Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Graph-Optimal T-Layering versus Replayable Local Realizability in Clifford+T Circuits

Csaba Biró

biro.csaba@uni-eszterhazy.hu

Eszterházy Károly Catholic University

Research Article

Keywords: Clifford+\((T)\) circuits, graph-based scheduling, SAT-based certification, replayable transformations, construction gaps, circuit realizability, graph-optimal \((T)\)-layering

Posted Date: May 26th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9732418/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Graph-Optimal T-Layering versus Replayable Local Realizability in Clifford+T Circuits

Csaba Biró[1,2*]

[1*]Faculty of Informatics, Eszterházy Károly Catholic University, Leányka 4., Eger, 3300, Heves, Hungary.
[2]Faculty of Informatics, Eötvös Loránd University, Pázmány Péter sétány 1/C, Budapest, Hungary, Country.

Corresponding author(s). E-mail(s): biro.csaba@uni-eszterhazy.hu;

Abstract

Optimizing the $\boldsymbol{T}$-depth of quantum circuits is a central challenge in fault-tolerant quantum computation, where non-Clifford resources dominate implementation cost. Graph-based abstractions provide lower bounds on achievable $\boldsymbol{T}$-depth by grouping commuting gates into parallel layers, typically computed via graph coloring. In this work, we investigate the extent to which such graph-optimal $\boldsymbol{T}$-layerings can be realized using a fixed, replayable local-commutation transformation framework. We introduce a formal distinction between graph-optimal depth, obtained via SAT-based certification, and the depth produced by a concrete transformation procedure. This leads to the notion of an observed construction gap. A replayable local-commutation constructor is defined, providing verifiable transformation sequences and a deterministic mapping from circuits to transformed circuit representations. Empirical evaluation across multiple circuit families indicates that the constructor frequently fails to realize graph-optimal layerings, even in structurally simple cases.
In particular, positive observed construction gaps are found in a substantial fraction of benchmark instances. These results highlight a systematic discrepancy between graph-level optimality and realizability under local transformation constraints. The findings suggest that graph-optimal $\boldsymbol{T}$-depth should be interpreted with caution when used as a proxy for achievable circuit performance under fixed replayable construction procedures.

Keywords: Clifford+$\boldsymbol{T}$ circuits, graph-based scheduling, SAT-based certification, replayable transformations, construction gaps, circuit realizability, graph-optimal $\boldsymbol{T}$-layering

1 Introduction

Fault-tolerant quantum computation places significant emphasis on the efficient use of non-Clifford resources, in particular T gates, which dominate implementation cost in many architectures [1, 2]. Consequently, reducing the T-depth of quantum circuits—that is, the number of sequential layers of T gates—has become a central objective in quantum circuit optimization.

A broad range of approaches has been proposed to reduce T-count and T-depth in Clifford+T circuits. Early work by Selinger [3] investigated circuit constructions with T-depth one, while polynomial-time optimization techniques based on matroid partitioning were introduced in [4]. Subsequent methods explored heuristic and synthesis-based optimization strategies [5–7], as well as compiler-level certificate-guided realization frameworks for large-scale quantum circuits [8].

In parallel, graphical techniques based on the ZX-calculus have enabled powerful rewrite-based simplification procedures by translating quantum circuits into graph-like representations and applying algebraic rewrite rules [9, 10]. These methods, implemented in tools such as PyZX [11], can explore large transformation spaces and often achieve substantial reductions in non-Clifford resources [12]. Modern quantum compilers such as Qiskit [13] and tket [14] similarly incorporate heuristic rewriting and scheduling strategies to reduce circuit depth in practice.

Another important direction uses SAT- and synthesis-based techniques to derive depth-optimal circuits or certify optimality for restricted instances. SAT encodings have been proposed for Clifford circuit synthesis [15, 16], $\{\mathrm{CNOT}, T\}$ synthesis [17], and more general synthesis formulations based on satisfiability and counting techniques [18]. These approaches provide strong optimality guarantees, but typically operate at the level of global search and synthesis rather than constrained replayable circuit transformations.

Complementary to optimization and synthesis methods, formal verification techniques have been developed to ensure correctness of quantum circuit transformations. Equivalence-checking frameworks [19, 20] provide rigorous guarantees that optimized circuits preserve circuit semantics, while formally verified certificate-guided realization frameworks such as VOQC [21] demonstrate that certified transformation pipelines are feasible in practice.

Despite the diversity of these approaches, a common abstraction underlying many certificate-guided realization frameworks is the notion of T-layering, where commuting T gates are grouped into parallelizable layers. Graph-based formulations of this abstraction naturally lead to graph coloring problems, in which the chromatic number of a dependency graph defines a graph-optimal T-depth.

However, such graph-level optimality implicitly assumes that the corresponding layering can be realized through valid circuit transformations. In practice, certificate-guided realization frameworks are typically restricted to finite sets of local rewrite rules that preserve semantic equivalence but may not enable all globally admissible reorderings. Structural constraints arising from locality, intermediate dependencies, or restricted transformation systems may therefore obstruct realization of graph-optimal layerings.

This motivates the following question: to what extent can graph-optimal T-layerings be realized under fixed replayable local-commutation transformation procedures?

Existing work primarily focuses on optimizing or synthesizing low-T-depth circuits. In contrast, the present work investigates whether graph-optimal T-layerings are realizable under a fixed replayable local-commutation transformation framework.

The framework distinguishes between graph-optimal T-depth and the depth produced by a fixed replayable local-commutation constructor. Rather than interpreting discrepancies between these quantities solely as implementation artifacts, the framework studies them as potentially structural phenomena arising from the interaction between graph-level abstraction and local realizability constraints.

To formalize this distinction, each circuit is associated with a dependency graph over its T gates, capturing pairwise commutation constraints. The chromatic number of this graph defines a graph-optimal layering obtainable via SAT-based certification. In contrast, a replayable local-commutation constructor produces a concrete transformed circuit whose achievable T-depth may differ from this graph-level optimum.

This leads to the notion of an observed construction gap, defined as the difference between graph-optimal depth and the depth produced by the replayable constructor. The framework combines:

- SAT-based certification of graph-optimal T-layerings,
- replayable local transformation sequences preserving circuit equivalence,
- empirical evaluation across multiple circuit families.

