Classical Clifford+T Sampling without Computing Marginals

Mark Koch
Department of Computer Science
University of Oxford
mark.koch@cs.ox.ac.uk

We introduce a new exact classical sampling algorithm for Clifford+T circuits, called T-by-T sampling. For a circuit with T-count t, it reduces sampling to evaluating $2t$ amplitudes of circuits with successively increasing T-counts $1,2,...,t$, without the need to compute costly marginal probabilities. Crucially, the number of amplitude evaluations is completely independent of the surrounding Clifford structure. This makes T-by-T sampling well suited for use together with low-rank stabiliser decompositions, which enable efficient amplitude evaluations as long as the T-count is not too high. In that sense, T-by-T sampling is an improvement over the gate-by-gate sampling technique introduced by Bravyi, Gosset, and Liu [6], where the number of amplitudes instead scales with the number of non-monomial operations such as Hadamard gates, which can remain large even in circuits with low stabiliser rank. Concretely, T-by-T sampling offers significant speedups in regimes where T gates are embedded within large Clifford computations. This makes it well suited for applications such as simulating quantum error correction experiments, which typically have lower T complexity but involve extensive Clifford operations and measurements. In particular, we show how T-by-T sampling allows us to take samples from recent end-to-end magic state cultivation circuits with escape up to distance 19 in a couple of seconds.

1 Introduction

It is widely believed that simulating universal quantum circuits on a classical computer has an exponential runtime cost. Nevertheless, classical simulation of quantum systems continues to play an important role in the field, with applications ranging from the design and validation of quantum algorithms to benchmarking and verification of hardware. Consequently, there is significant interest in accelerating classical simulation methods and coming up with techniques that work well for specific classes of circuits. One such example is the class of *Clifford+T circuits* that allow for fault-tolerant compilation while at the same time being approximately universal. As a result, Clifford+T serves as the standard compilation framework for many fault-tolerant architectures, and the number of T gates, also known as the *T-count*, is widely regarded as a key resource measure. An effective technique for simulating Clifford+T circuits with relatively low T-counts is the *low-rank stabiliser decomposition* method [5, 4]. Given a circuit C, the idea is to decompose $C|0\rangle$ into a linear combination of stabiliser states. The number of terms in this decomposition, also known as the *stabiliser rank*, is believed to scale as $2^{\alpha t}$ where t denotes the T-count of C and $\alpha < 1$ is a constant. The best known strategy achieves $\alpha \approx 0.396$ and was discovered in [12, 11]. This decomposition then allows us to compute an output probability $P(\vec{x}) := |\langle \vec{x}|C|0\rangle|^2$ by evaluating the corresponding amplitude for each stabiliser term and summing up the contributions. The task of computing such probabilities $P(\vec{x})$ is also known as *strong simulation*.

In many applications, however, the objective is not to compute individual probabilities, but rather to *sample* from the output distribution P. This task is also known as *weak simulation*. For example, in the context of quantum error correction (QEC), one typically performs Monte Carlo simulations over many noisy shots to estimate the logical error rate of fault-tolerant protocols or gadgets. A standard

To appear in EPTCS.

method to reduce weak simulation to strong simulation proceeds via sequential marginalisation. The idea is to calculate marginal probabilities $P_j(x_1,...,x_j) = \langle 0|C^\dagger(|\vec{x}\rangle\langle\vec{x}| \otimes I_{n-j})C|0\rangle$ and then sample the qubit outcomes one by one, conditioning on the previously sampled bits and marginalising over the ones that have not been sampled yet. However, this approach comes with a significant runtime overhead. In the case of Clifford+T circuits, computing a marginal P_j corresponds to evaluating an amplitude of a circuit with *twice* the number of T gates. This leads to an exponential runtime increase from $O(2^{\alpha t})$ to $O(4^{\alpha t})$, making it very costly in practice.

Several approaches have been proposed to mitigate this overhead. For example, Refs. [5, 4] approximate marginal probabilities via a norm estimation technique based on overlaps with carefully chosen random stabiliser states. [4] also proposed a Metropolis-Hastings Markov Chain Monte Carlo method for approximate sampling, however without rigorous error bounds. To the best of our knowledge, the current state-of-the-art method is the *gate-by-gate* algorithm proposed by Bravyi, Gosset, and Liu [6]. The high-level idea is to begin with a trivial sample $\vec{x} = 0$ and update it by applying the gates of the circuit one by one, using amplitude evaluations to resample affected bits. Concretely, if C is expressed in the CNOT+Hadamard+T gate set, then the number of required amplitude evaluations will scale linearly with the number of Hadamard gates. While this avoids the doubling of the T-count inherent in marginal-based sampling, the number of Hadamard gates can still be very large even for circuits with low stabiliser rank. For example, this situation is typical in QEC circuits, which are dominated by extensive Clifford subroutines, such as syndrome extraction, lattice surgery etc., while non-Clifford resources are primarily concentrated in magic state preparation. As a consequence, the gate-by-gate method may still require a large number of expensive amplitude evaluations despite a relatively low T-count.

These considerations motivate the development of a sampling method in which the number of amplitude evaluations depends primarily on the T-count, rather than the surrounding Clifford operations. This work introduces a new exact sampling algorithm for Clifford+T circuits that achieves precisely this goal. The new method, which is called *T-by-T sampling*, requires only $2t$ amplitude evaluations for a circuit with T-count t, completely independent of the Clifford structure of the circuit. Moreover, most of these amplitude evaluations involve circuits with significantly smaller T-count than t. Concretely, the algorithm proceeds by evaluating amplitude pairs of circuits with increasing T-counts $1,2,...,t$, so that only the very last step incurs the full strong simulation cost of $O(2^{\alpha t})$. As a result, T-by-T sampling provides a substantial practical speed-up for circuits where T gates are embedded within large Clifford computations. After reviewing the gate-by-gate method in Section 2 and introducing T-by-T sampling in Section 3, this advantage is demonstrated through numerical benchmarks in Section 4, including experiments on random circuits as well as magic state cultivation circuits from the recent QEC literature [9].

