(C, r)}} \Pr[\text{Imp}_{C,r}^s] = O_\varepsilon\left(\frac{\log(K_{\text{base}} + 2)}{K_{\text{base}}}\right).$$

Since there are at most  $O(m)$  short-clause endpoints,

$$\mathbb{E}[W_{\text{imp}}] \leq C_\varepsilon^* \frac{m \log(K_{\text{base}} + 2)}{K_{\text{base}}}$$

for some constant  $C_\varepsilon^*$  depending only on  $\varepsilon$ . □

The same geometric argument also bounds the number of scales in  $\mathcal{R}(C, r)$ : there is a constant  $C_\varepsilon^{\text{sc}} > 0$  such that

$$|\mathcal{R}(C, r)| \leq C_\varepsilon^{\text{sc}} \log(K_{\text{base}} + 2) \quad \text{for every endpoint } (C, r). \quad (18)$$

Indeed, applying the same argument to a fixed endpoint  $(C, r)$ , with  $d = d_{v_r(C)}^{\leq C}$ , shows that low-degree wrong scales contribute only  $O_\varepsilon(1)$  possibilities, while the bound above leaves only  $O_\varepsilon(\log(K_{\text{base}} + 2))$  possible high-degree scales.

On  $\mathcal{G}$ , Lemma 8 says that each queried label is live exactly when its threshold inequality is true. The bounds for fixed coordinates proved in (13), (15), (16), and (17) give the following facts:

1. every high-degree one-endpoint coordinate output is unbiased for its true coordinate;
2. a low-degree one-endpoint coordinate output can differ from its target only when the queried endpoint satisfies the impersonation condition for the queried scale;
3. a two-endpoint coordinate output can differ from its target only when an endpoint satisfies the impersonation condition for one of the queried scales, or when a true high-degree endpoint overflows.

We now combine these fixed-coordinate bounds into a bound for the full linear form. Recall that  $\text{Tgt}_C^{\ell,j,\alpha}$  denotes the event that  $C$  is the target-type for  $U_{\ell,j,\alpha}^{(a)}$ ;  $\text{Tgt}_C^{\ell,j,t,\alpha,\beta}$  denotes the event that  $C$  is the target-type for  $P_{\ell,j,t,\alpha,\beta}^{(a,b)}$ .

**One-endpoint contribution.** The high-degree one-endpoint coordinates are unbiased, and the counter  $M_3$  is exact on  $\mathcal{G}$ . Therefore only the low-degree one-endpoint coordinates can contribute here. (15) gives, for every fixed low-degree coordinate  $(\ell, j, \alpha, a)$ ,

$$\left| \mathbb{E} \left[ \widehat{U}_{\ell,j,\alpha}^{(a)} \right] - U_{\ell,j,\alpha}^{(a)} \right| \leq \sum_{C \in \Phi_{\leq 3}} \mathbf{1} \left[ \text{Tgt}_C^{\ell,j,\alpha} \right] \cdot \mathbf{1} \left[ \text{Imp}_{C,j}^a \right].$$

Fix an event that  $(C, r)$  impersonates scale  $a$ , where  $a$  is a wrong queried scale in  $\mathcal{I}(C, r)$ . Once the clause length, the endpoint position, and the sign of that literal are fixed, only a constant number of one-endpoint coordinates can use this event: the bucket index ranges over a constant-size set, and the coefficients  $q_\alpha$  and  $u_j(\alpha)$  in (4) are fixed constants. Therefore there is a constant  $C_\varepsilon^{\text{one}} > 0$  such that the total weighted one-endpoint error is at most  $C_\varepsilon^{\text{one}} W_{\text{imp}}$ .

**Two-endpoint error from wrong scale pairs.** (16) and (17) give, for every fixed coordinate  $(\ell, j, t, \alpha, \beta, a, b)$ ,

$$\begin{aligned} & \left| \mathbb{E} \left[ \widehat{P}_{\ell,j,t,\alpha,\beta}^{(a,b)} \right] - P_{\ell,j,t,\alpha,\beta}^{(a,b)} \right| \\ & \leq \sum_{C \in \Phi_{\leq 3}} \mathbf{1} \left[ \text{Tgt}_C^{\ell,j,t,\alpha,\beta} \right] \left( \mathbf{1} \left[ \text{Imp}_{C,j}^a \right] + \mathbf{1} \left[ \text{Imp}_{C,t}^b \right] + \mathbf{1} \left[ a = a(C, j), D_a \geq \kappa/2, s_{v_j(C)}^{\leq C} \geq \kappa \right] \right. \\ & \quad \left. + \mathbf{1} \left[ b = a(C, t), D_b \geq \kappa/2, s_{v_t(C)}^{\leq C} \geq \kappa \right] \right). \end{aligned}$$

We first sum the part caused by endpoints counted by the impersonation condition for a scale that is not their true scale.