The experimental results show that positive observed construction gaps occur frequently in practice, including in structurally simple circuit regimes with low graph complexity.

Unlike prior optimization-oriented approaches, the present work does not aim to improve circuit depth directly. Instead, it investigates the realizability of graph-optimal T-layerings under constrained replayable transformation procedures, thereby separating abstract graph-level optimality from locally constructible circuit realizations.

These findings suggest that graph-based optimality notions should be interpreted with caution when used as proxies for achievable circuit performance under restricted local transformation models.

2 Related Work

2.1 T-Depth and Non-Clifford Resource Optimization

The optimization of non-Clifford resources, in particular T gates, is a central problem in fault-tolerant quantum computation due to the high implementation cost of magic-state distillation and fault-tolerant T gate execution [2]. As a consequence, reducing T-count and T-depth has become a major objective in quantum circuit optimization.

Early investigations of Clifford+T circuits with T-depth one established that substantial reductions in sequential non-Clifford layers are possible under suitable ancilla assumptions [3]

Algebraic techniques based on meet-in-the-middle search and phase-polynomial optimization were subsequently developed for depth-oriented Clifford+T synthesis [1, 4].

A variety of additional optimization strategies have since been proposed. Heuristic synthesis procedures targeting T-depth optimality have also been investigated [5, 6].

Other approaches focus on reducing non-Clifford resource usage through compiler-level or structural simplifications [7, 22]. Large-scale parameterized circuit optimization has also been explored in practical compiler frameworks [8].

These works primarily address the problem of constructing or synthesizing circuits with reduced T-depth. In contrast, the present work investigates whether graph-level optimal layerings are realizable under constrained replayable local transformation systems.

2.2 ZX-Calculus and Rewrite-Based Optimization

Graphical reasoning frameworks based on the ZX-calculus have become a powerful tool for quantum circuit optimization and simplification.

Graph-theoretic simplification procedures derived from ZX-diagram rewriting have been shown to substantially reduce circuit complexity [9].

PyZX [11] provides an automated large-scale implementation of such rewrite procedures and has become a widely used certificate-guided realization framework for Clifford+T circuits.

Non-Clifford gate reduction using ZX-calculus-based rewriting strategies has also been studied [12], while expository treatments such as [10] further established the ZX-calculus as a practical framework for quantum circuit optimization.

These approaches typically explore large rewrite spaces using algebraic or graph-based simplifications. However, the relationship between graph-level parallelization and realizability under restricted replayable local transformations is usually not analyzed explicitly. The framework instead focuses on the realizability of graph-optimal T-layerings under constrained local transformation procedures.

2.3 SAT-Based Synthesis and Optimality Certification

SAT- and synthesis-based methods provide another important direction for quantum circuit optimization. SAT encodings have been used for exact or depth-optimal synthesis of Clifford and Clifford+T circuits [15–17]. More recently, satisfiability-counting formulations based on #SAT have been proposed for quantum circuit synthesis and optimization [18].

These approaches are particularly valuable because they provide explicit optimality guarantees and certified constructions. In many cases, SAT-based formulations can determine provably minimal circuit depth or gate count for restricted circuit classes.

SAT-based techniques are also employed here, but with a substantially different objective. Rather than using SAT solving to synthesize optimized circuits directly, SAT-based graph coloring is used to certify graph-optimal T-layerings. The primary focus is then placed on whether such graph-optimal layerings can be realized through replayable local transformations.

2.4 Verification, Equivalence, and Realizability Constraints

As quantum circuit optimization pipelines become increasingly complex, formal verification and equivalence checking have become important complementary directions. Advanced equivalence-checking techniques for quantum circuits have been developed together with practical verification tooling [19, 20].

These methods provide rigorous guarantees that optimized circuits preserve the intended unitary semantics.

Verified certificate-guided realization frameworks have also emerged. Formally verified optimization pipelines have also been shown to be feasible in proof-assistant-based frameworks such as VOQC [21].

In parallel, recent work has highlighted the intrinsic computational hardness of quantum circuit optimization problems. Several important quantum circuit optimization problems, including tasks related to non-Clifford resource minimization, have been shown to be computationally hard in general [23] .

A different question is addressed here. Rather than proving global hardness or constructing formally verified optimizers, the focus here is on the realizability of graph-optimal layerings under a fixed replayable local-commutation constructor. The resulting observed construction gap characterizes a discrepancy between abstract graph-level optimality and locally realizable circuit transformations.

In contrast to prior optimization-, synthesis-, and verification-oriented approaches, the present work focuses on the realizability of SAT-certified graph-optimal T-layerings under a fixed replayable local-commutation constructor.

The present work instead focuses on the discrepancy between graph-level optimality and replayable constructive realizability under constrained local transformation systems.

3 Formal Model

This section introduces the formal framework used to distinguish graph-optimal T-layering from constructor-induced circuit depth under a fixed replayable transformation procedure.

3.1 Circuit Model

Definition 1 (Clifford+T circuit) A Clifford+T circuit C is a quantum circuit composed of gates from the Clifford group together with the T gate. Let $\mathcal{T}(C)$ denote the set of all T gates appearing in C.

Definition 2 (T-depth) The T-depth of a circuit C, denoted $\mathrm{depth}_T(C)$, is the minimum number of sequential layers of T gates required to implement C, where gates in the same layer can be executed in parallel.

3.2 Dependency Graph

Definition 3 (Dependency graph) Given a circuit C, the *dependency graph* is an undirected graph

$$G(C) = (V, E),$$

where:

- each vertex corresponds to a T gate in $\mathcal{T}(C)$,
- an edge $(u, v) \in E$ exists if the corresponding gates are incompatible under the considered scheduling abstraction and therefore cannot be assigned to the same T-layer.

The dependency graph captures sufficient non-commutation constraints for graph-level certification, while constructive realizability is treated separately at the circuit-transformation level.

Remark 1 The dependency graph captures pairwise incompatibility constraints used by the graph-level scheduling abstraction. It does not encode realizability under a specific transformation procedure.