2 Background: Gate-by-Gate Sampling

Before introducing T-by-T sampling, it will be useful to first review Bravyi, Gosset, and Liu's [6] gate-by-gate method in detail. To this end, let $C = U_m \cdots U_1$ be a quantum circuit (not necessarily Clifford+T) consisting of a sequence of m unitary gates U_i. The goal is to sample from the distribution $P(\vec{x}) := |\langle\vec{x}|C|0\rangle|^2$. Let $C_k = U_k \cdots U_1$ be the subcircuit containing only the first k gates. In particular, we have $C_0 = I$ and $C_m = C$. Finally, define the output distribution at time step k as $P_k(\vec{x}) := |\langle\vec{x}|C_k|0\rangle|^2$. In particular, $P_0(\vec{x}) = |\langle\vec{x}|0\rangle|^2 = \delta_{\vec{x}=0}$ and $P_m = P$. The key idea of gate-by-gate sampling is to start with a sample from the trivial distribution P_0 (i.e. the all-zero bit string) and updating it one gate at a time until we arrive at a sample that is distributed according to $P_m = P$. This works because adjacent distributions P_{k-1} and P_k are closely related: they only differ by a single gate U_k that moreover only acts on a small

subset $S \subseteq \{1,...,n\}$ of qubits. On the remaining qubits $\neg S := \{1,...,n\} \setminus S$, the gate acts trivially. In particular, the distribution of outcomes x_i for these trivial qubits must be the same for P_k and P_{k-1}. In other words, the marginal distribution $P_k(\vec{x}_{\neg S}) := \sum_{\vec{y}_S} P_k(\vec{x}_{\neg S}, \vec{y}_S)$ including only the qubits in $\neg S$ satisfies $P_k(\vec{x}_{\neg S}) = P_{k-1}(\vec{x}_{\neg S})$. Therefore,

$$P_k(\vec{x}) = P_k(\vec{x}_S \mid \vec{x}_{\neg S}) P_k(\vec{x}_{\neg S}) = P_k(\vec{x}_S \mid \vec{x}_{\neg S}) P_{k-1}(\vec{x}_{\neg S}).$$

Thus, in order to turn a sample $\vec{y} \sim P_{k-1}$ into a sample $\vec{y}' \sim P_k$, the trivial bits can be kept unchanged, $\vec{y}'_{\neg S} := \vec{y}_{\neg S}$, and only the bits $\vec{y}'_S$ that U_k acts on need to be resampled according to the conditional distribution

$$P_k(\vec{y}'_S \mid \vec{y}_{\neg S}) = \frac{P_k(\vec{y}'_S, \vec{y}_{\neg S})}{\sum_{\vec{x}_S} P_k(\vec{x}_S, \vec{y}_{\neg S})}.$$

For example, if U_k is a single-qubit gate, then $S = \{q\}$ for some qubit q. To resample y'_q, one only needs to compute the two probabilities $p_0 = P_k(x_q = 0, \vec{x}_{\neg q} = \vec{y}_{\neg q})$ and $p_1 = P_k(x_q = 1, \vec{x}_{\neg q} = \vec{y}_{\neg q})$, and then set $y'_q := 1$ with probability $p_1/(p_0 + p_1)$. In general, updating a sample after applying an ℓ-qubit gate requires 2^ℓ amplitude evaluations, making the gate-by-gate algorithm scale exponentially in the maximum gate size. However, in many cases this size is bounded, for example when considering circuits made up of standard single- and two-qubit gates.

Furthermore, there is a special case where amplitude evaluations can be skipped completely: Suppose U_k is a *monomial gate*, meaning it maps computational basis states to other computational basis states up to a phase.[1] In other words, suppose there are functions $f : \mathbb{F}_2^n \to \mathbb{F}_2^n$ and $\varphi : \mathbb{F}_2^n \to \mathbb{R}$ such that $U_k|\vec{x}\rangle = e^{i\varphi(\vec{x})}|f(\vec{x})\rangle$ for all $\vec{x}$. Examples of these monomial gates include CNOT, Toffoli, T gates and (controlled) Z rotations. Note that f must be injective since U_k is unitary with $U_k^\dagger|\vec{x}\rangle = e^{-i\varphi(\vec{x})}|f^{-1}(\vec{x})\rangle$ such that

$$P_{k-1}(\vec{x}) = |\langle\vec{x}|C_{k-1}|0\rangle|^2 = \left|\langle\vec{x}|U_k^\dagger U_k C_{k-1}|0\rangle\right|^2 = \left|e^{-i\varphi(\vec{x})}\langle f^{-1}(\vec{x})|U_k C_{k-1}|0\rangle\right|^2 = P_k(f^{-1}(\vec{x})).$$

Thus, given a sample $\vec{y} \sim P_{k-1}$, it can be deterministically updated to $\vec{y}' := f(\vec{y})$, which satisfies $P_k(\vec{y}') = P_{k-1}(\vec{y})$ without needing to perform amplitude calculations. Therefore, expensive resampling is only required when we encounter non-monomial gates such as Hadamards. In particular, if the circuit is given in the CNOT+Hadamard+T gate set, then gate-by-gate sampling requires exactly $2h$ amplitude evaluations, where h is the number of Hadamards. The goal of the T-by-T sampling procedure that will be introduced in the next section is to remove this dependence on h and instead make it scale only with the T-count t.

3 T-by-T Sampling

At a high level, T-by-T sampling follows a structure similar to the gate-by-gate method. However, instead of processing the circuit one general gate at a time, it is sampled one *T gate* at the time. Concretely, the idea is to start with a pure Clifford circuit that does not contain any T gates. Obtaining a sample from this initial circuit can be done efficiently using standard methods [3, 8]. This sample is then refined by reintroducing the T gates into the computation one by one, until we obtain a sample from the original circuit. In Section 3.1 below, we give a description of the algorithm using a framework similar to Section 2.

[1] We call these gates monomial since they correspond to monomial matrices, i.e. matrices where each row and column has exactly one non-zero entry.

The complexity of the algorithm is analysed in Section 3.2 Finally, Section 3.3 gives an alternative, more circuit-centric interpretation of T-by-T sampling and points out some connections to measurement based quantum computation.