Fix a clause  $C$  and one ordered pair of relevant endpoints  $(C, j)$  and  $(C, t)$  with  $1 \leq j < t \leq |C|$ . A queried scale pair  $(a, b)$  can receive a nonzero term from this clause even though  $(a, b)$  is not the true scale pair only in one of the following two ways:

- the first endpoint impersonates the queried first scale, so  $a \in \mathcal{I}(C, j)$ , while the second queried scale must still belong to  $\mathcal{R}(C, t)$ ;
- the second endpoint impersonates the queried second scale, so  $b \in \mathcal{I}(C, t)$ , while the first queried scale must belong to  $\mathcal{R}(C, j)$ .

Hence the total number of queried scale pairs that can receive such an extra nonzero term from this ordered pair is at most

$$|\mathcal{I}(C, j)| |\mathcal{R}(C, t)| + |\mathcal{I}(C, t)| |\mathcal{R}(C, j)|.$$

Using (18), this is at most

$$C_\varepsilon^{\text{sc}} \log(K_{\text{base}} + 2) (|\mathcal{I}(C, j)| + |\mathcal{I}(C, t)|).$$

The clause length, the ordered pair  $(j, t)$ , and the bucket indices  $(\alpha, \beta)$  all range over constant-size sets, and the coefficients in (4) are fixed constants. Therefore there is a constant  $C_\varepsilon^{\text{two}} > 0$  such that the total weighted wrong-scale error in all two-endpoint coordinates is at most

$$C_\varepsilon^{\text{two}} \log(K_{\text{base}} + 2) W_{\text{imp}}.$$

**Overflow contribution.** The weight  $W_{\text{of}}$  counts the high-degree endpoints whose sampled positive-prefix count is at least  $\kappa$ . Overflow does not affect one-endpoint coordinates. For two-endpoint coordinates, an overflowing endpoint matters only on its true queried scale, not on wrong scales. Therefore each overflowing endpoint can affect only the constantly many true two-endpoint coordinates that use that endpoint inside its clause. Since all coefficients in (4) are fixed constants, there is a constant  $C_{\text{of}} > 0$  such that the total weighted overflow error is at most  $C_{\text{of}} W_{\text{of}}$ .

Combining the one-endpoint, two-endpoint, and overflow bounds, there is a constant  $C_{\text{imp}} = O_\varepsilon(\log(K_{\text{base}} + 2))$  such that for every  $\omega \in \mathcal{G}$ ,

$$|\mathbb{E}[Z \mid \omega] - L_{\leq 3}(\text{WPsSnap}(\Phi_{\leq 3}))| \leq C_{\text{imp}} W_{\text{imp}} + C_{\text{of}} W_{\text{of}}. \quad (19)$$

Choose  $K_{\text{base}}$  large enough so that the  $O_\varepsilon(\log^2(K_{\text{base}} + 2)/K_{\text{base}})$  bound from Lemma 13 gives

$$C_{\text{imp}} \mathbb{E}[W_{\text{imp}}] \leq \frac{\varepsilon m}{320}, \quad \text{hence} \quad \mathbb{E}[W_{\text{imp}}] \leq \frac{\varepsilon m}{320 C_{\text{imp}}}.$$

The initial choice of  $\kappa = \text{poly}(1/\varepsilon)$  also ensures

$$\mathbb{E}[W_{\text{of}}] \leq \frac{\varepsilon m}{320 C_{\text{of}}};$$

this follows from Lemma 12.

Now define

$$\mathcal{S} := \left\{ W_{\text{imp}} \leq \frac{\varepsilon m}{4 C_{\text{imp}}} \right\}, \quad \text{and} \quad \mathcal{O} := \left\{ W_{\text{of}} \leq \frac{\varepsilon m}{4 C_{\text{of}}} \right\}.$$

By Markov's inequality,

$$\Pr[\mathcal{S}^c] \leq \frac{1}{80}, \quad \text{and} \quad \Pr[\mathcal{O}^c] \leq \frac{1}{80}.$$

On  $\mathcal{S} \cap \mathcal{O}$ , (19) gives

$$|\mathbb{E}[Z \mid \omega] - L_{\leq 3}(\text{WPsSnap}(\Phi_{\leq 3}))| \leq \frac{\varepsilon m}{2} \quad \text{for every } \omega \in \mathcal{G} \cap \mathcal{S} \cap \mathcal{O}. \quad (20)$$

We now bound the variance of one full run, still with the public randomness fixed. There are  $A = O(\log(n + m))$  scales. The number of one-endpoint coordinates is  $O(\log(n + m))$ , and the number of two-endpoint coordinates is  $O(\log^2(n + m))$ , because the clause lengths, endpoint positions, signs, and bucket indices all range over constant-size sets.