3.3 Graph-Optimal T-Layering

Definition 4 (Graph-optimal T-depth) Let $G(C)$ be the dependency graph of a circuit C. The graph-optimal T-depth is defined as

$$L^*(C) = \chi(G(C)).$$

Proposition 1 *For any circuit C, the following lower bound holds:*

$$\mathrm{depth}_T(C) \geq L^*(C).$$

Proof Any valid T-layering induces a proper coloring of $G(C)$, where each layer corresponds to a color class. Since adjacent vertices cannot share the same color, every realizable layering requires at least $\chi(G(C))$ layers. □

Remark 2 The graph-optimal depth characterizes parallelism within the dependency-graph abstraction rather than directly realizable circuit schedules. Consequently, small graph-optimal depths may coexist with large constructor-induced depths under constrained replayable transformation systems.

3.4 SAT-Based Optimal T-Layering

Computing the chromatic number $\chi(G(C))$ is NP-hard. To obtain graph-optimal T-layerings in practice, a SAT-based encoding is used. The encoding is expressed using pseudo-Boolean constraints for clarity.

Let $G(C) = (V, E)$ and let k be a candidate number of layers. Introduce Boolean variables

$$x_{v,\ell} \in \{0, 1\}, \quad v \in V,\ \ell \in \{1, \ldots, k\},$$

where $x_{v,\ell} = 1$ indicates that vertex v is assigned to layer ℓ.

The constraints are:

- **Assignment:**

$$\sum_{\ell=1}^{k} x_{v,\ell} = 1 \quad \forall v \in V.$$

- **Conflict:**

$$x_{u,\ell} + x_{v,\ell} \leq 1 \quad \forall (u, v) \in E,\ \ell = 1, \ldots, k.$$

The formula is satisfiable if and only if $G(C)$ admits a proper coloring with k colors. The smallest such k yields $L^*(C)$. The search for k can be performed incrementally, starting from simple lower bounds. In practice, the above constraints are translated into conjunctive normal form (CNF) before being passed to a SAT solver.

Remark 3 This formulation provides both the optimal depth and an explicit certificate of the corresponding layering.

3.5 Constructor-Induced Depth

Definition 5 (Replayable local-commutation constructor) Let $\mathcal{A}$ denote a fixed transformation procedure that applies a sequence of replayable local commutation rules to a circuit C, producing a circuit $\mathcal{A}(C)$.

Definition 6 (Constructor-induced T-depth) The constructor-induced T-depth is defined as

$$\widehat{L}_{\mathcal{A}}(C) = \mathrm{depth}_T(\mathcal{A}(C)).$$

Remark 4 The value $\widehat{L}_{\mathcal{A}}(C)$ is not claimed to be optimal over all circuits reachable under local commutation rules. It reflects the depth achieved by the specific replayable transformation procedure used in this work.

3.6 Observed Construction Gap

Definition 7 (Observed construction gap) The observed construction gap is defined as

$$\widehat{\Delta}(C) = \widehat{L}_{\mathcal{A}}(C) - L^*(C).$$

Remark 5 A positive value of $\widehat{\Delta}(C)$ indicates that the SAT-certified graph-optimal T-layering is not realized by the considered replayable local-commutation constructor.

3.7 Problem Statement

The central problem investigated in this work is whether the graph-optimal T-depth $L^*(C)$ is realized by a fixed replayable local-commutation constructor $\mathcal{A}$, and if not, to characterize the resulting observed construction gap.

4 Transformation Framework

This section describes the replayable local-commutation constructor $\mathcal{A}$ used to evaluate the constructibility of graph-optimal T-layerings.

4.1 Overview

The constructor $\mathcal{A}$ transforms an input circuit C into a new circuit $\mathcal{A}(C)$ by applying a sequence of local commutation rules. Each transformation step is explicitly recorded, resulting in a replayable sequence that certifies semantic equivalence between C and $\mathcal{A}(C)$.

The objective of the constructor is to attempt the realization of the SAT-certified graph-optimal T-layering through replayable local commutation transformations. The SAT layer assignment is defined over the dependency-graph abstraction, whereas the constructive realization operates over the ordered circuit instruction sequence under replayable local commutation constraints. The constructor is deterministic given a fixed input circuit and rule set, ensuring reproducibility of the resulting circuit and its depth.

4.2 Local Commutation Rules

The transformation procedure is restricted to a fixed set of local rewrite rules. Each rule operates on adjacent gates and allows their reordering if the corresponding operations commute.

Formally, a transformation step has the form

$$C_i \;\rightarrow\; C_{i+1},$$

where C_{i+1} is obtained from C_i by swapping two adjacent gates that satisfy a valid commutation condition.

These rules are:

- **Local:** they operate only on neighboring gates,
- **Sound:** they preserve the unitary semantics of the circuit,
- **Replayable:** each step can be recorded and verified independently.

4.3 Replayable Transformation Sequences

Each execution of $\mathcal{A}$ produces a sequence

$$C = C_0 \rightarrow C_1 \rightarrow \cdots \rightarrow C_k = \mathcal{A}(C),$$

where each transition corresponds to a valid application of a local commutation rule.

This sequence serves as an experimentally verifiable transformation witness between C and $\mathcal{A}(C)$, since each local rewrite step can be independently replayed and checked. The replayability property therefore provides a concrete verification mechanism for the constructor output, avoiding reliance on black-box optimization procedures. Every accepted transformation is represented as a replayable adjacent swap justified by an explicit local commutation rule. Global equivalence follows by composition of replay-validated locally sound adjacent commutations.

4.4 Greedy Retiming Strategy

The constructor follows a greedy replayable strategy that attempts to realize the target graph-optimal layering through admissible adjacent local commutations.

At each step:

- a target layering, typically derived from the SAT-based optimal solution, is considered,
- gates are examined in circuit order,
- local commutation rules are applied to move T gates toward earlier positions,
- transformations are accepted only if they are valid under the local rule set.

If no valid sequence of local commutations allows a gate to be moved further, the gate is considered *blocked*.

4.5 Blocking and Structural Limitations

A key feature of the constructor is the presence of blocking configurations.

Definition 8 (Blocking) A T gate is said to be blocked if it cannot be moved to an earlier layer using any sequence of admissible local commutation steps.