3.1 The Algorithm

Let C be a Clifford+T circuit with T-count t. Similar to gate-by-gate sampling, the algorithm proceeds by considering a sequence $\{C_k\}$ of circuits derived from C. In particular, let C_k be a modified version of C that keeps only the first k T gates and replaces the remaining $t-k$ ones with Z basis measurements. In particular, C_0 is a Clifford computation and $C_t = C$. Sampling from C_k yields the final outcomes $\vec{x} \in \mathbb{F}_2^n$, but also $t-k$ additional bits $\vec{a} \in \mathbb{F}_2^{t-k}$ for the intermediate measurement results. We write $C_k^{\vec{a}}$ for the circuit where these intermediate measurements are replaced with projectors $\Pi^{a_i} := |a_i\rangle\langle a_i|$ onto the outcomes $\vec{a}$. This yields a distribution $P_k(\vec{x},\vec{a}) = \left|\langle\vec{x}|C_k^{\vec{a}}|0\rangle\right|^2$ over the outcomes $\vec{x},\vec{a}$. The idea behind T-by-T sampling is to start with a sample from P_0 which can be computed efficiently. Then, we refine this sample by inserting T gates into the computation one by one until we arrive at a sample from the final distribution $P_t = P$. Similar to gate-by-gate sampling, this works because adjacent distributions P_{k-1} and P_k are closely related. Consider the following lemma:

Lemma 1. *There exist bit strings $\vec{\gamma}_k \in \mathbb{F}_2^n$ and $\vec{\delta}_k \in \mathbb{F}_2^{t-k}$ such that for all $\vec{x} \in \mathbb{F}_2^n$ and $\vec{a} \in \mathbb{F}_2^{t-k}$*

$$P_{k-1}(\vec{x},\vec{a},a_k=0) + P_{k-1}(\vec{x},\vec{a},a_k=1) = \frac{1}{2}\left(P_k(\vec{x},\vec{a}) + P_k(\vec{x}\oplus\vec{\gamma}_k,\vec{a}\oplus\vec{\delta}_k)\right). \tag{1}$$

Proof. By definition

$$C_{k-1}^{\vec{a},a_k} = V^{\vec{a}}\Pi_q^{a_k}U \qquad\qquad C_k^{\vec{a}} = V^{\vec{a}}T_qU$$

where U is the Clifford+T prefix of $C_{k-1}^{\vec{a},a_k}$ containing the first $k-1$ T gates, Π^{a_k} is the projector replacing the k-th T gate on qubit q, and $V^{\vec{a}}$ is the remaining Clifford suffix of $C_{k-1}^{\vec{a},a_k}$ with projectors replacing the remaining $t-k$ T gates according to outcomes $\vec{a}$. Using the fact that

$$\Pi^{a_k} = |a_k\rangle\langle a_k| = \frac{1}{2}e^{-ia_k\frac{\pi}{4}}(T + (-1)^{a_k}ZT), \tag{2}$$

we get

$$\begin{aligned}
\sum_{a_k} P_{k-1}(\vec{x},\vec{a},a_k) &= \sum_{a_k}\left|\langle\vec{x}|V^{\vec{a}}\Pi_q^{a_k}U|0\rangle\right|^2 \\
&\overset{(2)}{=} \sum_{a_k}\left|\frac{1}{2}e^{-ia_k\frac{\pi}{4}}\langle\vec{x}|V^{\vec{a}}(T+(-1)^{a_k}ZT)_qU|0\rangle\right|^2 \\
&= \sum_{a_k}\frac{1}{4}\left|\langle\vec{x}|V^{\vec{a}}(T+(-1)^{a_k}ZT)_qU|0\rangle\right|^2 \\
&= \sum_{a_k}\frac{1}{4}\Big(\langle\vec{x}|V^{\vec{a}}T_qU|0\rangle\langle 0|U^\dagger T_q^\dagger V^{\vec{a}\dagger}|\vec{x}\rangle + (-1)^{a_k}\langle\vec{x}|V^{\vec{a}}T_qU|0\rangle\langle 0|U^\dagger T_q^\dagger Z_qV^{\vec{a}\dagger}|\vec{x}\rangle \\
&\qquad + (-1)^{a_k}\langle\vec{x}|V^{\vec{a}}Z_qT_qU|0\rangle\langle 0|U^\dagger T_q^\dagger V^{\vec{a}\dagger}|\vec{x}\rangle + \langle\vec{x}|V^{\vec{a}}Z_qT_qU|0\rangle\langle 0|U^\dagger T_q^\dagger Z_qV^{\vec{a}\dagger}|\vec{x}\rangle\Big) \\
&= \frac{1}{2}\langle\vec{x}|V^{\vec{a}}T_qU|0\rangle\langle 0|U^\dagger T_q^\dagger V^{\vec{a}\dagger}|\vec{x}\rangle + \frac{1}{2}\langle\vec{x}|V^{\vec{a}}Z_qT_qU|0\rangle\langle 0|U^\dagger T_q^\dagger Z_qV^{\vec{a}\dagger}|\vec{x}\rangle \\
&= \frac{1}{2}\left|\langle\vec{x}|V^{\vec{a}}T_qU|0\rangle\right|^2 + \frac{1}{2}\left|\langle\vec{x}|V^{\vec{a}}Z_qT_qU|0\rangle\right|^2
\end{aligned}$$

Note that while the first term already corresponds to $\frac{1}{2}P_k(\vec{x},\vec{y})$, the second term has the wrong form since Z_qT_q is applied instead of the desired T_q. However, since $V^{\vec{a}}$ is Clifford, the Pauli Z can be pushed through it, potentially flipping some of the projector outcomes. In particular, there is a Pauli string P and flip vector $\vec{\delta}_k \in \mathbb{F}_2^{t-k}$ such that $V^{\vec{a}}Z_q = PV^{\vec{a}\oplus\vec{\delta}_k}$ up to a phase. Similarly, there is a flip vector $\vec{\gamma}_k \in \mathbb{F}_2^n$ such that $\langle\vec{x}|P = \langle\vec{x}\oplus\vec{\gamma}_k|$ up to a phase. Therefore,

$$\left|\langle\vec{x}|V^{\vec{a}}Z_qT_qU|0\rangle\right|^2 = \left|\langle\vec{x}\oplus\vec{\gamma}_k|V^{\vec{a}\oplus\vec{\delta}_k}T_qU|0\rangle\right|^2 = P_k(\vec{x}\oplus\vec{\gamma}_k,\vec{a}\oplus\vec{\delta}_k). \qquad \square$$