For a fixed coordinate, the number of elementary sub-sketches is  $O_\varepsilon(1)$ : for one-endpoint coordinates it is  $2\kappa + 2$ , and for two-endpoint coordinates it is  $4|\Sigma_a||\Sigma_b| = O_\varepsilon(1)$ . Every elementary sub-sketch outputs either 0 or a number whose absolute value is at most  $M = O_\varepsilon(m)$ . Hence the output for one fixed coordinate has variance  $O_\varepsilon(m^2)$ . Since different coordinates use independent work registers,

$$\text{Var}(Z \mid \omega) = O_\varepsilon(m^2 \log^2(n + m)) \quad \text{for every } \omega \in \mathcal{G}. \quad (21)$$

Take  $R = C_\varepsilon^{\text{rep}} \log^2(n + m)$  independent full runs, in parallel during the same pass, with the same public randomness  $\omega$ , and average their outputs:

$$\bar{Z} := \frac{1}{R} \sum_{r=1}^R Z_r.$$

Then, for every  $\omega \in \mathcal{G}$ ,

$$\mathbb{E}[\bar{Z} \mid \omega] = \mathbb{E}[Z \mid \omega], \text{ and } \text{Var}(\bar{Z} \mid \omega) = O_\varepsilon(m^2).$$

Choosing  $C_\varepsilon^{\text{rep}}$  large enough, Chebyshev's inequality gives

$$\Pr \left[ \left| \bar{Z} - \mathbb{E}[Z \mid \omega] \right| > \frac{\varepsilon m}{4} \mid \omega \right] \leq \frac{1}{40} \quad \text{for every } \omega \in \mathcal{G}.$$

Hence on the event  $\mathcal{G} \cap \mathcal{S} \cap \mathcal{O} \cap \{|\bar{Z} - \mathbb{E}[Z \mid \omega]| \leq \varepsilon m/4\}$ , the triangle inequality and (20) give

$$\left| \bar{Z} - L_{\leq 3}(\text{WPsSnap}(\Phi_{\leq 3})) \right| \leq \frac{3\varepsilon m}{4} < \varepsilon m.$$

Also, as shown above,

$$\Pr[\mathcal{G}^c] \leq \frac{1}{80}.$$

Even on  $\mathcal{G}^c$ , the algorithm still runs legally. What fails there is only the estimation guarantee stated above. Therefore

$$\begin{aligned} & \Pr \left[ \left| \bar{Z} - L_{\leq 3}(\text{WPsSnap}(\Phi_{\leq 3})) \right| > \varepsilon m \right] \\ & \leq \Pr[\mathcal{G}^c] + \Pr[\mathcal{S}^c] + \Pr[\mathcal{O}^c] + \mathbb{E}_\omega \left[ \mathbf{1}[\omega \in \mathcal{G} \cap \mathcal{S} \cap \mathcal{O}] \Pr \left[ \left| \bar{Z} - \mathbb{E}[Z \mid \omega] \right| > \frac{\varepsilon m}{4} \mid \omega \right] \right]. \end{aligned}$$

Using the four bounds above,

$$\Pr \left[ \left| \bar{Z} - L_{\leq 3}(\text{WPsSnap}(\Phi_{\leq 3})) \right| > \varepsilon m \right] \leq \frac{1}{80} + \frac{1}{80} + \frac{1}{80} + \frac{1}{40} = \frac{1}{16}.$$

So the averaged output  $\hat{L}_{\leq 3} := \bar{Z}$  succeeds with probability at least 15/16.

**Space complexity.** One sub-sketch uses a quantum register of size  $O(\log(n + m))$  qubits and, after a successful query, only  $O(\log(n + m))$  additional classical bits for storing the variable name, the relevant indices, and the suffix total/positive degree. One full run contains  $O_\varepsilon(\log^2(n + m))$  elementary sub-sketches, so one full run uses  $O_\varepsilon(\log^3(n + m))$  qubits and the same order of classical working bits. The averaging factor  $R = \Theta_\varepsilon(\log^2(n + m))$  multiplies this by another factor  $\Theta_\varepsilon(\log^2(n + m))$ . Therefore, the overall algorithm uses  $O_\varepsilon(\log^5(n + m))$  qubits, and the classical working space is of the same asymptotic order. Public random coins are shared and are not charged to the working space, as in the rest of the paper. This proves Lemma 6.

## 6 Quantum algorithm for Max- $k$ SAT

We now prove the main theorem.

*Proof of Theorem 1.* Let  $\rho = 0.717275$ . Since the theorem only needs a constant 0.7172, we fix  $\varepsilon > 0$  small enough to absorb all  $O(\varepsilon)$  losses below.

Let  $\Phi_{\text{orig}}$  be the input instance. As the stream arrives, the algorithm applies the online simplifications from Subsection 3: it removes repeated literals, discards tautological clauses while counting them in an exact counter  $T$ , and keeps an exact counter  $m_{\geq 4}$  for simplified clauses of length at least four. These counters use  $O(\log m) = O(\log n)$  classical bits, and we let  $\Phi$  denote the remaining nonconstant instance.

We first describe one constant-success trial. The trial samples fresh public randomness for the hash functions and smoothing noise used in the pseudosnapshot construction. It then runs the quantum estimator of Lemma 6 with a fixed constant failure probability, say  $1/16$ , obtaining an estimate  $\widehat{L}_{\leq 3}$ . The trial outputs

$$Y := \max \left\{ 0, \widehat{L}_{\leq 3} + \rho m_{\geq 4} - C\varepsilon m \right\},$$

where  $C$  is a sufficiently large constant depending only on the fixed certificate and on the constants in Lemma 5.