Blocking arises when:

- required reorderings involve non-adjacent gates,
- intermediate transformations are not permitted by the local rule set,
- circuit structure enforces ordering constraints not captured by pairwise commutation alone.

These situations may prevent the constructor from realizing a graph-optimal T-layering.

4.6 Depth Evaluation

After applying the transformation, the T-depth of the resulting circuit $\mathcal{A}(C)$ is computed using a greedy scheduling procedure consistent with the dependency constraints. This yields the constructor-induced depth

$$\widehat{L}_{\mathcal{A}}(C) = \mathrm{depth}_T(\mathcal{A}(C)).$$

4.7 Relation to Graph-Optimal Layering

The constructor $\mathcal{A}$ operates independently of the SAT-based optimal layering. The graph-optimal value $L^*(C)$ provides a lower bound on achievable depth, while $\widehat{L}_{\mathcal{A}}(C)$ reflects what can be realized using the restricted transformation system.

A discrepancy between these values indicates that the optimal layering is not realized by the constructor.

Remark 6 The constructor does not explore the full space of circuits reachable under all possible local commutation sequences. Instead, it provides a controlled, replayable, and implementation-aligned procedure for evaluating constructibility.

The constructor is intentionally conservative. Its role in this work is not to serve as a competitive T-depth optimization heuristic, but to provide a replayable constructive realizability test relative to the SAT-certified graph optimum. Consequently, the inability of the constructor to realize the graph-optimal layering is interpreted as evidence of a construction gap under the fixed local commutation system, rather than as a failure of the graph certificate itself.

5 Theoretical Results

This section establishes the key theoretical properties of the replayable local-commutation framework and clarifies the relationship between graph-optimal layering and constructor-induced realizability under a fixed replayable transformation system.

5.1 Equivalence under Local Commutation

Theorem 2 *Let C' be obtained from a circuit C through a finite sequence of replayable local commutation transformations. Then C and C' are unitary equivalent.*

Proof Each local commutation rule preserves circuit semantics by reordering adjacent commuting operations. Since semantic equivalence is preserved at every transformation step, the composition of these locally sound transformations also preserves the implemented unitary. Therefore, the overall replayable transformation sequence preserves unitary equivalence between C and C'. □

Remark 7 The equivalence claim applies specifically to the replayable transformation sequence produced by the constructor $\mathcal{A}$. The framework does not assume correctness of external optimization or transpilation procedures.

5.2 Limitations of Local Commutation

Lemma 3 Local commutation transformations do not, in general, realize all gate reorderings compatible with pairwise commutation constraints.

Proof Local transformations operate only on adjacent gates. Reordering non-adjacent gates may require intermediate transformations that are not permitted by the considered local rule system. Consequently, pairwise compatibility alone does not guarantee realizability through replayable local transformations. □

Proposition 4 *Pairwise compatibility in the dependency graph does not, in general, imply replayable realizability under adjacent local commutation systems.*

Proof The dependency graph abstraction captures pairwise incompatibility constraints between T gates. However, realizability under replayable local commutation depends on the existence of a valid sequence of admissible adjacent transformations in the ordered circuit representation. Intermediate blocking configurations may prevent such realizations even when the corresponding graph-level layering is combinatorially valid. □

Remark 8 The graph-optimal layering is defined over the dependency-graph abstraction, whereas replayable realizability is defined over constrained circuit transformations. These notions therefore characterize different levels of abstraction.

5.3 Observed Separation

Proposition 5 *There exist benchmark instances for which*

$$\widehat{L}_{\mathcal{A}}(C) > L^{*}(C).$$

Proof This follows from the empirical evaluation presented in Section 6, where multiple circuit instances exhibit strictly positive observed construction gaps. □

Remark 9 The observed separation characterizes the behavior of the specific replayable constructor $\mathcal{A}$ considered in this work and does not imply limitations of all possible transformation systems or rewrite frameworks.

6 Experimental Results

This section presents the empirical evaluation of the replayable local-commutation constructor $\mathcal{A}$ and analyzes the relationship between SAT-certified graph-optimal T-layerings and replayable constructive realizability.

6.1 Experimental Design and Evidence Structure

The experimental evaluation is organized into three complementary evidence regimes designed to isolate different aspects of the framework:

1. a circuit-derived benchmark suite used to study replayable constructive realizability in Clifford+T circuits,

2. a large-scale separation search used to identify and characterize positive observed construction gaps,
3. a graph-stress benchmark used to evaluate the scalability of the SAT-based certification layer independently of circuit realizability.

For each circuit C, the following quantities were computed:

- the dependency graph $G(C)$,
- the graph-optimal T-layering value $L^*(C)$ obtained using the SAT-based formulation from Section 3,
- the constructor-induced depth $\widehat{L}_{\mathcal{A}}(C)$ obtained using the replayable transformation framework from Section 4,
- the observed construction gap

$$\widehat{\Delta}(C) = \widehat{L}_{\mathcal{A}}(C) - L^*(C).$$

All SAT instances in the circuit-derived benchmark suite were solved to optimality, yielding certified graph-optimal layerings together with explicit coloring certificates. Replayable transformation logs were generated for every transformed circuit instance.

Table 1 Summary of the experimental evidence regimes.

Block	Inst.	Pos.	Max gap	Max tw	Role
Circuit suite	140	78	32	1	Circuit-derived realizability evidence
Separation search	1620	1036	30	1	Targeted separation-witness search
Graph stress	42	–	–	50	Certification scalability analysis

The experimental design intentionally separates graph-level certification from replayable constructive realization. Consequently, graph-optimal layerings are interpreted as certified graph-theoretic lower bounds, while the constructor output characterizes realizability under the fixed replayable local transformation system.

The low graph-optimal depths observed in several benchmark families reflect the aggressive parallelism permitted by the dependency-graph abstraction rather than low overall circuit complexity. The experimental objective is therefore not to interpret the graph-optimal layering as a directly realizable circuit schedule, but to study the discrepancy between graph-level compatibility and replayable constructive realizability.

6.2 Certificate and Replay Validation

The correctness of both the SAT-certification layer and the replayable transformation framework was first validated.