Lemma 1 tells us that the marginal distribution $P_{k-1}(\vec{x},\vec{a})$ over a_k is an equal mixture of two copies of P_k, where one of them is shifted by the bit-flip vectors $\vec{\gamma}_k$ and $\vec{\delta}_k$. Therefore, to refine a sample $(\vec{y},\vec{b},b_k) \sim P_{k-1}$ to the new distribution P_k, it suffices to choose between these two copies with the appropriate relative weight. To this end, let

$$p(\vec{y},\vec{b}) := \frac{P_k(\vec{y}\oplus\vec{\gamma}_k,\vec{b}\oplus\vec{\delta}_k)}{P_k(\vec{y},\vec{b}) + P_k(\vec{y}\oplus\vec{\gamma}_k,\vec{b}\oplus\vec{\delta}_k)} \tag{3}$$

be the flip probability for the sample $(\vec{y},\vec{b})$. Similar to the gate-by-gate algorithm, $p(\vec{y},\vec{b})$ can be evaluated by computing the two probabilities $P_k(\vec{y},\vec{b})$ and $P_k(\vec{y}\oplus\vec{\gamma}_k,\vec{b}\oplus\vec{\delta}_k)$. Then, with probability $p(\vec{y},\vec{b})$, set $\vec{y}' := \vec{y}\oplus\vec{\gamma}_k$ and $\vec{b}' := \vec{b}\oplus\vec{\delta}_k$. Otherwise, keep $\vec{y}' := \vec{y}$ and $\vec{b}' := \vec{b}$. The claim is that $(\vec{y}',\vec{b}')$ is now correctly distributed w.r.t. P_k. To see this, let $Q(\vec{x}',\vec{a}')$ be the distribution of outcomes that we obtain using this process. Note that there are exactly two ways we can end up with a given final configuration $(\vec{x}',\vec{a}')$:

(i) either, we already sampled $(\vec{x}',\vec{a}')$ from P_{k-1} and did not apply the flip, or

(ii) we sampled $(\vec{x}'\oplus\vec{\gamma}_k,\vec{a}'\oplus\vec{\delta}_k)$ from P_{k-1} and then applied the flip.

Thus, we have

$$\begin{aligned}
Q(\vec{x}',\vec{a}') &= P_{k-1}(\vec{x}',\vec{a}')(1-p(\vec{x}',\vec{a}')) + P_{k-1}(\vec{x}'\oplus\vec{\gamma}_k,\vec{a}'\oplus\vec{\delta}_k)p(\vec{x}'\oplus\vec{\gamma}_k,\vec{a}'\oplus\vec{\delta}_k) \\
&= \frac{P_{k-1}(\vec{x}',\vec{a}')P_k(\vec{x}',\vec{a}')}{P_k(\vec{x}',\vec{a}') + P_k(\vec{x}'\oplus\vec{\gamma}_k,\vec{a}'\oplus\vec{\delta}_k)} + \frac{P_{k-1}(\vec{x}'\oplus\vec{\gamma}_k,\vec{a}'\oplus\vec{\delta}_k)P_k(\vec{x}',\vec{a}')}{P_k(\vec{x}',\vec{a}') + P_k(\vec{x}'\oplus\vec{\gamma}_k,\vec{a}'\oplus\vec{\delta}_k)} \\
&\overset{(1)}{=} \frac{P_{k-1}(\vec{x}',\vec{a}')P_k(\vec{x}',\vec{a}')}{2P_{k-1}(\vec{x}',\vec{a}')} + \frac{P_{k-1}(\vec{x}'\oplus\vec{\gamma}_k,\vec{a}'\oplus\vec{\delta}_k)P_k(\vec{x}',\vec{a}')}{2P_{k-1}(\vec{x}'\oplus\vec{\gamma}_k,\vec{a}'\oplus\vec{\delta}_k)} \\
&= \frac{1}{2}P_k(\vec{x}',\vec{a}') + \frac{1}{2}P_k(\vec{x}',\vec{a}') \\
&= P_k(\vec{x}',\vec{a}'),
\end{aligned}$$

showing that $\vec{y}',\vec{b}'$ is indeed a correct sample from P_k. To summarise, for each T gate k that is added back into the circuit, the current sample is randomly flipped by some vectors $\vec{\gamma}_k,\vec{\delta}_k$. In particular, the flip probability p at each step is computed from the relative weight between the probabilities $P_k(\vec{x},\vec{a})$ and $P_k(\vec{x}\oplus\vec{\gamma}_k,\vec{a}\oplus\vec{\delta}_k)$ as specified in equation (3). At the end, we arrive at a sample that is correctly distributed w.r.t. the target distribution $P_t = P$ of the original Clifford+T circuit. Pseudocode for the entire sampling procedure is given in Algorithm 1.

Algorithm 1 T-by-T Sampling

Input: An n-qubit Clifford+T circuit C with T-count t
Output: Sample $\vec{x}$ with probability $|\langle\vec{x}|C|0\rangle|^2$

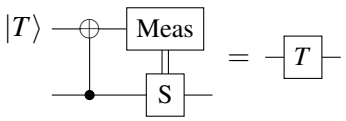
$C_0 \leftarrow$ Replace all T gates in C with Z measurements
$(\vec{x},\vec{a}) \leftarrow$ Sample from Clifford C_0
for $k = 1$ **to** t **do**
    $(\vec{a}, a_k) \leftarrow \textsc{Pop}(\vec{a})$ ▷ Drop outcome a_k
    $C_k \leftarrow$ Add next T gate back into C_{k-1}
    $(\vec{\gamma}_k, \vec{\delta}_k) \leftarrow$ Compute flip vectors according to Lemma 1
    $A \leftarrow \textsc{Prob}(C_k, \vec{x}, \vec{a})$ ▷ Compute $P_k(\vec{x},\vec{a})$
    $B \leftarrow \textsc{Prob}(C_k, \vec{x}\oplus\vec{\gamma}_k, \vec{a}\oplus\vec{\delta}_k)$ ▷ Compute $P_k(\vec{x}\oplus\vec{\gamma}_k,\vec{a}\oplus\vec{\delta}_k)$
    $p \leftarrow B/(A+B)$
    if $\textsc{RandomSample}(p)$ **then**
        $(\vec{x},\vec{a}) \leftarrow (\vec{x}\oplus\vec{\gamma}_k, \vec{a}\oplus\vec{\delta}_k)$ ▷ Apply flip with probability p
    end if
end for
return $\vec{x}$