Call a trial valid if both Lemma 5 holds and the estimator error is at most  $\varepsilon m$ . By choosing  $\kappa = \text{poly}(1/\varepsilon)$  large enough, the first event holds with probability at least  $15/16$ , and Lemma 6 gives the second event with probability at least  $15/16$ . Thus, a trial is valid with probability at least  $9/10$ .

On every valid trial, the estimator error is absorbed by the shift  $C\varepsilon m$ . Lemma 5 gives

$$\rho \text{OPT}(\Phi) - O(\varepsilon m) \leq Y \leq \text{OPT}(\Phi).$$

Every remaining nonconstant clause is satisfied by a uniformly random assignment with probability at least  $1/2$ , so  $\text{OPT}(\Phi) \geq m/2$ . Thus the additive loss is an  $O(\varepsilon)$   $\text{OPT}(\Phi)$  loss, and our choice of  $\varepsilon$  gives

$$0.7172 \text{OPT}(\Phi) \leq Y \leq \text{OPT}(\Phi)$$

on every valid trial.

To boost the success probability, run  $R_{\text{out}} = \Theta(\log \frac{1}{\delta})$  independent trials in parallel during the same pass, using independent public seeds and independent quantum estimator randomness. Return the median of their outputs, denoted  $Z_0$ . By a Chernoff bound, with probability at least  $1 - \delta$ , a strict majority of the trials are valid. In that case, the median also lies in the interval

$$0.7172 \text{OPT}(\Phi) \leq Z_0 \leq \text{OPT}(\Phi).$$

Finally, return  $Z := T + Z_0$ .

Since the removed tautological clauses contribute exactly  $T$  to every assignment, adding  $T$  preserves both inequalities:

$$0.7172 \text{OPT}(\Phi_{\text{orig}}) \leq Z \leq \text{OPT}(\Phi_{\text{orig}}).$$

One trial uses  $O(\log^5 n)$  qubits because Lemma 6 is invoked with constant failure probability. The outer amplification runs  $R_{\text{out}}$  trials in parallel, so the total space is  $O(\log^5 n \log \frac{1}{\delta})$  qubits, plus  $O(\log n)$  classical bits for the exact counters. Public random coins are shared and are not charged to the working space.  $\square$

## References

- [AMS99] Noga Alon, Yossi Matias, and Mario Szegedy. The space complexity of approximating the frequency moments. *Journal of Computer and System Sciences*, 58(1):137–147, 1999.
- [Ass23] Sepehr Assadi. Recent advances in multi-pass graph streaming lower bounds. *ACM SIGACT News*, 54(3):48–75, 2023.
- [BBD<sup>+</sup>02] Brian Babcock, Shivnath Babu, Mayur Datar, Rajeev Motwani, and Jennifer Widom. Models and issues in data stream systems. In *Proceedings of the Twenty-First ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 1–16, 2002.
- [BMN<sup>+</sup>21] Ryan Babbush, Jarrod R. McClean, Michael Newman, Craig Gidney, Sergio Boixo, and Hartmut Neven. Focus beyond quadratic speedups for error-corrected quantum advantage. *PRX Quantum*, 2(1):010103, 2021.
- [BV97] Ethan Bernstein and Umesh Vazirani. Quantum complexity theory. *SIAM Journal on Computing*, 26(5):1411–1473, 1997.
- [BYKS02] Ziv Bar-Yossef, Ravi Kumar, and D. Sivakumar. Reductions in streaming algorithms, with an application to counting triangles in graphs. In *Proceedings of the Thirteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 623–632, 2002.
- [CGV20] Chi-Ning Chou, Alexander Golovnev, and Santhoshini Velusamy. Optimal streaming approximations for all boolean Max-2CSPs and Max- $k$ SAT. In *Proceedings of the 61st IEEE Symposium on Foundations of Computer Science*, pages 330–341, 2020.
- [FGG14] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm, 2014.
- [FM85] Philippe Flajolet and G. Nigel Martin. Probabilistic counting algorithms for data base applications. *Journal of Computer and System Sciences*, 31(2):182–209, 1985.
- [FMMC12] Austin G. Fowler, Matteo Mariantoni, John M. Martinis, and Andrew N. Cleland. Surface codes: Towards practical large-scale quantum computation. *Physical Review A*, 86(3):032324, 2012.
- [Gal06] François Le Gall. Exponential separation of quantum and classical online space complexity. In *Proceedings of the Eighteenth Annual ACM Symposium on Parallelism in Algorithms and Architectures*, pages 67–73, 2006.
- [GKK<sup>+</sup>08] Dmitry Gavinsky, Julia Kempe, Iordanis Kerenidis, Ran Raz, and Ronald de Wolf. Exponential separation for one-way quantum communication complexity, with applications to cryptography. *SIAM Journal on Computing*, 38(5):1695–1708, 2008.
- [JN14] Rahul Jain and Ashwin Nayak. The space complexity of recognizing well-parenthesized expressions in the streaming model: The index function revisited. *IEEE Transactions on Information Theory*, 60(10):6646–6668, 2014.