Across the full circuit-derived benchmark suite of 140 instances:

- graph-optimal layerings were successfully certified in all cases (140/140),
- replayable transformation certificates were valid in all cases (140/140),
- replay-validated global equivalence between the encoding-input circuit and the conservatively retimed circuit was established in all cases (140/140).

These results validate both the correctness of the SAT-based graph-layer certification procedure and the semantic soundness of the replayable local-commutation constructor. Importantly, the equivalence guarantees are derived from replayable sequences of locally verified adjacent commutations rather than from black-box optimization procedures.

6.3 Main Circuit Suite: Observed Construction Gaps

Non-zero observed construction gaps occur in a substantial fraction of the circuit-derived benchmark suite. Specifically, 78 out of 140 circuits exhibit

$$\widehat{\Delta}(C) > 0,$$

corresponding to 55.7% of the evaluated instances. The maximum observed construction gap is 32.

These results provide empirical evidence that SAT-certified graph-optimal T-layerings are frequently not realized by the replayable constructor $\mathcal{A}$.

Importantly, the conservative constructor does not improve the input T-depth within the main benchmark suite: in all 140 instances,

$$\widehat{L}_{\mathcal{A}}(C) = \mathrm{depth}_T(C).$$

This is consistent with its intended role in the present work: the constructor serves as a replayable realizability test relative to the SAT-certified graph optimum rather than as a competitive optimization heuristic.

The observed construction gap should therefore be interpreted as evidence of a discrepancy between graph-level optimality and replayable constructive realizability under the fixed local commutation system.

6.4 Family-Level Realizability Structure

The observed construction gaps exhibit strong family-dependent structure.

Grover-like, QFT-like, and complete-QAOA circuits exhibit particularly large observed construction gaps, while VQE-like and clique-based circuits show no separation in the main circuit suite. This indicates that replayable realizability depends strongly on circuit structure and cannot be inferred solely from graph-level layering properties.

Table 2 Family-level realizability statistics in the circuit-derived benchmark suite.

Family	Instances	Positive gaps	Mean gap	Max gap
grover	7	6	16.86	32
qft_phase_like	12	12	8.00	18
qaoa_complete	14	14	7.64	20
random_ct	21	20	2.43	5
brickwork_ct	12	6	1.00	2
random_interaction_ct	12	6	1.00	2
dense_phase	18	6	0.33	1
qaoa_ring	24	8	0.33	1
clique_stress	12	0	0.00	0
vqe_su2	8	0	0.00	0

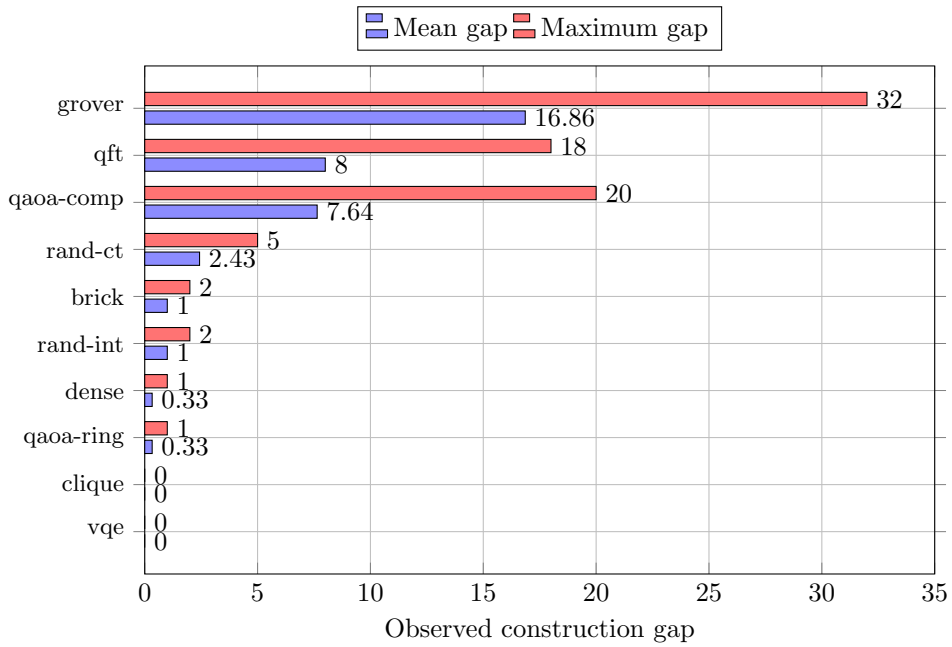


Fig. 1 Family-level realizability structure in the circuit-derived benchmark suite. Bars report mean and maximum observed construction gaps for each circuit family. Zero-length bars indicate families in which no construction gap was observed in the main circuit suite.

6.5 Obstruction Certificates

The replayable framework additionally provides obstruction-level evidence explaining why graph-optimal layerings fail to be realized.

Across the main circuit suite:

- obstruction certificates are present in 127 instances,
- a total of 2968 failed replayable swaps were recorded.

The obstruction analysis reveals that replayability limitations are dominated by a small number of recurring blocking mechanisms. Across all circuit-derived instances, the failed swaps are distributed as follows:

$$CX : 2061, \qquad H : 616, \qquad RY : 167, \qquad RX : 124.$$

Thus, controlled-X operations constitute the dominant obstruction source, while Hadamard and rotation-induced ordering constraints also contribute in several circuit families.