3.2 Analysis

The time complexity of T-by-T sampling heavily depends on the method used for evaluating the probabilities $P_k(\vec{x},\vec{a})$. In general, for a circuit with T-count t, the algorithm will make exactly $2t$ calls to this strong simulation procedure. Gate-by-gate sampling on the other hand requires two amplitude evaluations per non-monomial gate in the circuit. Assuming that we are working with polynomially sized circuits, this corresponds to a polynomial number of strong simulations. For simulation via low rank stabiliser decompositions, computing an amplitude for a circuit with T-count t requires $2^{\alpha t}$ stabiliser terms. In the case of gate-by-gate sampling, the worst case would be a circuit where the non-monomial gates occur after all T have already been applied, making every evaluation cost the maximum of $2^{\alpha t}$ stabiliser terms, resulting in $O(\text{poly}(n)2^{\alpha t})$ terms in total. For T-by-T sampling, the T-count increases gradually in each step, requiring only $\sum_{k=1}^{t} 2\cdot 2^{\alpha k} = O(2^{\alpha t})$ terms in total, thus yielding a polynomial advantage over gate-by-gate sampling. Of course, the asymptotic growth is still dominated by the exponential scaling in t, in particular for the final evaluation of the circuit containing all T gates. However, in situations where the input circuit contains a lot more non-monomial gates than T gates, T-by-T sampling requires significantly fewer calls to this exponentially costly procedure than gate-by-gate sampling, which makes it a lot faster in practice.

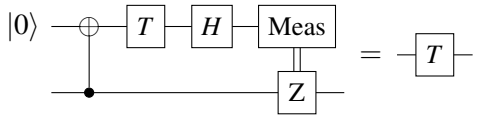
3.3 MBQC Perspective

Finally, we give an alternative perspective on T-by-T sampling using ideas from *measurement based quantum computation* (MBQC) [13] and further relate it to gate-by-gate sampling. First, note that Clifford+T circuits can be viewed as pure Clifford computations that consume $|T\rangle := \frac{1}{\sqrt{2}}(|0\rangle + e^{i\frac{\pi}{4}}|1\rangle)$ magic states. In particular, those magic states can be turned back into T gates using the following injection circuit:



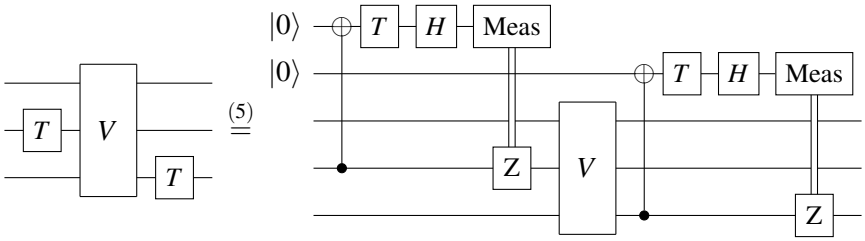
(4)

Instead of state injection in its standard form, we may however also consider the time-reversed version of this protocol where, instead of preparing the ancilla in $|T\rangle$ and measuring in the $\{|0\rangle,|1\rangle\}$ basis, we prepare the ancilla in $|0\rangle$ and measure in the $\{|T_+\rangle,|T_-\rangle\}$ basis where $|T_\pm\rangle := \frac{1}{\sqrt{2}}(|0\rangle \pm e^{i\frac{\pi}{4}}|1\rangle)$. Equivalently, this corresponds to first applying HT and then measuring in the computational basis:

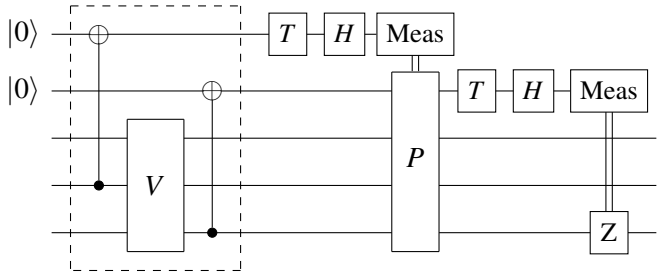


(5)

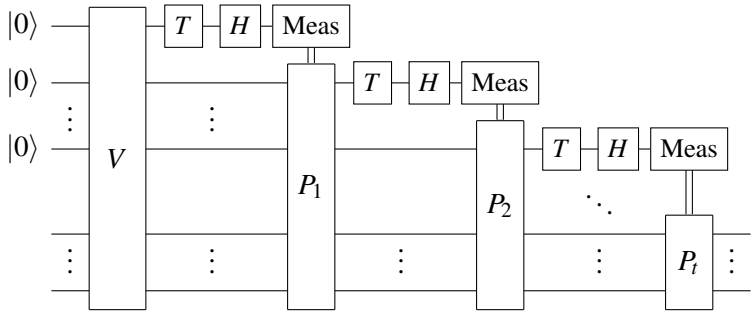
Notably, the correction for the -1 outcome is now Z as opposed to S in (4). As an example, apply this transformation to the following circuit with two T gates separated by a Clifford V:



Since V is Clifford, the first Z correction can actually be delayed by pushing it through V and the following CNOT, resulting in some Pauli string P on the remaining qubits:



Note that the beginning of the circuit in the dashed box is now purely Clifford and the T gates are applied at the very end, only interleaved with measurements and Pauli corrections. By iterating this argument, it is easy to see that any Clifford+T circuit with T-count t can be represented as some Clifford W with t additional ancillas initialised in $|0\rangle$, followed by HT measurements and Pauli corrections:



(6)

This transformation can be viewed as a limited form of MBQC, where Cliffords remain as regular gates in the circuit model, but all T gates are realised via ancilla measurements. Interestingly, the MBQC computation in (6) is closely related to the steps we described in Section 3.1. First, note that a sample

from the initial Clifford V exactly corresponds to a sample from the initial distribution $P_0(\vec{x},\vec{a})$: All T gates are replaced with CNOTs according to (5) with the target initialised in $|0\rangle$. Thus, measuring one of the ancillas in the Z basis corresponds to applying a projector Π^{a_k} instead of the T gate and yields the additional outcome a_k. After applying the change of basis HT, the measurement injects either a T or a ZT gate into the circuit, which is similar to equation (2) in the proof of Lemma 1. Finally, the X components of the Pauli corrections P_k exactly describe the bit flips $\vec{\gamma}_k, \vec{\delta}_k$ from Lemma 1.