- [Kal22] John Kallaugher. A quantum advantage for a natural streaming problem. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science*, pages 897–908, 2022.
- [KP22] John Kallaugher and Ojas Parekh. The quantum and classical streaming complexity of quantum and classical Max-Cut. In *Proceedings of the 63rd IEEE Symposium on Foundations of Computer Science*, pages 498–506, 2022.
- [KPV24] John Kallaugher, Ojas Parekh, and Nadezhda Voronova. Exponential quantum space advantage for approximating maximum directed cut in the streaming model. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, pages 1805–1815. Association for Computing Machinery, 2024.
- [McG14] Andrew McGregor. Graph stream algorithms: A survey. *SIGMOD Record*, 43(1):9–20, 2014.
- [Mut05] S. Muthukrishnan. Data streams: Algorithms and applications. *Foundations and Trends in Theoretical Computer Science*, 1(2):117–236, 2005.
- [NT17] Ashwin Nayak and Dave Touchette. Augmented index and quantum streaming algorithms for DYCK(2). In *32nd Computational Complexity Conference*, volume 79 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 23:1–23:21. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017.
- [Sho99] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Review*, 41(2):303–332, 1999.
- [Sin25] Noah G. Singer. Nine lower bound conjectures on streaming approximation algorithms for CSPs. *ACM SIGACT News*, 56(4):25–36, 2025.
- [SSSV23] Raghuvansh R Saxena, Noah G Singer, Madhu Sudan, and Santhoshini Velusamy. Improved streaming algorithms for maximum directed cut via smoothed snapshots. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 855–870. IEEE, 2023.
- [Sud22] Madhu Sudan. Streaming and sketching complexity of CSPs: A survey. In *Proceedings of the 49th International Colloquium on Automata, Languages, and Programming*, pages 5:1–5:20, 2022.
- [Wat18] John Watrous. *The Theory of Quantum Information*. Cambridge University Press, 2018.

## A Quantum space lower bound for Max- $k$ SAT

We use the following promise-gap form of the quantum streaming lower bound for Max-Cut of Kallaugher and Parekh [KP22].

**Theorem 3.** [KP22] *For every constant  $\varepsilon > 0$ , any one-pass quantum streaming algorithm that, on an  $n$ -vertex graph with  $m_E$  edges, distinguishes*

$$\text{MaxCut}(G) = m_E \quad \text{from} \quad \text{MaxCut}(G) \leq (1/2 + \varepsilon)m_E$$

*with constant advantage requires  $\Omega(n)$  qubits of space.*

*Proof of Theorem 2.* We reduce from this Max-Cut promise problem. For a graph  $G = (V, E)$  and an assignment  $\sigma : V \rightarrow \{0, 1\}$ , define

$$\text{Cut}_G(\sigma) := |\{\{u, v\} \in E : \sigma(u) \neq \sigma(v)\}|, \quad \text{MaxCut}(G) := \max_{\sigma} \text{Cut}_G(\sigma).$$

Let  $m_E := |E|$ . Given the edge stream of  $G$ , we construct a Max- $k$ -SAT stream  $\Phi_G$  online by replacing each edge  $\{u, v\}$  with the two clauses

$$(x_u \vee x_v), \quad (\neg x_u \vee \neg x_v).$$

Since  $k \geq 2$ , these are valid Max- $k$ -SAT clauses. The reduction is one-pass and online, and uses only  $O(\log(n + m_E))$  additional classical space. The resulting instance has  $M = 2m_E$  clauses. In what follows,  $m_E$  always denotes the number of Max-Cut edges, not the number of Max- $k$ SAT clauses.

For every assignment  $\sigma$ , each edge contributes  $1 + \mathbf{1}[\sigma(u) \neq \sigma(v)]$  satisfied clauses: if  $\sigma(u) \neq \sigma(v)$ , both clauses are satisfied, whereas if  $\sigma(u) = \sigma(v)$ , exactly one is satisfied. Summing over all edges gives

$$\text{Val}_{\Phi_G}(\sigma) = m_E + \text{Cut}_G(\sigma).$$

Taking the maximum over  $\sigma$ , we obtain  $\text{OPT}(\Phi_G) = m_E + \text{MaxCut}(G)$ .

Now suppose, toward contradiction, that there is a one-pass quantum streaming algorithm  $\mathcal{A}$  using  $o(n)$  qubits and achieving an approximation ratio  $3/4 + \gamma$  for Max- $k$ -SAT. Choose  $\varepsilon < 2\gamma$ , and run  $\mathcal{A}$  on the stream  $\Phi_G$  produced above.