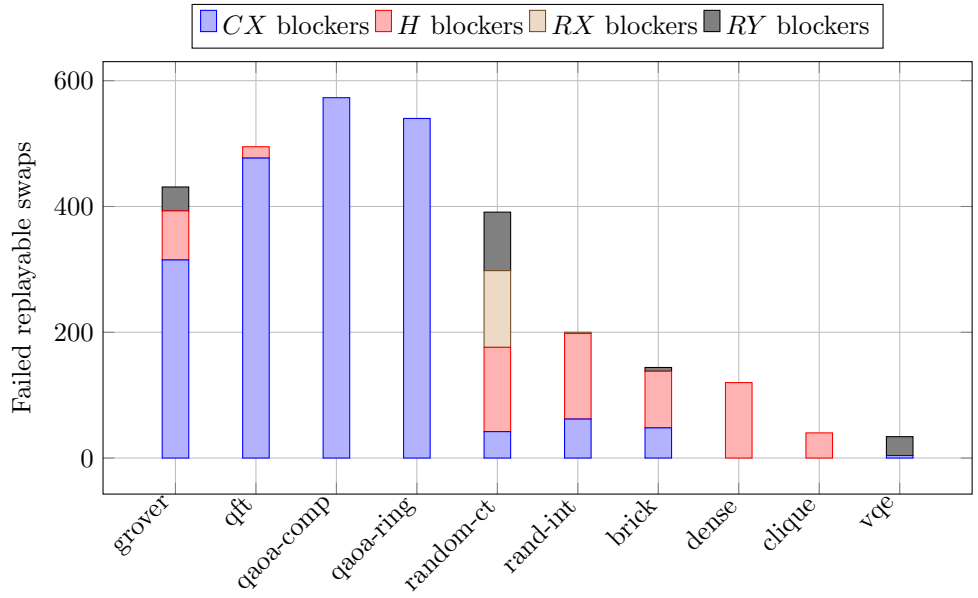


Fig. 2 Obstruction certificate structure by circuit family. The stacked bars report the total number of failed replayable swaps attributed to the dominant blocking gate types.

These results suggest that the observed construction gap is associated with recurring structural replayability limitations rather than isolated implementation artifacts.

6.6 Separation Search

A larger targeted search over 1620 generated circuit instances was performed to evaluate the robustness of the separation phenomenon.

The search produced:

- 1036 instances with $\widehat{\Delta}(C) > 0$,
- a separation rate of 63.95%,
- a maximum observed gap of 30,
- maximum observed treewidth 1,

- maximum graph-optimal depth 2.

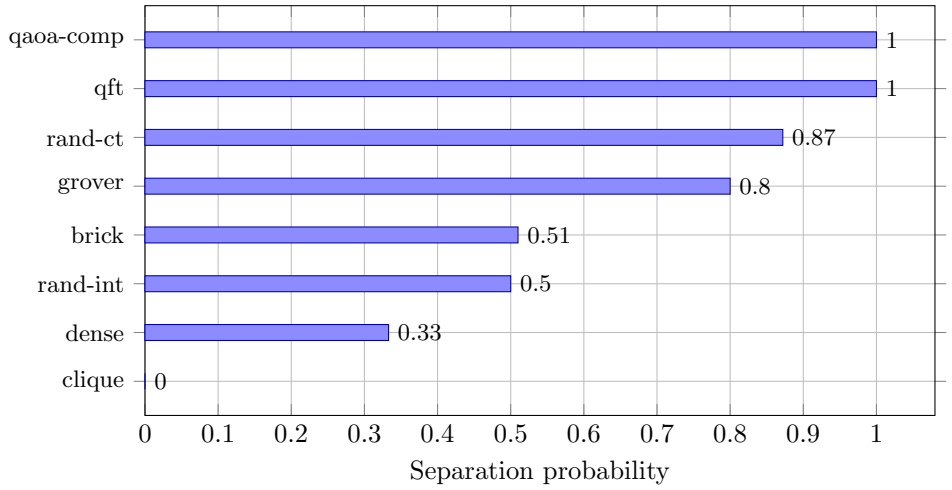


Fig. 3 Separation probability in the targeted search. Values report the probability that the replayable constructor fails to realize the SAT-certified graph-optimal layering.

Representative separation witnesses from the circuit-derived benchmark suite are shown in Table 3. These examples demonstrate that even circuits with extremely small graph-optimal depth may exhibit large replayability-induced construction gaps.

Table 3 Representative circuits exhibiting large observed construction gaps.

Circuit	Family	$\lvert\mathcal{T}\rvert$	L^*	$\widehat{L}_{\mathcal{A}}$	$\widehat{\Delta}$
grover_n5	grover	42	2	34	32
grover_n12	grover	140	2	32	30
qaoa_comp._n12	qaoa	132	2	22	20
qft_phase_n10	qft	110	2	20	18

6.7 Near-Tree Structural Regime

An important structural observation is that the circuit-derived benchmark suite remains in an extremely low-complexity graph regime.

Across the main dataset:

- observed treewidth values never exceed 1,
- the maximum graph-optimal depth is 2,
- nevertheless, 78 out of 140 instances exhibit positive observed construction gaps.

Despite this structural simplicity, large construction gaps occur frequently. Figure 4 reports the actual T-count/gap pairs from the full circuit-derived benchmark suite.

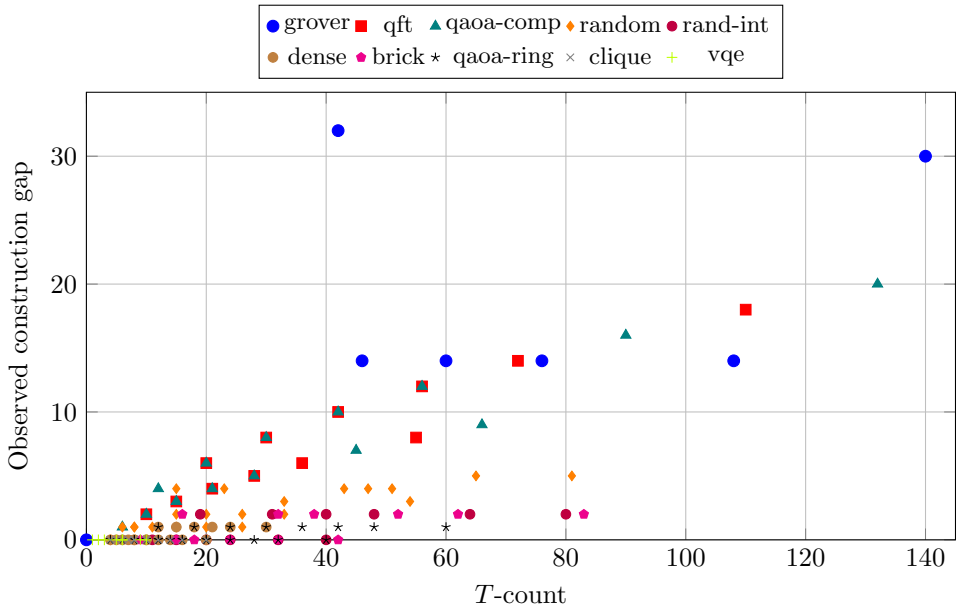


Fig. 4 Actual T-count versus observed construction gap for all 140 circuit-derived benchmark instances. All corresponding dependency graphs remain in the near-tree regime, with treewidth at most 1 and graph-optimal depth at most 2.