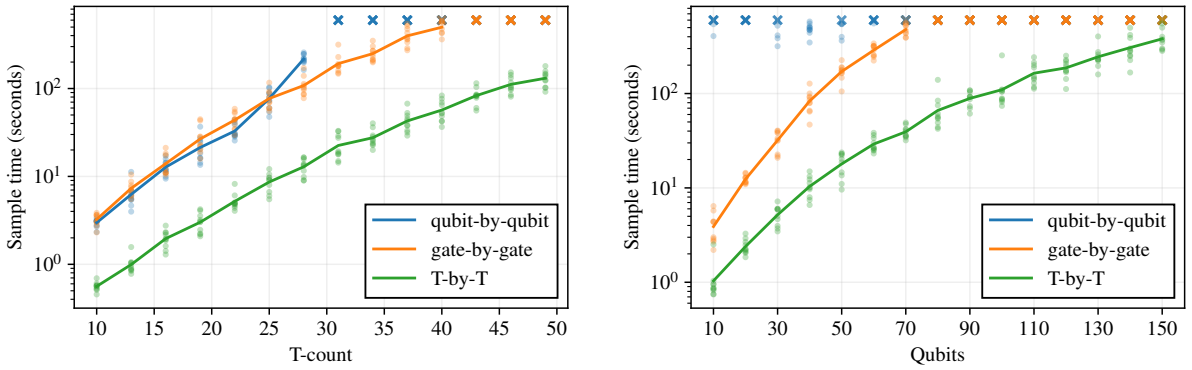
From this perspective, once we have sampled from the initial Clifford circuit W, the remaining computation in (6) can be interpreted using the gate-by-gate framework. The injected T gate itself is monomial and therefore preserves computational basis states. The subsequent Hadamard, however, is non-monomial and requires resampling. Resampling after the Hadamard amounts to evaluating two amplitudes corresponding to projections onto $\langle T_+|$ and $\langle T_-|$. These amplitudes coincide with those of the circuit in which either a T or a ZT gate is injected. After commuting the Z through the Clifford suffix, these are precisely the amplitudes $P_k(\vec{x},\vec{a})$ and $P_k(\vec{x}\oplus\vec{\gamma}_k,\vec{a}\oplus\vec{\delta}_k)$ from Section 3.1. Thus, Lemma 1 tells us that P_{k-1} can be interpreted as the distribution coming from the equal mixture of the branches where either T or ZT is injected. Then, the flip probability (3) is exactly the posterior probability that a given sample arose from the ZT branch rather than the T branch. If the ZT branch is selected, we apply the deterministic bit flip that maps the sample back into the T branch. Thus, we can view the T-by-T update rule as projecting the ancilla measurements in (6) onto the +1 outcome.

Interestingly, the circuit (6) also shows us a slight alternative implementation of T-by-T sampling. Instead of projecting onto the +1 outcome, we could keep the -1 outcome but then modify the circuit by adding the Pauli correction P_k. Since Paulis are monomial gates, they just correspond to deterministic flips in the outcome and do not require resampling. The main difference is that these Pauli corrections would remain in the circuit when we move on to the next T gates. In particular, X components in the correction frame would correspond to flipping future T gates into $T^\dagger$ gates. This complication is avoided by following the postselected algorithm as described in Section 3.1.

4 Experiments

To compare T-by-T sampling against the gate-by-gate and qubit-by-qubit marginalisation method, we ran numerical experiments measuring the sampling time for both random circuits (Section 4.1) and structured QEC circuits for magic state cultivation (Section 4.2). Since the goal of this section is to compare sampling strategies rather than the underlying strong simulation techniques, the stabiliser decomposition backend is kept identical across all experiments. Any observed performance differences therefore arise from the sampling strategy, specifically, from how many amplitude evaluations are required and at which effective T-count they are performed. In particular, all experiments use the BSS + cat state stabiliser decomposition algorithm [11] implemented in the `quizx` library [10, 1], achieving the state of the art worst-case scaling factor of $\alpha \approx 0.396$. All tests were performed on a consumer laptop, imposing a 10 minute time limit per sample. All data points represent running times on a single CPU core. Implementations of all sampling methods, together with code to generate the circuits and reproduce the results are open source and available at:

`https://github.com/mark-koch/t-by-t-sampling`



(a) Random 50-qubit circuits with increasing T-counts (b) Random T-count 30 circuits with increasing qubits

Figure 1: Runtime for obtaining a sample from random Clifford+T circuits. The solid lines indicate an average over 10 runs. Cross marks denote runs that exceeded the 10 minute time limit.

4.1 Random Clifford+T Circuits

First, we consider random Clifford+T circuits constructed from multiple layers of exponentiated Pauli unitaries of the form $e^{-i\frac{\pi}{8}P}$ for some dense Pauli string P whose entries are uniformly sampled from $\{I,X,Y,Z\}$. Each Pauli exponential contributes exactly one T gate to the circuit, so the T-count t is equal to the number of layers. For the qubit-by-qubit marginalisation method, this T-count will effectively double to $2t$ since computing marginals corresponds to evaluating amplitudes of a circuit with twice as many T gates. For the gate-by-gate method, the T-count remains t, but each layer requires resampling of qubit outcomes x_i for all positions where the Pauli support P_i is X or Y as these require a non-monomial change of basis. For dense Pauli strings, this occurs on approximately half of the qubits in expectation, so each layer requires $O(n)$ resampling steps, where n is the number of qubits. Over the entire circuit, this results in $O(nt)$ amplitude evaluations with the T-count increasing after each layer. In contrast, T-by-T sampling only requires $2t$ amplitude evaluations in total, independent of n.