If  $\text{MaxCut}(G) = m_E$ , then

$$\text{OPT}(\Phi_G) = 2m_E = M,$$

so a successful run of  $\mathcal{A}$  outputs

$$Z \geq \left(\frac{3}{4} + \gamma\right) 2m_E = \left(\frac{3}{2} + 2\gamma\right) m_E.$$

If instead  $\text{MaxCut}(G) \leq (1/2 + \varepsilon)m_E$ , then

$$\text{OPT}(\Phi_G) \leq m_E + \left(\frac{1}{2} + \varepsilon\right) m_E = \left(\frac{3}{2} + \varepsilon\right) m_E,$$

and every valid approximation output satisfies  $Z \leq \text{OPT}(\Phi_G)$ .

Because  $\varepsilon < 2\gamma$ , the two output ranges are separated by a threshold. Hence  $\mathcal{A}$  would yield an  $o(n)$ -space quantum streaming algorithm distinguishing the corresponding Max-Cut gap instances. This contradicts Theorem 3. Therefore, any one-pass quantum streaming algorithm beating  $3/4$  for Max- $k$ SAT requires  $\Omega(n)$  qubits of space.  $\square$

## B Exact finite LP certificate for Max- $k$ SAT

The proof of Lemma 1 uses a finite rational certificate. The complete certificate data and verifier are available in the repository: <https://github.com/Guangxu-Yang/maxksat-lp-certificate>.

The verifier contains the exact rational data

$$\mathcal{I}, \quad r, \quad c_0, \quad \{u_j\}_{j=1}^3, \quad \{p_{jt}\}_{1 \leq j < t \leq 3}, \quad \{\lambda_{i,\tau}^{\text{lo}}, \lambda_{i,\tau}^{\text{up}}\}_{i,\tau}.$$

All arithmetic in the verifier is performed over  $\mathbb{Q}$ , using rational integer numerators and denominators. Floating-point numbers are printed only for readability. The bucket partition and rounding vector used by the certificate are

$i$	$I_i$	$r_i$
0	$[-1, -3/4)$	0.27083
1	$[-3/4, -1/2)$	0.27083
2	$[-1/2, -1/4)$	0.27083
3	$[-1/4, 0)$	0.4069
4	$[0, 1/4)$	0.5931
5	$[1/4, 1/2)$	0.72917
6	$[1/2, 3/4)$	0.72917
7	$[3/4, 1]$	0.72917.

Thus  $L = 8$ , and the final bucket is closed at 1. As in Definition 3, a positive literal in bucket  $I_i$  is satisfied by the independent rounding with probability  $r_i$ , and a negative literal in bucket  $I_i$  is satisfied with probability  $1 - r_i$ . We use codes to check all inequalities exactly over  $\mathbb{Q}$ . We can get  $\rho = 0.717275$ .

## C Finite LP certificate for approximation ratio of Max-2OR

We only focus on the step that reduces Max-2OR to the problem of estimating a snapshot. This appendix improves only the approximation constant for the Max-2OR; the streaming and quantum estimation arguments are inherited from the Max- $k$ SAT framework.

The optimization is instead over a Max-2SAT-specific rounding certificate: the unary endpoint coefficient, the nonuniform bias partition, and the finite list of nine rounding profiles are chosen so that the best profile certifiably beats every feasible mixture of unary and binary typed atoms.

We consider simplified unweighted Max-2SAT instances, meaning finite multisets of unary clauses ( $x_v = s$ ) and binary clauses  $(x_u = s) \vee (x_v = t)$  with  $s, t \in \{+, -\}$ , after replacing repeated-literal binary clauses by the corresponding unary clause and removing tautologies. Let  $|\Phi|$  denote the number of remaining clauses.

Before defining the "snapshot" of instances, we first define the bias of the variables in  $\Phi$ . The certificate uses the unary endpoint coefficient  $w_u := \frac{67}{20}$ ; this is part of the bias definition, not an input clause weight.

**Definition 13.** For each variable  $v$ , let  $P_v^{(1)}$ ,  $N_v^{(1)}$  and  $P_v^{(2)}$ ,  $N_v^{(2)}$  be its positive/negative unary occurrence counts and positive/negative binary-endpoint counts. Define

$$p_v := w_u P_v^{(1)} + P_v^{(2)} \quad q_v := w_u N_v^{(1)} + N_v^{(2)} \quad d_v := p_v + q_v,$$

and the signed bias

$$b_v := \begin{cases} \frac{p_v - q_v}{d_v}, & d_v > 0, \\ 0, & d_v = 0. \end{cases}$$

Fix the nonuniform partition  $-1 = t_0 < t_1 < \dots < t_{450} = 1$  from the artifact file <sup>3</sup>. Let  $I_i = [t_i, t_{i+1})$  ( $0 \leq i \leq 448$ ) and  $I_{449} = [t_{449}, t_{450}]$ .

**Definition 14.** For  $s, t \in \{+, -\}$  and  $0 \leq i, j \leq 449$ , define the unary and binary snapshot coordinates

$$U_{s,i}(\Phi) := \sum_{C=(x_v=s) \in \Phi} \mathbf{1}[b_v \in I_i],$$

$$S_{s,t,i,j}(\Phi) := \sum_{C=(x_u=s) \vee (x_v=t) \in \Phi} \mathbf{1}[b_u \in I_i] \mathbf{1}[b_v \in I_j].$$

Define the snapshot as

$$\text{WSnap}_t(\Phi) := \left( U_{s,i}(\Phi), S_{s,t,i,j}(\Phi) \right)_{s,t \in \{+,-\}, 0 \leq i,j \leq 449}$$

Now, we can define the snapshot score.