This indicates that replayability-induced separation is not merely a consequence of highly complex dependency structures. Instead, the phenomenon already emerges in structurally simple graph regimes.

Large observed construction gaps also cannot be explained solely by circuit size, since several benchmark families with comparable T-counts exhibit substantially different replayability behavior. In particular, Grover-like and QFT-like circuits frequently exhibit large construction gaps, whereas other families with similar or larger T-counts may show only negligible separation.

6.8 Graph-Stress Certification Analysis

To evaluate the scalability of the SAT-based certification layer independently of circuit realizability, graph-only stress instances were generated. These instances are not interpreted as circuit-level construction-gap witnesses. Instead, they isolate the graph-coloring certification component.

The graph-stress benchmark contains 42 graph instances. The maximum observed graph treewidth is 50, and the maximum certified graph-layer value is 32.

The graph-stress experiments show that increasing structural complexity can substantially increase SAT encoding size. These experiments are included primarily to

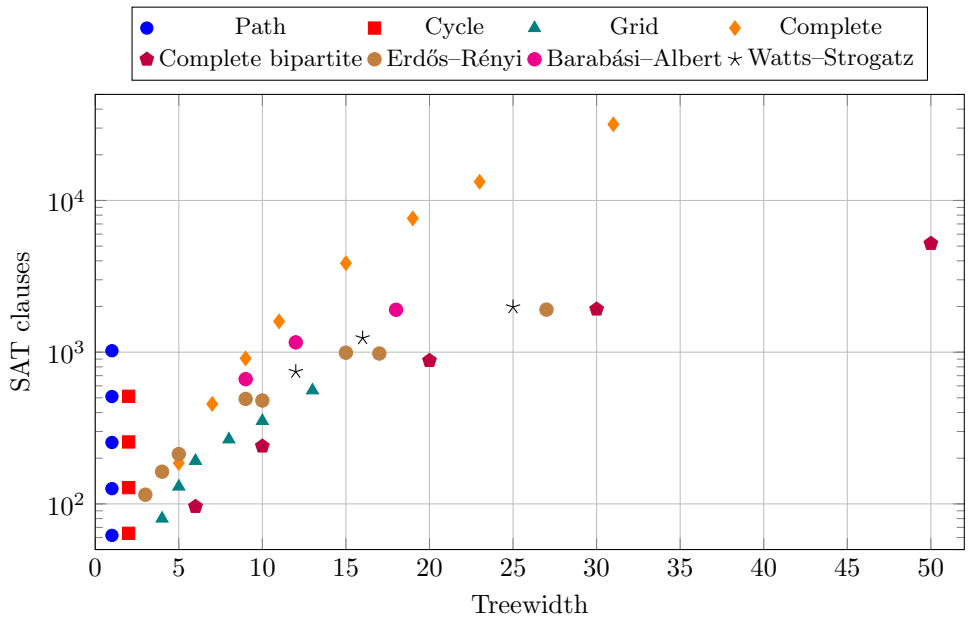


Fig. 5 Graph-stress instances: dependency-graph treewidth versus SAT encoding size. Different graph families exhibit substantially different SAT-growth regimes as structural complexity increases. These graph-only instances characterize the certification layer and are not interpreted as circuit-level construction-gap witnesses.

characterize the scalability behavior of the certification layer rather than replayable constructive realizability.

6.9 Interpretation

The experimental results collectively demonstrate that:

- graph-optimal T-layerings can be certified reliably using SAT-based graph coloring,
- replayable local transformations preserve circuit semantics,
- positive observed construction gaps occur frequently,
- the separation phenomenon depends strongly on circuit structure,
- replayability-induced obstruction persists even in near-tree graph regimes.

The evidence therefore indicates a systematic discrepancy between graph-level optimality and replayable constructive realizability under constrained local transformation systems.

Remark 10 The observed construction gaps are relative to the specific replayable local-commutation constructor considered in this work. Consequently, the results do not exclude the possibility that more expressive rewrite systems, alternative dependency abstractions, or non-local transformation procedures could realize smaller gaps for some circuit families.

7 Conclusion

The relationship between graph-optimal T-layering and realizability under replayable local-commutation transformations was investigated.

A formal distinction was introduced between graph-optimal T-depth, obtained via SAT-based certification, and the depth produced by a concrete replayable transformation procedure. This distinction was captured by the notion of an observed construction gap, which measures the discrepancy between these two quantities.

A deterministic and replayable constructor was defined, enabling verifiable circuit transformations based on local commutation rules. The formal framework together with the empirical evaluation provides evidence that locality constraints can limit the realizability of graph-optimal layerings under constrained replayable transformation procedures.

Empirical evaluation demonstrated that the constructor frequently fails to realize graph-optimal layerings, even in circuits with simple dependency structures. The observed gaps are consistent across multiple circuit families and parameter regimes, indicating that the phenomenon is not isolated.

The persistence of large observed construction gaps even under extremely low graph complexity suggests that replayable constructive realizability may depend on structural properties not captured by pairwise graph compatibility alone.

These findings indicate that graph-level abstractions may systematically overestimate replayable realizable parallelism under constrained local transformation systems.

Future work includes extending the transformation framework, exploring more expressive rewrite systems, and investigating structural conditions under which graph-optimal layerings become realizable.

Supplementary information. Supplementary material, including replay certificates, benchmark artifacts, plotting data, and experimental logs, is available in the public repository associated with this work.

Acknowledgements. This research was supported by the National Research, Development and Innovation Office within the Quantum Information National Laboratory of Hungary (Grant No. 2022-2.1.1-NL-2022-00004).

Declarations

Funding. This research was supported by the National Research, Development and Innovation Office within the Quantum Information National Laboratory of Hungary (Grant No. 2022-2.1.1-NL-2022-00004).

Ethics, Consent to Participate, and Consent to Publish declarations. Not applicable.