Figure 1a plots the runtime of these three methods for random 50-qubit circuits with increasing T-counts. As discussed in Section 3.2, any sampling method based on stabiliser decompositions is expected to scale exponentially in t. This behaviour is clearly visible in Figure 1a: all three curves exhibit exponential growth in t. Moreover, the gate-by-gate and T-by-T methods have essentially identical slopes. This is because both methods perform amplitude evaluations at T-count t, which dominates the asymptotic scaling. Nevertheless, T-by-T sampling consistently outperforms gate-by-gate sampling by roughly an order of magnitude across the tested range, since it only requires two evaluations at full T-count t. Finally, qubit-by-qubit marginalisation remains competitive with gate-by-gate sampling only up to about $t \approx 25$, after which the exponential penalty from doubling the T-count becomes dominant.

In Figure 1b, the T-count is fixed to $t = 30$ and instead the number of qubits n is varied, thereby also increasing the number of Clifford operations. This isolates the scaling of each method in relation to the overall circuit size rather than the T-count. In this setting, the advantage of T-by-T sampling becomes even clearer: the runtime gap between T-by-T and gate-by-gate sampling widens as the number of qubits increases, reflecting the $O(nt)$ resampling overhead of the gate-by-gate method. Qubit-by-qubit marginalisation is no longer competitive in this setting, with most runs exceeding the allotted time limit.

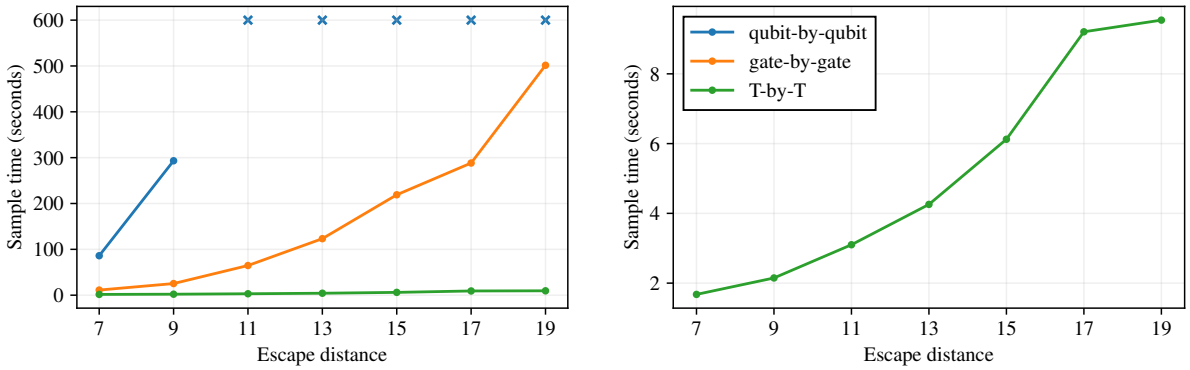


Figure 2: Runtime for obtaining a (noiseless) sample from distance 3 magic state cultivation circuits including escape into codes of increasing distance. The left side shows a zoomed in view of the scaling of T-by-T sampling. Cross marks denote runs that exceeded the 10 minute time limit.

Distance	Qubits	Gates	Measurements	H Gates	T Gates
7	102	2,290	520	520	15
9	166	3,675	840	825	15
11	246	5,424	1,240	1,206	15
13	342	7,573	1,720	1,663	15
15	454	10,014	2,280	2,196	15
17	582	12,855	2,920	2,805	15
19	726	16,060	3,640	3,490	15

Table 1: Magic state cultivation circuit sizes for different escape distances.

4.2 Magic State Cultivation Circuits

While random Clifford+T circuits are useful for studying generic scaling behaviour, they do not necessarily reflect the structure of real-world circuits encountered in practice. To evaluate performance in a more realistic setting, we now turn to Clifford+T circuits from the recent quantum error correction (QEC) literature. Concretely, we consider magic state cultivation circuits [9] implementing a recent protocol for constructing high fidelity $|T\rangle$ magic states within QEC codes. These circuits consist of an initial injection and cultivation step that prepares and verifies a logical $|T\rangle$ state in a small low-distance code. For the following experiments, we use the distance 3 injection and cultivation circuits from [9], which have a T-count of only $t = 15$. Then, the circuit is composed with an escape stage that rapidly grows the code size. Notably, the escape stage is purely Clifford, but it involves a large number of qubits, gates, and measurements. Following [9], we consider escape circuits with three QEC rounds during the growth step and five rounds in the final code. Overall circuit statistics for increasing escape distances are summarised in Table 1.

Figure 2 shows the sampling time for noiseless cultivation circuits with increasing escape distances. Because the escape stage contains many qubits and a large number of Hadamard gates, both qubit-by-qubit marginalisation and gate-by-gate sampling incur substantial runtime overheads as the distance increases. In contrast, the T-by-T method requires only $2t = 30$ amplitude evaluations, independent of escape distance and the size of the surrounding Clifford computation. As a result, the sampling time

increases only moderately, from approximately 1.5s to 10s, as the escape distance grows from 7 to 19. Gate-by-gate sampling, on the other hand, suffers an increase from roughly 11s to 500s over the same range. Finally, the qubit-by-qubit method already times out after distance 9. These results highlight that T-by-T sampling is particularly well suited for large-scale QEC simulations, where a small number of non-Clifford resources are embedded in large Clifford computations, making it a promising tool for simulating fault-tolerant protocols.

5 Discussion

In this work, we have shown that exact weak simulation of Clifford+T circuits can be reduced to a number of strong simulations that depends only on the T-count, and is independent of the surrounding Clifford structure. We demonstrated the advantage of this new T-by-T sampling approach over existing methods for a class of random circuits as well as magic state cultivation circuits [9], where other baselines are outperformed by one to two orders of magnitude. However, it is important to emphasise that this advantage is not universal. There exist classes of circuits for which alternative approaches may be more suitable. For example, dense single-layer IQP circuits [7] can contain significantly more T gates than Hadamard gates. In such regimes, the gate-by-gate method may be preferable, as its cost scales with the number of Hadamard gates rather than the T-count. T-by-T sampling, by contrast, is most effective in settings where a moderate number of T gates are embedded within large Clifford-dominated computations.