**Definition 15** (Snapshot score). The certificate evaluates nine fixed rounding profiles

$$p^{(1)}, \dots, p^{(9)} \in [0, 1]^{450}$$

from the codes <sup>4</sup>. For each rule  $r$ , let  $q_+^{(r)}(i) := p_i^{(r)}$  and  $q_-^{(r)}(i) := 1 - p_i^{(r)}$ , define the rule- $r$  snapshot score by

$$A_r(\text{WSnap}_t(\Phi)) := \sum_{s \in \{+,-\}} \sum_{i=0}^{449} q_s^{(r)}(i) U_{s,i}(\Phi) + \sum_{s,t \in \{+,-\}} \sum_{i,j=0}^{449} \left( 1 - (1 - q_s^{(r)}(i))(1 - q_t^{(r)}(j)) \right) S_{s,t,i,j}(\Phi),$$

and let

$$A_{\max}(\text{WSnap}_t(\Phi)) := \max_{1 \leq r \leq 9} A_r(\text{WSnap}_t(\Phi)).$$

We use nine profiles rather than one because no single oblivious rounding curve is best for all mixtures of unary and binary clauses. The profiles are fixed in advance by the certificate. Each  $A_r$  is the expected number of satisfied clauses under the corresponding independent rounding rule, expressed as a linear function of the snapshot;  $A_{\max}$  then chooses the best of these constant many certified rules.

**Lemma 14.** *For every nonempty simplified unweighted Max-2SAT instance  $\Phi$ ,*

$$\rho_2 \text{OPT}(\Phi) \leq A_{\max}(\text{WSnap}_t(\Phi)) \leq \text{OPT}(\Phi),$$

where  $\rho_2 = 0.74252$ .

*Proof.* The upper bound is immediate: for every profile  $r$ ,  $A_r(\text{WSnap}_t(\Phi))$  is the expected number of satisfied clauses in  $\Phi$  under the corresponding independent rounding rule, and is therefore at most  $\text{OPT}(\Phi)$ . Hence  $A_{\max}(\text{WSnap}_t(\Phi)) \leq \text{OPT}(\Phi)$ .

Fix an optimal assignment  $x^*$ . Refine every unary or binary clause according to its signs, bucket labels, and endpoint bits under  $x^*$ . Thus, a typed atom is either

$$c = (\text{U}, s, i, \tau) \quad \text{or} \quad c = (\text{S}, s, t, i, j, \tau, v),$$

<sup>3</sup>artifacts/max2sat/nonuniform\_L450\_focus0\_str36\_sig012\_plus009\_s6\_w335.json in the codes

<sup>4</sup>The file artifacts/max2sat/multirounding\_L450\_9\_rounding\_curves.json

where  $s, t \in \{+, -\}$ ,  $i, j \in \{0, \dots, 449\}$ , and  $\tau, v \in \{0, 1\}$ . Let  $x_c$  be the number of clauses of type  $c$ .

Write  $\chi(+)=+1$ ,  $\chi(-)=-1$ , and

$$\chi_x(\tau) = \begin{cases} +1, & \tau = 1, \\ -1, & \tau = 0. \end{cases}$$

Define the optimum contribution of an atom by

$$o(\mathbf{U}, s, i, \tau) = \mathbf{1}[\chi(s) = \chi_x(\tau)]$$

and

$$o(\mathbf{S}, s, t, i, j, \tau, v) = 1 - \mathbf{1}[\chi(s) \neq \chi_x(\tau)]\mathbf{1}[\chi(t) \neq \chi_x(v)].$$

For rounding profile  $r$ , define its atom contribution  $R_r(c)$  by

$$R_r(\mathbf{U}, s, i, \tau) = q_s^{(r)}(i)$$

and

$$R_r(\mathbf{S}, s, t, i, j, \tau, v) = 1 - (1 - q_s^{(r)}(i))(1 - q_t^{(r)}(j)).$$

Then, by construction,

$$\sum_c x_c o(c) = \text{OPT}(\Phi), \quad \sum_c x_c R_r(c) = A_r(\text{WSnap}_t(\Phi)).$$

We also need the bucket feasibility constraints. For each bucket  $a$  and bit  $\theta$ , let  $D_{a,\theta}(c)$  and  $B_{a,\theta}(c)$  be the total bias-weighted endpoint count and signed bias-weighted endpoint count contributed by atom  $c$  to bucket  $a$  and bit class  $\theta$ . For unary atoms, this contribution uses the coefficient  $w_u/2$ , and for binary endpoints, the coefficient  $1/2$ . If  $I_a = [\ell_a, u_a]$ , set

$$g_{a,\theta}^{\text{lo}}(c) := \ell_a D_{a,\theta}(c) - B_{a,\theta}(c), \quad g_{a,\theta}^{\text{up}}(c) := B_{a,\theta}(c) - u_a D_{a,\theta}(c).$$