Conflict of interest/Competing interests. The author declares no competing interests.

Data availability. The experimental data, replay certificates, benchmark artifacts, and plotting data generated during this study are publicly available at:

https://github.com/drcsababiro-alt/certified-t-layer-realizability

Code availability. The implementation used to generate the experimental results and replay-validation artifacts is publicly available at:

https://github.com/drcsababiro-alt/certified-t-layer-realizability

Author contribution. Cs. Biró conceived the study, developed the formal framework, implemented the experimental pipeline, performed the computational experiments, analyzed the results, and wrote the manuscript.

References

[1] Amy, M., Maslov, D., Mosca, M., Roetteler, M.: A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **32**(6), 818–830 (2013) https://doi.org/10.1109/TCAD.2013.2244643

[2] Bravyi, S., Smith, G., Smolin, J.A.: Trading classical and quantum computational resources. Phys. Rev. X **6**, 021043 (2016) https://doi.org/10.1103/PhysRevX.6.021043

[3] Selinger, P.: Quantum circuits of T-depth one. Physical Review A **87**(4), 042302 (2013) https://doi.org/10.1103/PhysRevA.87.042302

[4] Amy, M., Maslov, D., Mosca, M.: Polynomial-time T-depth optimization of Clifford+T circuits via matroid partitioning. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **33**(10), 1476–1489 (2014) https://doi.org/10.1109/TCAD.2014.2341953

[5] Gheorghiu, V., Mosca, M., Mukhopadhyay, P.: A (quasi-)polynomial time heuristic algorithm for synthesizing T-depth optimal circuits. npj Quantum Information **8**, 110 (2022) https://doi.org/10.1038/s41534-022-00624-1

[6] Niemann, P., Gupta, A., Drechsler, R.: T-depth optimization for fault-tolerant quantum circuits. In: 2019 IEEE 49th International Symposium on Multiple-Valued Logic (ISMVL), pp. 108–113 (2019). https://doi.org/10.1109/ISMVL.2019.00027

[7] Heyfron, L.E., Campbell, E.T.: An efficient quantum compiler that reduces T count. Quantum Science and Technology **4**(1), 015004 (2018) https://doi.org/10.1088/2058-9565/aad604

[8] Nam, Y., Ross, N.J., Su, Y., Childs, A.M., Maslov, D.: Automated optimization

of large quantum circuits with continuous parameters. npj Quantum Information **4**(1), 23 (2018) https://doi.org/10.1038/s41534-018-0072-4

[9] Duncan, R., Kissinger, A., Perdrix, S., Wetering, J.: Graph-theoretic simplification of quantum circuits with the zx-calculus. Quantum **4**, 279 (2020) https://doi.org/10.22331/q-2020-06-04-279

[10] Wetering, J.: ZX-calculus for the working quantum computer scientist. arXiv preprint arXiv:2012.13966 (2020) https://doi.org/10.48550/arXiv.2012.13966 arXiv:2012.13966 [quant-ph]

[11] Kissinger, A., Wetering, J.: Pyzx: Large scale automated diagrammatic reasoning. Electronic Proceedings in Theoretical Computer Science **318**, 229–241 (2020) https://doi.org/10.4204/EPTCS.318.14

[12] Kissinger, A., Wetering, J.: Reducing the number of non-Clifford gates in quantum circuits. Physical Review A **102**(2), 022406 (2020) https://doi.org/10.1103/PhysRevA.102.022406

[13] IBM Quantum: Qiskit: An Open-source Framework for Quantum Computing. https://qiskit.org (2021)

[14] Sivarajah, S., Dilkes, S., Cowtan, A., Simmons, W., Edgington, A., Duncan, R.: t—ket⟩: a retargetable compiler for nisq devices. Quantum Science and Technology **6**(1), 014003 (2020) https://doi.org/10.1088/2058-9565/ab8e92

[15] Peham, T., Brandl, N., Kueng, R., Wille, R., Burgholzer, L.: Depth-optimal synthesis of clifford circuits with sat solvers. In: 2023 IEEE International Conference on Quantum Computing and Engineering (QCE), vol. 01, pp. 802–813 (2023). https://doi.org/10.1109/QCE57702.2023.00095

[16] Schneider, S., Burgholzer, L., Wille, R.: A SAT encoding for optimal Clifford circuit synthesis. In: Proceedings of the 28th Asia and South Pacific Design Automation Conference, pp. 190–195 (2023). https://doi.org/10.1145/3566097.3567929

[17] Meuli, G., Soeken, M., Micheli, G.D.: SAT-based $\{\text{CNOT}, T\}$ quantum circuit synthesis. In: Reversible Computation. Lecture Notes in Computer Science, vol. 11106, pp. 175–188. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-99498-7_12

[18] Zak, D., Mei, J., Lagniez, J.-M., Laarman, A.: Reducing quantum circuit synthesis to #sat. arXiv preprint arXiv:2208.00416 (2025) https://doi.org/10.48550/arXiv.2508.00416

[19] Burgholzer, L., Wille, R.: Advanced equivalence checking for quantum circuits. IEEE Transactions on Computer-Aided Design of Integrated Circuits and

Systems **40**(9), 1810–1824 (2021) https://doi.org/10.1109/TCAD.2020.3032630

[20] Burgholzer, L., Wille, R.: QCEC: A JKQ tool for quantum circuit equivalence checking. Software Impacts **7**, 100051 (2021) https://doi.org/10.1016/j.simpa.2020.100051

[21] Hietala, K., Rand, R., Hung, S.-H., Wu, X., Hicks, M.: A verified optimizer for quantum circuits. Proceedings of the ACM on Programming Languages **5**(POPL), 1–29 (2021) https://doi.org/10.1145/3434318

[22] Abdessaied, N., Soeken, M., Drechsler, R.: Quantum circuit optimization by Hadamard gate reduction. In: Reversible Computation. Lecture Notes in Computer Science, vol. 8507, pp. 149–162. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-08494-7_12

[23] Wetering, J., Amy, M.: Optimising quantum circuits is generally hard. arXiv preprint arXiv:2310.05958 (2024) https://doi.org/10.48550/arXiv.2310.05958 arXiv:2310.05958 [quant-ph]