One such setting is simulation of large scale QEC experiments. The magic state cultivation benchmarks in Section 4.2 are very promising and suggest that T-by-T sampling could become a valuable tool for this application. In that context it is worth stressing that the simulations in Section 4.2 were single noiseless samples, whereas actual QEC simulations would typically insert stochastic noise across many shots, apply corrections conditioned on detected syndromes, and evaluate the overlap with the ideal logical state in order to determine the logical error rate. However, the goal of this work was not to reproduce such an end-to-end QEC study, but rather to demonstrate that T-by-T sampling is particularly well suited to the circuit structures that arise in that context.

A natural next step for future work would therefore be to integrate T-by-T sampling into a complete QEC simulation pipeline. Notably, there has been substantial recent progress in simulating Clifford+T QEC experiments, including techniques for pushing noise though circuits [14], improved stabiliser decompositions for cultivation circuits [16], and parametric decompositions that allow many shots to be sampled in parallel [2, 15]. To the best of our knowledge, however, all of these works still rely on the qubit-by-qubit marginalisation approach for sampling and none of them have simulated full end-to-end circuits including the escape stage. In that sense, the contribution of this work is largely orthogonal to these developments. By combining T-by-T sampling with some of these other ideas, we can hopefully obtain further significant speedups for QEC simulations.

Finally, although this work focuses primarily on Clifford+T circuits, the T-by-T algorithm actually trivially generalises to arbitrary Clifford+R_Z circuits. However, in that broader setting, stabiliser decomposition techniques are typically only approximate rather than exact [4]. Investigating the robustness of T-by-T sampling to approximation errors and applying it to Clifford+R_Z circuits would therefore be another interesting direction for future work.

Acknowledgements We would like to thank Benjamin Rodatz for useful discussions that led to the discovery of this algorithm. Furthermore, MK is supported by the Clarendon Fund Scholarship and the Sloane Robinson Scholarship.

References

[1] (2026): *QuiZX*. https://github.com/zxcalc/quizx.

[2] (2026): *tsim*. https://github.com/QuEraComputing/tsim.

[3] Scott Aaronson & Daniel Gottesman (2004): *Improved simulation of stabilizer circuits*. Physical Review A 70(5), p. 052328.

[4] Sergey Bravyi, Dan Browne, Padraic Calpin, Earl Campbell, David Gosset & Mark Howard (2019): *Simulation of quantum circuits by low-rank stabilizer decompositions*. Quantum 3, p. 181, doi:10.22331/q-2019-09-02-181. Available at https://doi.org/10.22331/q-2019-09-02-181.

[5] Sergey Bravyi & David Gosset (2016): *Improved Classical Simulation of Quantum Circuits Dominated by Clifford Gates*. Phys. Rev. Lett. 116, p. 250501, doi:10.1103/PhysRevLett.116.250501. Available at https://link.aps.org/doi/10.1103/PhysRevLett.116.250501.

[6] Sergey Bravyi, David Gosset & Yinchen Liu (2022): *How to Simulate Quantum Measurement without Computing Marginals*. Phys. Rev. Lett. 128, p. 220503, doi:10.1103/PhysRevLett.128.220503. Available at https://link.aps.org/doi/10.1103/PhysRevLett.128.220503.

[7] Michael J. Bremner, Ashley Montanaro & Dan J. Shepherd (2016): *Average-Case Complexity Versus Approximate Simulation of Commuting Quantum Computations*. Phys. Rev. Lett. 117, p. 080501, doi:10.1103/PhysRevLett.117.080501. Available at https://link.aps.org/doi/10.1103/PhysRevLett.117.080501.

[8] Craig Gidney (2021): *Stim: a fast stabilizer circuit simulator*. Quantum 5, p. 497, doi:10.22331/q-2021-07-06-497. Available at https://doi.org/10.22331/q-2021-07-06-497.

[9] Craig Gidney, Noah Shutty & Cody Jones (2024): *Magic state cultivation: growing T states as cheap as CNOT gates*. arXiv:2409.17595.

[10] Aleks Kissinger & John van de Wetering (2022): *Simulating quantum circuits with ZX-calculus reduced stabiliser decompositions*. Quantum Science and Technology 7(4), p. 044001, doi:10.1088/2058-9565/ac5d20. Available at https://dx.doi.org/10.1088/2058-9565/ac5d20.

[11] Aleks Kissinger, John van de Wetering & Renaud Vilmart (2022): *Classical Simulation of Quantum Circuits with Partial and Graphical Stabiliser Decompositions*. In François Le Gall & Tomoyuki Morimae, editors: *17th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2022)*, *Leibniz International Proceedings in Informatics (LIPIcs)* 232, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, pp. 5:1–5:13, doi:10.4230/LIPIcs.TQC.2022.5. Available at https://drops.dagstuhl.de/opus/volltexte/2022/16512.

[12] Hammam Qassim, Hakop Pashayan & David Gosset (2021): *Improved upper bounds on the stabilizer rank of magic states*. Quantum 5, p. 606, doi:10.22331/q-2021-12-20-606. Available at https://doi.org/10.22331/q-2021-12-20-606.

[13] Robert Raussendorf & Hans J. Briegel (2001): *A One-Way Quantum Computer*. Phys. Rev. Lett. 86, pp. 5188–5191, doi:10.1103/PhysRevLett.86.5188. Available at https://link.aps.org/doi/10.1103/PhysRevLett.86.5188.

[14] Samyak Surti, Lucas Daguerre & Isaac H. Kim (2025): *Efficient simulation of logical magic state preparation protocols*. arXiv:2512.23799.

[15] Matthew Sutcliffe & Aleks Kissinger (2025): *Fast Classical Simulation of Quantum Circuits via Parametric Rewriting in the ZX-Calculus*. Electronic Proceedings in Theoretical Computer Science 426, pp. 247–269, doi:10.4204/EPTCS.426.10.

[16] Kwok Ho Wan & Zhenghao Zhong (2026): *Simulating magic state cultivation with few Clifford terms*. arXiv:2509.08658.