For the real instance, after grouping variables with  $b_v \in I_a$  and  $x_v^* = \theta$ , the sums

$$D_{a,\theta} := \sum_c x_c D_{a,\theta}(c), \quad B_{a,\theta} := \sum_c x_c B_{a,\theta}(c)$$

are common scalar multiples of

$$\sum_{v: b_v \in I_a, x_v^* = \theta} d_v \quad \text{and} \quad \sum_{v: b_v \in I_a, x_v^* = \theta} d_v b_v.$$

The common scalar is  $1/2$ , since unary endpoints carry coefficient  $w_u/2$  and binary endpoints carry coefficient  $1/2$  in the atom bookkeeping. Hence  $B_{a,\theta}/D_{a,\theta}$ , when  $D_{a,\theta} > 0$ , is a weighted average of biases lying in  $I_a$ . Therefore

$$\sum_c x_c g_{a,\theta}^{\text{lo}}(c) \leq 0, \quad \sum_c x_c g_{a,\theta}^{\text{up}}(c) \leq 0.$$

The finite factor-revealing LP behind the certificate is

$$\begin{aligned}
\eta^* &:= \min && z \\
\text{s.t.} &&& \sum_c o(c)x_c = 1, \\
&&& \sum_c g_{i,\tau}^{\text{lo}}(c)x_c \leq 0 \quad \forall i, \tau, \\
&&& \sum_c g_{i,\tau}^{\text{up}}(c)x_c \leq 0 \quad \forall i, \tau, \\
&&& \sum_c R_r(c)x_c \leq z \quad \forall r = 1, \dots, 9, \\
&&& x_c \geq 0 \quad \forall c.
\end{aligned} \tag{22}$$

Here, the normalization  $\sum_c o(c)x_c = 1$  scales the optimum value to one, and  $z$  upper-bounds the value achieved by each of the nine rounding rules. This finite LP is solved by the accompanying search code<sup>5</sup>. The numerical solution is used only to find a candidate certificate; the proof below uses the rational Farkas-dual witness checked by the exact verifier:

$$\begin{aligned}
\max & \quad \rho \\
\text{s.t.} & \quad \sum_{r=1}^9 \alpha_r R_r(c) - \rho o(c) + \sum_{i,\tau} \lambda_{i,\tau}^{\text{lo}} g_{i,\tau}^{\text{lo}}(c) + \sum_{i,\tau} \lambda_{i,\tau}^{\text{up}} g_{i,\tau}^{\text{up}}(c) \geq 0 \quad \forall c, \\
& \quad \alpha_r \geq 0 \quad (1 \leq r \leq 9), \quad \sum_{r=1}^9 \alpha_r = 1, \\
& \quad \lambda_{i,\tau}^{\text{lo}}, \lambda_{i,\tau}^{\text{up}} \geq 0.
\end{aligned} \tag{23}$$

In particular, the checker certifies nonnegative numbers  $\alpha_1, \dots, \alpha_9$ , with  $\sum_r \alpha_r = 1$ , and nonnegative multipliers  $\lambda_{a,\theta}^{\text{lo}}, \lambda_{a,\theta}^{\text{up}}$ , such that for every typed atom  $c$ ,

$$\sum_{r=1}^9 \alpha_r R_r(c) - \rho_2 o(c) + \sum_{a,\theta} \lambda_{a,\theta}^{\text{lo}} g_{a,\theta}^{\text{lo}}(c) + \sum_{a,\theta} \lambda_{a,\theta}^{\text{up}} g_{a,\theta}^{\text{up}}(c) \geq 0.$$

Multiplying by  $x_c$  and summing over all atoms gives

$$\sum_{r=1}^9 \alpha_r A_r(\text{WSnap}_t(\Phi)) - \rho_2 \text{OPT}(\Phi) + \sum_{a,\theta} \lambda_{a,\theta}^{\text{lo}} \sum_c x_c g_{a,\theta}^{\text{lo}}(c) + \sum_{a,\theta} \lambda_{a,\theta}^{\text{up}} \sum_c x_c g_{a,\theta}^{\text{up}}(c) \geq 0.$$

The two bucket-feasibility sums are nonpositive, and all multipliers are nonnegative. Hence

$$\sum_{r=1}^9 \alpha_r A_r(\text{WSnap}_t(\Phi)) \geq \rho_2 \text{OPT}(\Phi).$$

Since  $\alpha$  is a convex combination,

$$A_{\max}(\text{WSnap}_t(\Phi)) \geq \sum_{r=1}^9 \alpha_r A_r(\text{WSnap}_t(\Phi)) \geq \rho_2 \text{OPT}(\Phi).$$

The complete certificate data and verifier are also available in the repository: <https://github.com/Guangxu-Yang/maxksat-lp-certificate>.  $\square$

<sup>5</sup>`scripts/max2sat_multi_rounding_certificate_search.py` in codes